# USING DEPTH INFORMATION TO AID STEREOSCOPIC IMAGE FORENSICS

BY

MARK-ANTHONY FOUCHÉ

Submitted in partial fulfilment of the requirements for the degree

**Master of Science (Computer Science)**

in the

**Faculty of Engineering, Build Environment and Information Technology**

**UNIVERSITY OF PRETORIA**

**September 2014**

© University of Pretoria

© University of Pretoria

# SUMMARY

## USING DEPTH INFORMATION TO AID STEREOSCOPIC IMAGE FORENSICS

by

**Mark-Anthony Fouché**

**Supervisor:** Prof MS Olivier

**Department:** Computer Science

**University:** University of Pretoria

**Degree:** Master of Science (Computer Science)

**Keywords:** Stereoscopic Image, Stereo 3D, Image Forensics, Splicing, Depth, Disparity Map, Forgery Detection

With the advances in image manipulation software, it has become easier to manipulate digital images. These manipulations can be used to increase image quality, but can also be used to depict a scene that never occurred. One of the purposes of digital image forensics is to identify such manipulations. There is however a lack of research on the detection of manipulated stereoscopic images. Stereoscopic images are images which create an illusion of depth for the viewer by showing an image pair that correlates to a person's left and right eye.

This dissertation investigates how depth information can be used to detect stereoscopic image manipulations. Two techniques were developed and tested through experimentation.

The first technique used disparity maps to highlight large areas without internal depth. These areas can be the product of non-stereoscopic to stereoscopic splicing techniques. Experimentation results showed that areas without internal depth can be detected. However, the detected areas can be the product of natural occurrences in images and not only non-stereoscopic to stereoscopic splicing. Post investigation of detected areas is thus required to verify the results.

© University of Pretoria

The second technique used a derived formula to determine the distance an area will lose internal depth. Experimentation results showed that the formula is fairly accurate. This information can be used to aid the detection of internal depth inconsistencies in stereoscopic images. These inconsistencies can arise due to stereoscopic image splicing or other image manipulation techniques that may modify the internal depth of a stereoscopic image.

© University of Pretoria

# TABLE OF CONTENTS

© University of Pretoria

© University of Pretoria

© University of Pretoria

# LIST OF FIGURES

© University of Pretoria

# LIST OF TABLES

© University of Pretoria

# LIST OF ABBREVIATIONS

| | |
|---|---|
| **2D** | Two Dimensional |
| **3D** | Three Dimensional |
| **ADVENT** (project) | All-Digital Video Encoding Network, and Transmission |
| **APPn** | Application n |
| **CCD** | Charged-Coupled Device |
| **CFA** | Colour Filter Array |
| **CGI** | Computer-Generated Imagery |
| **CMOS** | Complementary Metal-Oxide-Semiconductor |
| **CMYK** | Cyan, Magenta, Yellow and Key |
| **CS5.1** | Creative Suite 5.1 |
| **DCT** | Discrete Cosine Transformation |
| **DSLR** | Digital Single Lens Reflex |
| **EOI** | End of Image |
| **EOF** | End of File |
| **EXIF** | Exchangeable Image File |
| **FMT** | Furier-Mellin Transform |
| **FPN** | Fixed Pattern Noise |
| **GIF** | Graphics Interchange Format |
| **HCF** | Histogram Characteristic Function |
| **HDR** | High Dynamic Range |
| **HSL** | Hue, Saturation and Lightness |
| **HSV** | Hue, Saturation and Value |
| **HTC** (company) | High Tech Computer |
| **IFD** | Image File Directories |
| **JPEG** | Joint Photographic Experts Group |
| **KMQ** (company) | Koschnitzke, Mehnert and Quick |

© University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

| | |
|---|---|
| **LG** (company) | Lucky-Goldstar |
| **LSB** | Least Significant Bit |
| **MATLAB** (software) | Matrix Laboratory |
| **MD5** | Message Digest 5 |
| **MP** | Multi Picture |
| **MPO** | Multi Picture Object |
| **PNU** | Pixel Non-Uniformity |
| **PoV** | Pairs of Values |
| **PRNU** | Photo-Response Non-Uniformity |
| **PVD** | Pixel Value Difference |
| **QT** | Quantization Table |
| **RGB** | Red, Green and Blue |
| **SHA-1** | Secure Hashing Algorithm 1 |
| **SIFT** | Scale Invariant Feature Transformation |
| **SOI** | Start of Image |
| **SPOT** (satellite) | Satellite Probatoire d'Observation de la Terre |
| **SSD** | Sum of Squared Differences |
| **STEREO** (spacecraft) | Solar TErrestrial RElations Observatory |
| **SVM** | Support Vector Machine |
| **TPM** | Transition Probability Matrix |
| **TV** | Television |
| **YASS** | Yet Another Steganographic Scheme |
| **ZDP** | Zero Disparity Plane |

© University of Pretoria

# CHAPTER 1 - INTRODUCTION

## 1.1 INTRODUCTION

Advances in image manipulation software have made it easy for users to manipulate digital images. Image manipulation can be used to improve image quality, but can also be used to depict a scene that never occurred. This creates a problem when digital images are used for media, scientific research or as evidence in a court of law. Digital image forensics is a field that explores this problem. By searching for inconsistencies in manipulated images, it is possible to determine whether an image has been manipulated. Inconsistencies include those found in lighting [1], shadows [2], geometric shapes [3], colour [4], focal blur [5] and compression [6]. The amount of existing research in digital image forensics can be seen from multiple surveys that have been compiled on the topic [7][8][9][10][11]. There is however a lack of research in detecting inconsistencies which are unique to stereoscopic images.

A stereoscopic image is an image pair, which is often referred to as a 3D image. This image pair is created using the same scene but at slightly different viewpoints. These viewpoints correlate to a person’s left and right eye. Stereoscopic images are usually viewed by separating the image pair for the user’s eyes. Viewing stereoscopic images this way, creates the illusion of seeing a scene in 3D instead of a scene projected onto a flat surface. Stereoscopic images have become more convenient to create and view with the creation of autostereoscopic displays. There are existing digital cameras [12], digital video cameras [13], a mobile game console [14], mobile phones [15] and even a tablet [16], which allow for the creation and viewing of stereoscopic images. Stereoscopic images are widely used. Applications for stereoscopic images include entertainment [17], education [18], medical sciences [19], forensic sciences [20], astronomy [21], robotics [22] and advertisements [23].

© University of Pretoria

Stereoscopic images can be manipulated in a similar way as non-stereoscopic images. These manipulations can cause the same image inconsistencies found in normal manipulated images, which allow normal digital image forensics techniques to be used. Since stereoscopic images contain more information about a scene, it is possible that additional inconsistencies can arise from image manipulations. The more inconsistencies that can arise from image manipulations, the more detection techniques can be developed and the harder it becomes to create undetectable image forgeries.

Two techniques are produced in this dissertation, which try to use stereoscopic depth information to aid the detection of stereoscopic image manipulation. One uses disparity maps, while the other uses triangulation.

## 1.2 PROBLEM STATEMENT

This dissertation investigates how stereoscopic depth information can be used to aid the detection of stereoscopic image manipulations.

Stereoscopic depth information can be obtained from image disparity. Image disparity is the difference of a point in the left and right stereoscopic image parts. This disparity can be used to retrieve relative depth information in the form of disparity maps. If camera information is available, image disparity can be used to get precise depth information with the use of triangulation. Both disparity maps and triangulation will be looked at in order to detect stereoscopic image manipulations.

There are many image manipulation techniques, but this dissertation will focus on image splicing. Image splicing combines two or more images to create a composite image. An example of image splicing is copying the head of one person onto the torso of another.

© University of Pretoria

Two stereoscopic image manipulation detection techniques are created and tested. The first technique uses disparity maps to highlight areas without internal depth. Objects which are copied from non-stereoscopic images do not have internal depth. By detecting areas without internal depth it is possible to detect these objects. The second approach uses triangulation to determine the distance at which objects loses internal depth. Objects beyond a certain distance from the camera do not have internal depth, since these objects are represented by a limited amount of pixels. If an object is copied from a distance where it should have internal depth to a distance where it should not, it is possible that the object will still have internal depth and vice versa. The next section explains the experiment design of these two techniques.

## 1.3 METHODOLOGY

The first technique is tested with an experiment on a set of 50 non-stereoscopic to stereoscopic spliced images. A mask is created that shows which area has been copied. For each image, a disparity map is generated and an algorithm is used to highlight large areas with no internal depth. The detected areas are compared with the mask and the correct an incorrect amount of detected pixels are investigated.

The second technique is also tested with an experiment. Using an existing formula for calculating the distance of objects, a new formula is derived to determine the distance at which an object loses internal depth. A set of stereoscopic images are captured with objects at different distances. The internal depth of the objects are calculated and compared with the derived formula, to confirm the correctness of the formula. An object with internal depth is then copied, scaled and moved to a position where it should not have internal depth and vice versa. The change in internal depth information is then investigated.

© University of Pretoria

## 1.4 STRUCTURE OF DISSERTATION

This dissertation is structured as follows: Chapter 2 provides an overview of digital image forensics and the research that has been done in the field. Chapter 3 provides information on stereoscopic images, to provide a better understanding of the topic. The connection between stereoscopic images and digital image forensics is also looked at in Chapter 3. Chapter 4 provides detailed information, results and discussion of the first detection technique tested. Chapter 5 provides detailed information, results and discussion of the second detection technique tested. Chapter 6 provides the conclusion for this dissertation.

© University of Pretoria

## Comics by MFouché

Your dissertation looks good so far, but it seems a bit short.
I can fix that. I'll use max font size, tons of pictures and comics.
Comics?
Comics will give me an excuse to use the world's most hated font: Comic Sans!

If this dissertation was about encryption, I would have used Wingdings.

© University of Pretoria

# CHAPTER 2 - DIGITAL IMAGE FORENSICS

## 2.1 INTRODUCTION

Digital images are widely used in today's society in both printed and digital form. With recent developments in digital imaging software, it has become easier for anyone to modify these images. These modifications can be used to enhance the quality of an image, but it can also be used to depict a scene which never occurred. It has been shown that image manipulation of past public events can influence memory, attitudes and behavioural intentions [24]. Image manipulation can be seen as a problem for the media, law-enforcement and scientific research.

The media has a strong influence on the general public, since it influences ideologies [25]. Magazine advertisements sometimes contain manipulated images of models that make them appear thinner [26]. These types of manipulated images have aided the ideology that a person needs to be thin in order to be attractive. Exposure to this ideology has been linked to psychological responses such as depression and eating disorders [27]. In March 2012, an Israeli law was created that bans the showing of overly thin models in local advertisements [28]. This new law also requires publishers to disclose when images have been manipulated to make models appear thinner.

Another issue that involves the law and manipulated images is child pornography. In 2006, Stafford Sven Tudor-Miles scanned photographs of adult porn stars and manipulated the images to make them look like young girls [29]. He was charged and found guilty of the possession of indecent pseudo-images of children, which is illegal in the UK.

© University of Pretoria

Political parties often use the media to aid their political campaigns. Image manipulation of political candidates has been shown to influence voters' preference [30]. Another politically related incident was the announced death of Osama bin Laden in 2011. Two images have been combined to create a composite image that supposedly showed his dead body. Only after it has been published in multiple newspapers, was it discovered that the image was actually a composite of two different images [31]. Neither of these two images confirmed the death of Osama bin Laden.

Scientific research results are sometimes displayed with the use of images. Manipulating these images to fabricate or change results is a form of scientific fraud. An example can be found in the 2005 scientific article by Hwang et al. [32], which has been retracted by the publisher. In this article, images of 11 stem cells have been manipulated to fabricate positive results [33].

As seen in this section, there are many reasons why it can be useful to be able to determine whether an image really depicts the scene it claims to depict. Digital image forensics is a field which deals with the authenticity and credibility of digital images. This includes finding the device that produced the image and determining what types of modifications have been made to a digital image. This chapter provides an overview of some of the research done in digital image forensics. Section 2.2 discusses different techniques to determine the device that produced the image. Section 2.3 looks at various image manipulation techniques and methods to detect these techniques. Section 2.4 discusses techniques used to hide information in digital images as well as ways to determine whether an image has hidden data.

© University of Pretoria

## 2.2 IMAGE SOURCE IDENTIFICATION

There are different ways to create digital images. Images can be captured with a digital camera, scanned with a scanner, printed or even partially or fully created with the use of CGI (Computer-generated imagery). Knowing how an image was created can help determine the authenticity of the image. This is known as image source identification and will be looked at in this section.

### 2.2.1 SOURCE CAMERA IDENTIFICATION

There are a wide variety of digital cameras on the market. Most mobile devices also contain a digital camera. Before looking at the different methods of determining the source camera, it is important to know how images are captured. The image capturing process can be described as a pipeline [11]. Light enters through the camera lens. Some cameras then contain an extra filter, such as a filter that removes infrared or ultra-violet light. Colour cameras then contain a CFA (Colour Filter Array), such as a RBG Bayer Pattern filter or a CMYK subtractive colour filter. After the light has passed through the filters, it is captured with a sensor. Examples of sensors include a silicon CCD and CMOS. The captured light is then processed with the use of algorithms and hardware within the camera. Each of these steps adds anomalies to the image, which can be used to aid source camera identification.

The goal of source camera identification is to determine which camera captured an image. Camera source identification can be divided into device class identification and specific device identification. With device class identification, the model and manufacturer of the device is determined. If there are two cameras from the same type and manufacturer, specific device identification is used to determine which of these two devices captured a specific image.

© University of Pretoria

## DEVICE CLASS IDENTIFICATION

Most cameras use an EXIF header for images. In these headers information is stored that can be used to identify the device type that captured the image. These headers can however be maliciously modified and might not be available if the image is resaved or recompressed [34].

Watermarks can be used to store camera information in an image so that the device can later be identified. A watermark is information that can be imbedded in the image itself instead of the header. Watermarks are mostly designed to protect copyright and withstand attacks aimed to remove or destroy the watermark [35]. Watermarks can also be used to determine whether an image has been modified [36]. Any modifications applied to an image before the watermark has been added, will not be detected. The watermark thus needs to be added when the image is captured. There is however only a few cameras that contain a watermarking chip for this purpose. This method is thus more suited for cameras used to capture forensic evidence [37].

Another way to identify the type of camera used is to use a machine learning algorithm. Different image characteristics caused by cameras can be used to train an algorithm for classification. Kharrazi et al. [38] proposed 34 image properties that can be used for classification. These properties are based on the CFA configuration and demosaicing algorithm, as well as the colour processing and transformation performed by the camera. Choi et al. [39] used intrinsic lens radial distortions to aid classification. In the majority of cameras, manufacturers use spherical surfaces for lenses, to reduce manufacturing costs. These lenses have radial distortions, which can be used as a property in a machine learning algorithm for device class identification. Choi et al. used radial distortions to compliment the 34 properties of Kharrazi et al. Another detection method that uses machine learning was proposed by Long and Huang [40]. They used the inter-pixel correlation information, caused

© University of Pretoria

by the demosaicing of the CFA, to train an artificial neural network. Filler et al. [41] showed that sensor pattern noise, which is used in specific device identification, can be used in device class identification.

In general, the machine learning approach does have a flaw. Since manufactures sometimes use the same components in different cameras, some machine learning techniques will be limited in the number of devices that can be differentiated [11].

## SPECIFIC DEVICE IDENTIFICATION

Like device class identification, image headers and watermarks can be used for specific device identification. Information like serial numbers can be stored in one of these ways. Unfortunately, headers and watermarks have the same problems as identified with device class identification. Headers can easily be removed and watermarks are not commonly used in devices. Another approach is to identify abnormalities introduced in images, which is unique to each device.

During the manufacturing process of CCDs, different defects can be introduced. Geradts et al. [42] showed that these defects can be used to aid specific device identification. These include hot point defects, dead pixels, pixel traps and cluster defects as defined by manufacturers. Hot point defects are pixels with very high output voltages. Dead pixels are pixels with poor response to light, which causes dots on a captured image. Pixel traps is a problem with the charge transfer, which can cause a partial or complete line of white or dark pixels. Cluster defects are clusters of pixels that generally differ ±6-20% in responsiveness, in comparison to the mean value of all the pixels in the neighbourhood. The deciding numbers depend on the manufacturer. There are potential problems with using these defects. More expensive CCDs have fewer defects and some cameras remove these defects by post processing of images.

© University of Pretoria

Image sensor defects also introduce pattern noise in captured image. Two main components of pattern noise are FPN (fixed pattern noise) and PRNU (photo-response non-uniformity) noise. FPN is caused by dark currents, which primarily refer to the pixel-to-pixel difference when sensors are not exposed to light. This noise can be compensated for in cameras by subtracting a dark frame from every image that is captured [43]. PRNU is primarily caused by PNU (pixel non-uniformity), which is the non-uniformity of pixel sensitivity during the sensor manufacturing process. Even sensors of the same silicon wafer differ, because of PNU. Other factors of PRNU noise are low-frequency defects, like dust particles on sensors and zoom settings. Lukas et al. [44] proposed a method to use pattern noise to create a type of fingerprint for sensors, which can be used for specific device identification. A reference pattern noise was created for multiple cameras by averaging the noise obtained from multiple images, with the use of a denoising filter. Each reference pattern noise was then correlated to the noise residual of an image to identify the camera used. Goljan and Fridrich [45] created a more general approach to use pattern noise, by allowing images to be scaled and cropped. To show the viability of using pattern noise for specific camera detection, Goldan et al. [46] performed a large scaled test. In this test more than one million images were used, spanning 6896 individual cameras covering 150 models. Results showed a false rejection rate of less than 0.0238% and a false acceptance rate below $2.4 \times 10^{-5}$ %. The most important factor that contributed to missed detections was caused by the quality of the images used for fingerprint estimation.

Dust particles on its own can be used for specific device identification. DSLR (digital single lens reflex) cameras have interchangeable lenses. When these lenses are changed, dust particles can enter the film chamber, stick to the rear of the lens and often make their way to the camera sensor. When images are captured with the camera, the dust particles leave artefacts on the image. Dirik et al. [47] proposed a method for DSLR camera identification based on sensor dust characteristics. Their

© University of Pretoria

approach detects dust particles on an image, using intensity variations and shape features, to form the dust pattern of the DSLR camera. Olivier [48] proposed a matching techniques that uses grid overlay instead. Olivier also investigated several factors that affect the rendering of dust particles on images. These factors include different focal lengths, sensor speeds and apertures. Results showed promise for using the technique for specific device identification.

### 2.2.2 SCANNER SOURCE IDENTIFICATION

Similar to cameras, scanners can also capture images. Photographs captured on film, as well as documents, can be converted in this manner to digital images. Some of the techniques used in camera source identification can also be used for scanner source identification.

As mentioned in the previous subsection, sensor pattern noise can be used to identify specific cameras. Similar techniques can be used for scanners, since scanners also use sensors to capture images. Gou et al. [49] used statistical noise features to determine the brand and model of a set of scanned images. These statistical noise features include denoising algorithms, wavelet analysis and neighbourhood prediction. Khanna et al. [50] showed that improving the feature selecting process for denoising filters, accuracy of scanner source detection can be improved. Their results showed that specific scanners can be identified from scanned images at native resolution. For images scanned at lower resolutions, the make and model of scanners can be identified. Even images that have undergone JPEG compression with low quality factors, image sharpening and contrast stretching, performed well.

Using sensor pattern noise is not the only method used for scanner source detection. Dirik et al. [51] used traces of dust, dirt and scratches on scanner platens. These platen impurities are captured on scanned images and are also referred to as

© University of Pretoria

“trash marks”. It is shown that it is possible to identify the specific scanner that scanned an image, by comparing the scanner platens to the platen impurities found in scanned images.

### 2.2.3 COMPUTER GENERATED IMAGES

Images can be partially or fully generated with a computer application instead of being scanned or captured by a camera. Advances in computer generated images make it close to impossible for the human eye to distinguish natural photographic images from photorealistic computer generated images. Figure 1 shows an example of how realistic photorealistic computer generated objects can be. The image fragment is from a scene in the 2012 film Life of Pi and shows a digitally created Bengal tiger [52]. Techniques for detecting computer generated images can be categorized as using statistic wavelet features, physical models of images, or camera related characteristics [53].

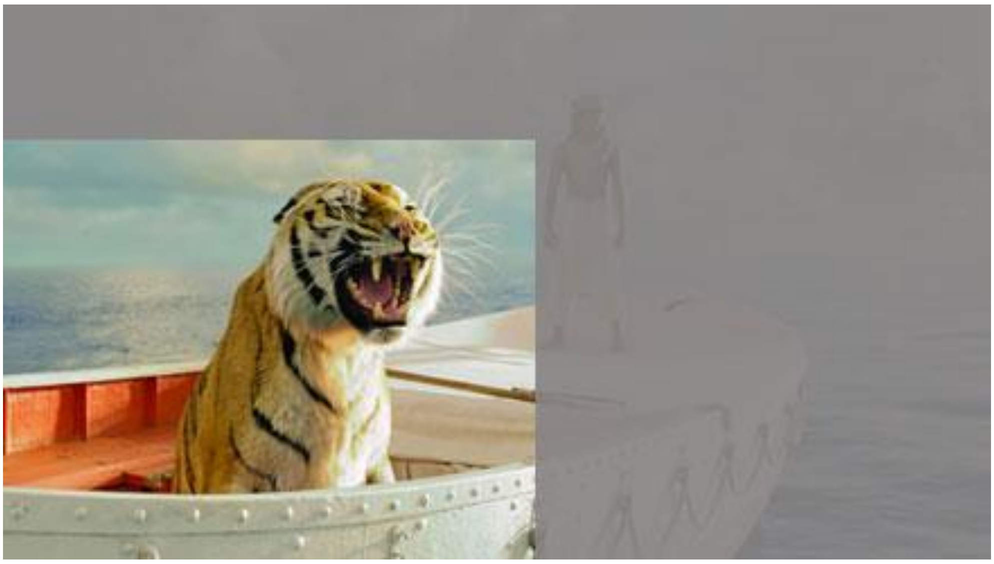


**Figure 1 - Photorealistic Computer Generated Bengal Tiger from Life of Pi**

Photographic and computer generated images have different wavelet statistics which can be used to distinguish the two from each other. Lyu and Farid [54] described a statistical model for photographic images that consists of first-order and

© University of Pretoria

higher order wavelet statistics. For feature extraction, the images are decomposed into multiple scales and orientations. For colour images, the decomposition is applied to each colour channel (RGB) independently. For the statistical model, two sets of statistics are used. The first set is the first order statistics (mean, variance, skewness, and kurtosis) of the sub-band coefficient histogram at each orientation, scale and colour channel. The second set is based on the errors in a linear predictor of coefficient magnitude. These statistic sets are used in a SVM classifier to differentiate photographic images from computer generated ones. The model was tested on a database of 40 000 photographic and 6000 computer generated images. Approximately 67% of the computer generated images were correctly classified, while only 1% of the photographic images were misclassified. The rest of the images could not be classified. Wang and Moulin [55] also used wavelet statistics, but improved the classifier to only require half of the feature extraction and testing time. Chen et al. [56] showed that classification using wavelet statistic can be improved by extracting features from the HSV colour space instead of the RGB colour space.

There are physical differences between photographic images and computer generated images that can be used to distinguish the two. Ng et al. [57] proposed five different feature sets that uses some of these physical properties for classification. The first feature set is local fractal dimensions, which captures the surface complexity of objects. Real-world surfaces are generally more complex than computer-generated ones. The second feature set is local patch vectors, which captures characteristics of local edge profiles. The third feature set is surface gradient, which is influenced by gamma correction. Photographic images often go through gamma correction to enhance contrast, which is not needed for computer-generated images. The fourth feature set is the second fundamental form, which uses the geometric properties of objects. Computer-generated images use polygons to represent objects. These polygons can cause sharper edges and corners than

© University of Pretoria

those found in photographic images. The fifth and last feature set is Beltrami flow vectors, which captures artefacts cause by the colour independence assumption in computer-graphics. Colour independence occurs when the light reflection properties are simplified for computer-generated objects. Ng et al. [57] used the previously mentioned five feature sets as input for a SVM classifier to achieve 83.5% accuracy on a set of 3200 images.

Camera characteristics used in device identification can be used to distinguish photographic from computer generated images. Dirik et al. [58] proposed using traces of two different properties to enhance detection accuracy, when detecting computer generated images. The first property is demosaicing. The colour filter array found in most cameras requires a demosaicing algorithm. This demosaicing algorithm leaves a Bayer pattern on images that are not present in computer generated images. Dirik et al. [58] showed 98.1% accuracy on a dataset of 3600 images, when exploiting the demosaicing property. Gallagher and Chen [59] further exploited the demosaicing property by considering the weighted linear combination of neighbouring pixels. Gallagher and Chen tested their approach on Columbia's ADVENT dataset [60] and achieved 98.4% accuracy. Using the demosaicing property relies on the quality of the images. Classification accuracy decreases with higher compressions. The second property proposed by Dirik et al. [59] is chromatic aberration. Chromatic aberration is anomalous colour shifts caused by variations in the optical glass formulation of camera lenses. Computer generated images do not require an optical lens and will not show traces of chromatic aberration. Dirik et al. [58] achieved 99.6% accuracy on a dataset of 3600 images. Unlike the demosaicing property, detection accuracy of chromatic aberration is consistent over a wide range of image compression values.

Techniques for generating photorealistic images are constantly improving. The image classification techniques mentioned in this subsection might not perform so

© University of Pretoria

well on more modern images. Images such as Figure 1, which are a mixture of photography and computer generated images, also causes problems for classifiers [53]. There are also anti-forensic techniques which can confuse classifiers even more. Anti-forensic techniques evolve as forensic techniques improve. Different anti-forensic techniques are discussed in the next subsection.

### 2.2.4 SOURCE IDENTIFICATION ANTI-FORENSICS

In digital forensics, there are attackers who will try to counter digital forensic techniques. For source identification, these attackers can try to change the detected image capturing device in order to frame someone. Anti-forensic techniques include changing statistical properties of images and re-imaging images.

Statistical properties of images can be used for camera source identification. As discussed in subsection 2.2.1, Lukas et al. [44] proposed a method to use pattern noise to create a type of fingerprint for camera sensors. This sensor fingerprint can be used for specific device identification. Gloe et al. [61] investigated an anti-forensic technique that uses inverse flat fielding to change the sensor finger prints of images. Flat fielding is a method used to minimize pattern noise of images and is typically used in astronomy or in flatbed scanners to enhance image quality [44]. Once an image's noise pattern has been suppressed by flat fielding, a different pattern noise can be added to the image with inverse flat bedding. The noise pattern added to an image can be an approximation of flat field frames from multiple images of a targeted device. Gloe et al. [61] showed that inverse flat bedding can be used to successfully forge the camera fingerprint of images. This technique can also be used to add pattern noise to computer generated images, to add confusion to photographic and computer generated image classifiers. Goljan et al. [62] discovered that planting a sensor fingerprint with inverse flat fielding will leave a trace. Planting a sensor fingerprint without a trace is thus more difficult than previously thought.

© University of Pretoria

Many source identification techniques use properties unique to image capturing devices for device identification. An anti-forensic technique which exploits this fact, is the re-imaging of images. A simple re-imaging technique is to take a photograph of an existing image. This technique can be used to confuse classifiers for photographic and computer generated images. Ng et al. [57] showed that re-imaging can be partially addressed by adding re-imaged images in the dataset used to train classifiers. Yu et al. [63] showed that the specularity of images can be used to detect re-photographed images. Specularity of an object is the brightest point on that object, caused by the direct reflection of a light source towards the viewer. Yu et al. [63] discovered that the specular ratio’s gradient distribution in natural images was found to be Laplacian-like, while that of re-captured images is Rayleigh-like. The Rayleigh-like distribution is caused by the mesostructure of the photo surface.

## 2.3 Detecting Manipulated Images

Advances in image manipulation software have made it easy for users to manipulate digital images. These manipulations can be used to enhance image quality or depict a scene that never occurred. The latter is important when the image is used as evidence in a court of law. In this section different image manipulation techniques are discussed as well as techniques which can detect manipulated images.

### 2.3.1 Image Manipulation Techniques

Recent image manipulation software provides many tools for digital image manipulation. These tools can be used to enhance the quality of images or modify the image to depict a scene that never occurred. Multiple image manipulation techniques can be used on a single image to enhance the end result. The author used Adobe Photoshop CS5.1 at time of writing, but there are free alternatives like Gimp. In Photoshop CS5.1 image manipulation techniques include transformations, colour manipulation, image filters, cloning and splicing. Text, drawings and images

© University of Pretoria

can also be combined to create a composite image. Examples of such a composite images can be found on magazine covers.

Transformation techniques modify the structure of an image or image part. These modifications include scaling, rotating, skewing, distorting, warping and flipping. Photoshop CS5.1 has an advance transformation technique called "Puppet Warp", which is similar to warping but allows more control of warped areas. Figure 2 shows a few transformation techniques.

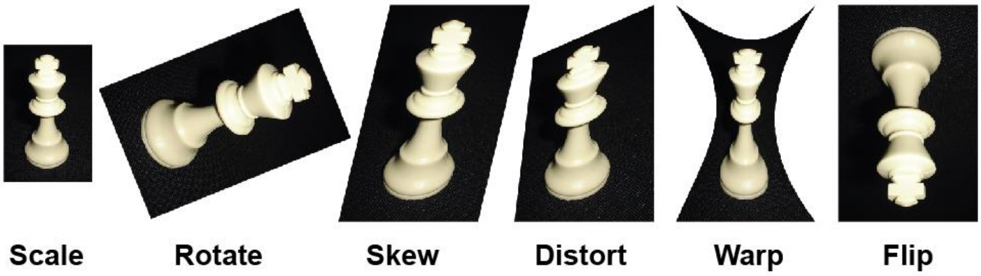


**Figure 2 - Examples of Image Transformations**

For colour manipulation, different colour models can be used. Digital images consist of pixels. One way to represent these pixels is with the red, green and blue (RGB) colour model that corresponds to the primary colours of light. A computer monitor pixel displays the RGB coloured lights separately, but the viewer sees the combination of the lights. Viewing a white monitor pixel through a magnifying glass can reveal the three separate colours. By modifying the light intensities of the RGB values, different colours are perceived by a viewer. If a pixel has an RGB value of (0, 0, 0) it will appear black, since there will be no light. If we assume the RGB values are stored in three bytes, a pixel with an RGB value of (255, 255, 255) will appear white. The RGB colour model is ideal for displaying images on a monitor, but makes colour modification difficult. It is hard to estimate which RGB values are needed to create a certain shade of colour. It is also hard to estimate the RGB values of a given colour. For this reason image manipulation software may also support a different colour model, like one that makes use of hue, saturation, and value (HSV) or

© University of Pretoria

lightness (HSL) [64]. HSV and HSL models represent colour in a similar way an artist mixes paint. Hue is one or a mixture of two primary colours. Saturation is the amount of white added to the colour. Value or lightness is the amount of black added to the colour. In order to display pixels using HSV or HSL, it is first converted to the RGB colour model.

There are a variety of algorithms and techniques to manipulate an image's colour. Adjusting the brightness and contrast of an image is used in photography to enhance image quality. Brightness modifies how close a colour is to black, while contrast determines the colour range. Desaturation converts the saturation of image colours to zero, resulting in a black and white image. Changing the hue and saturation of colours makes it possible to change the colour of objects in an image. If a RGB colour model is used, the colours can be inverted by subtracting each colour value from the maximum allowed value. Colours in an image can also be completely replaced by a gradient by desaturating the image and choosing a gradient to replace values from white to black. Figure 3 shows a few colour manipulation techniques.

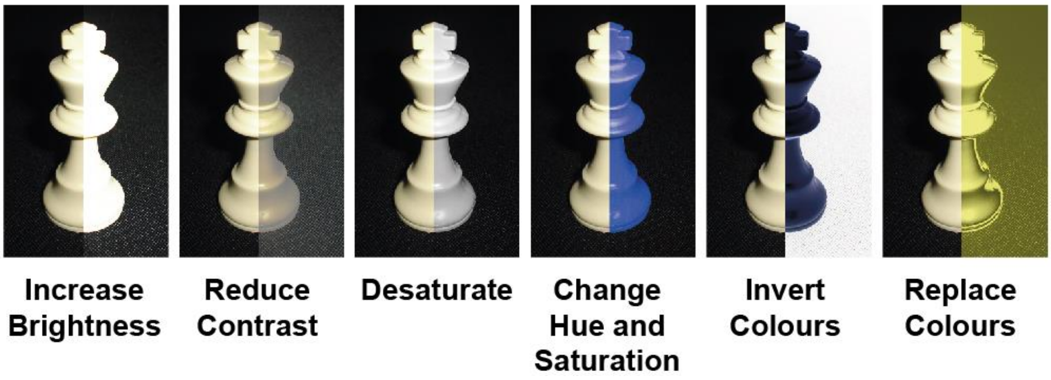


**Figure 3 - Examples of Image Colour Modifications**

Image filter algorithms are similar to colour manipulation algorithms, but use additional information like neighbouring pixels. It is convenient to first convert an image to signals (frequency domain) in order to apply filters. Some applications of

© University of Pretoria

filters include image enhancements, changing the appearance of images, and highlighting certain properties of an image for image tampering detection. Figure 4 shows a few examples of filters found in Photoshop CS5.1.

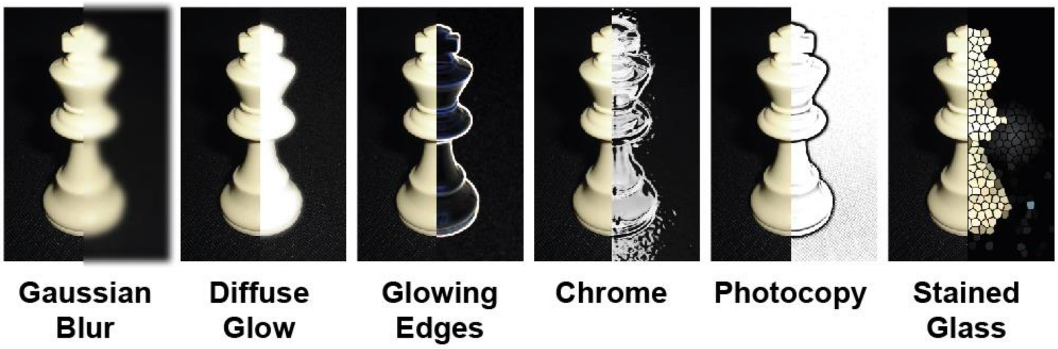


**Figure 4 - Examples of Image Filters**

Another image manipulation technique is cloning, also referred to as copy-move. Cloning is a technique that copies one part of an image to another part. If you have a photo of a poster on a wall, you can copy the parts of the wall onto the poster in order to hide the poster. Photoshop CS 5.1 has a tool called "content aware fill" which covers a selected area with parts of the given image, similar to cloning. Applications for cloning include the removal of unwanted artefacts, filling an area which has been deleted or simply duplicating objects in a scene. Figure 5 shows an example of how cloning is used to remove written information from a message.

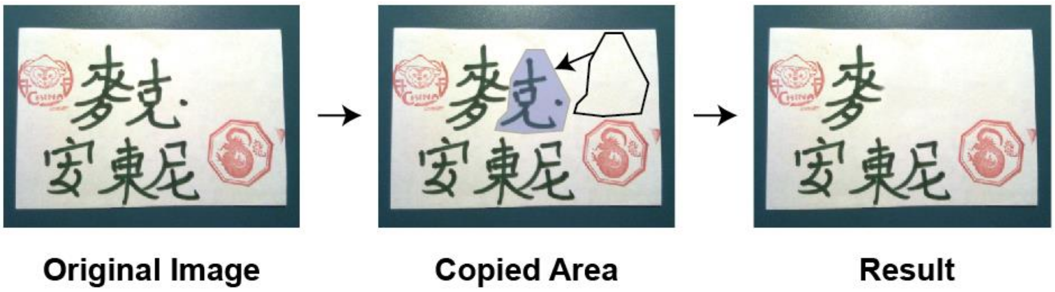


**Figure 5 - Example of Image Cloning**

© University of Pretoria

Splicing is an image manipulation technique where two or more images are combined to create a composite image. Splicing can be used to depict a scene that never occurred. Figure 6 shows an example of how the image of Figure 2 and Figure 5 are spliced to create a composite image. Image 2 is first slightly distorted and its colour manipulated, before it is added to Image 1. A shadow is added to the composite result by manipulating the colour at the base of the chess piece.

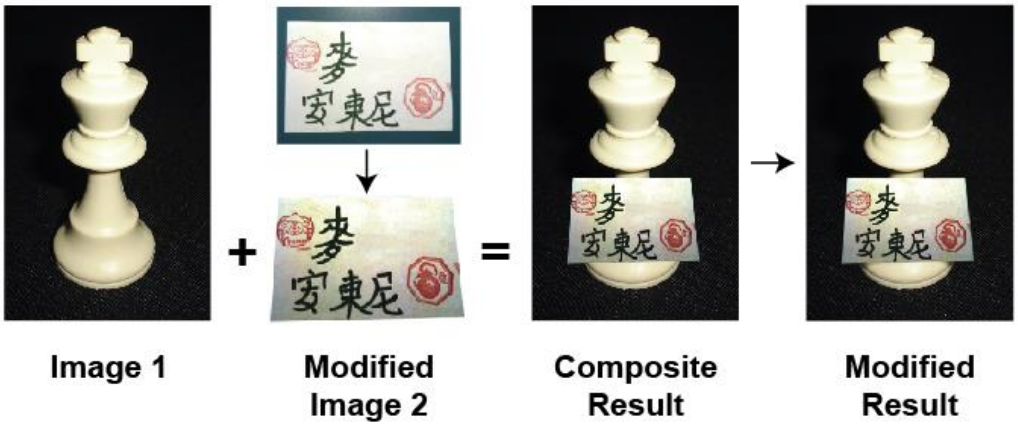


**Figure 6 - Example of Image Splicing**

As seen in Figure 6 it is relatively easy to manipulate images to depict a scene that never occurred. The next subsection discusses different techniques to detect manipulated images.

## 2.3.2 IMAGE MANIPULATION DETECTION

Methods used to detect manipulated images can be classified as informed and blind. Informed methods require access to the original image before modifications have been made. Blind methods do not require access to the original image. This subsection discusses a few active and passive methods that can be used to detect image manipulations.

© University of Pretoria

### INFORMED DETECTION METHODS

Informed detection methods require access to the original image in order to detect modifications. Informed detection methods include hash-based and watermark-based methods.

A hash function is an algorithm that maps data of variable length to data of fixed length called a signature. It is difficult to modify the original data so that the original and modified versions produce the same signature, which is why it can be used to aid data authentication. It is also possible to create multiple signatures with different hash functions to strengthen authentication. Cryptographic hash functions like MD5 or SHA-1 are used for message authentication, but can also be used to authenticate other digital information, such as images [65]. If the original image signature is available, it can be compared with the signature of an image in question to determine whether it has undergone any changes. Some image modifications do not change the content of an image, but will still cause a signature to change. Venkatesan et al. [66] introduced an image hashing method that is designed to be robust against image changes due to compression, geometric distortions and other content-preserving manipulations. If an image is detected as being manipulated, it is sometimes required to know which part of the image has been manipulated. Roy and Sun [67] presented an image hashing method which not only detect, but also localize tampering with the use of a small signature. A problem with hash-based methods is that the signature is designed to be stored and transferred separately from the image through a secure channel. The signature can also be stored in the header of an image, but can be easily removed. Watermark-based methods are similar to hash-based methods, but embed information into the image instead.

A digital watermark is a distinct mark that is embedded into digital media to ensure integrity and authentication of images. Watermarks are typically used to identify ownership of copyright. A content-fragile watermark is a type of watermark which is

© University of Pretoria

designed to be robust to perceptually irrelevant information, while being fragile to perceptually significant modifications. Possible modifications can be detected and localized by observing any damages that have been done to a watermark. This can be done in multiple ways. Wu and Liu [68] embedded a watermark, as well as a set of simple features in the frequency domain of an image via table look-up. Fridrich [69] divided an image into blocks and watermarked each block independently. Kundur and Hatzinakos [70] embedded a watermark in the discrete wavelet domain of an image by quantizing the corresponding coefficients. The previously mentioned watermarking methods are either vulnerable to vector quantization counterfeiting attacks, or sacrifice localization accuracy to improve security. Celik et al. [71] addressed these problems by proposing a hierarchical watermark. The image is divided into blocks in a multilevel hierarchy. Block signatures are then calculated in this hierarchy. The lower level block signatures provide good localization accuracy, while higher level block signatures provide increased resistance to vector quantization attacks. General problems with watermarks include the distortion of original image content and the potential increase in bit-rate required to compress a watermarked image. Watermarks can also be tampered with, as shown by Cox and Linnartz [72].

Hash-based and watermark-based methods only ensure authentication from the time the methods have been used. If the image has been manipulated beforehand, it will not be detected. To detect image manipulation without prior knowledge of an image, blind detection methods can be used.

## BLIND DETECTION METHODS

Unlike informed detection methods, blind detection methods only have access to the image in question. In order to detect image modifications, blind methods search for inconsistencies caused by image modifications. Some inconsistencies can be

© University of Pretoria

found in duplicated areas, blurred areas, lighting, image compression, chromatic aberrations and geometry perspectives.

Duplicated areas in an image can indicate that an image has been manipulated. Cloning, as discussed in subsection 2.3.1, is an image manipulation technique that duplicates an area of an image to another part in the same image. This technique can be used to hide areas of an image. An exhaustive search can be used to detect duplicated areas, but is computationally expensive. Fridrich et al. [73] used a technique that tiles the image by overlapping blocks. The DCT (discrete cosine transformation) coefficients of the image blocks are compared to detect duplications. This technique is more efficient than an exhaustive search since the DCT coefficients are first lexicographically sorted. Mahdian et al. [74] used blur moment invariants to compare blocks. Blur moment invariants can be used on images with blur degradation, additional noise, or arbitrary contrast changes. Huang et al. [75] used SIFT (scale, invariant, feature and transform) descriptors from an image for comparisons. SIFT descriptors are invariant to image changes that include changes to illumination, rotation and scale. Bayram et al. [6] extracted features with FMT (Furier-Mellin Transform) to compare blocks. FMT features are invariant to scaling and translation. FMT features are also robust to lossy JPEG compression, blurring and noise addition. Bayram et al. [6] also showed that computational efficiency can be improved by using bloom filters instead of sorting blocks lexicographically. All of the above mentioned techniques can detect duplicate areas, but have high computation time, produce a high number of false positives and require human interpretation of results.

Images can have inconsistent blur characteristics if image splicing is used. Images at the same distance from the camera should have the same focal blur. Blurring is also used to conceal traces of image manipulation. By detecting inconsistent areas of blur on the same image, it is possible to detect manipulated images. Hsiao and Pei

© University of Pretoria

[76] used frequency domain information for blur estimation. The scheme used was shown to be powerful in blurred region detection for defocused images, as well as images that were manipulated with a blur filter. Wang et al. [77] used Elder and Zucker's [78] method to estimate the blurriness of chosen image patches and used a threshold to judge consistency. Cao et al. [5] used an edge-based blur estimator. The linear consistency of blurred edges is assessed based on linear fitting and a statistical metric. Kakar et al. [79] used motion blur estimations through image gradients to detect spliced areas. The motion blur technique was shown to provide better results in selecting regions with inconsistent blur, than other blur-based techniques. Blur-based detection techniques in general require human interpretation of result.

Lighting conditions between two images can vary significantly. Even images of the same scene, but with different camera locations, can have different lighting conditions. Determining the direction of the light sources in a scene can be used to detect image manipulations, if different objects appear to be illuminated by different lights. Nillius and Eklundh [1] presented a fully automatic algorithm for estimating the direction of a projected light source. First a heuristic algorithm uses colour and edge information to choose potential occluding contours. Secondly, a shading model is used to estimate the light direction of each contour. Finally a Bayesian network is used to estimate the final light direction from the first two steps. Johnson and Farid [80] used the specular highlights of people's eyes to determine the direction of a light source. Kee et al. [2] described a geometric technique to detect inconsistent shadows. This technique was designed for a scene with a single dominant light source, like an outdoor scene lit by the sun. Gholap and Bora [4] used the colour illumination of objects to detect colour inconsistencies. Colour inconsistencies can be caused by different light sources or image manipulation techniques. The advantage of lighting-based detection techniques is

© University of Pretoria

that it is difficult to hide inconsistencies in lighting. Like most of the other techniques, human interpretation of results is required.

The compression history of images can be used to detect manipulated images. When an image is saved as an image file type like JPEG, it gets compressed. Every time an image is manipulated and saved, another level of compression is applied. If part of an image is manipulated, that part will most likely have a single level of compression, while the rest of the image will have more levels of compression. Lukas and Fridrich [81] presented a solution to detect double compressed JPEGs by detecting features in the image's DCT histogram. The DCT histogram will show double compression if different quantization matrixes were used for each compression level. A neural network classifier was used to determine the primary quantization step, with less than 1% error rate. Popescu and Farid [82] used an expectation/maximization algorithm to detect statistical correlations that are caused by resampling images. The approach was shown to work on uncompressed TIFF, JPEG and GIF images. Chen et al. [83] used a machine learning based scheme to distinguish between single and double compressed JPEGs. Features were formulated from JPEG 2D arrays, with different magnitudes and directions. A threshold technique is then used to reduce the size of each TPM (transition probability matrix), which is characterized by the Markov process used. Elements of the TPMs are then used for the machine learning algorithm. Compression based detection techniques are good for detecting re-saved images. Unfortunately images are also resaved after image modifications that do not change the content of a scene.

Chromatic aberration occurs when an optical system fails to perfectly focus light from all wavelengths. Manipulated image parts can cause chromatic aberration inconsistencies, which can be used to detect image manipulation. Johnson and Farid [84] proposed a technique that uses lateral chromatic aberrations that are

© University of Pretoria

approximated with a low-parameter model. Their technique is based on maximizing the mutual information between colour channels, but claim that other correlation metrics may be equally effective. Chromatic aberration based detection methods work well with non-compressed non-uniform parts of an image. Weak results can however be expected for uniform regions of typical JPEG images.

When images are spliced together, the geometric perspective of image parts can be different. Even manipulated image parts of the same image can have different geometric perspectives. Johnson and Farid [3] reviewed three techniques which are used for metric measurements on planar surfaces. These techniques use polygons, vanishing lines or circles. The measurements can be used to detect projective geometric distortion caused by image manipulation techniques, which are difficult to conceal. This approach is unfortunately difficult to automate.

Image manipulation detection techniques have their weaknesses, which can be exploited. In the next subsection different image detection anti-forensic techniques are looked at.

### 2.3.3 Image Manipulation Anti-Forensics

Image anti-forensics manipulates the inconsistencies used for image manipulation detection. The goal of image anti-forensics is to try and create a perfect forgery. In this subsection a few of these are looked at.

Watermarks are created with the goal to authenticate images and potentially detect image manipulations. Watermarks can be made unusable with anti-forensic techniques. Voloshynovskiy et al. [85] classified watermark attacks into four categories. The first category is removal attacks, which try to remove the watermark from the image. These techniques try to remove the watermark without trying to crack the security of the watermark algorithm. Some of these techniques might not

© University of Pretoria

fully remove the watermark, but does significant damage to it. The second category is geometric attacks, which try to distort the watermark instead of trying to remove it. Some watermark detectors can only recover watermark information if perfect synchronization is possible. By modifying the geometry of the image, the detector will not be able to achieve synchronization. Some of the more recent watermark techniques use special synchronization techniques, which are not affected by geometric attacks. The third category is cryptographic attacks, which try to crack the security methods used in watermarks. By cracking the security method, it becomes easy to remove the watermark. The fourth category is protocol attacks, which try to attack the whole concept of the watermark application. An example is to try and estimate watermark data in an image and copy it to another image. This way a watermark can be faked even if the watermark algorithm, watermark technology or watermark key is unknown.

Some detection methods use inconsistencies that arise when image parts are resized or rotated. Kirchner and Bohme [86] presented three approaches which can be used to defeat the resampling detection method, like the one proposed by Popescu and Farid [82]. The best performance was achieved by the complementary methods that used resampling with edge-modulated geometric distortion and a duel-path approach. Two general conclusions were made by Kirchner and Bohme [86]. The first conclusion was that attacks which are integrated in the manipulation technique are more effective than post-processing attack techniques. The second conclusion is that downscaling and rotation modifications are easier to conceal than upscaling.

Searching for JPEG image compression inconsistencies is another detection method, which was discussed in the previous subsection. Stamm et al. [87] showed how proper addition of noise to an image's DCT coefficient can remove quantization artefacts. Quantization artefacts are used as indicators of JPEG compression. Results

© University of Pretoria

showed that this approach was able to fool a DCT quantization based artefact classifier more than 95% of the time.

This section looked at methods that manipulate an image's appearance, as well as techniques to detect such manipulations. There are other ways to manipulate images, which do not significantly change the image's appearance. These techniques are used to hide information for covert communication channels, instead of fabricating a scene that did not occur. Techniques to hide information in images and detect images with hidden information are also part of image forensics. These techniques are looked at in the next section.

## 2.4 STEGANOGRAPHY AND STEGANALYSIS

Steganography are techniques which are used to inconspicuously hide information in other information. The goal of steganography is to send a message without a third party knowing that a message has been sent. Images are ideal for hiding messages since images usually contain a lot of data which can be changed without changing the appearance of the image. An image used for steganography is called a cover image. The message itself is encrypted and then inserted into the cover image to create a stego image. The stego image is sent though a communication channel which can be intercepted by a third party. When the receiver receives the message, the message can be extracted by reversing the algorithm used to hide the message. Covert communication will be successful if a third party is unable to detect the hidden message. Techniques used to detect steganography are called steganalysis. Figure 7 illustrates the model for steganography and steganalysis [88].

© University of Pretoria

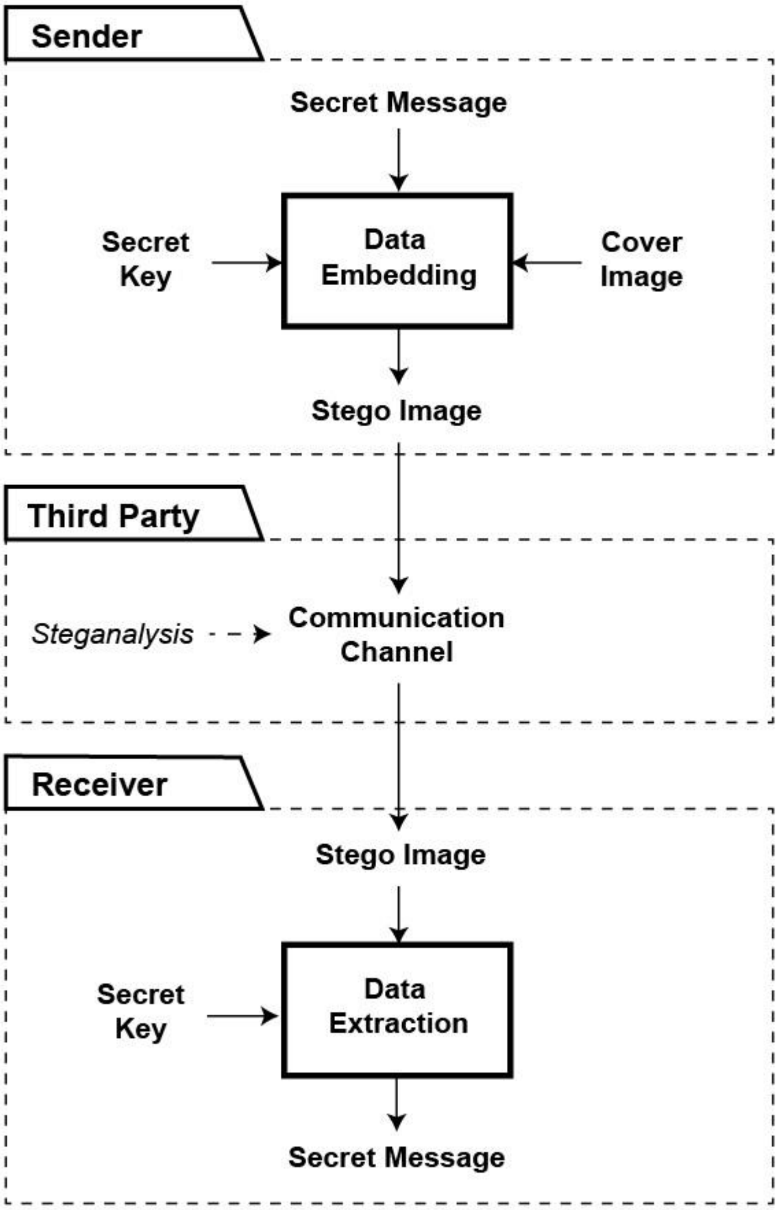


**Figure 7 - Model for Steganography and Steganalysis**

Steganography can be used to protect privacy but can also be used for illegal activities, like the distribution of illegal information. In this section different steganography and steganalysis techniques will be looked at, since it is also part of image forensics.

© University of Pretoria

### 2.4.1 STEGANOGRAPHY TECHNIQUES

There are different steganography techniques for hiding information in images. Steganography techniques can be evaluated on how hard it is to detect and remove, the amount of information which can be hidden, and whether the appearance of the image is changed in a visible way. Steganography techniques can be categorized into those which exploit an image's format, spatial domain, or frequency domain.

An image's file format has certain properties which can be exploited to embed information. Data can be appended at the end of the image file. This technique exploits the EOF (end of file) tag. When the image is opened in a viewer or image editor, everything after the EOF tag is ignored. Another place to insert data is in the EXIF data of images, which is used to store metadata of an image. These methods are simple and do not modify the appearance of the image in any way, but can be easily detected [89].

Spatial domain techniques modify the pixel values of an image. One method is to replace the LSBs (least significant bits) of predetermined pixels with hidden data. Modifying the LSB of a pixel, results in a small change in colour. Small changes in colour are hard to detect with the human eye. Increasing the amount of LSBs used per pixel increases the amount of data which can be hidden, but also increases the amount of image colour distortion that might be visible [89]. LSB matching is another technique that is a slight modification of LSB replacement. Instead of replacing the LSB, the LSB is increased, decreased or left the same depending on the data that should be hidden. Even pixel values are increased or left unchanged, while odd pixel values are decreased or left unchanged. Mielikainen [90] proposed a variation of this approach by using two pixels as a unit to hide information. The first pixels carries one bit while the combination of the two carries a second bit. This approach showed more resistance to steganalysis techniques. Wu and Tsai [91] proposed a PVD (pixel value difference) steganography technique that also uses

© University of Pretoria

pairs of pixels as a group to hide information. Data is hidden in the pixel difference of consecutive and non-overlapping pixel pairs. Some image formats, like GIFs, use a colour palette to restrict the amount of colours used in the image. If the LSBs are modified, too many colours might be required for the palette. Grey-scaled palette based images are thus more suitable for steganography, since the colour values differ slightly in the palette [92]. Fridrich and Du [93] proposed a palette based steganography technique that avoid areas of uniform colour and embed data into texture-rich portions of the image. Spatial domain techniques have weak resistance against steganalysis techniques, which led to the research of frequency domain techniques.

Frequency domain techniques use pixels in the frequency domain instead of the spatial domain. The frequency domain represents data as signals. The DCT (discrete cosine transformation) coefficients used for JPEG images is an example of frequency domain information. The compression of JPEG DCT coefficients is controlled by a QT (quantization table). Upham [94] created JSteg that embeds information in LSBs of non-zero quantized DCT coefficients. Westfeld [95] introduced the F5 algorithm, which does not embed information directly into the LSBs of the DCT coefficient. Instead, the F5 algorithm decreases the absolute values of the DCT coefficients by one if it needs to be modified. Chang et al. [96] proposed a steganography technique that first modifies the QT of a JPEG. Information is then embedded in the least two-significant bits of the quantized DCT coefficients, which are located in the middle-frequency part. Modifying the QT first, results in less distortion in the resulting JPEG image. Provos [97] created OutGuess-0.2 which uses two steps to embed information. First the message bits are embedded along a random walk into the LSBs of the quantized DCT coefficients, while skipping 0's and 1's. Coefficients that were not selected during embedding are then corrected, to make the global DCT histogram of the stego image match that of the cover image. This technique was designed to be resistant against statistical steganalysis techniques. Solanki et al.

© University of Pretoria

[98] proposed the YASS (Yet Another Steganographic Scheme) method, which does not embed data directly into the JPEG DCT coefficients. First the spatial representation of the image is divided into big blocks (B-blocks). An 8x8 block (H-block) is then randomly selected in each B-block, with a secret key, for performing DCT. Information is then encoded with error correction codes and is embedded in the DCT coefficients of the H-blocks. Lastly, inverse DCT is performed on the H-blocks and the whole image is compressed and distributed as a JPEG. YASS was shown to resist some blind steganalysis attacks and survive distortion constrained attacks.

Steganography techniques can leave traces, such as statistical anomalies, that make it possible to detect. Different detection techniques are looked at in the next subsection.

### 2.4.2 Steganalysis Techniques

Steganalysis techniques try to identify whether an image has hidden information or not and can be classified as targeted and blind techniques. Targeted techniques try to exploit a known steganography technique. Blind techniques try to identify whether any steganography technique was used, without any knowledge of the steganography algorithm that might have been used. This subsection will look at both.

#### Targeted Techniques

Targeted techniques try to exploit weaknesses in known steganographic algorithms. Some informed steganalysis techniques will be looked at for LSB embedding, LSB matching, PVD steganography, F5 algorithm, OutGuess and YASS.

LSB replacement replaces information in the least significant bits of an image. A common misconception of earlier LSB embedding techniques is that the LSB of an

© University of Pretoria

image is evenly distributed. Westfeld and Pfitzmann [99] proposed two steganalysis techniques that exploit this misconception. The first is a visual technique that creates a black and white image from the LSBs for the target image, where an LSB of 0 = white and LSB of 1 = black. A distinctive pattern could be observed in stego images, but the visibility of the pattern was determined by the original cover image. The second is a statistical technique called Chi-square ($\chi^2$). With this technique the frequencies of the PoV (Pair of Values) of pixel bytes are first calculated. PoV are bytes with different LBS, while the rest of the bits are similar. An example is 11111110 and 11111111. The PoV frequencies are then compared with a Chi-square test. If the PoV frequencies show a close to random distribution, there is a high probability that a message has been embedded. Fridrich et al. [100] proposed a quantitative analysis technique, which is known as RS analysis. Groups of neighbouring pixels are classified as regular (R), singular (S) or unusable (U), with the use of a discrimination function. Each class has its own flipping function which is determined by a mask (M). In a typical image $R_M \cong R_{-M}\ and\ S_M \cong S_{-M}$ will be true, but in a stego image it will be false. The RS analysis algorithm can also predict the amount if pixels that were changed. Dumitrescu et al. [101] proposed a sample pair analysis algorithm, which is based on some statistical measures of sample pairs that are highly sensitive to LSB replacement. This approach uses finite state machines for detection. Fridrich and Goljan [102] proposed a weighted stego analysis. The advantage of this method is its modular structure and clean mathematical derivations. Most of the steganalysis methods for LSB replacement exploit the fact that the pixel values are changed within the PoV.

LSB matching removes the equal trend of frequency in the PoV, making LSB replacement steganalysis invalid for LSB matching steganography. Harmsen and Jeremiah [103] showed that LSB matching affects the HCF (histogram characteristic function) center of mass. Ker [104] showed that HCF worked well for colour image LSB matching steganalysis, but were ineffective for grey-scale images. For this

© University of Pretoria

reason two different methods were proposed to apply the HCF. The first was to calibrate the output using a down-sampled image. The second was to compute the adjacency histogram instead of the usual histogram. These two methods were shown to be more reliable. Li et al. [105] proposed using HCF on the pixel difference of adjacent pixels. This approach showed improved performance over its predecessors. Zhang et al. [106] proposed an algorithm that reduces false positives in high-frequency noised images. This algorithm exploits the fact that LSB matching decreases the local maxima of an image's grey level and increases the local minima.

PVD steganography uses the difference in pixel pairs to hide information. Zhang and Wang [107] showed that PVD steganography can be detected by a histogram-based analysis of pixel-value differences. A method to increase security of PVD steganography was also proposed. This method introduces a pseudo-random dithering to the division of ranges of the PVD. Sabeti et al. [108] proposed a chi-square based method to detect PVD steganography.

The F5 algorithm preserves crucial characteristics of the DCT coefficients histogram, like monotonicity and symmetry.  However, the shape of the DCT coefficients histogram is not preserved. Fridrich et al. [109] exploited this inconsistency in the DCT coefficients histogram for steganalysis. To estimate the histogram the following steps were taken. First the image is decompressed and cropped by four pixels, to remove the quantization in the frequency domain. The image is then recompressed to its original compression level. Finally a least square fit comparison is used to determine the relative changes in the histograms of the original and cropped images. Results showed that even if at least 10% of the DCT coefficients were modified, it can be detected.

Unlike the F5 algorithm, OutGuess 0.2 preserves the shape of the DCT coefficients histogram. Fridrich et al. [110] proposed a technique which exploits the embedding

© University of Pretoria

mechanism of OutGuess. This is achieved by embedding another message into the stego image, which will partially cancel out the original message. This effect is different on the stego and cover images, which enables detection.

YASS randomizes the locations of H-blocks with the use of a key. Li et al. [111] showed that the randomized embedding of YASS was not random enough. YASS also caused detectable artefacts. Statistical features were detected in the domain used for data embedding, which showed high effectiveness in detection and identifying some embedding parameters.

## Blind Techniques

Unlike targeted techniques, blind techniques require less or even no information of the steganographic method used. The goal is to extract image features which can be used on a wide variety of steganographic methods. Blind techniques usually use some kind of classifier, which is trained on a set of image features from stego and cover images. The image features used can be classified as image quality, calibration, moment based, and correlation based features [88].

Steganography techniques can cause degradation in image quality, but degradation can also be caused by other factors. Avcibaş et al. [112] showed that image quality metric based distance, between a cover image and its filtered version, is different in comparison to a stego image and its filtered version. The image quality features were chosen with the use of the analysis of variance technique. These quality features provided the best discriminating power for an image classifier. Results showed approximately 70% correct classification on a variety of steganography techniques.

Calibration features are features that can be used to approximately recover cover image parameters from a stego image. Fridrich [113] used calibration features to

© University of Pretoria

create a blind steganographic detector for JPEG images. A set of 23 calibration features were used from the image’s DCT domain. Using calibration features, the image-to-image variations can be decreased and thus enabled more accurate detection. Pevny and Fridrich [114] extended the 23 calibration features, with the use of Markov features [115], to produce a 274-dimentional feature vector. The feature vector was used with an SVM multi-classifier. Results showed more reliable results than those achieved with the original 23 calibration features. The problem with using DCT is that DCT uses first order statics. Counter-measures that match first order statistics can foil detection.

Moment based features can be found in the higher-order statistics of an image’s wavelet domain. These statistical moments can be used to detect stego-noise, caused by steganography techniques. Lyu and Farid [116] showed strong higher-order statistical regularities within the wavelet domain of a set of natural images. By embedding information into an image, these statistics are significantly modified. These statistics were used in a linear and non-linear classifier and showed high detection rates for large messages. However, when a small enough message was embedded it was not detected.

Steganography techniques can disrupt the local correlations of an image. Local correlation can be found between pixels in the spatial domain, or DCT coefficients blocks in the frequency domain. Sullivan et al. [117] modelled the inter-pixel correlations of images as a Markov chain. A detection-theoretical divergence measurement can be calculated from the Markov model and used with a classifier for detection. Shi et al. [115] also used a Markov technique to model differences, but used it for JPEG DCT coefficient instead. JPEG 2D arrays are first formed from the magnitudes of quantized block DCT coefficients. These JPEG 2D arrays are then used to enhance changes caused be JPEG steganography. The differences are then modelled with a Markov process to that it can be used for steganalysis.

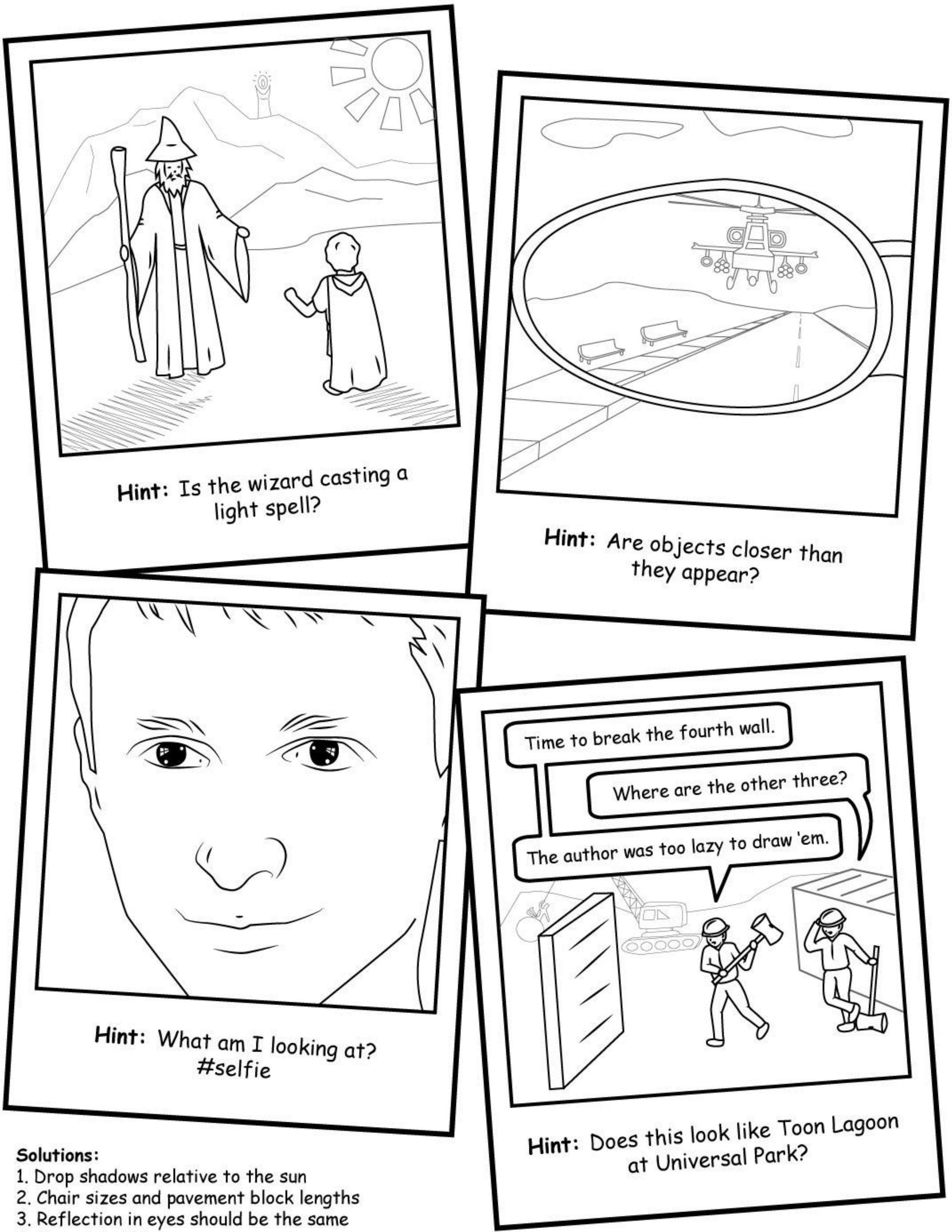
© University of Pretoria

Blind steganalysis techniques can be used for more general steganography detection, but mostly showed weaker detection rates than targeted techniques.

This chapter gave an overview of digital image forensics. In the next chapter, stereoscopic images are investigated. Stereoscopic images contain more information than normal images, which can be exploited for digital image forensic purposes.

© University of Pretoria

# What's wrong with these pictures? by MFouché


Hint: Is the wizard casting a light spell?
Hint: Are objects closer than they appear?
Hint: What am I looking at? #selfie
Time to break the fourth wall.
Where are the other three?
The author was too lazy to draw 'em.
Hint: Does this look like Toon Lagoon at Universal Park?

**Solutions:**
1. Drop shadows relative to the sun
2. Chair sizes and pavement block lengths
3. Reflection in eyes should be the same
4. Speech bubbles

© University of Pretoria

# CHAPTER 3 - STEREOSCOPIC IMAGES

## 3.1 INTRODUCTION

Everything most people see in the world is seen from two perspectives. These perspectives are from a left and right eye. There is a large amount of overlap between what the two eyes see. This overlap is often referred to as binocular vision [118]. Binocular vision helps a person's brain to estimate the depth and distance of objects. This is also known as stereopsis. Leonardo da Vinci was the first known person to record this phenomenon [119]. Leonardo noticed that objects in a painting will appear at the same depth as its surroundings. This prevents a viewer, with two eyes and at a close distance, to see the objects as it would appear in nature. The reason for this is because the viewer won't be able to see parts behind an object that should be visible to one eye and not to the other. A few centuries after Leonardo, Wheatstone [120] found that not only the visibility behind objects differ, but also the objects themselves appear to differ to each eye. This led to the creation of photographic stereoscopic images.

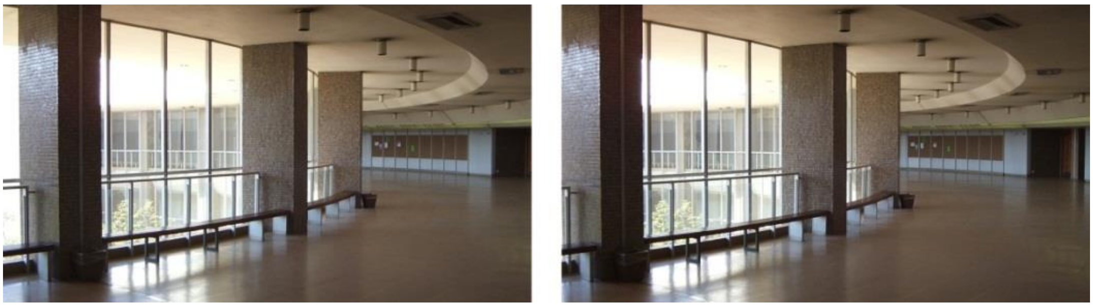


**Figure 8 - Example of a Stereoscopic Image**

A stereoscopic image is an image pair that is sometimes referred to as a 3D image. Figure 8 shows an example of a stereoscopic image. This image pair usually consists of a left and right image, mostly taken from approximately the same position and angle as would be seen by a person's left and right eye. Section 3.2 gives more

© University of Pretoria

information on creating these images. To view a stereoscopic image, some kind of technique must be used to project the one image to the left eye and the other image to the right eye. This will create the illusion of depth to the viewer. Section 3.3 discusses the different techniques used to view stereoscopic images. Section 3.4 gives a few applications for stereoscopic images. Section 3.5 discusses stereoscopic images and digital image forensics. Section 3.6 looks at ways to splice and fabricate stereoscopic images. Section 3.7 looks at methods to extract and view stereoscopic depth information.

## 3.2 CAPTURING STEREOSCOPIC IMAGES

Capturing a stereoscopic image require two different images of the same scene. There are multiple ways to achieve this. If for example two separate cameras were used, the physical horizontal position (baseline) and the view direction of the cameras can be adjusted in different ways. Subsection 3.2.1 discusses these different approaches.

The devices to capture the stereoscopic image pair can also differ. A single camera can be used to capture one image, moved and then used to capture the second image. This will not really work so well when capturing objects in motion, since the object’s position will have changed by the time the camera has moved. A second option can be to use two separate cameras to capture the two images at the same time. Some kind of synchronization mechanism will most likely be required if the images need to be taken at exactly the same time. A third option is to use a single camera to capture both images at the same time. These types of cameras will most likely have two separate lenses or a specialized lens attachment that makes use of mirrors. Subsection 3.2.2 gives more information on the different type of digital devices which can be used to capture digital stereoscopic images. Subsection 3.2.3 gives more information on digital file formats used to store digital stereoscopic images.

© University of Pretoria

### 3.2.1 VIEW DIRECTION AND BASELINE

When capturing a stereoscopic image pair, the view direction of the image capturing devices determines whether a toed-in or parallel camera configuration is used [121]. Figure 9 illustrates the toed-in camera configuration, while Figure 10 illustrates the parallel camera configuration. If the stereoscopic image was displayed on a 3D-TV, objects at the ZDP (zero disparity plane) will appear to be at the same depth as the screen [122]. Objects between the ZDP and the image capturing devices will appear to be in-front of the screen. The rest of the objects will appear to be behind the screen.

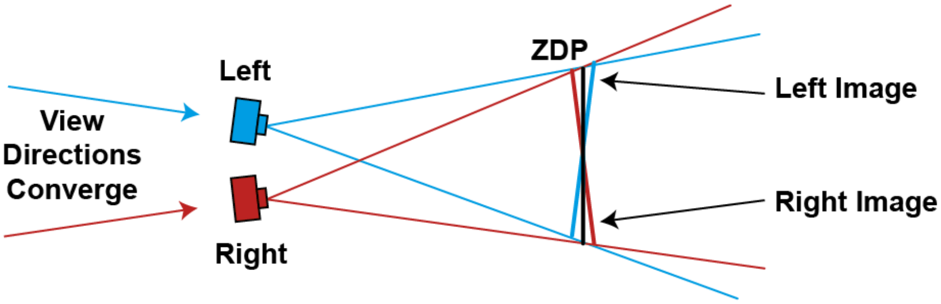


**Figure 9 - Toed-in Camera Configuration**

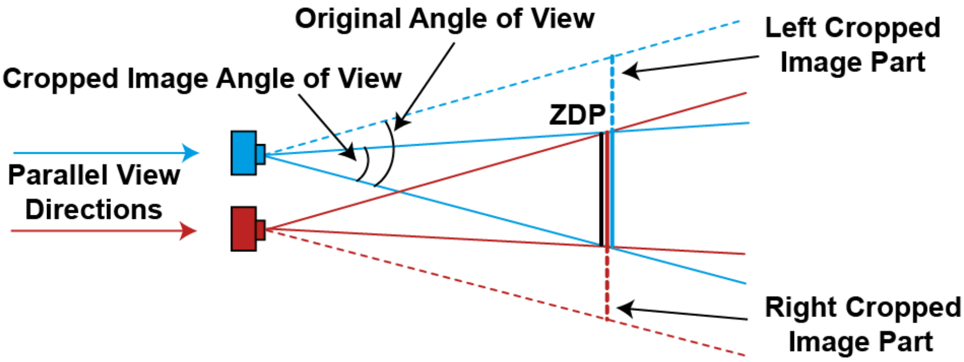


**Figure 10 - Parallel Camera Configuration with Image Cropping**

© University of Pretoria

With toed-in camera configuration, the view directions of the image capturing devices converge. By changing the angle of the image capturing devices, the ZDP can be easily adjusted. The problem with a toed-in camera configuration is that the images suffer from vertical disparity caused by keystone distortion [121]. Vertical disparity is when a point on the left image is at a different vertical position as the same point on the right image of a stereoscopic image pair. In order to correct the vertical disparity, a stereoscopic image rectification algorithm can be used [123]. Figure 11 illustrates vertical disparity.

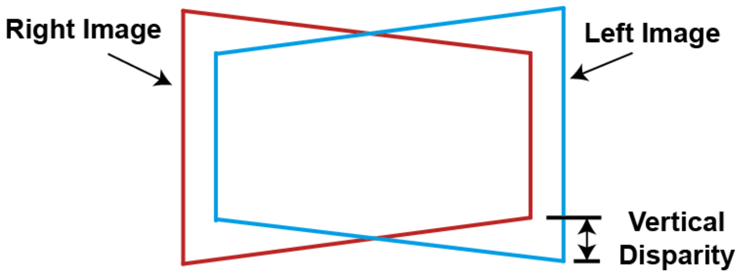


**Figure 11 - Vertical Disparity caused by Keystone Distortion**

With the parallel camera configuration, the view directions of the image capturing devices are parallel.  This cause the ZDP to be at infinity, which means everything in the image, will appear in front of the screen if viewed with a 3D-TV. One way to correct this is to crop the left image on the left side and the right image on the right side. The more the images are cropped, the closer the ZDP will be to the viewer. Cropping also reduces the angle of view of the resulting image, which can be seen in Figure 10. Another method to control the ZDP is to use specialized shift-sensor cameras [124]. With this method, the CCD image sensors of the cameras are shifted. This changes the view angle of the camera while keeping the camera lenses parallel. Figure 12 illustrates this method.

© University of Pretoria

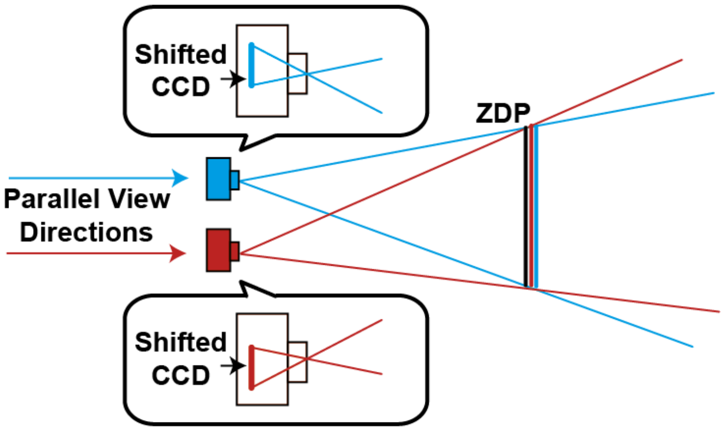


**Figure 12 - Parallel Camera Configuration using Shifted Sensors**

In addition to changing the view direction of the image capturing devices, the baseline can also be adjusted. If two separate cameras were used to capture a stereoscopic image pair, the baseline would be the horizontal distance between the centre of the cameras' lenses. The mean adult IPD (inter-pupillary distance) is approximately 63 mm [125]. A baseline around this distance is the usually used for capturing stereoscopic images. Increasing or decreasing the baseline can be used to alter the perceived size and depth of objects, since the baseline represents the viewer's size. The larger the baseline, the larger the viewer must be to have the same distance between the viewer's eyes. A large baseline can be used to capture a scene that would normally be too far away for the user to see any depth, like a mountain range. A smaller baseline can be used to capture a small object. This "enlarges" the object for the viewer. These techniques are commonly used in creating stereoscopic 3D films [126].

## 3.2.2 Digital Image Capturing Devices

A variety of image capturing devices can be used to capture digital stereoscopic images. Any device with a digital camera can potentially be used. This section discusses different digital image capturing devices and methods for capturing digital stereoscopic images.

© University of Pretoria

One of the methods to capture a stereoscopic image is to use a single device with a single lens camera to capture both images. First one image can be captured. After moving the camera, the second image can be captured. These images can be combined with software to form a single stereoscopic image. Since it takes time to move the device, this method is not ideal for capturing moving objects. For this method, any device with a digital camera can be used to capture a stereoscopic image.

Another method to capture stereoscopic images is to use two separated image capturing devices. One device is used to capture the left image and other used for the right image. These devices are usually synchronised to capture both images at the same time, which allows the capturing of moving objects. Synchronisation can be achieved in various ways. One way is to use a mechanical trigger, which presses both shutter release buttons at the same time [127]. Another way is to use a mechanism that triggers the shutter release buttons electronically [128]. If the devices have remote triggers, the triggers can be linked to trigger both devices simultaneously [129]. With this stereoscopic capturing method it would be ideal to have two identical cameras. This will insure that the images are of the same quality and with the same properties.

Instead of capturing both images separately, a single non-stereo image capturing device can be modified to capture both stereoscopic image-parts. A camera lens adaptor can be used to split the captured image, so that both image-parts are captured side-by-side in a single image. These adapters make use of a prism and mirrors to capture light from two different positions [130].

Recently, capturing digital stereoscopic images has become more convenient. Specialized devices have been developed which have two separate lenses to capture digital stereoscopic images. In addition, most of these devices have auto-

© University of Pretoria

stereoscopic displays. These displays allow a user to instantly view the captured stereoscopic image without the need of special eyewear. There are a few image capturing devices with these functionalities, like the Fujifilm FinePix Real 3D W3 digital camera [12]. Not only digital cameras, but also specialized video cameras exist that can capture and display stereoscopic images, like Sony's HDR-TD10 3D Camcorder [13]. Some smart phones have the same capabilities, such as the LG Optimus 3D and the HTC Evo 3D [15]. Nintendo's 3DS is a handheld game console with stereoscopic image capturing and viewing capabilities [14]. There are also tablets that can capture and view stereoscopic images, like the 3D tablet from Hampoo [16].

### 3.2.3 DIGITAL FILE FORMATS

Stereoscopic images can be stored as single file or two separate files. When storing the stereoscopic image in separate files, any image file format can be used to store the stereoscopic image-parts. The image-parts can also be placed next to each other in a single image and saved as any image file format. A third option is to use the MP (Multi Picture) format [131].

The MP format is used in multiple stereoscopic image capturing devices, which include Nintendo's 3DS handheld game console and Fujifilm's FinePix Real 3D W3 digital camera. Stereoscopic images using this file format can be identified by the “MPO" file extension, which stands for Multi Picture Object. This file extension can be changed to “JPG”, which will enable most devices to view the first image-part of the stereoscopic image. Figure 13 illustrates the basic MP format data structure for a stereoscopic image.

© University of Pretoria

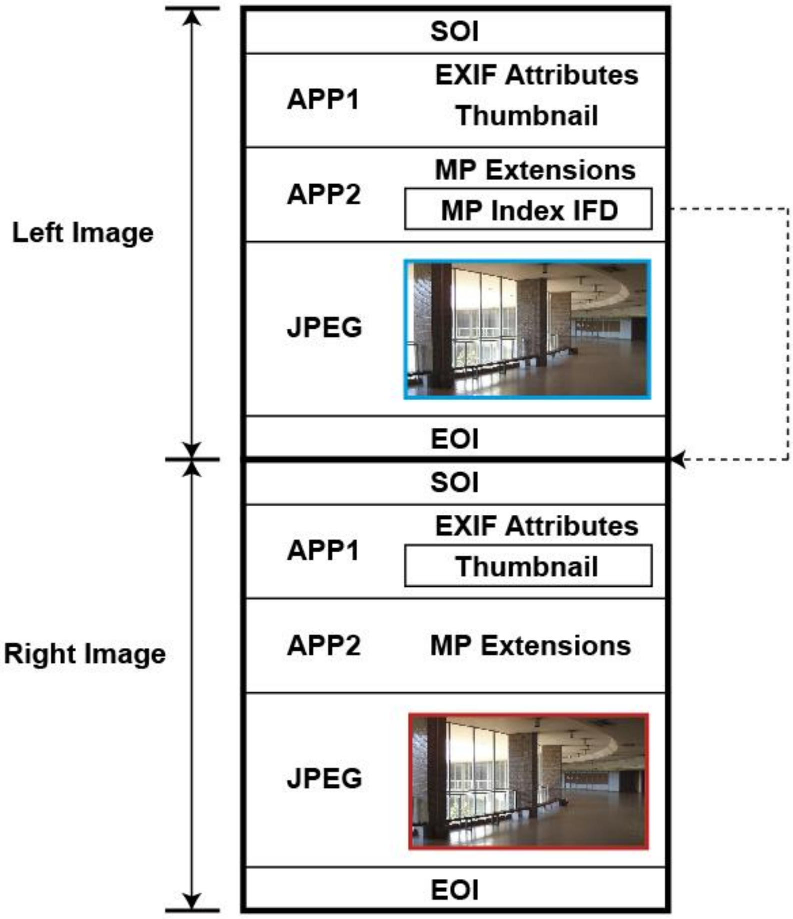


**Figure 13 - Basic MP File Format Data Structure for a Stereoscopic Image**

The MP format uses the same structure as EXIF JPEG for individual images, which are linked with additional MP extensions. Each individual image is a JPEG (Joint Photographic Experts Group) compliant image as specified in the EXIF (Exchangeable image file format) specification. These individual images are delimited by SOI (Start of image – 0xFFD8) and EOI (End of image – 0xFFD9) markers. Images stored in APPn (JPEG application marker segments), like thumbnails, do not have these markers and are not considered "individual images". APP1 contains EXIF attributes, in the form of tags, as specified in the EXIF 2.21 specification. These tags include metadata, like the camera settings and the conditions under which the image was created. APP2 contains MP extensions, which include metadata for the set of images and how they are related. For stereoscopic images, APP2 will contain information like the baseline length. The first individual image's APP2 also contains

© University of Pretoria

a MP Index IFD (Image File Directories), which contains information about the overall file structure.

More detailed information about the MP format can be found in the MP format file specification [131].

## 3.3 VIEWING METHODS

The goal of stereoscopic viewing methods is to project one stereoscopic image-part to the one eye, while projecting the other stereoscopic image-part to the other eye. Several methods have been developed to achieve this. Some require the viewers to wear special eyewear, while others do not. This section gives an overview of eyewear, autostereoscopic and other methods for viewing stereoscopic images.

### 3.3.1 STEREOSCOPIC WITH EYEWEAR

Stereoscopic methods that require eyewear can be classified into four categories. The categories are position multiplex, colour multiplex, polarisation multiplex and time multiplex.

With position multiplex, the two image parts are displayed at different positions. Special eyewear is used to change the view direction of each eye when the images are viewed. This allows the viewer to see both images as though they were placed at the same position. The problem with this approach is that it is highly dependent on the position and the distance of the viewer's eyes from the images. An example of a product that uses this approach is the KMQ prism glasses. For these glasses, stereo image-parts are displayed one above the other as seen in Figure 14.

© University of Pretoria

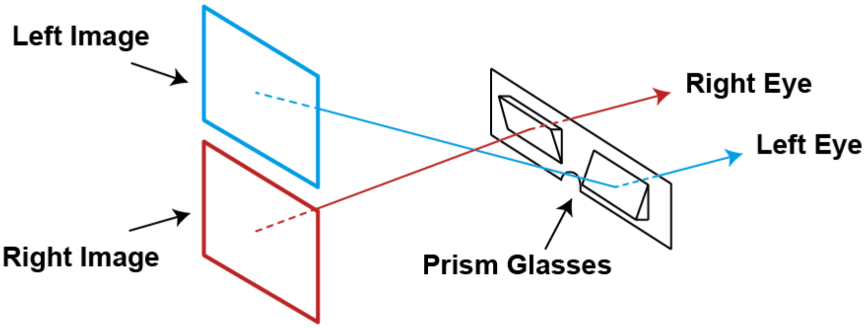


**Figure 14 - Position Multiplex with KMQ Prism Glasses**

With colour multiplex, the two-image parts are merged into a single image using a complementary colour coding technique. An anaglyph is an example that uses eyewear with different coloured lenses to filter the complementary colours. The complementary colour combinations are red / cyan, blue / yellow and green / magenta [132]. For example, let us assume red / cyan is used to create the merged image. The red colours of the one image-part and the blue and green colours of the other image-part will then be used. This causes a loss of colour information in the resulting anaglyph image [122]. Another problem with anaglyphs is that the image suffers from crosstalk [132]. Crosstalk causes one eye to see some of the image-part intended for the other eye. Figure 15 shows an example of a red / cyan anaglyph created from Figure 8.

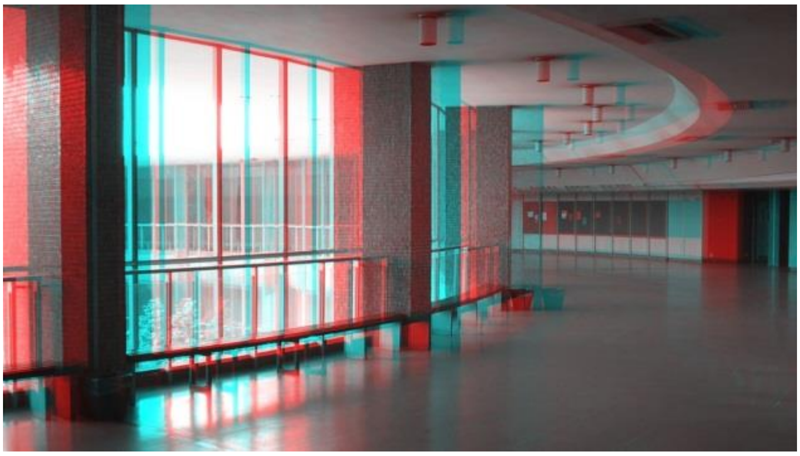


**Figure 15 - Example of an Anaglyph**

© University of Pretoria

Polarization multiplex is a popular viewing method used in 3D cinemas. This method super-imposes the two stereoscopic image-parts by using different polarities for each part. Polarised glasses are then used to filter the different polarised images for each eye. Polarisation of images can be circular or linear. Unlike linear polarisation, circular polarisation does not depend on the orientation of the viewer's head to properly separate the image-parts for the viewer's eyes [133]. The advantage of this stereoscopic viewing method is that the eyewear is inexpensive.

Time multiplex is a technique where the image-parts are alternated on a single display. These image-parts are alternated at a high refresh rate, which is typically twice the refresh rate of a normal display, to prevent the appearance of flickering [134]. Shutter glasses that are synchronized with the display, are then used by the viewer. These glasses block the view of one eye, depending on the image part that is currently showing on the display. Shutter glasses are more expensive than other eyewear, because they require synchronization electronics and LCD lenses to block images.

## 3.3.2 AUTOSTEREOSCOPIC

Autostereoscopic methods do not require eyewear to view stereoscopic images. The filtering of the two-image pairs occurs at the display itself, instead of requiring special eyewear. Two filtering methods that are commonly used in autostereoscopic displays are parallax-barrier and lenticular lenses [135].

One of the first implementations of parallax barriers dates back to 1692 [136]. The French painter, G. A. Bois-Clair, created paintings with alternating strips of two images. He placed a grid of vertical laths in front of the image to block the view of one of the images, depending on the position of the viewer. Some autostereoscopic displays use the same principle to display stereoscopic images. The image-parts are interlaced in vertical strips and a parallax barrier is placed in front of the image. The

© University of Pretoria

barrier blocks the one image to the one eye and the other image to the other eye, depending on the position of the viewer. Nintendo's 3DS handheld game console is an example of a device that uses parallax barriers [14]. Figure 16 illustrates this autostereoscopic method.

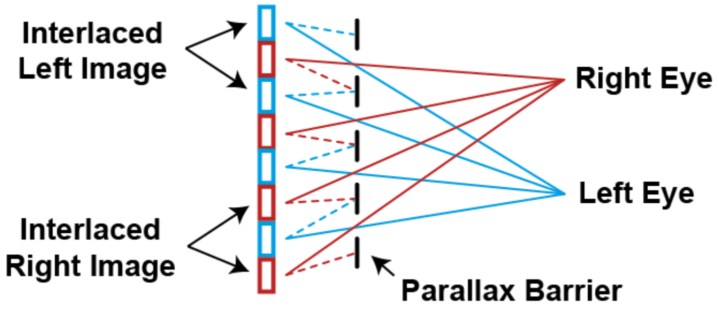


**Figure 16 - Autostereoscopic Method using Parallax Barrier**

The lenticular method is commonly used in printing stereoscopic images, as well as printing images that appear to be animated when the viewing angle is changed [137]. With this method, the image-parts are interlaced similar to the parallax barrier method. Instead of using a parallax barrier, cylindrical lenses are used. These lenses magnify image parts for the viewer, depending on the viewing angle. Fujifilm's FinePix 3D W3 digital camera is an example of a device that uses an autostereoscopic lenticular display [12]. Figure 17 illustrates this autostereoscopic method.

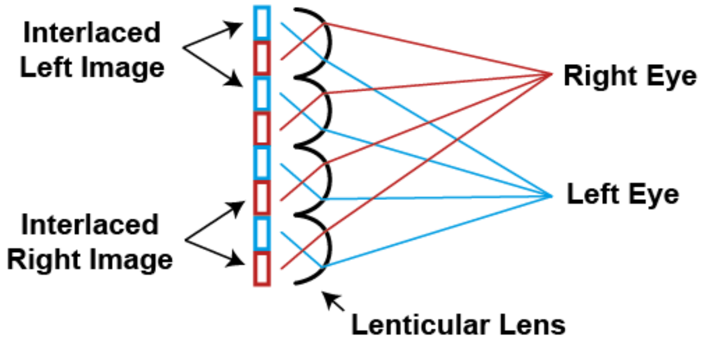


**Figure 17 - Autostereoscopic Method using a Lenticular Lens**

© University of Pretoria

### 3.3.3 OTHER METHODS

Other methods exist which allow for the viewing of stereoscopic images. This subsection discusses a few of them.

Some stereographic displays can be worn like a pair of glasses. Wearable stereoscopic devices project the correct stereo image-part to each eye. This can be achieved with the use of a small high resolution screen, or by projecting light directly into the eye. When using a screen, the stereoscopic image-parts are displayed side-by-side. Lenses or mirrors are then used to project the correct image part to the correct eye. A separate screen for each eye can also be used in a similar way. An example of an electronic device that uses a screen, is Vuzix's stereoscopic augmented reality glasses [138]. Another example is the Oculus Rift virtual reality headset, which was acquired by Facebook in 2014 [139]. Instead of using screens to display the stereoscopic image, light can be projected to the back of the viewer's eyes with the use of microscopic mirrors. Avegant's Glyph is an example of headset that uses this technique [140].

Wigglegrams are another way to view stereoscopic images. Wigglegrams are animated images that alternate between stereoscopic image-parts. This creates the illusion of the camera constantly moving slightly left and right. This movement helps the brain perceive depth, since objects closer to the viewer appear to move more than objects further away.

Certain eye techniques can be used to view stereoscopic images, where the image-parts are placed next to each other. The placement of the images can be parallel or cross-eyed. With parallel placement, the image-part intended for the left eye is placed on the left side. By relaxing the eyes, as if viewing something in the distance, each eye will look at a different image. This will create the illusion of a third centre image that is an overlap of the image-parts and will appear to have depth. To aid

© University of Pretoria

parallel viewing, a piece of cardboard can be placed between the eyes, to enable each eye to only see one of the image-parts. With cross-eyed placement, the image-part for the left eye is place on the right side. When viewing the image, the eyes need to cross as if focussing on an object between viewer and the image. This will create the illusion of a centre image, which appears to have depth, similar to parallel viewing. To aid cross-eyed viewing, an object can be placed in front of the viewer and moved towards the image. Figure 18 illustrates both methods.

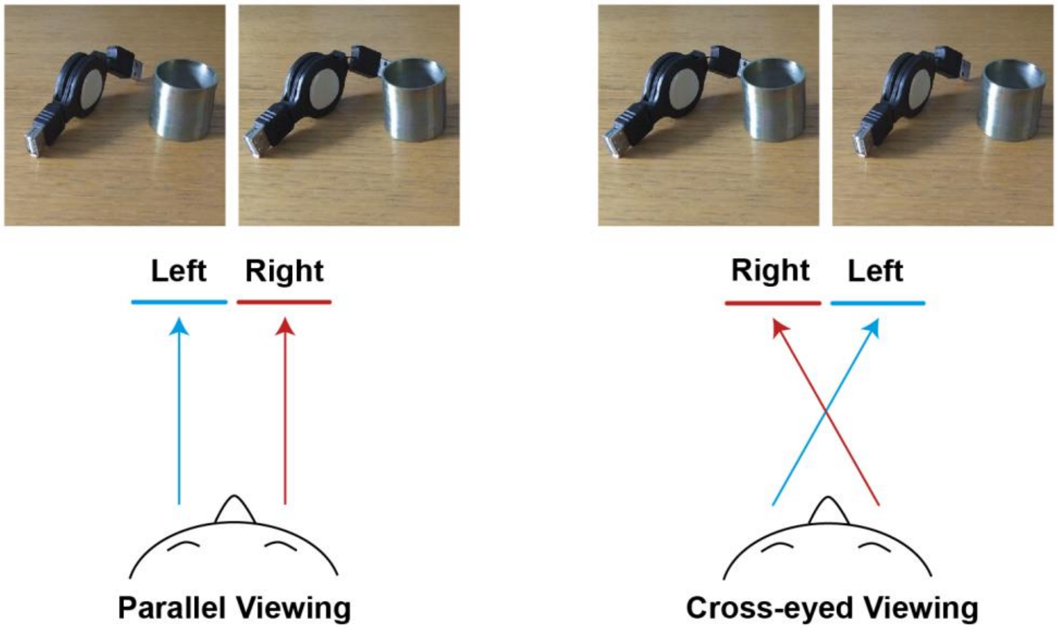


**Figure 18 - Parallel and Cross-eyed Viewing Techniques**

Constant use of eye-techniques can cause strain on a person's eyes.

## 3.4 APPLICATIONS

There are many applications for stereoscopy and stereoscopic images. This section gives a few of these applications, which include entertainment, education, medical sciences, forensic sciences, astronomy, robotics, altered reality and advertisement.

© University of Pretoria

Stereoscopic images are commonly used in the entertainment industry. This can be seen by the amount of stereoscopic 3D films that are released each year. In addition, films that were not originally created in stereoscopic 3D, like Titanic and the Star Wars films, are converted to stereoscopic 3D [141]. Television programs can also be converted to stereoscopic 3D [142]. There are already a few television sets which can achieve this in real-time, like Samsung's Smart TVs [143]. Video games are another form of entertainment which use stereoscopic images [17].

Education benefits from stereoscopic images, because it aids in the understanding of spatial relationships. The GeoWall project [18] is an example of how stereoscopic images help with Geoscience research and education. A great number of physics processes are better understood when shown in stereoscopic 3D [144]. Medical education uses stereoscopic images to help students visualize the human anatomy [145]. Stereoscopic images also help enhance training simulations. Examples include dental [146] and neurosurgeon training [147].

Stereoscopic images are not only used in medical education, but also in medical sciences. It has been shown to help improve surgical navigation [148]. Stereoscopic X-ray imaging systems [19] demonstrate another medical application for stereoscopic images.

There are different applications for stereoscopic images in forensics science. Historical stereoscopic aerial photography can aid environmental forensic investigations [20]. With the aid of stereoscopic microscopy, a forensic analysis of black coral has been performed [149]. A stereoscopic microscope has also been used in fingerprint-based forensics [150].

Stereoscopic images have been used multiple times in astronomy and are still used frequently. In 1969, both the Apollo 11 and Apollo 12 missions used a special

© University of Pretoria

stereoscopic camera to take pictures of the lunar surface [151]. In 1976, the Viking mission used two facsimile cameras for taking stereoscopic images of Mars [152]. In 1986, the first SPOT satellite was launched, which was used to take stereoscopic images of the earth [153]. In 2006, the twin STEREO spacecraft were launched, which at the time of writing were still being used to take stereoscopic images of the sun and solar phenomena [21].

There are more applications for stereoscopic images. In robotics stereoscopic images are used for robot vision [22]. Stereoscopic images can also be used to enhance Augmented Reality [138] and Virtual Reality [154]. With the use of lenticular printing, stereoscopic images can be printed for advertisements [23]. As technology advances, more uses for stereoscopy and stereoscopic images will most likely be discovered.

## 3.5 STEREOSCOPIC IMAGES AND DIGITAL IMAGE FORENSICS

Regular digital image forensics can be applied to stereoscopic images. The difference is that stereoscopic images contain additional information that can improve digital image forensic techniques. This additional information also makes digital image counter forensics more challenging.

Image source identification tries to identify the device or method used to create an image. For stereoscopic images, there are additional factors to take into account. The stereo image parts could have been taken by separate image capturing devices, a single device with two lenses or a single device for both images. When separate image capturing devices are used, the make and model of the capturing devices will most likely be the same. This ensures that the image quality remains constant over the two image parts. Using this information, it can help to validate device class

© University of Pretoria

identification for the image parts. Specific device identification information will be the same or different, depending on the technique used to capture the image parts.

Image forgery detection tries to detect image manipulations on an image. For stereoscopic images, there are two images to edit instead of one. This makes it harder to create an undetectable forgery. Image inconsistencies could arise from each image part separately, as well as the combination of the image parts. Because there are two images of the same scene, it is possible to extract depth information. Section 3.7 discusses a few techniques to extract depth information. Inconsistencies in depth information can help identify image manipulations, which is explored in this dissertation. Depth information can also be used to automate other image forgery detection techniques. An example is the detection of defocus blur [77]. Objects at the same distance from a camera will have the same blur kernel sizes. When an image is modified with a technique like splicing, blur inconsistences can arise. This method requires the investigator to select areas with similar distance from the camera. The selection process can potentially be automated for stereoscopic images, with the use of depth information.

Steganography tries to inconspicuously hide information, while steganalysis tries to detect hidden information. Stereoscopic images can be used for steganography in a similar way as normal images. There is more space to hide information because of the additional image part. Information can be hidden in one or both image parts. Another rather naïve place to hide information, can be the space between the EOI and SOI tags in a MPO file. The problem with stereoscopic steganography is that the two image parts provide additional statistical information for steganalysis. Each image part has its own statistical information. Comparing the statistical information of the two image parts can provide even more statistical information for steganalysis. Additional counter-measures may thus be required in order to counter steganalysis techniques.

© University of Pretoria

## 3.6 FORGING STEREOSCOPIC IMAGES

Stereoscopic images can be manipulated in non-stereoscopic image manipulation software. If the stereoscopic image is in the MP format, it might first need to be converted to two separate images, or placed side-by-side in a single image. When manipulating stereoscopic images, everything done to the one image part should be done to the other image part to minimize visual inconsistencies. This section will look at image splicing for stereoscopic images and techniques to synthesize stereoscopic images.

### 3.6.1 STEREOSCOPIC IMAGE SPLICING

Image splicing is when multiple images are combined to form a single composite image. For stereoscopic images, splicing can be categorised as non-stereoscopic to stereoscopic splicing and stereoscopic to stereoscopic splicing.

Non-stereoscopic to stereoscopic splicing can be used when the subject that needs to be spliced is only available in non-stereoscopic image form. An image of a famous actor is an example. When the subject is copied to the stereoscopic image, it needs to be copied to both the left and right image part. The vertical position of the subject should stay the same. The horizontal position is slightly adjusted to place the subject at the correct depth in the scene. Using this splicing method can cause the subject to appear flat. The reason for this is that objects from non-stereoscopic images have no internal depth. Internal depth is the depth difference of points on a subject. The nose of a person facing the camera should appear closer than the person’s ears. This inconsistency is only visible to viewers when there is a large enough difference in the expected and perceived internal depth. Figure 19 shows an example of non-stereoscopic to stereoscopic image splicing, which uses the stereoscopic image from in Figure 8.

© University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

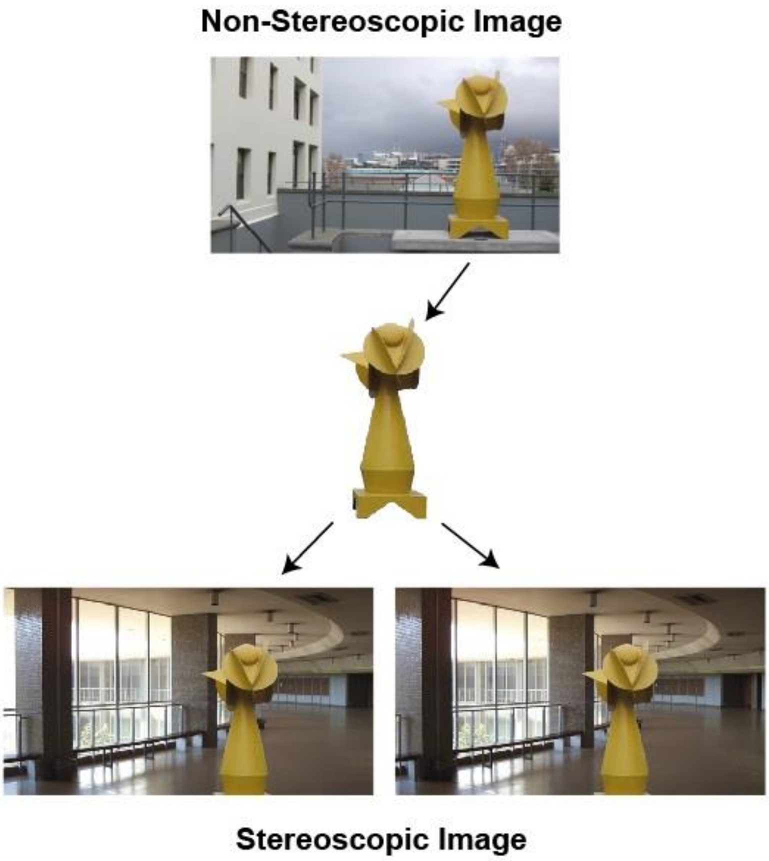


**Figure 19 - Non-Stereoscopic to Stereoscopic Image Splicing**

Stereoscopic to stereoscopic splicing can be used when both images are stereoscopic. When a subject is copied from a stereoscopic image, the subject in the left image part is copied to the left image part of the combined stereoscopic image. The same subject in the right image part is copied to the right image part of the combined stereoscopic image. The goal is the preserve the internal depth of spliced image parts. It should be note that non-stereoscopic to stereoscopic image splicing can be used with stereoscopic images, if only one of the image parts of a stereoscopic image is used. Figure 20 shows an example of non-stereoscopic to stereoscopic image splicing, which uses the stereoscopic image from in Figure 8.

© University of Pretoria

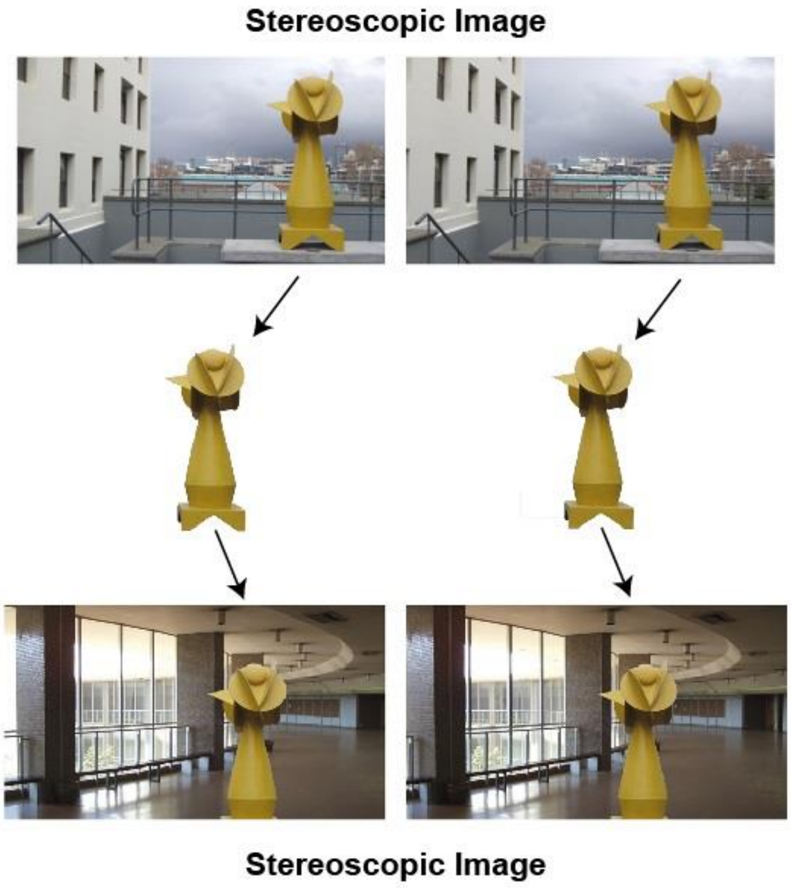


**Figure 20 - Stereoscopic to Stereoscopic Image Splicing**

Non-stereoscopic to stereoscopic image splicing is more likely to be used, than stereoscopic to stereoscopic image splicing. The reason for this is that non-stereoscopic images are more common than stereoscopic images. It is easier to find non-stereoscopic images of subjects on the internet, than stereoscopic images of subjects. The advantage of stereoscopic to stereoscopic images is that subjects contain internal depth, unlike non-stereoscopic to stereoscopic image splicing. There are however methods to synthesize internal depth for non-stereoscopic image parts, which are discussed in the next subsection.

### 3.6.2 SYNTHESIZING STEREOSCOPIC IMAGES

It is possible to synthesize a stereoscopic image from a non-stereoscopic image. An example of such synthesized stereoscopic images can be found in old non-

© University of Pretoria

stereoscopic films that are converted to stereoscopic 3D [141]. Stereoscopic images can also be synthesized for film scenes which would be too expensive or impractical to capture with stereoscopic cameras. In this subsection, non-stereoscopic to stereoscopic conversion will be further referred to as 2D to stereo 3D conversion. 2D to stereo 3D conversion methods can be classified as manual, semi-automatic and automatic conversion. These methods scale from quality to speed.

### MANUAL 2D TO STEREO 3D CONVERSION

Manual methods require the shifting of object or region pixels horizontally to create an artistically chosen depth. When pixels are shifted, empty areas will appear. The reason for this is that objects in one stereoscopic image part cover areas not visible in the second stereoscopic image part. These areas can be painted in, or filled with image cloning or a similar technique. The newly created image can then be used as the second part of the stereoscopic image.  Figure 21 shows an example of simple manual 2D to stereo 3D conversion. In the example the object was first separated from the background. The background and object were then skewed horizontally to opposite directions. Lastly, the empty areas were filled with Photoshop CS 5.1's content aware fill, which is similar to image cloning.

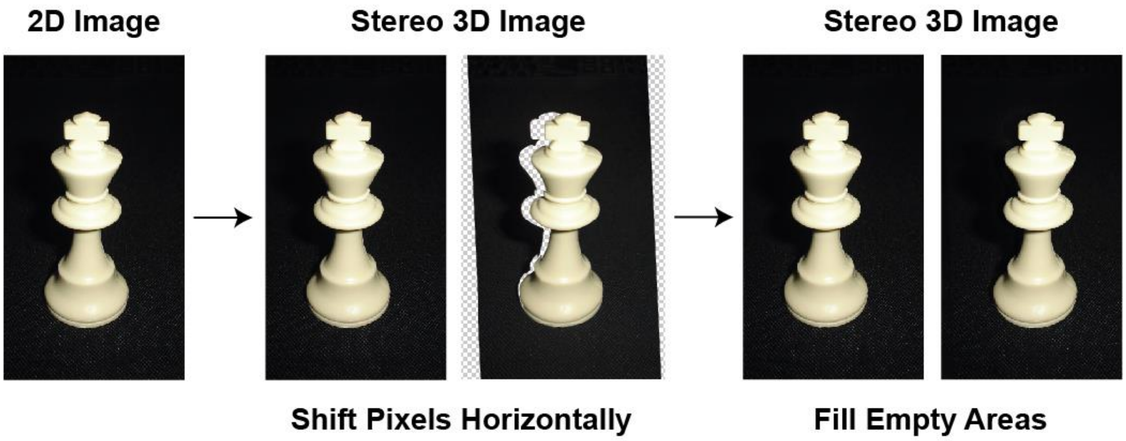


**Figure 21 - Manual 2D to Stereo 3D Conversion**

© University of Pretoria

Manual type methods can produce high quality depth, but are very time consuming and expensive. In order to improve conversion time, semi-automatic methods can be used.

## SEMI-AUTOMATIC 2D TO STEREO 3D CONVERSION

Semi-automatic methods convert 2D to stereo 3D automatically but still require human guidance. A common method for conversion is with the use of depth maps. This method usually requires three steps [142]. First the depth map is created and processed. Then the image is warped to create the stereoscopic image pair. Finally, if any holes are present on the image, they are filled.

The first step is to create a depth map. Depth maps can be represented as grey scale images that illustrate the different depths in an image. Objects closer to the camera are illustrated as a lighter grey than objects further away. Creating depth maps usually require human guidance to specify the depth of surfaces. Surfaces are first selected, isolated and then given a relative depth. The selection process can be simplified by drawing a line over an area and having an algorithm determine the boundary of the surface [155]. The boundary can then be modified as desired. The depths of objects are chosen depending on where they should appear in relation to the viewing surface. An optional step is to smooth the edges of the depth map, to reduce block artefacts and sharp transitions between object boundaries.

The second step is to convert the created depth maps to a stereoscopic image pair. This is achieved by rendering a virtual image by projecting the pixels of the original image to their proper 3D locations and then projecting them to the virtual image plane. The virtual camera is set up to form a parallel stereoscopic setup with the original camera. With this approach the original image is used as one of the stereoscopic image parts. Alternatively, two separate images can be created using the original only as a reference.

© University of Pretoria

The third step is to fill any empty areas that may arise from step two. Empty areas are areas which are covered by surfaces in the original image, but should be revealed when viewed from a slightly different position. There are multiple ways to fill these empty areas. The pixel colour information of nearby pixels can be horizontally interpolated to fill the holes [156]. If no depth information is taken into account the foreground and background will appear to be fused. To avoid this fusing problem, the pixel colours of only the background can be used instead. Another filling method is to diffuse boundary pixel values by solving the Laplacian equation on the boundaries of empty areas [157]. Inpainting techniques [158], which are used for image restoration and removing of artefacts, can also be used to fill empty areas. All of the filling methods mentioned so far can cause texture artefacts, caused by the duplication of neighbouring pixel colour information. To reduce texture artefacts, the depth maps can be pre-processed. Empty areas arise when there are sharp edges in the depth map. By smoothing the depth map, these empty areas can be avoided [159]. Smoothing can however cause geometric distortion in the resulting stereoscopic image. Straight vertical lines in one image part could be rendered as curves in the second image part.

After a stereoscopic image has been created, post-processing can be used to improve image quality.

## AUTOMATIC 2D TO STEREO 3D CONVERSION

Automatic 2D to Stereo 3D conversion attempts to automate all of the steps used to synthesize stereoscopic images. Automatic conversion can be done in real time if speed is more important than quality. One application is the conversion of real time TV broadcasting.

© University of Pretoria

Creating depth maps for depth map based conversions is one step that can be automated. Instead of estimating depth with the aid of a human, depth information can be estimated with the use of pictorial depth cues. Pictorial depth cues are information the human visual system uses to perceive depth in a 2D image. Focal blur, geometric cues, and colour intensity cues are examples of pictorial depth cues.

Focal blur can be used to estimate the relative distance of objects from the camera. Objects at the same focus levels will have the same depth. Some methods to calculate blur levels include a wavelet-based approach [160], a high order statistical approach [161] and a de-convolution process in the frequency domain which uses inverse filtering [162]. Using focus information is relatively simple, but out-of-focus areas in front and behind the point of focus can appear the same.

Geometric pictorial depth cues include linear perspective, the height of objects and the textures of objects. Linear perspective refers to parallel lines that appear to converge at a distances, eventually reaching a vanishing point. Using the position of these lines and vanishing points, it is possible to estimate depth [163]. Height of objects can also be used to determine depth. Objects closer to the bottom of an image are usually closer to the camera. To extract depth, a line-tracing algorithm can be used to create horizontal lines from the one side of the image to the other side [164]. It is possible to extract depth from textures of certain images. The shape of a surface can be estimated, by looking at the markings or texture of the surface [165].

Image colour intensity pictorial depth cues include atmospheric shattering, light and shadows distribution, and figure-ground perception. Atmospheric shattering refers to the shattering of light rays in the atmosphere. This shattering effect causes objects far away to have less contrast and a blue-ish tint. Using a simple warm/cool colour theory, objects with a warm colour can be estimated to be closer than those

© University of Pretoria

with a cool colour [166]. Light and shadow distribution is the shading of objects and the shadows they cause due to lighting. The grey level of a pixel depends on the light source direction and a surface normal. There are multiple algorithms which can determine the shape of objects from an image's shading [167]. Figure ground perception is provided by the edges and regions in a scene. Two overlapping regions can be separated by a single edge. The edge usually belongs to only one of the regions, which will be in the front. The back region is assumed to go on behind the front region. This principle can be used to estimate relative depth of edges and regions [168].

There are conversion methods that do not require the creation of depth maps. If the original 2D image was created from a fully computer generated 3D scene, access to the 3D scene can simplify 2D to stereo 3D conversion. The 3D scene can be rendered from two different views, simulating a stereoscopic camera setup. The same can be done for any computer generated 3D objects that are combined with other images.

Other non-depth map based methods include those that construct 3D scenes using structure from motion [169] and those that calculate the planar transformations between images in sequence [170]. Both these methods require multiple video frames in sequence and thus not suitable for single image conversion.

## 3.7 EXTRACTING STEREOSCOPIC DEPTH INFORMATION

Stereoscopic depth information can be useful for stereoscopic image forgery detection. If a stereoscopic image is manipulated, inconsistencies in the image's depth can occur. Object depth can be internal or external. Internal depth is the depth difference of separate points on the same object. External depth is the position of the object relative to the scene and other objects. There are multiple

© University of Pretoria

ways to extract and view depth information, which include triangulation, disparity maps, and 3D modelling.

Triangulation can be used to determine the distance of a point from the image capturing device(s) [171]. Triangulation uses the positions of the point in both stereoscopic image parts, together with viewing angle and camera baseline distance. Calculating the distance of points can be useful, when investigating depth inconsistencies. However, the required information for triangulation might not be available. The stereoscopic image metadata can be removed, or the image's file format might not support the required metadata.

Disparity maps can be used to extract and view relative depth information, without the need to know specific camera properties. Disparity is the difference in horizontal position of a pixel in the left and right stereoscopic image part. With the use of a stereo-matching algorithm, each pixel is assigned a colour value according to its disparity. A disparity map can be dense or sparse. A dense disparity-map tries to show a match for every pixel, while a sparse depth-map does not. Some pixels may be occluded in one image part and will thus not have a match. Figure 22  shows an example of a dense disparity map created from Figure 8. The example was created from a simple SSD (sum of squared differences) algorithm [172]. Disparity map creation can be challenging when images contain semi-transparent areas, reflective areas, smooth surface areas, strong specular lighting, and areas that only appear in one of the image parts. The quality of Figure 22 was affected by these image properties.

© University of Pretoria

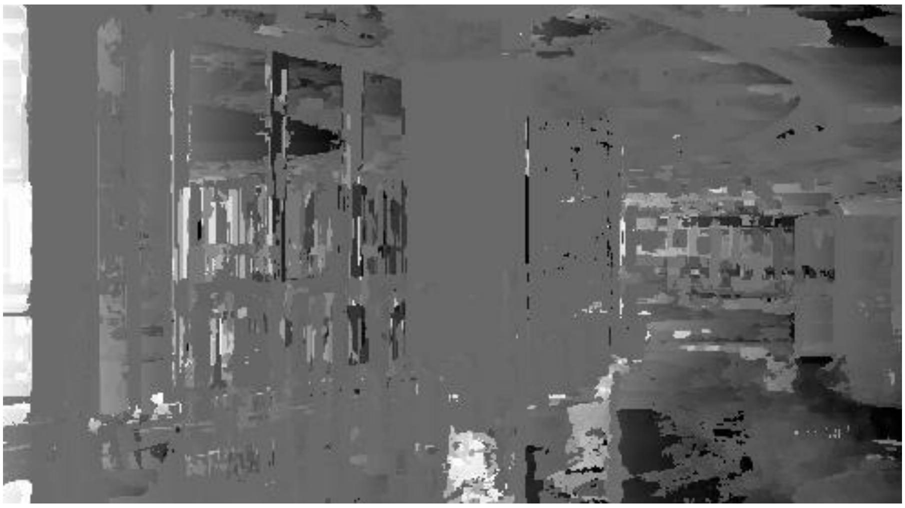


Figure 22 - A SSD Dense Disparity Map

Stereo-matching algorithms use a combination of one or more of the following steps to create a disparity map [172]. First matching costs are evaluated for the stereo image pair. Next the matching costs are aggregated. Disparities are then computed or evaluated. Finally the disparity is refined.

Stereo-matching algorithms can be categorized as local or global [172]. Local algorithms use the image colour values of a squared window around each pixel in order to find a match. An example of a local algorithm is the SSD algorithm. For the SSD algorithm, matching cost is the squared difference of pixel intensity values at a given disparity. The matching cost values are then aggregated over square windows with constant disparity. The disparity is determined by the match with the smallest aggregated value for each pixel. Other matching costs include absolute differences [173], adaptive normalized cross correlation [174] and rank transformation [175]. There are local algorithms that change the size of the squared window to deal with occluded areas. These include adaptive windows [173] and shiftable windows [176]. Unlike local algorithms, global algorithms usually try to find a disparity match by minimizing a global cost function. Examples of global algorithms include those that use graph cuts [177], nonlinear diffusion [178] and belief propagation [179].

© University of Pretoria

Instead of displaying relative depth as a 2D image, a 3D model can be created. 3D models can be represented in multiple ways which include volume, functions and polygons. Volume can be represented by cubes in a discrete 3D grid, like voxels [180]. Function representations encode distance to the closest surface, like level-sets [181]. Polygons are planar facets which can be connected to form a 3D surface, called polygon meshes [182]. Figure 23 shows an example of a 3D model created from Figure 8, with the use of AgiSoft StereoScan software.

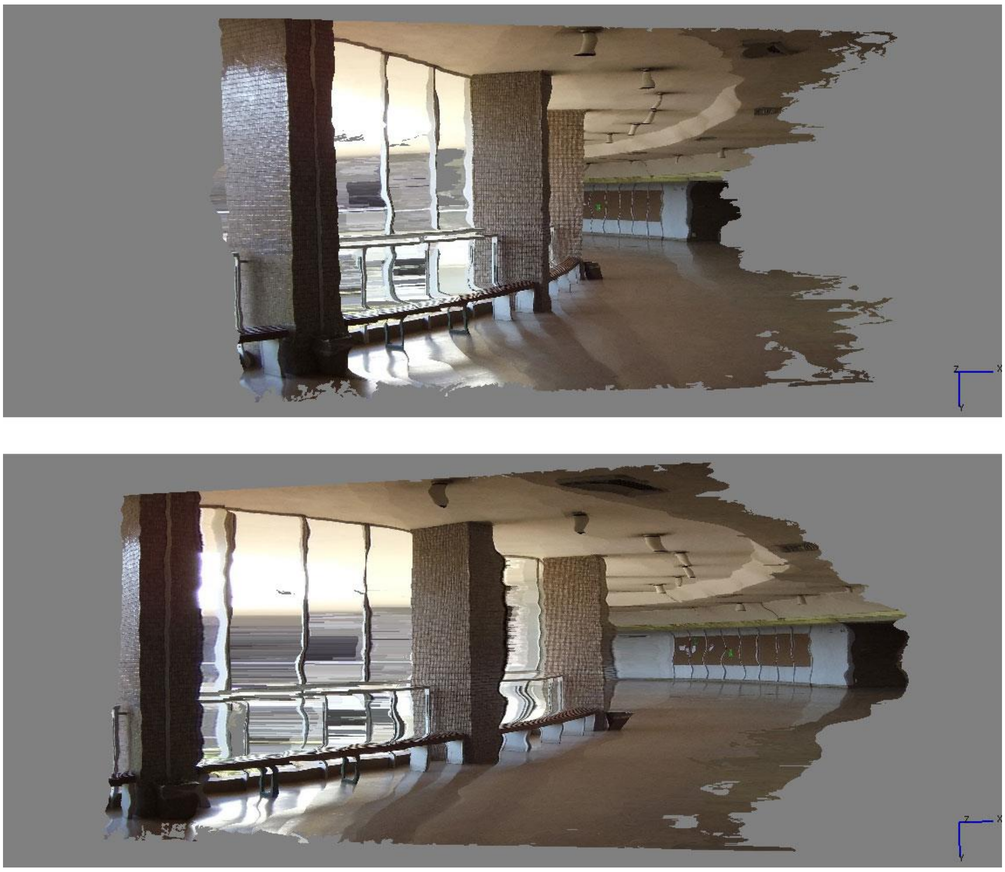


**Figure 23 - 3D Model Created from a Stereoscopic Image**

The 3D reconstruction of multiple 2D images can be categorized into four classes of techniques [183]. The first class computes a cost function on an estimated 3D volume and then extracts a surface from this volume. An example is the voxel colouring algorithm [180]. The second class iteratively evolves a surface to minimize a cost function. An example is an evolving mesh that moves as a function of internal

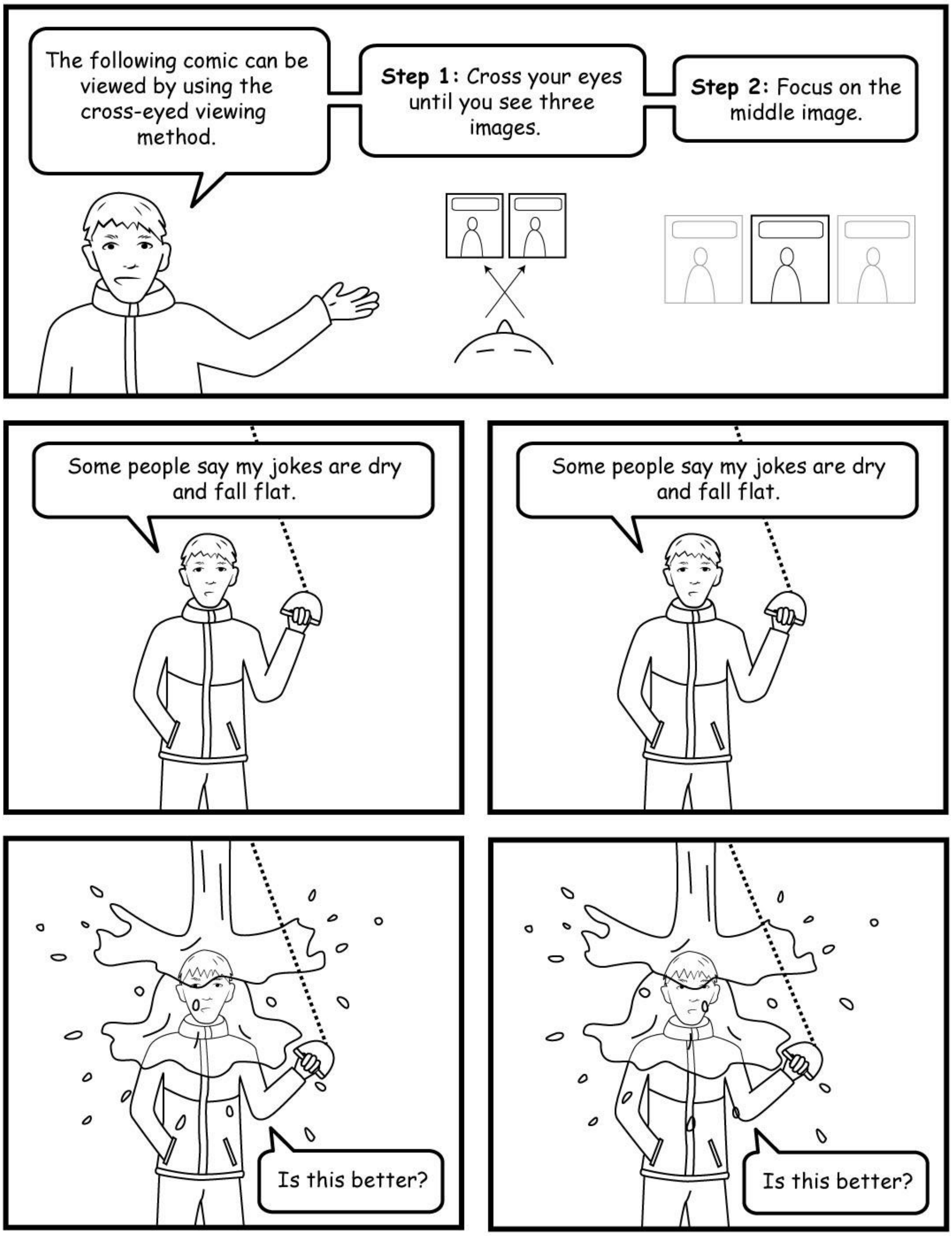
© University of Pretoria

and external forces [182]. The third class generates disparity maps and uses the disparity maps to create a 3D scene [184]. The final class extracts and matches a set of feature points and then creates a 3D surface from these points [185].

This chapter gave an overview of stereoscopic images. The next chapter investigates how disparity maps can be used to aid the detection of stereoscopic image forgery.

© University of Pretoria

## Stereoscopic Comic by MFouché


The following comic can be viewed by using the cross-eyed viewing method.
Step 1: Cross your eyes until you see three images.
Step 2: Focus on the middle image.
Some people say my jokes are dry and fall flat.
Some people say my jokes are dry and fall flat.
Is this better?
Is this better?

My slapstick humour doesn't always work, neither does my self-defeating humour.

# CHAPTER 4 - USING DISPARITY MAPS TO AID STEREOSCOPIC SPLICING DETECTION

## 4.1 INTRODUCTION

Non-stereoscopic to stereoscopic image splicing copies a 2D area into both stereoscopic image parts. Modifying the horizontal position of the 2D area, changes the depth at which the object appear in the scene. The 2D area itself will not have internal depth, which may be detected with the use of a disparity map.

This chapter investigates whether it is possible to detect simple non-stereoscopic to stereoscopic splicing, by highlighting flat areas in the disparity map of a stereoscopic image. An assumption is made that internal depth has not been fabricated for the spliced area.

An experiment is performed on a set of 50 spliced stereoscopic images. Section 4.2 describes the methodology of the experiment. Section 4.3 discusses the results. Section 4.4 provides a conclusion.

## 4.2 METHODOLOGY

An experiment was created to test whether the following is true:

> The 2D spliced area, in non-stereoscopic to stereoscopic splicing, can be detected by highlighting flat regions in the disparity map of the image. An assumption is made that internal depth has not been fabricated for the spliced area.

© University of Pretoria

The four steps used for the experiment are as follows:

1. Create a dataset of 50 spliced images which mimic non-stereoscopic to stereoscopic image splicing.
2. Create a mask which shows the correct spliced areas.
3. Create a disparity map for each spliced image.
4. Highlight large areas of similar disparity. The highlighted areas are copied from the mask onto the original image to illustrate detected areas. The amounts of correctly and incorrectly identified pixels are recorded.

Figure 24 illustrates the steps taken for the experiment. The illustration shows the mask as black and white for clarity. In the result image, black shows correctly identified spliced areas, while white shows false positives.

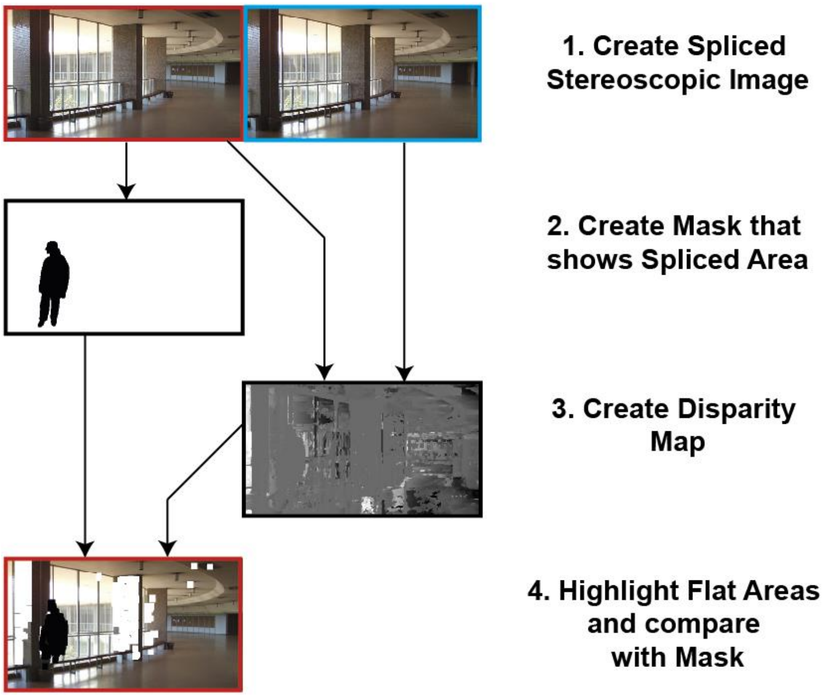

1. Create Spliced Stereoscopic Image
2. Create Mask that shows Spliced Area
3. Create Disparity Map
4. Highlight Flat Areas and compare with Mask

**Figure 24 - Experiment Steps to Detect Non-Stereoscopic to Stereoscopic Splicing**

MATLAB 7.9.0 was used to implement the disparity map creation and highlighting algorithms. The preparation of the data set, algorithms used and measurements taken are discussed in the next subsections.

© University of Pretoria

### 4.2.1 PREPARING THE DATASET

A set of 50 stereoscopic images were chosen from self-captured images using Fujifilm's FinePix Real 3D W3 digital camera. Self-captured images were used instead of images from the internet to ensure integrity. The image set provided a variety of subjects at different distances, with different lighting conditions. Figure 25 shows the left image part of images used for the experiment.

**Figure 25 - Thumbnails of the Stereoscopic Images used for the Experiment**

© University of Pretoria

Each stereoscopic image was first converted from the original MPO file to a side-by-side image, for easier image manipulation. AgiSoft StereoScan was used to extract the stereoscopic image parts from the file. Photoshop CS 5.1 was used to create the side-by-side image and further manipulations. The right image part was used as the first image part, so that images can be viewed using a cross-eye viewing method. The side-by-side image size before splicing was 4000 x 1125 pixels.

To simulate non-stereoscopic to stereoscopic image splicing, an area in the right image part was copied to the left image part. The copied area had the same vertical position, but a different horizontal position in the left image part. The horizontal position was adjusted to place the copied area at a depth which was consistent with the background it originally was part of. By duplicating an area in this way the internal depth of the chosen area was removed, but the external depth was kept consistent with the image. This simulated the effect caused by non-stereoscopic to stereoscopic image splicing. Using part of the original image to simulate splicing ensured the consistency of some image properties which might have influence the experiment, such as colour difference. Figure 26 shows the spliced area in black, which is the silhouette of a person. Figure 27 shows an example of the splicing simulation process.

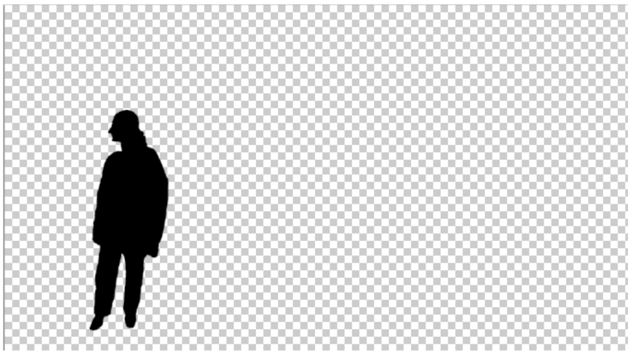


**Figure 26 - Silhouette used to Simulate Stereoscopic Image Splicing**

© University of Pretoria

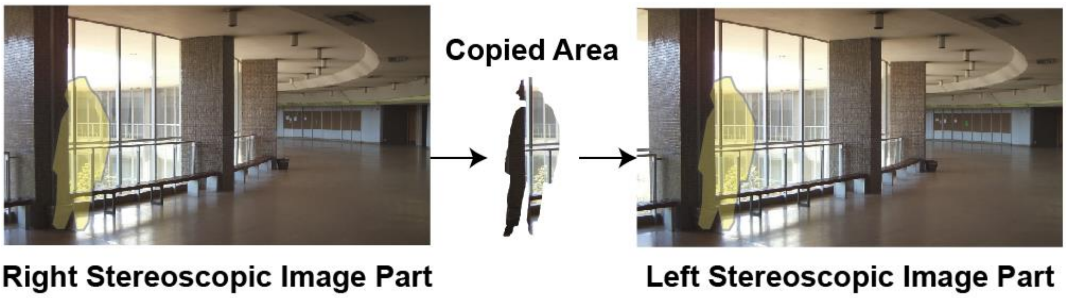


**Figure 27 - Simulating Non-Stereoscopic to Stereoscopic Image Splicing**

The same area was used to simulate splicing for every stereoscopic image. The only difference was the horizontal placement of the copied area in the left image part. This allowed the use of a single mask to validate results of the experiment. The copied area for each image also had different textures. This simulated different types of areas that might be copied from a 2D image.

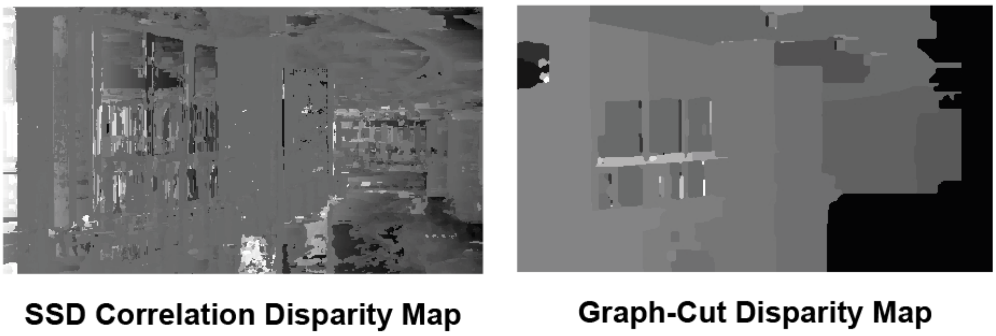
After the stereoscopic image had been manipulated, the image was saved as a 1000 x 281 pixel JPEG with 80% compression. This step was used to add noise to the image and simulate the type of transformation the image may receive before reaching a viewer. Image hosting websites, or social media websites that host images, may resize and compress images to save storage space and bandwidth.

The mask used to validate results was created from the silhouette of Figure 26, by adding a mono-colour background. For the experiment, the silhouette was converted to blue and the background to red. This simplified validation slightly, by checking the red and blue colour channels in MATLAB. For images in this dissertation, the colours in images are changed to black from blue and to white from red. This allows better visibility if this dissertation is not viewed in colour. The rest of the dissertation will refer to the black pixels as correctly identified areas and white pixels as false positives.

© University of Pretoria

### 4.2.2 CREATING THE DISPARITY MAP

An SSD correlation-based algorithm [172] was used to create a dense disparity map. A correlation based algorithm was chosen for simplicity. Other algorithms might smoothen or flatten areas to improve visual appearance. Smoothing and flattening can improve the appearance of external depth information, but remove or lessen internal depth information. Figure 28 shows a disparity map of Figure 8 created with a SSD correlations algorithm and an algorithm that uses graph cuts.


SSD Correlation Disparity Map

Graph-Cut Disparity Map

**Figure 28 - Disparity Maps Created from Different Algorithms**

For the SSD correlation algorithm used, the side-by-side stereoscopic image $I$ was first converted to grey scale with the use of MATLAB's rgb2grey function. $I$ was then split into a left image part $I_{left}$ and a right image part $I_{right}$. A window around each pixel in the left image was compared to the window of a set of pixels in the right image. The algorithm used is given as follows:

For each pixel $\{I_{left}(x_1,y_1): x_1 \in \left[b, I_{left_{width}} - b\right], y_1 \in \left[b, I_{left_{height}} - b\right]\}$ find a pixel $\{I_{right}(x_2,y_2): x_2 \in \left[\max(b, x_1 - x_{offset}), \min(I_{right_{width}} - b, x_1 + x_{offset}\right], y_2 \in \left[\max(b, y_1 - y_{offset}), \min(I_{right_{height}} - b, y_1 + y_{offset}\right]\}$ so that the SSD formula (4.1) is minimized.

© University of Pretoria

$$\sum_{u=-b}^{b}\sum_{v=-b}^{b}\left(I_{left}(u+x_1,v+y_1)-I_{right}(u+v_2,v+y_2)\right)^2 \tag{4.1}$$

The value $(x_1 - x_2)$ is calculated from $I_{left}(x_1,y_1)$ and $I_{right}(x_2,y_2)$ that provided the closest match. $(x_1 - x_2)$ is stored in a result matrix at position $(x_1,y_1)$. After the best matches for all the pixels are found, the values in the result matrix are normalized. The normalized values are used to create a grey scale disparity map, by using the normalized values as the RGB colour values.

For the formula, $b$ indicates the window size around each pixel that should be used for comparison. $x_{offset}$ determines the number of pixels left and right of $I_{right}(x_1,y_1)$ that should be used for comparison. $y_{offset}$ determines the number of pixels above and below $I_{right}(x_1,y_1)$ that should be used for comparison.

For the algorithm $b$ was chosen as $\min\left(\frac{I_{left_{width}}}{50},\frac{I_{left_{height}}}{50}\right)$, since it provided adequate results. $x_{offset}$ was chosen as $\frac{I_{left_{width}}}{5}$, since none of the images showed a horizontal disparity greater than that. $y_{offset}$ was chosen as 1, for any small vertical disparities which might be present.

### 4.2.3 HIGHLIGHTING FLAT AREAS

Highlighting large flat areas can be done in multiple ways. For the experiment each pixel in the disparity map was compared to the values of its neighbouring pixels. The size of the area around the pixel was chosen to be four times the size of the area used in the creation of the disparity map. This size was chosen, since it is small enough to highlight areas as wide as the legs of the silhouette in Figure 26. Calculating whether the area is flat was done with the following algorithm:

© University of Pretoria

For every point, $\{D(x_D, y_D): x_D \in [2b, D_{width} - 2b], y_D \in [2b, D_{height} - 2b]\}$, in disparity map $D$ calculate the result of formula (4.2).

$$\sum_{u=-2b}^{2b} \sum_{v=-2b}^{2b} |D(x_D, y_D) - D(x_D + u, y_D + v)| \tag{4.2}$$

Where $b$ is from formula (4.1). If the result of formula (4.2) is equal to zero, the area around $D(x_D, y_D)$ has no internal depth. If the area has no internal depth, copy the selected area from the mask $M$, $\{M(x_M, y_M): x_M \in [x_D - 2b, x_D + 2b], y_M \in [y_D - 2b, y_D + 2b]\}$ onto the original image $\{I(x_I, y_I): x_I \in [x_D - 2b, x_D + 2b], y_I \in [y_D - 2b, y_D + 2b]\}$.

### 4.2.4 TAKING MEASUREMENTS

Two measurements were taken during the experiment. The first measurement was the percent of correctly detected spliced area. This was calculated by comparing the number of black pixels in the highlighted image area to the number of black pixels in the mask. A high value indicates a high rate of detection. The second measurement was the percentage of false positives. This was calculated by comparing the number of white pixels in the highlighted image area to the number of white pixels in the mask.

The next section gives the experiment results and discusses these results.

## 4.3 DISCUSSION

Table 1 shows the results of the experiment. The correctly spliced area is significantly higher than the false positives.

© University of Pretoria

**Table 1 - Experiment Results for Test Set of 50 Spliced Images**

| Property | Value (mean ± $\sigma$) |
|---|---|
| Correctly Detected Spliced Area | 50.77% ± 18% of spliced image area |
| False Positives | 10.12% ± 8.32% of non-spliced image area |

From the results of Table 1 it can be seen that only around 50% of the spliced image areas were detected. Reasons for the low detection rate include the highlighting algorithm used and the textures of spliced areas.

The highlighting algorithm searches for square areas with no depth. As seen in Figure 26, the silhouette has a lot of curves, which may not be completely filled if only covered with squares. The legs of the silhouette are also barely large enough to fit a few squares. Since the disparity map algorithm searches for square areas to compare depth, the areas near the borders of the silhouette might have a different depth than the rest of the silhouette. This causes the leg areas to be even smaller than expected and can thus not be filled by the chosen block size. The feet could also not be filled by the chosen block size because they were too small. Another area of interest is the area between the legs. This area has a different depth, and will not be filled. The highlighting algorithm used thus has a few problems. It cannot fully highlight spliced areas with a lot of curves. A spliced area will not be highlighted if the spliced area is smaller than the highlighting squares used. A spliced area with holes, or a lot of discontinuities, might not be highlighted.

The area which was copied sometimes contained textures with solid colours, smooth surfaces or repeated parts. These textures make it difficult for the disparity algorithm used to identify the correct disparity of an area. Figure 29 shows an example of a copied area with such textures. The walls in the image have the same texture and make it difficult to find a matching pair of pixels. The window area does not have this problem and was successfully detected.

© University of Pretoria

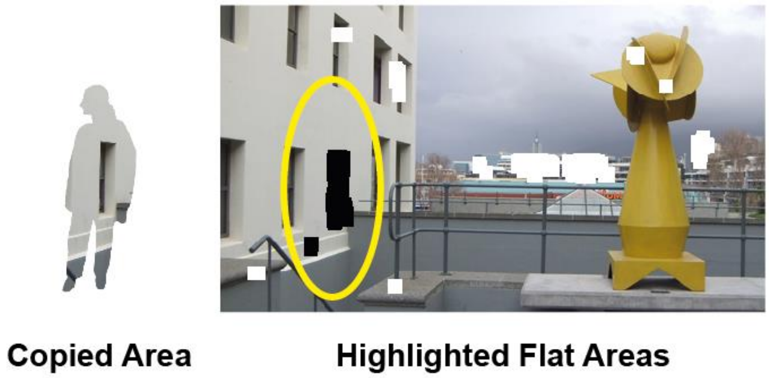


Figure 29 - Simple Textured Areas Influencing Detection Rate

From the results of Table 1 it can be seen that there was a high false positive rate of around 10%. A reason for this is that the internal depth of some areas appears flat due to the limited amount of pixels used to represent the area. Examples of such areas include, a wall facing the camera and objects at a faraway distance. Figure 30 shows examples of a few flat surfaces that are facing the camera. Figure 31 shows examples of areas at a far distance.

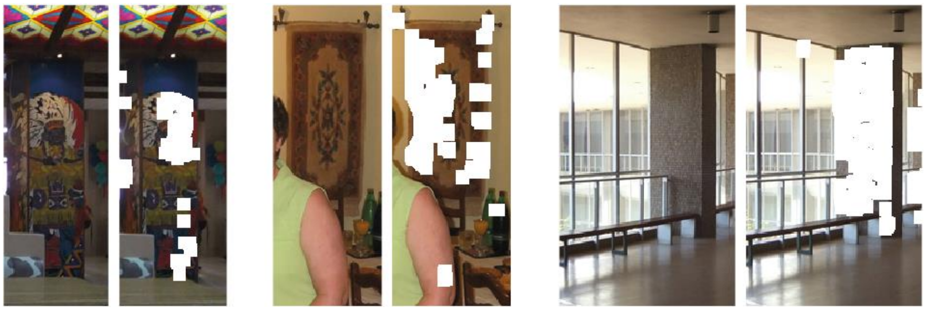


Figure 30 - False Positives from Flat Areas Facing the Camera

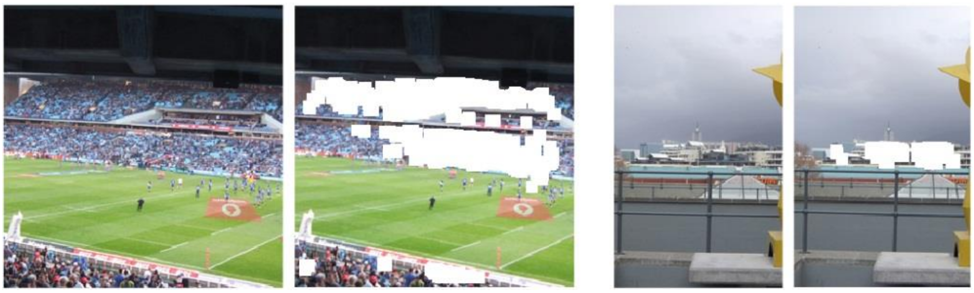


Figure 31 - False Positives from Areas at a Far Distance

© University of Pretoria

It may be possible to determine whether an area should be flat with post investigation. A concern for post investigation is the chance of false negatives. Some spliced areas, such as a poster facing the camera, may be spliced into a stereoscopic image. Detecting flat areas will reveal the poster but determining whether an area should be flat may cause the result to be dismissed as a false negative.

## 4.4 CONCLUSION

In this chapter a technique is proposed to detect simple non-stereoscopic to stereoscopic image splicing detection. The proposed technique creates a disparity map of the stereoscopic image and highlights large areas without internal depth. The technique was tested on a set of 50 spliced stereoscopic images. Results showed 50.77% of the spliced areas were detected and 10.12% of non-spliced areas were incorrectly detected.

Reasons for the low detection rate include the limited areas which can be filled by the highlighting algorithm and the texture areas of the spliced image. The highlighting algorithm struggles to fill areas with a lot of curves. The highlighting algorithm also requires spliced areas to be large enough and not contain too many holes or discontinuities. Sliced areas can also contain textures that are smooth, or repetitive, which causes problems when the disparity map is created. Using a different highlighting and disparity map creation algorithm may improve detection rate.

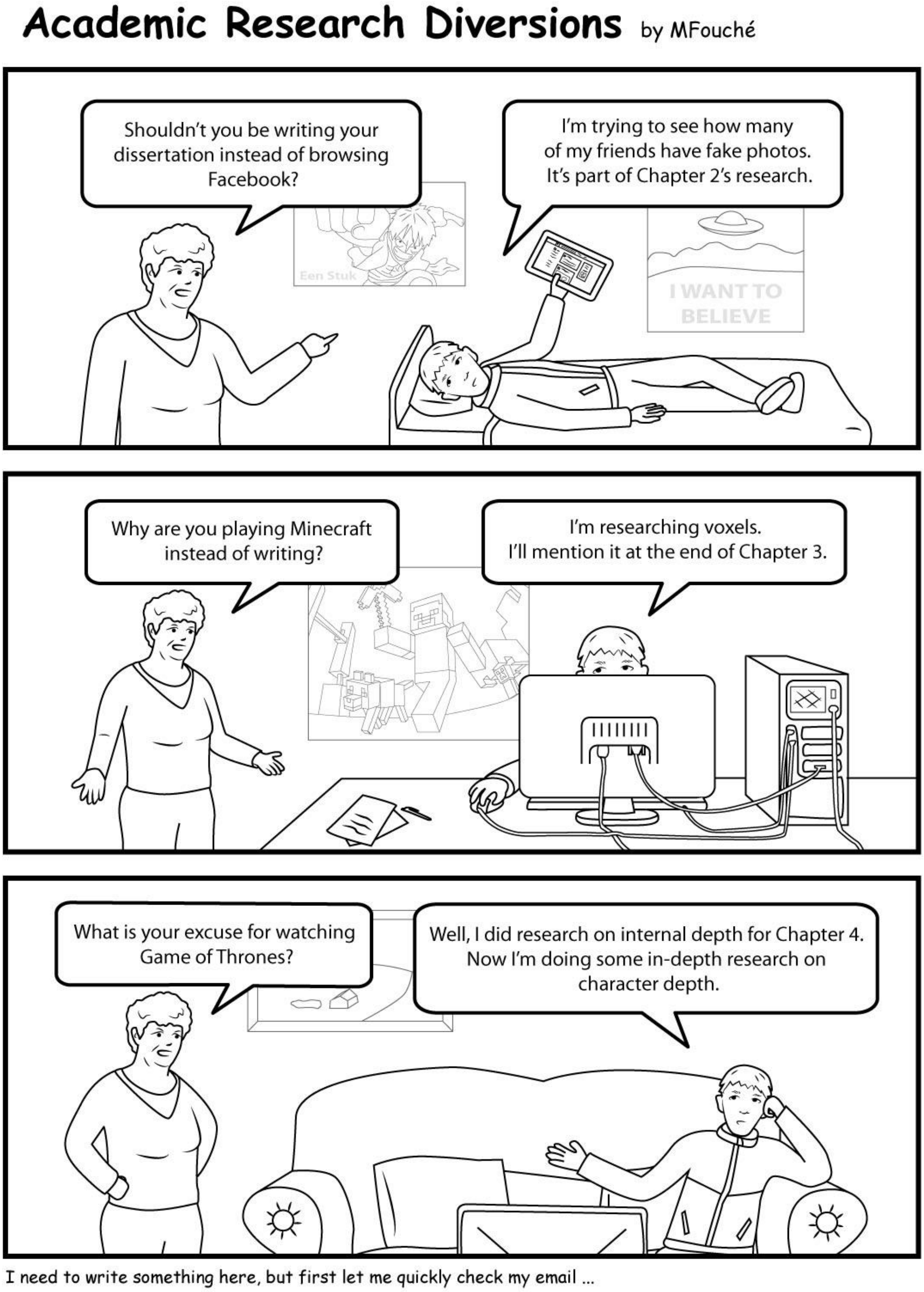
A reason for the high number of false positives is that some areas in an image may appear to have no internal depth. The lack of internal depth can be attributed to the limited amount of pixels used to represent the internal depth. Example of such areas includes flat surfaces facing the camera and areas which are far away from the camera.

© University of Pretoria

Detecting areas without internal depth, using the technique described in this chapter, has limited uses and requires post investigation of results. The technique in this chapter showed a higher correct detection rate than a false positive rate, which indicates that this technique might be useful when combined with other splicing detection techniques. On its own, the technique of this chapter is not a reliable way to detect non-stereoscopic to stereoscopic image splicing.

It is not enough to just detect the lack of internal depth. Some areas in an image will have no internal depth due to the limited amount of pixels used to represent these areas. In order to use internal depth for detection, it is necessary to know when areas are expected to have internal depth. The next chapter investigates the distance at which objects loses internal depth.

© University of Pretoria


Academic Research Diversions by MFouché
Shouldn't you be writing your dissertation instead of browsing Facebook?
I'm trying to see how many of my friends have fake photos. It's part of Chapter 2's research.
Een Stuk
I WANT TO BELIEVE
Why are you playing Minecraft instead of writing?
I'm researching voxels. I'll mention it at the end of Chapter 3.
What is your excuse for watching Game of Thrones?
Well, I did research on internal depth for Chapter 4. Now I'm doing some in-depth research on character depth.
I need to write something here, but first let me quickly check my email ...

# CHAPTER 5 - DETERMINING DISTANCE OBJECTS LOOSES INTERNAL DEPTH

## 5.1 INTRODUCTION

The previous chapter tried to identify areas without internal depth, which can be caused by simple non-stereoscopic to stereoscopic splicing. It was shown that detecting the lack of internal depth is not enough to detect spliced areas. There were areas which had no internal depth, because of the limited amount of pixels that represent the internal depth. By determining the distance at which an object loses internal depth, it is possible to know when internal depth is expected or not. Knowing when internal depth is expected can be used to aid simple non-stereoscopic to stereoscopic image splicing, as well as stereoscopic to stereoscopic image splicing.

Internal depth of an object can be calculated by calculating the difference in the distance of the nearest and furthest visible point on the object with regards to the camera. In digital stereoscopic images, a point can be represented by a single pixel. The distance of a point can be calculated by using the difference in horizontal pixel positions of the point on the left and right image of a stereoscopic image pair. At a certain distance from the camera any two points on an object would appear to have the same distance, because the difference will be less than one pixel. In this case, we will say that the object has no internal depth. If an object is spliced from a distance where it has internal depth to a point where it should not have internal depth, one may be able to still detect internal depth. This will indicate that the image was spliced. Splicing an image with no internal depth to a position where it should have internal depth could also lead to a similar anomaly, which can be used to detect splicing.

© University of Pretoria

In this chapter, a formula is derived to calculate the distance at which objects lose internal depth. This formula is tested on a set of stereoscopic images. The stereoscopic images contained chairs with target signs in the front and back to represent internal depth. These chairs were placed at different distances and photographed. The internal depth of the chairs is measured and results compared to the derived formula. A chair is then spliced from a distance where it had internal depth, to an area where it should not have internal depth and vice versa. This is done with stereoscopic to stereoscopic image splicing. The spliced chair's internal depth is measured and compared to the expected internal depth.

Section 5.2 gives the experiment design and shows the steps taken to derive the formula. Section 5.3 discusses the results.  Section 5.4 provides a conclusion.

## 5.2 METHODOLOGY

A formula was first derived to determine the depth at which objects loses internal depth. In order to test the derived formula, stereoscopic images with different properties were created. The distance at which an object loses internal depth in these images were compared with the results of the formula. Once the results of the derived formula were confirmed, steps were taken to test the use of the formula for stereoscopic image splicing detection. The steps taken for deriving the formula and the rest of the experiment are given in this section.

### 5.2.1 DERIVING THE ALGORITHM

An object will have no internal depth if the calculated distance of any two points on the object is the same. The following formula, from Mrovlje and Vrančić [171], can be used to calculate the distance ($D$), from the camera, of a single point in a digital stereoscopic image:

© University of Pretoria

$$D = \frac{Bx_0}{2\tan\left(\frac{\varphi_0}{2}\right)(x_R - x_L)} \tag{5.1}$$

$B$ is the horizontal distance between the lenses of the camera or cameras used to capture the stereoscopic image. $x_0$ is the horizontal image resolution in pixels. $\varphi_0$ is the viewing angle of the camera. $(x_R - x_L)$ is the number of horizontal pixels the point differs in the left and right image of a stereoscopic image pair. It should be noted that formula 5.1 assumes that the view direction of both lenses are parallel.

As seen in formula 5.1, the distance of a point is dependent on the number of horizontal pixels the point differs in the two images of the stereo image pair. When two points have the same difference in pixels, those points will be the same distance from the camera. If those two points are the nearest and furthest point on an object, with regards to the camera position, the object will have no internal depth. To derive a formula to determine the distance which an object loses internal depth, requires the horizontal pixel difference of two points, to be less than a pixel.

Let $D$ be the distance of the closest point on an object with a depth of $\Delta D$. The distance of the furthest point on the object will thus be $(D + \Delta D)$. Let $(x_R - x_L)$ be the number of pixels the point at $D$ differs in the left and right image of a stereoscopic image pair. Let $(x_{R2} - x_{L2})$ be the number of pixels the point at $(D + \Delta D)$ differs in the left and right image of a stereoscopic image pair. The number of pixels $(P)$ that represents the smallest amount of pixels needed, to represent internal depth for an object, can be given as:

$$P = (x_R - x_L) - (x_{R2} - x_{L2}) \tag{5.2}$$

Using (5.1), we get:

$$D + (\Delta D) = \frac{Bx_0}{2\tan\left(\frac{\varphi_0}{2}\right)(x_{R2} - x_{L2})} \tag{5.3}$$

Substituting (5.2) in (5.3) gives:

$$D + (\Delta D) = \frac{Bx_0}{2\tan\left(\frac{\varphi_0}{2}\right)\left((x_R - x_L) - P\right)} \tag{5.4}$$

Simplifying (5.4) with the use of (5.1) gives:

$$0 = \left[2P\tan\left(\frac{\varphi_0}{2}\right)\right]D^2 + \left[2P(\Delta D)\tan\left(\frac{\varphi_0}{2}\right)\right]D + \left[-B(\Delta D)x_0\right] \tag{5.5}$$

$D$ can be solved with the use of the Quadratic Formula. Since we are only interested in the positive distance in-front of the camera, solving and simplifying $D$ gives:

$$D = \frac{(\Delta D)}{2}\left[\left(\sqrt{1 + \frac{2Bx_0}{P(\Delta D)\tan\left(\frac{\varphi_0}{2}\right)}}\right) - 1\right] \tag{5.6}$$

With object depth $(\Delta D)$, difference in horizontal camera lens positions $(B)$, horizontal pixel resolution $(x_0)$, the camera viewing angle $(\varphi_0)$ and max internal depth $(P)$. The distance $(D)$ where $P = 1$, is the approximate distance at which an object starts losing internal depth.

More detailed steps for (5.4) to (5.5) and (5.5) to (5.6) can be found in Appendix A.

© University of Pretoria

### 5.2.2 TESTING THE FORMULA

The following steps were taken to create stereoscopic images to test the distance at which an object loses internal depth.

Five chairs were set up with two targets each. One target was placed in the front of the chair and one against the back of the chair. The distance between the targets were measured as 37cm and represents the internal depth of the chair.

The five chairs were placed in an arc in front of the camera position. The angle between each subsequent chair was approximately 10°, so that the chairs would fit in the frame of the cameras used. The chairs were moved and photographed 5m, 10m, 15m and 20m from the camera position. These distances were chosen since formula 5.6 indicated that the chairs will lose internal depth somewhere in that range. Figure 32 illustrates the chair setup. Figure 33 shows a photograph example of the chairs placed at 5m.

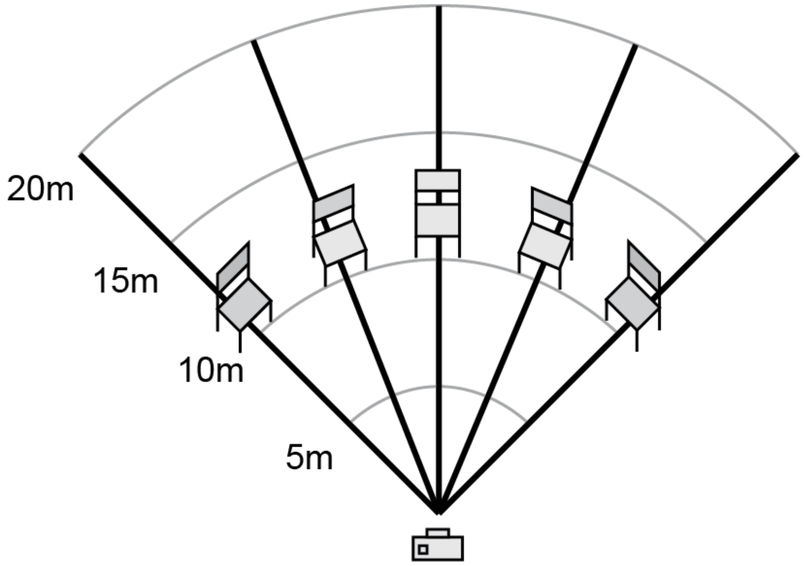


**Figure 32 - Illustration of Chair Placement at 10m**

© University of Pretoria

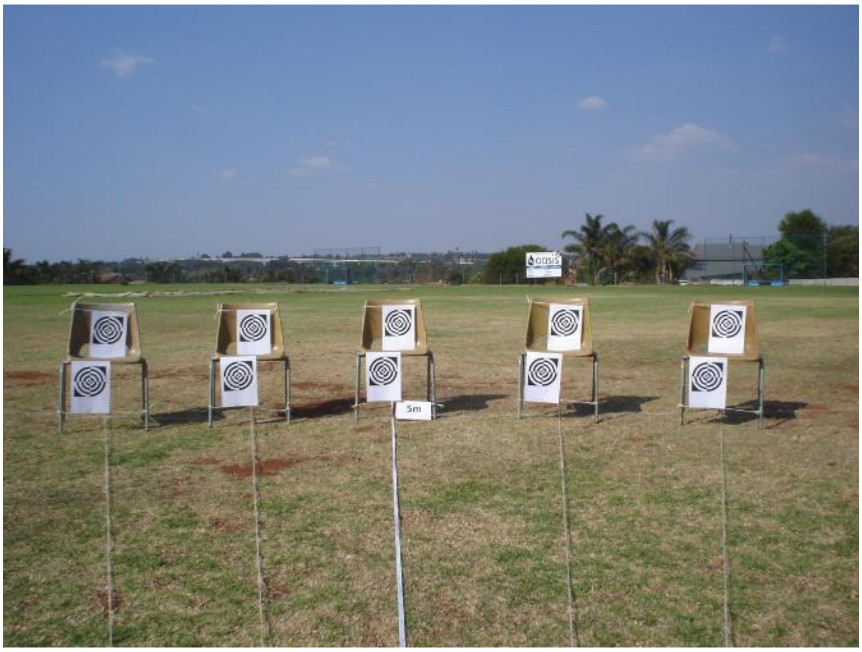


**Figure 33 - Photo of Chair Placement at 5m**

Two cameras were used to photograph the chairs. The first was a Fujifilm FinePix REAL 3D W3. Both 3D and 2D photos were taken with this camera. The second camera used was a Sony Cybershot DSC-P43. These cameras were chosen, since they provided different variables to test formula 5.6 with. When the chairs were photographed with a single lens, the camera was moved on a horizontal line at 0mm, 25mm, 50mm, 75mm and 100mm. Stereoscopic images were created using the 0mm image as the left image and the 25mm, 50mm, 75mm and 100mm images separately as the right image in the stereoscopic image pair. The reason for having stereoscopic images with different lens distances is because the distance of lenses differs on different stereoscopic image capturing devices.  This provided a way to emulate that difference. Table 2 gives more information about the cameras used. The distance at which the chairs are expected to lose internal depth, using formula 5.6, is given in Table 3.

© University of Pretoria

Table 2 - Information of Cameras used in the Experiment

| **Make** | Sony | Fujifilm | |
|---|---|---|---|
| **Model** | DSC-P43 | FinePix REAL 3D W3 | |
| **Mode** | 2D (Single Lens) | 2D (Single Lens) | 3D (Double Lenses) |
| **Resolution** | 2304 x 1728 | 3584 x 2016 | 3584 x 2016 |
| **Focal Length** | 5 mm | 6.3 mm | 6.3 mm |
| **Sensor** | 1/2.7″ CCD | 1/2.3″ CCD | 1/2.3″ CCD x2 |
| **Sensor Size** | 5.37 x 4.04 mm | 6.16 x 4.62 mm | 6.16 x 4.62 mm |
| **Approx. Angle of View** | 56.47° | 52.11° | 45° |
| **Distance of Lenses** | - | - | 75 mm |

Table 3 - Distance Chairs $(\Delta D = 0.37m)$ Start Losing Internal Depth

| **Make** | Sony | | | | Fujifilm | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Model** | DSC-P43 | | | | FinePix REAL 3D W3 | | | | |
| **Mode** | 2D | | | | 2D | | | | 3D |
| **Image Width** $(x_0)$ | 2304 | | | | 3584 | | | | 3584 |
| **Approx. Angle of View** $(\varphi)$ | 56.47° | | | | 52.11° | | | | 45° |
| **Distance of Lenses** $(B)$ | 25 mm | 50 mm | 75 mm | 100 mm | 25 mm | 50 mm | 75 mm | 100 mm | 75 mm |
| **Distance chair** $(\Delta D = 37cm)$ **start losing internal depth, using formula 5.6** $(P = 1)$ | 4.27 m | 6.12 m | 7.53 m | 8.73 m | 5.64 m | 8.05 m | 9.90 m | 11.46 m | 10.77 m |

After the chairs have been photographed at the four distances, the images were downloaded. In each image the horizontal pixel distance of the targets on each chair were measured separately and recorded. Measurements were taken in pixel units.

© University of Pretoria

### 5.2.3 Testing the Detection of Splicing

In order to test the detection of splicing, two stereoscopic images of chairs were used. Both images were taken with the Fujifilm FinePix REAL 3D W3 camera in 3D mode.

According to formula 5.6 and the information in Table 3, the distance a chair with a depth of 37cm will start to lose internal depth is approximately:

$$D = \frac{(0.37)}{2}\left[\left(\sqrt{1+\frac{2(0.075)(3584)}{(0.5)(0.37)\tan\left(\frac{45}{2}\right)}}\right)-1\right] \approx 10.8m$$

For this reason, images of chairs at 5m and 15m were used.

Two forms of stereoscopic to stereoscopic splicing were tested. First the center chair at 5m was spliced to a distance of 15m. Then a chair originally at 15m was spliced to a position of 5m. The spliced images were scaled to be the same size as the non-spliced chairs. The internal depth of the spliced chairs were measured and compared. Measurements were taken in pixel units.

## 5.3 Discussion

The discussion of results is separated into two parts. The first is the testing of formula 5.6. The second is the testing of stereoscopic to stereoscopic image splicing detection.

### 5.3.1 Testing the Formula

Table 4 gives the average measured internal depth of the five chairs for the different cameras, camera settings and chair distances. For example, the average measured

© University of Pretoria

distance of the chairs at 5m, taken with the Fujifilm camera in 3D mode, is 4.8 pixels. The asterisks (***) indicates the point where the chairs should be losing internal depth according to formula 5.6 and Table 3. All the pixel distance values above the asterisks (***) are expected to be above 1.0 and those below are expected to be less than 1.0.

**Table 4 - Average Measured Internal Depth of Chairs at Different Distances**

| **Make** | | **Sony** | | | | **Fujifilm** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | | **DSC-P43** | | | | **FinePix REAL 3D W3** | | | | |
| **Mode** | | **2D** | | | | **2D** | | | | **3D** |
| **Distance of Lenses $(B)$** | | **25 mm** | **50 mm** | **75 mm** | **100 mm** | **25 mm** | **50 mm** | **75 mm** | **100 mm** | **75 mm** |
| **Average Measured Internal Depth of the Five Chairs in Pixels** | | | | | | | | | | |
| **Chairs' Distance from Camera** | | *** | | | | | | | | |
| | **5m** | 0.6 | 1.0 | 1.8 | 2.2 | 1.0 | 1.4 | 2.6 | 3.4 | 4.8 |
| | | | *** | *** | *** | *** | *** | *** | | |
| | **10m** | 0.2 | 0.6 | 0.6 | 0.6 | 0 | 0 | 0.6 | 0.8 | 1.6 |
| | | | | | | | | | *** | *** |
| | **15m** | 0 | 0 | 0 | 0 | 0 | 0 | 0.4 | 0.6 | 0.6 |
| | | | | | | | | | | |
| | **20m** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| *** Distance chairs should loose internal depth according to formula 5.6 and Table 2. | | | | | | | | | | |

Table 4 indicates that formula 5.6 accurately predicted the point at which objects lose internal depth for most cases. There was an exception when the Fujifilm camera was used in 2D mode. When the distance of the lenses was 100mm, and the chairs were at 10m, the average internal depth was expected to be more than 1 pixel. As seen in the table, it was measured as 0.8 pixels which is less than expected. A possible explanation for this inconsistency could be due to camera lens distortion [186] and the position of the chairs relative to this distortion.

© University of Pretoria

Something else that should be noted is the internal depth values at 20m. All these values are 0, which means that none of the chairs had any internal depth at this point. Accuracy of the formula can be improved by modifying the value of $P$ in formula 5.6. A $P$ value less than 1 may be chosen to find the point at which an object will always have no internal depth. A value greater than 1 can be chosen for P to ensure that detected areas will have internal depth.

In the next subsection, the result for using formula 5.6 in stereoscopic image splicing detection is discussed.

## 5.3.2 TESTING THE DETECTION OF SPLICING

This subsection gives and discusses the results of splicing stereoscopic images before and after the point in which an object loses internal depth. The point where the object loses internal depth was calculated with the use of formula 5.6.

### SPLICING AN OBJECT FROM INTERNAL DEPTH TO NO INTERNAL DEPTH

For this experiment the center chair was spliced from a distance of 5m to a distance of 15m. The spliced chair was scaled to be the same size as the non-spliced chair in the same image. Before the spliced image was scaled, it had an internal depth of 5 pixels. The non-spliced chair had no internal depth.

After splicing and scaling, the non-spliced chair still had no internal depth, but the spliced chair had an internal depth of 2 pixels. The spliced chair thus had internal depth at a point where internal depth was not expected. This indicates that a spliced object can be detected if the object has internal depth beyond the distance calculated by formula 5.6.

© University of Pretoria

## SPLICING AN OBJECT FROM NO INTERNAL DEPTH TO INTERNAL DEPTH

For this experiment the center chair was spliced from a distance of 15m to a distance of 5m. The spliced chair was scaled to be the same size as the non-spliced chair in the same image. Before the spliced image was scaled, it had no internal depth. The non-spliced chair at 5m had an internal depth of 5 pixels.

After splicing and scaling, the non-spliced chair's internal depth was still 5 pixels. The spliced chair's internal depth was now measured to be 2 pixels, where it originally was measured to be 0 pixels. The reason for this increased internal depth can be explained with the help of graphs shown in Figure 34. When the internal depth was measured, it was done in pixel units with the use of markers, like the darkest point on an edge. When the darkest point on an edge is estimated with graphs, the distance of these points shows the increase in internal depth. The first graph shows the difference in pixel distance of the left (L) and right (R) edge, which is less than 1 pixel and thus have no internal depth. When the edges are scaled, the difference in internal depth is also scaled to be more than one pixel. Thus, when an image has an internal depth less than 1 pixel, scaling can cause it to have an internal depth of more than 1 pixel.

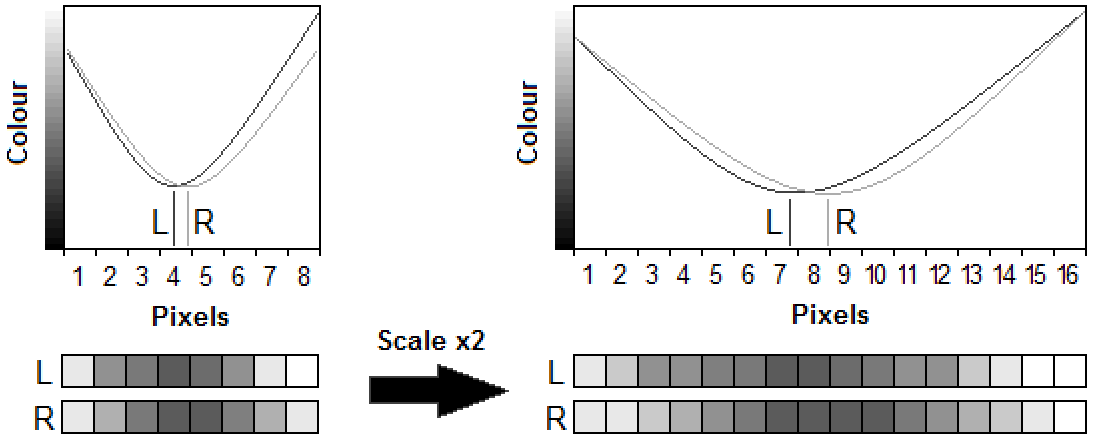


**Figure 34 - Increasing Scale Increases Measured Internal Depth**

© University of Pretoria

Results of the experiment thus indicate that an image which was spliced from a distance, where it was measured to have no internal depth, may have internal depth after scaling. However, the internal depth of the spliced object may be significantly smaller than when it was not spliced.

## 5.4 CONCLUSION

This chapter investigated how the relation between an object's distance and internal depth can aid in the detection of spliced stereoscopic images.

A formula was derived to determine the distance at which an object loses internal depth. The calculated values of the formula were compared to a set of stereoscopic images. The formula was shown to calculate a good estimate for determining at which distance an object loses internal depth.

An object was spliced from a point where it has internal depth to a point where it should not have internal depth and vice versa. The derived formula was used to calculate the point at which the object loses internal depth. Results indicate that an object scaled and spliced from an area with internal depth to an area without internal depth, can be detected. An object that was spliced from an area without internal depth to an area with internal depth, might gain internal depth if scaled, but can have a significantly smaller internal depth than expected. These results indicate that the relation of an object's physical depth to its calculated internal depth can aid the detection of stereoscopic image splicing.

There are limitations to the technique proposed in this chapter. The formula which was derived assumed a parallel camera configuration. Another formula will be needed for toed-in camera configuration. A second limitation is that the points chosen in the image need to be compared manually with the actual distance between the points. It may be possible to automatically highlight objects and search

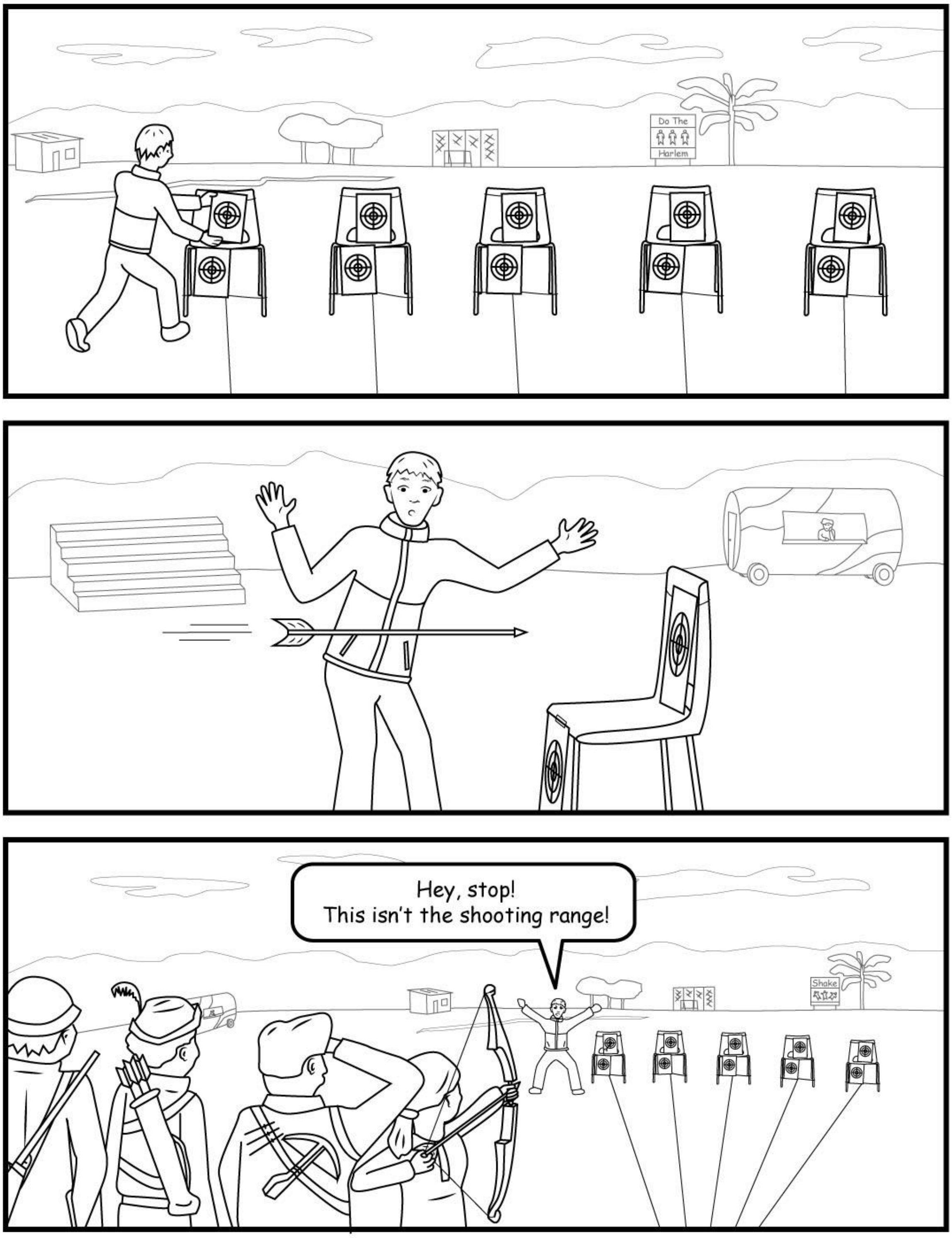
© University of Pretoria

for the closest and farthest point, but post inspection will still be required. The actual distance of the points needs to be known. A third limitation is the image properties required by the formula. These camera properties might not be available in the image's metadata, such as the camera angle of view. The angle of view will change in cropped images and also if some sort of zoom is used when capturing an image. It might however be possible to predict the necessary properties by identifying the camera used. Goljan and Fridrich [45] have shown that camera identification can be done with cropped and scaled images. They also showed that previous image dimensions of zoomed and cropped images can be predicted. A fourth limitation is the distance an object is spliced to. If the distance of an object isn't changed significantly, the proposed technique won't be able to detect the difference.

The proposed technique is not just useable for spliced images, but can also be used to aid the detection of stereoscopic image manipulation techniques which significantly change the depth of areas in an image. It may be possible to combine the proposed technique with other image manipulation techniques to strengthen detection rates.

© University of Pretoria

**Unidentified Research Risks** by MFouché



I used to be a researcher, but then I took an arrow to the knee. This excuse won't work for Stephen Hawking.

© University of Pretoria

# CHAPTER 6 - CONCLUSION

## 6.1 CONCLUSION

This dissertation investigated how stereoscopic depth information can be used to aid the detection of stereoscopic image manipulations.

Two detection techniques were created and tested with experiments. The first used disparity maps, while the second used triangulation to get depth information. These techniques focused on detecting stereoscopic image splicing. The first focused only on non-stereoscopic to stereoscopic image splicing while the second can also be used for stereoscopic to stereoscopic image splicing.

The first technique used a disparity map to highlight large areas without internal depth. When a 2D image is copied onto a stereoscopic image, the 2D image will have no internal depth. By detecting large areas without internal depth, these areas could be detected. An experiment was created with 50 spliced stereoscopic images. Large areas without internal depth were highlighted and compared with the copied area. Results showed that approximately 50% of the copied area was detected. Reasons for the low detection rate include the highlighting algorithm used and the textures of the spliced area. It may be possible to improve these results with the use of different highlighting and stereo matching algorithms. Results also showed that approximately 10% of the original stereoscopic image area was incorrectly detected. A reason for the high false positive rate is that some areas have internal depth that cannot be represented in the limited amount of pixels used in the image. These areas will also appear flat, similar to spliced areas from non-stereoscopic images. Examples of such areas include objects at a far enough distance and flat areas facing the camera, such as a wall. This technique can thus not reliably detect copied areas on its own. It can potentially be enhanced with the use of other techniques. An

© University of Pretoria

example is using this technique with depth from defocus [187]. Depth from defocus uses focal blur information of images to estimate depth information. This information may help to lower the amount of false positives that were detected by the first technique.

For the second technique, a formula was derived to determine the distance at which an object loses internal depth. The formula was tested on images with known internal depth to determine whether the formula accurately predicts the loss of depth. Results showed that the formula can predict the point at which an object loses internal depth. An object with internal depth was then copied and scaled to a depth where it was expected to have no internal depth and vice versa. The new internal depth was then compared to the expected internal depth calculated with the formula. Results showed that the internal depth of scaled and moved objects were inconsistent with the internal depth of other objects captured at the same depth. This technique is limited to objects which have been moved to a depth which is significantly different from its original depth. This technique is reliable but requires information that might not be available. Such information includes specific image capturing properties and the expected internal depth of an object. These values can however be estimated. It is possible to determine the camera used with the use of camera source identification. Values, like the angle of view, can then be estimated from the camera specification. Goljan and Fridrich [45] showed that camera identification can be done, even if the image was scaled or cropped. The scaling or cropping factor can also be determined to help estimate the camera properties needed for the second technique in this dissertation.

From the results of the two experiments it can be concluded that it is possible to use inconsistent depth information to detect stereoscopic image splicing. Other image manipulation techniques that modify depth may also be detected. The research in this dissertation is not just limited to stereoscopic images. It can potentially be used

© University of Pretoria

for any set of images that cover a part of the same scene from a similar direction. It should be noted that it is possible to fabricate the correct depth to counter the detection of depth inconsistencies.

## 6.2 PUBLICATIONS

Chapter 4 and Chapter 5 is based on the author's work which have been previously published [188][189].

## 6.3 FUTURE WORK

The techniques in this dissertation focused only on using depth information derived from disparity. It may be possible to combine this depth information with other depth cues to enhance detection. An example is focal blur, which can be used to estimate depth in a 2D image. Comparing the disparity map of a stereoscopic image with the focal blur information may show interesting results.

For the second technique, the formula was derived for a parallel camera configuration. Modifications must be made to this formula or a different formula should be derived for toed-in camera configuration. The different values required for the formula can also be investigated. If some of these values are unavailable they will need to be estimated. The accuracy of the estimations and the required accuracy for the formula will play a large role in determining the accuracy of the formula.

Only depth was looked at in this dissertation. Other properties of stereoscopic images may be investigated. An example is the difference in overall colour of the stereoscopic image parts. The image parts could have been taken by separate camera lenses, or the lighting conditions might have slightly changed when a single

© University of Pretoria

camera was used. The resulting colour variations may be used to detect image manipulations which do not take these into account.

© University of Pretoria

**The Future** by MFouché

So what do you want to do after you finish this degree?
I was thinking of maybe getting a PhD.
Oh?
Then when people ask me who I am, I can reply: "The Doctor".

At the pace I was doing this dissertation, I'll be lucky if I end up as the 20th Doctor.

© University of Pretoria

# REFERENCES

[1] P. Nillius and J. O. Eklundh, “Automatic Estimation of the Projected Light Source Direction,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001, vol. 1, pp. 1076–1083.

[2] E. Kee, J. O’brien, and H. Farid, “Exposing Photo Manipulation with Inconsistent Shadows,” *ACM Trans. Graph.*, vol. 32, no. 3, pp. 1–12, Jun. 2013.

[3] M. K. Johnson and H. Farid, “Metric Measurements on a Plane from a Single Image,” Dept. Computer Science, Dartmouth College, Technical Report TR2006-579, 2006.

[4] S. Gholap and P. K. Bora, “Illuminant Colour Based Image Forensics,” in *TENCON 2008 - 2008 IEEE Region 10 Conference*, 2008, pp. 1–5.

[5] G. Cao, Y. Zhao, and R. Ni, “Edge-Based Blur Metric for Tamper Detection,” *J. Inf. Hiding Multimed. Signal Process.*, vol. 1, no. 1, pp. 20–27, 2010.

[6] S. Bayram, H. Taha Sencar, and N. Memon, “An Efficient and Robust Method for Detecting Copy-Move Forgery,” in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 1053–1056.

[7] W. Luo, Z. Qu, F. Pan, and J. Huang, “A Survey of Passive Technology for Digital Image Forensics,” *Front. Comput. Sci. China*, vol. 1, no. 2, pp. 166–179, May 2007.

[8] T. Van Lanh, K.-S. Chong, S. Emmanuel, and M. S. Kankanhalli, “A Survey on Digital Camera Image Forensic Methods,” in *Multimedia and Expo, 2007 IEEE International Conference on*, 2007, pp. 16–19.

[9] B. Mahdian and S. Saic, “Blind Methods for Detecting Image Fakery,” in *2008 42nd Annual IEEE International Carnahan Conference on Security Technology*, 2008, vol. 25, no. 6, pp. 280–286.

[10] H. Farid, “Image Forgery Detection,” *IEEE Signal Process. Mag.*, vol. 26, no. 2, pp. 16–25, Mar. 2009.

[11] A. Rocha, W. Scheirer, T. Boult, and S. Goldenstein, “Vision of the Unseen: Current Trends and Challenges in Digital Image and Video Forensics,” *ACM Comput. Surv.*, vol. 43, no. 4, pp. 1–42, Oct. 2011.

[12] M. Harris, “Your Next Camera Will Shoot 3-D [Tools & Toys],” *IEEE Spectr.*, vol. 48, no. 1, pp. 24–24, Jan. 2011.

© University of Pretoria

[13] J. Stamas, “Sony Unveils HDR-TD10 3D Camcorder,” *CamcorderInfo.com*, 2011. [Online]. Available: http://www.camcorderinfo.com/News/Sony-Unveils-HDR-TD10-3D-Camcorder.htm. [Accessed: 29-Sep-2011].

[14] M. Anderson, “Can Mario Make the Leap? [Tools & Toys],” *IEEE Spectr.*, vol. 48, no. 6, pp. 24–26, Jun. 2011.

[15] C. Martin, “LG Optimus 3D vs HTC Evo 3D Head to Head Review,” *TheInquirer.net*, 2011. [Online]. Available: http://www.theinquirer.net/inquirer/review/2104310/lg-optimus-3d-vs-htc-evo-3d-head-head-review. [Accessed: 29-Sep-2011].

[16] “Full HD Glasses-free 3D Tablet,” *Hampoo*, 2014. [Online]. Available: http://pad.hampoo.com/product/3d. [Accessed: 03-Jul-2014].

[17] J. Schild, J. LaViola, and M. Masuch, “Understanding User Experience in Stereoscopic 3D Games,” in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI ’12*, 2012, p. 89.

[18] A. Johnson, J. Leigh, P. Morin, and P. Keken, “GeoWall: Stereoscopic Visualization for Geoscience Research and Education,” *IEEE Comput. Graph. Appl.*, vol. 26, no. 6, pp. 10–14, Nov. 2006.

[19] J. P. O. Evans, M. Robinson, and S. X. Godber, “A New Stereoscopic X-Ray Imaging Technique using a Single X-Ray Source: Theoretical Analysis,” *NDT E Int.*, vol. 29, no. 1, pp. 27–35, Feb. 1996.

[20] W. M. Grip, R. W. Grip, and R. D. Morrison, “Application of Aerial Photography and Photogrammetry in Environmental Forensic Investigations,” *Environ. Forensics*, vol. 1, no. 3, pp. 121–129, Jan. 2000.

[21] M. L. Kaiser, T. A. Kucera, J. M. Davila, O. C. Cyr, M. Guhathakurta, and E. Christian, “The STEREO Mission: An Introduction,” *Space Sci. Rev.*, vol. 136, no. 1–4, pp. 5–16, Nov. 2007.

[22] P. C. Leger, R. G. Deen, and R. G. Bonitz, “Remote Image Analysis for Mars Exploration Rover Mobility and Manipulation Operations,” in *2005 IEEE International Conference on Systems, Man and Cybernetics*, 2005, vol. 1, pp. 917–922.

[23] E. Anderton, “See LG’s Awesome 3D Poster Ads for Their 3D Sound Home Theater,” *Firstshowing.net*, 23-Apr-2012. [Online]. Available: http://www.firstshowing.net/2012/see-lgs-awesome-3d-poster-ads-for-their-3d-sound-home-theater/. [Accessed: 12-May-2012].

© University of Pretoria

[24] D. L. M. Sacchi, F. Agnoli, and E. F. Loftus, “Changing History: Doctored Photographs Affect Memory for Past Public Events,” *Appl. Cogn. Psychol.*, vol. 21, no. 8, pp. 1005–1022, Dec. 2007.

[25] T. Gitlin, *The Whole World is Watching: Mass Media in the Making & Unmaking of the New Left*. University of California Press (Berkeley), 1980.

[26] S. Reaves, J. Bush Hitchon, S.-Y. Park, and G. Woong Yun, “If Looks Could Kill: Digital Manipulation of Fashion Models,” *J. Mass Media Ethics*, vol. 19, no. 1, pp. 56–71, Mar. 2004.

[27] E. Stice and H. E. Shaw, “Adverse Effects of the Media Portrayed Thin-Ideal on Women and Linkages to Bulimic Symptomatology,” *J. Soc. Clin. Psychol.*, vol. 13, no. 3, pp. 288–308, Sep. 1994.

[28] D. Hadid, “New Israeli Law Bans Underweight Models in Ads,” *The Associated Press*, 2012. [Online]. Available: http://www.businessweek.com/ap/2012-03/D9TK8JBG0.htm. [Accessed: 09-Jul-2012].

[29] “Man Made ‘Pseudo’ Child Sex Pics,” *BBC News*, 2006. [Online]. Available: http://news.bbc.co.uk/2/hi/uk_news/england/tees/5327826.stm. [Accessed: 09-Jul-2012].

[30] S. W. Rosenberg, S. Kahn, and T. Tran, “Creating a Political Image: Shaping Appearance and Manipulating the Vote,” *Polit. Behav.*, vol. 13, no. 4, pp. 345–367, Dec. 1991.

[31] A. Hill, “Osama bin Laden Corpse Photo is Fake,” *The Guardian*, 2011. [Online]. Available: http://www.guardian.co.uk/world/2011/may/02/osama-bin-laden-photo-fake. [Accessed: 09-Jul-2012].

[32] W. S. Hwang, S. Il Roh, B. C. Lee, S. K. Kang, D. K. Kwon, S. Kim, S. J. Kim, S. W. Park, H. S. Kwon, C. K. Lee, J. B. Lee, J. M. Kim, C. Ahn, S. H. Paek, S. S. Chang, J. J. Koo, H. S. Yoon, J. H. Hwang, Y. Y. Hwang, Y. S. Park, S. K. Oh, H. S. Kim, J. H. Park, S. Y. Moon, and G. Schatten, “Patient-Specific Embryonic Stem Cells Derived from Human SCNT Blastocysts.,” *Science*, vol. 308, no. 5729, pp. 1777–83, Jun. 2005.

[33] S. Hong, “The Hwang Scandal that ‘Shook the World of Science,’” *East Asian Sci. Technol. Soc. an Int. J.*, vol. 2, no. 1, pp. 1–7, Jul. 2008.

[34] K. S. Choi, E. Y. Lam, and K. K. Y. Wong, “Source Camera Identification using Footprints from Lens Aberration,” in *Proceedings of SPIE*, 2006, vol. 6069, pp. 172–179.

© University of Pretoria

[35] C. Song, S. Sudirman, M. Merabti, and D. Llewellyn-Jones, “Analysis of Digital Image Watermark Attacks,” in *2010 7th IEEE Consumer Communications and Networking Conference*, 2010, pp. 1–5.

[36] M. M. Yeung and F. Mintzer, “An Invisible Watermarking Technique for Image Verification,” in *Proceedings of International Conference on Image Processing*, 1997, pp. 680–683.

[37] P. Blythe and J. Fridrich, “Secure Digital Camera,” in *Digital Forensics Research Workshop*, 2004, pp. 11–13.

[38] M. Kharrazi, H. T. Sencar, and N. Memon, “Blind Source Camera Identification,” in *International Conference on Image Processing, ICIP ’04.*, 2004, vol. 1, pp. 709–712.

[39] K. S. Choi, E. Y. Lam, and K. K. Y. Wong, “Automatic Source Camera Identification using the Intrinsic Lens Radial Distortion,” *Opt. Express*, vol. 14, no. 24, pp. 11551–11565, 2006.

[40] Y. Long and Y. Huang, “Image Based Source Camera Identification using Demosaicking,” in *2006 IEEE Workshop on Multimedia Signal Processing*, 2006, pp. 419–424.

[41] T. Filler, J. Fridrich, and M. Goljan, “Using Sensor Pattern Noise for Camera Model Identification,” in *2008 15th IEEE International Conference on Image Processing*, 2008, pp. 1296–1299.

[42] Z. Geradts, J. Bijhold, M. Kieft, K. Kurosawa, K. Kuroki, and N. Saitoh, “Methods for Identification of Images acquired with Digital Cameras,” in *Enabling Technologies for Law Enforcement and Security*, 2001, vol. 4232, pp. 505–512.

[43] A. C. Curtis, M. C. Kevin, and J. B. Edward, “Dark Frame Subtraction,” US6101287.

[44] J. Lukas, J. Fridrich, and M. Goljan, “Digital Camera Identification from Sensor Pattern Noise,” *IEEE Trans. Inf. Forensics Secur.*, vol. 1, no. 2, pp. 205–214, Jun. 2006.

[45] M. Goljan and J. Fridrich, “Camera Identification from Cropped and Scaled Images,” in *Proceedings of SPIE*, 2008, vol. 6819, p. 68190E–68190E–13.

[46] M. Goljan, J. Fridrich, and T. Filler, “Large Scale Test of Sensor Fingerprint Camera Identification,” in *Proceedings of SPIE*, 2009, p. 72540I–72540I–12.

© University of Pretoria

[47] A. E. Dirik, H. T. Sencar, and N. Memon, "Source Camera Identification Based on Sensor Dust Characteristics," in *IEEE Workshop on Signal Processing Applications for Public Security and Forensics*, 2007, pp. 1–6.

[48] M. S. Olivier, "Using Sensor Dirt for Toolmark Analysis of Digital Photographs," in *Advances in Digital Forensics IV*, 2008, pp. 193–206.

[49] H. Gou, A. Swaminathan, and M. Wu, "Robust Scanner Identification based on Noise Features," in *Proceedings of SPIE*, 2007, vol. 6505, p. 65050S–65050S–11.

[50] N. Khanna, A. K. Mikkilineni, and E. J. Delp, "Scanner Identification Using Feature-Based Processing and Analysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 4, no. 1, pp. 123–139, Mar. 2009.

[51] A. E. Dirik, H. T. Sencar, and N. Memon, "Flatbed Scanner Identification based on Dust and Scratches over Scanner Platen," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 1385–1388.

[52] J. Bayever, "Behind the Scenes: LIFE OF PI VFX," *ARRIChannel*, 2013. [Online]. Available: http://www.youtube.com/watch?v=pNoEYvNXO0M. [Accessed: 05-Sep-2013].

[53] T.-T. Ng and S.-F. Chang, "Identifying and Prefiltering Images," *IEEE Signal Process. Mag.*, vol. 26, no. 2, pp. 49–58, Mar. 2009.

[54] S. Lyu and H. Farid, "How realistic is photorealistic?," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 845–850, Feb. 2005.

[55] Y. Wang and P. Moulin, "On Discrimination between Photorealistic and Photographic Images," in *2006 IEEE International Conference on Acoustics Speed and Signal Processing Proceedings*, 2006, vol. 2, pp. II–161–II–164.

[56] W. Chen, Y. Q. Shi, and G. Xuan, "Identifying Computer Graphics using HSV Color Model and Statistical Moments of Characteristic Functions," in *Multimedia and Expo, 2007 IEEE International Conference on*, 2007, pp. 1123–1126.

[57] T.-T. Ng, S.-F. Chang, J. Hsu, L. Xie, and M.-P. Tsui, "Physics-Motivated Features for Distinguishing Photographic Images and Computer Graphics," in *Proceedings of the 13th annual ACM international conference on Multimedia - MULTIMEDIA '05*, 2005, p. 239.

[58] A. E. Dirik, S. Bayram, H. T. Sencar, and N. Memon, "New Features to Identify Computer Generated Images," in *2007 IEEE International Conference on Image Processing*, 2007, pp. IV – 433–IV – 436.

© University of Pretoria

[59] A. C. Gallagher, "Image Authentication by Detecting Traces of Demosaicing," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008, pp. 1–8.

[60] T.-T. Ng, S.-F. Chang, J. Hsu, and M. Pepeljugoski, "Columbia Photographic Images and Photorealistic Computer Graphics Dataset," ADVENT Technical Report #205-2004-5, Columbia University, 2005.

[61] T. Gloe, M. Kirchner, A. Winkler, and R. Böhme, "Can We Trust Digital Image Forensics?," in *Proceedings of the 15th international conference on Multimedia - MULTIMEDIA '07*, 2007, p. 78.

[62] M. Goljan, J. Fridrich, and M. Chen, "Sensor Noise Camera Identification: Countering Counter-Forensics," 2010, p. 75410S–75410S–12.

[63] H. Yu, T.-T. Ng, and Q. Sun, "Recaptured Photo Detection using Specularity Distribution," in *2008 15th IEEE International Conference on Image Processing*, 2008, pp. 3140–3143.

[64] A. R. Smith, "Color Gamut Transform Pairs," in *Proceedings of the 5th annual conference on Computer graphics and interactive techniques - SIGGRAPH '78*, 1978, pp. 12–19.

[65] M. Bellare, R. Canetti, and H. Krawczyk, "Keying Hash Functions for Message Authentication," in *CRYPTO '96 Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, 1996, pp. 1–15.

[66] R. Venkatesan, S.-M. Koon, M. H. Jakubowski, and P. Moulin, "Robust Image Hashing," in *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, 2000, pp. 664–666.

[67] S. Roy and Q. Sun, "Robust Hash for Detecting and Localizing Image Tampering," in *2007 IEEE International Conference on Image Processing*, 2007, pp. VI – 117–VI – 120.

[68] M. Wu and B. Liu, "Watermarking for Image Authentication," in *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269)*, 1998, vol. 2, pp. 437–441.

[69] J. Fridrich, "Image Watermarking for Tamper Detection," in *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269)*, 1998, vol. 2, pp. 404–408.

[70] D. Kundur and D. Hatzinakos, "Digital Watermarking for Telltale Tamper Proofing and Authentication," *Proc. IEEE*, vol. 87, no. 7, pp. 1167–1180, Jul. 1999.

© University of Pretoria

[71] M. U. Celik, G. Sharma, E. Saber, and A. M. Murat Tekalp, "Hierarchical Watermarking for Secure Image Authentication with Localization," *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 585–95, Jan. 2002.

[72] I. J. Cox and J.-P. M. G. Linnartz, "Some General Methods for Tampering with Watermarks," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 4, pp. 587–593, May 1998.

[73] J. Fridrich, D. Soukal, and J. Lukas, "Detection of Copy-Move Forgery in Digital Images," in *Proceedings of Digital Forensic Research Workshop*, 2003.

[74] B. Mahdian and S. Saic, "Detection of Copy-Move Forgery using a Method Based on Blur Moment Invariants," *Forensic Sci. Int.*, vol. 171, no. 2–3, pp. 180–9, Sep. 2007.

[75] H. Huang, W. Guo, and Y. Zhang, "Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm," in *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, 2008, pp. 272–276.

[76] D.-Y. Hsiao and S.-C. Pei, "Detecting Digital Tampering by Blur Estimation," in *First International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE'05)*, 2005, pp. 264–278.

[77] X. Wang, B. Xuan, and S. Peng, "Digital Image Forgery Detection Based on the Consistency of Defocus Blur," in *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2008, pp. 192–195.

[78] J. H. Elder and S. W. Zucker, "Local Scale Control for Edge Detection and Blur Estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 7, pp. 699–716, Jul. 1998.

[79] P. Kakar, N. Sudha, and W. Ser, "Exposing Digital Image Forgeries by Detecting Discrepancies in Motion Blur," *IEEE Trans. Multimed.*, vol. 13, no. 3, pp. 443–452, Jun. 2011.

[80] M. K. Johnson and H. Farid, "Exposing Digital Forgeries Through Specular Highlights on the Eye," *Inf. Hiding, 9th Int. Work.*, vol. 4567, pp. 311–325, 2007.

[81] J. Lukas and J. Fridrich, "Estimation of Primary Quantization Matrix in Double Compressed JPEG Images," in *Digital Forensic Research Workshop*, 2003, pp. 5–8.

[82] A. C. Popescu and H. Farid, "Exposing Digital Forgeries by Detecting Traces of Resampling," *Signal Process. IEEE Trans.*, vol. 53, no. 2, pp. 758–767, 2005.

© University of Pretoria

[83] C. Chen, Y. Q. Shi, and W. Su, "A Machine Learning Based Scheme for Double JPEG Compression Detection," in *2008 19th International Conference on Pattern Recognition*, 2008, pp. 1–4.

[84] M. K. Johnson and H. Farid, "Exposing Digital Forgeries Through Chromatic Aberration," in *Proceeding of the 8th workshop on Multimedia and security - MM&Sec '06*, 2006, p. 48.

[85] S. Voloshynovskiy, S. Pereira, T. Pun, J. J. Eggers, and J. K. Su, "Attacks on Digital Watermarks: Classification, Estimation Based Attacks, and Benchmarks," *IEEE Commun. Mag.*, vol. 39, no. 8, pp. 118–126, Aug. 2001.

[86] M. Kirchner and R. Bohme, "Hiding Traces of Resampling in Digital Images," *IEEE Trans. Inf. Forensics Secur.*, vol. 3, no. 4, pp. 582–592, Dec. 2008.

[87] M. C. Stamm, S. K. Tjoa, W. S. Lin, and K. J. R. Liu, "Anti-Forensics of JPEG Compression," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 1694–1697.

[88] B. Li, J. He, J. Huang, and Y. Q. Shi, "A Survey on Image Steganography and Steganalysis," *J. Inf. Hiding Multimed. Signal Process.*, vol. 2, no. 2, pp. 142–172, 2011.

[89] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital Image Steganography: Survey and Analysis of Current Methods," *Signal Processing*, vol. 90, no. 3, pp. 727–752, 2010.

[90] J. Mielikainen, "LSB Matching Revisited," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 285–287, May 2006.

[91] D.-C. Wu and W.-H. Tsai, "A Steganographic method for Images by Pixel-Value Differencing," *Pattern Recognit. Lett.*, vol. 24, no. 9–10, pp. 1613–1626, 2003.

[92] N. F. Johnson and S. Jajodia, "Exploring Steganography: Seeing the Unseen," *IEEE Comput.*, vol. 31, no. 2, pp. 26–34, 1998.

[93] J. Fridrich and R. Du, "Secure Steganographic Methods for Palette Images," in *Information Hiding, Third International Workshop*, 2000, vol. 1768, pp. 47–60.

[94] D. Upham, "JSteg," *Zooid.Org*. [Online]. Available: http://zooid.org/~paul/crypto/jsteg/. [Accessed: 18-Sep-2013].

[95] A. Westfeld, "F5—A Steganographic Algorithm," in *Information Hiding, 4th International Workshop*, 2001, vol. 2137, pp. 289–302.

© University of Pretoria

[96] C.-C. Chang, T.-S. Chen, and L.-Z. Chung, “A Steganographic Method based upon JPEG and Quantization Table Modification,” *Inf. Sci. (Ny).*, vol. 141, no. 1–2, pp. 123–138, 2002.

[97] N. Provos, “Defending Against Statistical Steganalysis,” in *Usinix Security Symposium*, 2001, pp. 323–336.

[98] K. Solanki, A. Sarkar, and B. S. Manjunath, “YASS: Yet Another Steganographic Scheme That Resists Blind Steganalysis,” in *Information Hiding, 9th International Workshop*, 2007, pp. 16–31.

[99] A. Westfeld and A. Pfitzmann, “Attacks on Steganographic Systems,” in *Information Hiding, Third International Workshop*, 2000, pp. 61–76.

[100] J. Fridrich, M. Goljan, and R. Du, “Reliable Detection of LSB Steganography in Color and Grayscale Images,” in *Proceedings of the 2001 Workshop on Multimedia and Security New Challenges - MM&Sec ’01*, 2001, pp. 27–30.

[101] S. Dumitrescu, X. Wu, and Z. Wang, “Detection of LSB Steganography via Sample Pair Analysis,” *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1995–2007, Jul. 2003.

[102] J. Fridrich and M. Goljan, “On Estimation of Secret Message Length in LSB Steganography in Spatial Domain,” in *Proc. SPIE 5306, Security, Steganography, and Watermarking of Multimedia Contents VI*, 2004, pp. 23–34.

[103] J. J. Harmsen and W. A. Pearlman, “Steganalysis of Additive-Noise Modelable Information Hiding,” in *Proc. SPIE 5020, Security and Watermarking of Multimedia Contents V*, 2003, pp. 131–142.

[104] A. D. Ker, “Steganalysis of LSB Matching in Grayscale Images,” *IEEE Signal Process. Lett.*, vol. 12, no. 6, pp. 441–444, Jun. 2005.

[105] X. Li, T. Zeng, and B. Yang, “Detecting LSB Matching by Applying Calibration Technique for Difference Image,” in *Proceedings of the 10th ACM workshop on Multimedia and security - MM&Sec ’08*, 2008, p. 133.

[106] J. Zhang, I. J. Cox, and G. Doerr, “Steganalysis for LSB Matching in Images with High-Frequency Noise,” in *2007 IEEE 9th Workshop on Multimedia Signal Processing*, 2007, pp. 385–388.

[107] X. Zhang and S. Wang, “Vulnerability of Pixel-Value Differencing Steganography to Histogram Analysis and Modification for Enhanced Security,” *Pattern Recognit. Lett.*, vol. 25, no. 3, pp. 331–339, 2004.

© University of Pretoria

[108] V. Sabeti, S. Samavi, M. Mahdavi, and S. Shirani, "Steganalysis of Pixel-Value Differencing Steganographic Method," in *2007 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 2007, pp. 292–295.

[109] J. Fridrich, M. Goljan, and D. Hogea, "Steganalysis of JPEG Images: Breaking the F5 Algorithm," in *Information Hiding, 5th International Workshop*, 2003, pp. 310–323.

[110] J. Fridrich, M. Goljan, and D. Hogea, "Attacking the OutGuess," in *Proceedings of the ACM Workshop on Multimedia and Security*, 2002, pp. 3–6.

[111] B. Li, J. Huang, and Y. Q. Shi, "Steganalysis of YASS," *Inf. Forensics Secur. IEEE Trans.*, vol. 4, no. 3, pp. 369–382, 2009.

[112] I. Avcibaş, N. Memon, and B. Sankur, "Steganalysis Using Image Quality Metrics," *IEEE Trans. Image Process. a Publ. IEEE Signal Process. Soc.*, vol. 12, no. 2, pp. 221–9, Jan. 2003.

[113] J. Fridrich, "Feature-Based Steganalysis for JPEG Images and Its Implications for Future Design of Steganographic Schemes," in *Information Hiding, 6th International Workshop*, 2004, pp. 67–81.

[114] T. Pevny and J. Fridrich, "Merging Markov and DCT Features for Multi-Class JPEG Steganalysis," in *Security, Steganography, and Watermarking of Multimedia Contents IX*, 2007, pp. 3–14.

[115] Y. Q. Shi, C. Chen, and W. Chen, "A Markov Process Based Approach to Effective Attacking JPEG Steganography," in *Information Hiding, 8th International Workshop*, 2007, pp. 249–264.

[116] S. Lyu and H. Farid, "Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines," in *Information Hiding, 5th International Workshop*, 2003, pp. 340–354.

[117] K. Sullivan, U. Madhow, S. Chandrasekaran, and B. S. Manjunath, "Steganalysis for Markov Cover Data With Applications to Images," *IEEE Trans. Inf. Forensics Secur.*, vol. 1, no. 2, pp. 275–287, Jun. 2006.

[118] I. P. Howard and B. J. Rogers, *Binocular Vision and Stereopsis*. Oxford University Press, 1995.

[119] L. da Vinci, *A Treatise on Painting*. trans. from the original Italian, London, 1721.

© University of Pretoria

[120] C. Wheatstone, “Contributions to the Physiology of Vision.--Part the First. On Some Remarkable, and Hitherto Unobserved, Phenomena of Binocular Vision,” *Philos. Trans. R. Soc. London*, vol. 128, pp. 371–394, 1838.

[121] A. Woods, T. Docherty, and R. Koch, “Image Distortions in Stereoscopic Video Systems,” *SPIE*, vol. 1915, pp. 36–48, 1993.

[122] S. Geng, “Perceived Depth Control in Stereoscopic Cinematography,” Durhan University, Doctoral thesis, 2012.

[123] A. Fusiello, E. Trucco, and A. Verri, “A Compact Algorithm for Rectification of Stereo Pairs,” *Mach. Vis. Appl.*, vol. 12, no. 1, pp. 16–22, Jul. 2000.

[124] L. Jovanov, A. Piurica, and W. Philips, “Wavelet-Based Stereo Images Reconstruction using Depth Images,” in *Proceedings of SPIE*, 2007, vol. 6701, p. 67012A–67012A–12.

[125] N. A. Dodgson, “Variation and Extrema of Human Interpupillary Distance,” in *Proceedings of SPIE*, 2004, vol. 5291, pp. 36–46.

[126] B. Mendiburu, *3d Movie Making: Stereoscopic Digital Cinema from Script to Screen*. Focal Press, 2009.

[127] “A Twin SLR Stereo Camera Rig Based on the Pentax K1000 Camera,” *The Victorian 3D Society, Inc*. [Online]. Available: http://www.44bx.com/vic3d/pentax_rig.html. [Accessed: 05-Sep-2013].

[128] “Devices to Synchronise Camera Shutters on Canon Cameras,” *Gentles Limited*, 2011. [Online]. Available: http://www.gentles.ltd.uk/gentstereo/sdm.htm. [Accessed: 05-Sep-2013].

[129] R. Crockett, “Twinning the Sony RM-VD1 Remote Commander for Digital Stereo Photography,” *Ledametrix.com*, 2008. [Online]. Available: http://www.ledametrix.com/remote/index.html. [Accessed: 05-Sep-2013].

[130] G. Somanath, M. Rohith, and C. Kambhamettu, “Single Camera Stereo System Using Prism and Mirrors,” in *Advances in Visual Computing*, 2010, vol. 6454, pp. 170–181.

[131] Camera & Imaging Products Association Standardization Committee, “Multi-Picture Format.” 2009.

[132] A. J. Woods and C. R. Harris, “Comparing Levels of Crosstalk with Red/Cyan, Blue/Yellow, and Green/Magenta Anaglyph 3D Glasses,” 2010, p. 75240Q–75240Q–12.

© University of Pretoria

[133] S. Klimenko, P. Frolov, L. Nikitina, and I. Nikitin, “Crosstalk Reduction in Passive Stereo-Projection Systems,” *Eurographics*, pp. 235–240, 2003.

[134] J.-C. Liou, K. Lee, F.-G. Tseng, J.-F. Huang, W.-T. Yen, and W.-L. Hsu, “Shutter Glasses Stereo LCD with a Dynamic Backlight,” in *Proceedings of SPIE*, 2009, p. 72370X–72370X–8.

[135] N. A. Dodgson, “Autostereoscopic 3D Displays,” *Computer (Long. Beach. Calif).*, vol. 38, no. 8, pp. 31–36, Aug. 2005.

[136] G. Oster, “Optical Art,” *Appl. Opt.*, vol. 4, no. 11, p. 1359, Nov. 1965.

[137] A. Hassan, “Evaluation of the Possibility to Obtain Angle Dependent Images with Inkjet Printing on Paper and Applying Lenticular and Spherical Lenses Sheets,” Royal Institute of Technology, 2009.

[138] A. Saenz, “Vuzix Displays Stereo Augmented Reality Glasses at CES 2010,” *Singularityhub.com*, 12-Jan-2010. [Online]. Available: http://singularityhub.com/2010/01/12/vuzix-displays-stereo-augmented-reality-glasses-at-ces-2010/. [Accessed: 12-May-2012].

[139] S. Dredge, “Facebook closes its $2bn Oculus Rift acquisition. What next?,” *The Guardian*, 2014. [Online]. Available: http://www.theguardian.com/technology/2014/jul/22/facebook-oculus-rift-acquisition-virtual-reality. [Accessed: 09-Aug-2014].

[140] D. Gross, “Meet Glyph, a Headset that Beams Video into Your Eyes,” *CNN*, 2014. [Online]. Available: http://edition.cnn.com/2014/02/06/tech/innovation/glyph-avegant-headset/. [Accessed: 09-Aug-2014].

[141] Z. Nauman, “Reborn Movies Making a Fortune in 3D,” *Dailytelegraph.com.au*, 06-May-2012. [Online]. Available: http://www.dailytelegraph.com.au/entertainment/insider/reborn-movies-making-a-fortune-in-3d/story-e6frewt9-1226346063826. [Accessed: 23-Sep-2013].

[142] C. Vazquez and S. Knorr, “3D-TV Content Creation: Automatic 2D-to-3D Video Conversion,” *IEEE Trans. Broadcast.*, vol. 57, no. 2, pp. 372–383, Jun. 2011.

[143] “2D to 3D Conversion Explained & Tips on Adjusting 3D Settings,” *Sumsung*, 2011. [Online]. Available: http://www.samsung.com/global/article/articleDetailView.do?atcl_id=57. [Accessed: 05-Sep-2013].

[144] S. Zabunov, “Stereo 3-D Vision in Teaching Physics,” *Phys. Teach.*, vol. 50, no. 3, p. 163, 2012.

© University of Pretoria

[145] C. Silén, S. Wirell, J. Kvist, E. Nylander, and O. Smedby, "Advanced 3D Visualization in Student-Centred Medical Education.," *Med. Teach.*, vol. 30, no. 5, pp. e115–24, Jun. 2008.

[146] A. D. Steinberg, P. G. Bashook, J. Drummond, S. Ashrafi, and M. Zefran, "Assessment of Faculty Perception of Content Validity of PerioSim, a Haptic-3D Virtual Reality Dental Training Simulator," *J. Dent. Educ.*, vol. 71, no. 12, pp. 1574–82, Dec. 2007.

[147] J. S. Henn, G. M. Lemole, M. A. T. Ferreira, L. F. Gonzalez, M. Schornak, M. C. Preul, and R. F. Spetzler, "Interactive Stereoscopic Virtual Reality: a New Tool for Neurosurgical Education. Technical Note," *J. Neurosurg.*, vol. 96, no. 1, pp. 144–9, Jan. 2002.

[148] T. A. N. Hernes, S. Ommedal, T. Lie, F. Lindseth, T. Langø, and G. Unsgaard, "Stereoscopic Navigation-Controlled Display of Preoperative MRI and Intraoperative 3D Ultrasound in Planning and Guidance of Neurosurgery: New Technology for Minimally Invasive Image-Guided Surgery Approaches," *Minim. Invasive Neurosurg.*, vol. 46, no. 3, pp. 129–37, Jun. 2003.

[149] E. O. Espinoza, M. D. Scanlan, P. J. McClure, and B. W. Baker, "Forensic Analysis of Black Coral (Order Antipatharia)," *Forensic Sci. Int.*, vol. 216, no. 1–3, pp. 73–7, Mar. 2012.

[150] M. Hanson, "Fingerprint-Based Forensics Identify Argentina's Desaparecidos," *IEEE Comput. Graph. Appl.*, vol. 20, no. 5, pp. 7–10, 2000.

[151] T. Gold, "Apollo 11 and 12 Close-up Photography," *Icarus*, vol. 12, no. 3, pp. 360–375, May 1970.

[152] H. P. Klein, J. Lederberg, A. Rich, N. H. Horowitz, V. I. Oyama, and G. V. Levin, "The Viking Mission Search for Life on Mars," *Nature*, vol. 262, no. 5563, pp. 24–27, Jul. 1976.

[153] K. Jacobsen, "High Resolution Satellite Imaging Systems - an Overview," *Photogramm. Fernerkundun Geoinf.*, vol. 2005, no. 6, pp. 487 – 496, 2005.

[154] Y. A. G. V. Boas, "Overview of Virtual Reality Technologies," University of Southhampton, Southampton, 2012.

[155] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, "Lazy Snapping," *ACM Trans. Graph.*, vol. 23, no. 3, p. 303, Aug. 2004.

[156] C. Vázquez, W. J. Tam, and F. Speranza, "Stereoscopic Imaging: Filling Disoccluded Areas in Depth Image-Based Rendering," in *SPIE 6392, Three-Dimensional TV, Video, and Display V*, 2006, p. 63920D–63920D–12.

© University of Pretoria

[157] A. Colombari, A. Fusiello, and A. Murino, "Continuous Parallax Adjustment for 3D-TV," in *IEEE European Conference on Visual Media Production*, 2005, pp. 194–200.

[158] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image Inpainting," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '00*, 2000, pp. 417–424.

[159] W. J. Tam, G. Alain, L. Zhang, T. Martin, and R. Renaud, "Smoothing Depth Maps for Improved Steroscopic Image Quality," in *SPIE 5599, Three-Dimensional TV, Video, and Display III*, 2004, pp. 162–172.

[160] S. Aguirre Valencia and R. M. Rodriguez-Dagnino, "Synthesizing Stereo 3D Views from Focus Cues in Monoscopic 2D Images," in *SPIE 5006, Stereoscopic Displays and Virtual Reality Systems*, 2003, pp. 377–388.

[161] J. Ko, M. Kim, and C. Kim, "2D-to-3D Stereoscopic Conversion: Depth-Map Estimation in a 2D Single-View Image," in *SPIE 6696, Applications of Digital Image Processing*, 2007, p. 66962A–66962A–9.

[162] A. P. Pentland, "A New Sense for Depth of Field," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 4, pp. 523–531, Jul. 1987.

[163] X. Huang, L. Wang, J. Huang, D. Li, and M. Zhang, "A Depth Extraction Method Based on Motion and Geometry for 2D to 3D Conversion," in *2009 Third International Symposium on Intelligent Information Technology Application*, 2009, pp. 294–298.

[164] Y. J. Jung, A. Baik, J. Kim, and D. Park, "A Novel 2D-to-3D Conversion Technique Based on Relative Height-Depth Cue," in *SPIE 7237, Stereoscopic Displays and Applications*, 2009, p. 72371U–72371U–8.

[165] D. A. Forsyth, "Shape from Texture without Boundaries," in *7th European Conference on Computer Vision*, 2002, pp. 225–239.

[166] K. Yamada, K. Suehiro, and H. Nakamura, "Pseudo 3D Image Generation with Simple Depth Models," in *2005 Digest of Technical Papers. International Conference on Consumer Electronics*, 2005, pp. 277–278.

[167] J. E. Cryer and M. Shah, "Shape-from-Shading: A Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 8, pp. 690–706, 1999.

[168] W. J. Tam, A. S. Yee, J. Ferreira, S. Tariq, and F. Speranza, "Stereoscopic Image Rendering Based on Depth Maps Created from Blur and Edge Information," in *SPIE 5664, Stereoscopic Displays and Virtual Reality Systems XII*, 2005, pp. 104–115.

© University of Pretoria

[169] S. Knorr and T. Sikora, "An Image-Based Rendering (IBR) Approach for Realistic Stereo View Synthesis of TV Broadcast Based on Structure from Motion," in *2007 IEEE International Conference on Image Processing*, 2007, pp. VI – 572–VI – 575.

[170] E. Rotem, K. Wolowelsky, and D. Pelz, "Automatic Video to Stereoscopic Video Conversion," in *SPIE 5664, Stereoscopic Displays and Virtual Reality Systems XII*, 2005, pp. 198–206.

[171] J. Mrovlje and D. Vrančić, "Distance Measuring Based on Stereoscopic Pictures," in *9th International PhD Workshop on Systems and Control*, 2008, vol. 2, no. October, pp. 1–6.

[172] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *Int. J. Comput. Vis.*, vol. 47, no. 1–3, pp. 7–42, 2002.

[173] T. Kanade and M. Okutomi, "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 9, pp. 920–932, 1994.

[174] S. Y. Heo, K. M. Lee, and S. U. Lee, "Robust Stereo Matching Using Adaptive Normalized Cross-Correlation," *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 33, no. 4, pp. 807–822, 2011.

[175] R. Zabih and J. Woodfill, "Non-Parametric Local Transforms for Computing Visual Correspondence," in *Third European Conference on Computer Vision*, 1994, pp. 151–158.

[176] A. F. Bobick and S. S. Intille, "Large Occlusion Stereo," *Int. J. Comput. Vis.*, vol. 33, no. 3, pp. 181–200, 1999.

[177] Y. Boykov, O. Veksler, and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, 2001.

[178] D. Scharstein and R. Szeliski, "Stereo Matching with Nonlinear Diffusion," *Int. J. Comput. Vis.*, vol. 28, no. 2, pp. 155–174, 1998.

[179] J. Sun, N.-N. Z. Zheng, and H.-Y. Shum, "Stereo Matching using Belief Propagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 787–800, Jul. 2003.

[180] S. M. Seitz and C. R. Dyer, "Photorealistic Scene Reconstruction by Voxel Coloring," *Int. J. Comput. Vis.*, vol. 35, no. 2, pp. 151–173, 1999.

© University of Pretoria

[181] O. Faugeras and R. Keriven, “Variational Principles, Surface Evolution, PDE’s, Level Set Methods and the Stereo Problem,” *IEEE Trans. Image Process. a Publ. IEEE Signal Process. Soc.*, vol. 7, no. 3, pp. 336–344, 1998.

[182] A. P. Rockwood and J. Winget, “Three-Dimensional Object Reconstruction from Two-Dimensional Images,” *Comput. Des.*, vol. 29, no. 4, pp. 279–285, 1997.

[183] S. M. Seitz, J. Diebel, D. Scharstein, and R. Szeliski, “A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 519–528.

[184] V. Kolmogorov and R. Zabih, “Multi-Camera Scene Reconstruction via Graph Cuts,” in *7th European Conference on Computer Vision*, 2002, pp. 82–96.

[185] A. Manessis, A. Hilton, P. Palmer, P. McLauchlan, and X. Shen, “Reconstruction of Scene Models from Sparse 3D Structure,” in *IEEE Conference Proceedings on Computer Vision and Pattern Recognition*, 2000, pp. 666–671.

[186] J. Wang, F. Shi, J. Zhang, and Y. Liu, “A New Calibration Model of Camera Lens Distortion,” *Pattern Recognit.*, vol. 41, no. 2, pp. 607–615, Feb. 2008.

[187] S. Zhuo and T. Sim, “Defocus Map Estimation from a Single Image,” *Pattern Recognit.*, vol. 44, no. 9, pp. 1852–1858, Sep. 2011.

[188] M.-A. Fouché and M. S. Olivier, “Detecting Non-Stereoscopic to Stereoscopic Image Splicing with the use of Disparity Maps,” in *Proceedings of the South African Institute of Computer Scientists and Information Technologists Conference on Knowledge, Innovation and Leadership in a Diverse, Multidisciplinary Environment - SAICSIT ’11*, 2011, pp. 271–274.

[189] M.-A. Fouché and M. S. Olivier, “Using Internal Depth to Aid Stereoscopic Image Splicing Detection,” in *Advances in Digital Forensics VIII*, 2012, pp. 319–333.

© University of Pretoria

# APPENDIXES

## APPENDIX A - DERIVING FORMULA 5.6

This appendix contains more detailed information for the steps between formula 5.4 and 5.5, as well as between formula 5.5 and 5.6.

### FROM (5.4) TO (5.5) THE FOLLOWING STEPS WERE TAKEN:

$$D+(\Delta D)=\frac{Bx_0}{2\tan\left(\frac{\varphi_0}{2}\right)((x_R-x_L)-P)} \qquad \text{...(5.4)}$$

$$\left((x_R-x_L)-P\right)=\frac{Bx_0}{2\tan\left(\frac{\varphi_0}{2}\right)(D+(\Delta D))}$$

$$(x_R-x_L)=\frac{Bx_0}{2\tan\left(\frac{\varphi_0}{2}\right)(D+(\Delta D))}+P$$

$$D+(\Delta D)=\frac{Bx_0}{2\tan\left(\frac{\varphi_0}{2}\right)(x_R-x_L)}+\frac{P(D+(\Delta D))}{(x_R-x_L)} \qquad \text{...multiplied } \frac{(D+(\Delta D))}{(x_R-x_L)} \text{ both sides}$$

$$D+(\Delta D)=D+\frac{P(D+(\Delta D))}{(x_R-x_L)} \qquad \text{...using (5.1)}$$

$$0=\frac{P(D+(\Delta D))}{(x_R-x_L)}-(\Delta D)$$

$$0=P(D+(\Delta D))-(\Delta D)(x_R-x_L) \qquad \text{...multiplied } (x_R-x_L) \text{ both sides}$$

$$0=P(D+(\Delta D))-(\Delta D)\left(\frac{Bx_0}{2\tan\left(\frac{\varphi_0}{2}\right)(D)}\right) \qquad \text{...using (5.1)}$$

$$0=P(D+(\Delta D))(2\tan\left(\frac{\varphi_0}{2}\right)(D))-(\Delta D)Bx_0 \qquad \text{...multiplied } \left(2\tan\left(\frac{\varphi_0}{2}\right)(D)\right) \text{ both sides}$$

$$0=\left[2P\tan\left(\frac{\varphi_0}{2}\right)\right]D^2+\left[2P(\Delta D)\tan\left(\frac{\varphi_0}{2}\right)\right]D+\left[-B(\Delta D)x_0\right] \qquad \text{... (5.5)}$$

## FROM (5.5) TO (5.6) THE FOLLOWING STEPS WERE TAKEN:

Let $a = 2P\tan\left(\frac{\varphi_0}{2}\right)$

Thus (5.5): $0 = [a]D^2 + [a(\Delta D)]D + [-B(\Delta D)x_0]$

Solving $D$ using the Quadric Formula: $0 = ax^2 + bx + c \quad => \quad \frac{-b\pm\sqrt{b^2-4ac}}{2a}$

$$D = \frac{1}{2a}\left[-a(\Delta D) \pm \sqrt{a^2(\Delta D)^2 - 4a(-B(\Delta D)x_0)}\right]$$

$$D = \frac{1}{2a}\left[-a(\Delta D) \pm \sqrt{a^2(\Delta D)^2 + 4aB(\Delta D)x_0}\right]$$

$$D = \frac{1}{2a}\left[-a(\Delta D) \pm \sqrt{a^2(\Delta D)^2(1 + \frac{4Bx_0}{a(\Delta D)})}\right]$$

$$D = \frac{1}{2a}\left[-a(\Delta D) \pm a(\Delta D)\sqrt{1 + \frac{4Bx_0}{a(\Delta D)}}\right]$$

$$D = \frac{(\Delta D)}{2}\left[-1 \pm \sqrt{1 + \frac{4Bx_0}{a(\Delta D)}}\right]$$

Since we are only interested in the positive distance in front of the camera:

$$D = \frac{(\Delta D)}{2}\left[\left(\sqrt{1 + \frac{4Bx_0}{a(\Delta D)}}\right) - 1\right]$$

Substituting back $a = 2P\tan\left(\frac{\varphi_0}{2}\right)$ gives:

$$D = \frac{(\Delta D)}{2}\left[\left(\sqrt{1 + \frac{4Bx_0}{2P\tan\left(\frac{\varphi_0}{2}\right)(\Delta D)}}\right) - 1\right]$$

$$D = \frac{(\Delta D)}{2}\left[\left(\sqrt{1 + \frac{2Bx_0}{P\,(\Delta D)\tan\left(\frac{\varphi_0}{2}\right)}}\right) - 1\right] \qquad \text{... (5.6)}$$

This is how the formula in Chapter 5 was derived.

© University of Pretoria