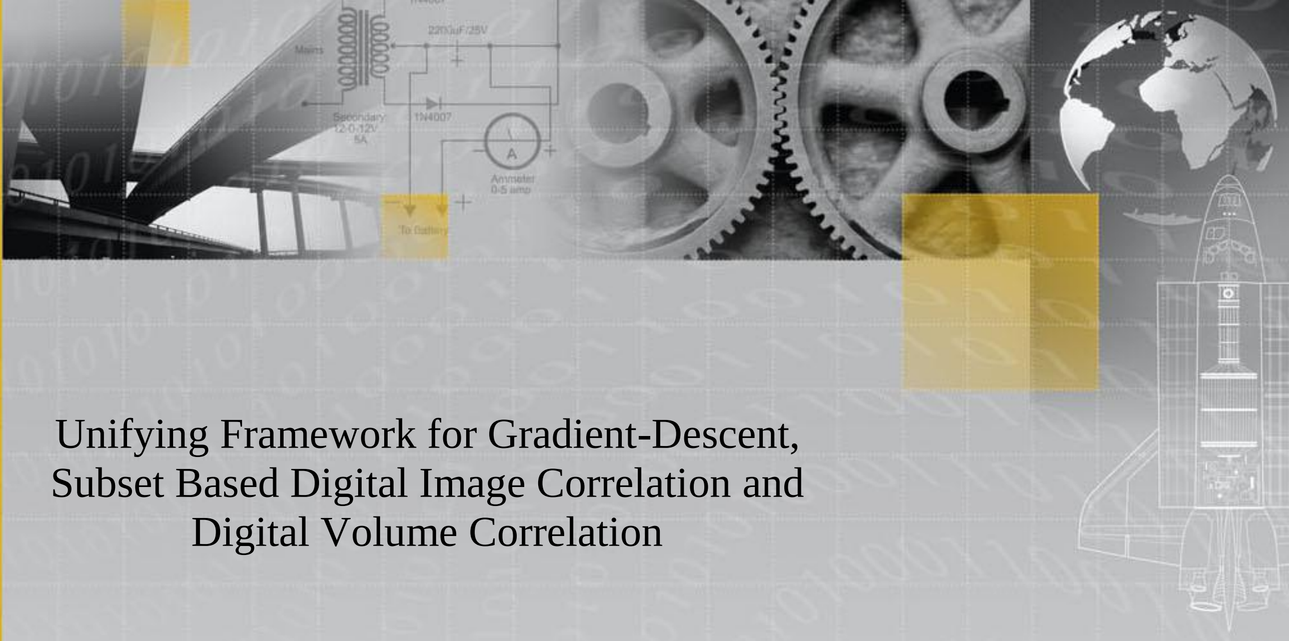
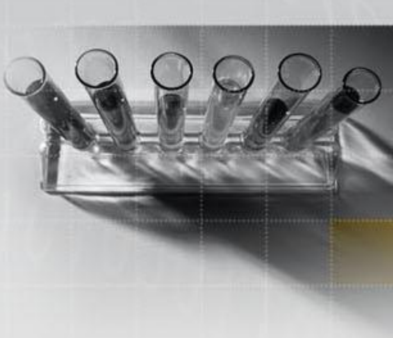


Unifying Framework for Gradient-Descent, Subset Based Digital Image Correlation and Digital Volume Correlation

Devan Atkinson
17732913



2017

Unifying Framework for Gradient-Descent, Subset Based Digital Image Correlation and Digital Volume Correlation

MEng Research Thesis Proposal

Supervisor: Dr Thorsten Becker

Devan Atkinson

Department of Mechanical and Mechatronic Engineering

2017

Department of Mechanical and Mechatronic Engineering
Stellenbosch University

Declaration

I know that plagiarism is wrong.

Plagiarism is to use another's work (even if it is summarised, translated or rephrased) and pretend that it is one's own.

This assignment is my own work.

Each contribution to and quotation (e.g. "cut and paste") in this assignment from the work(s) of other people has been explicitly attributed, and has been cited and referenced. In addition to being explicitly attributed, all quotations are enclosed in inverted commas, and long quotations are additionally in indented paragraphs.

I have not allowed, and will not allow, anyone to use my work (in paper, graphics, electronic, verbal or any other format) with the intention of passing it off as his/her own work.

I know that a mark of zero may be awarded to assignments with plagiarism and also that no opportunity be given to submit an improved assignment.

I know that students involved in plagiarism will be reported to the Registrar and/or the Central Disciplinary Committee.

Name: ....

Student no: ....

Signature: ....

Date: ....

Abstract

Digital Image Correlation (DIC) and Digital Volume Correlation (DVC) are becoming widely used tools, in the field of material science, to measure the displacements and deformations of specimens. However the high cost of commercial DIC and DVC software and the limited control offered over the correlation process in both commercial and open-source software limits its widespread adoption. This paper proposes a project which aims to create a Matlab based program capable of performing 2D DIC and DVC on a given set of images while allowing the user control over the correlation process. It is hypothesised that the different DIC algorithms use different methods to perform the same tasks in the correlation process. Thus by analysing these algorithms it is possible to combine these different methods into one program which will allow the user to select which methods to use for the various correlation tasks. If successful, this project will produce a freely available program, which offers many of the correlation methods of the different DIC algorithms, so that users will be able to perform in depth DIC and DVC analyses.

Contents

List of Figures

Chapter 1

Introduction

Chapter 2

Theory

This chapter outlines the theory that is relevant for Digital Image Correlation. First the digital camera is discussed in order to understand how it captures light and stores it as data. Thereafter the optical system, made of a system of lenses and apertures, which focuses the light for the digital camera is reviewed since it has a significant influence on how a three dimensional scene is converted into two dimensional data. Then the camera model that is used to mathematically relate three dimensional coordinates in the real world to two dimensional pixels in the image is presented. Lastly the distortions that are caused by the slight imperfections of lenses are explained.

2.1 Digital cameras

Cameras at the basic level rely upon using an aperture and a lens to focus light rays that originate from objects onto a plane within the camera called the sensor plane. At the sensor plane exist a Charge-Coupled Device (CCD) that consists of a matrix of light sensors. Each light sensor converts the light incident upon its surface into an electrical charge through the photoelectric effect. The charge is proportional to the light intensity. Then each light sensor's voltage is read by an analogue-to-digital converter which measures the voltage and assigns a digital value to it. These digital values are then stored at the corresponding position in a matrix which forms the digital image.

2.2 Camera optics

For an image of an object to be in focus the light incident upon each light sensor should originate from one point on the object's surface. However light is reflected by objects in many directions and so a means of focusing the light is required. This is accomplished through using lenses and an aperture. Throughout this project thin lenses are assumed. A thin lens is one in which its thickness is negligible in comparison with its focal length or radius of curvature [6]. Additionally the paraxial approximation is assumed which states that light rays passing through the lens do so with a small angle to the lens's optical axis and pass through the lens close to the optical axis. This leads to the small angle approximation.

$$\sin(\theta) \approx \tan(\theta) \approx \theta \tag{2.1}$$

Lenses are usually disk shaped pieces of glass with two convex surfaces. The convex surfaces are designed to bend light towards the optical axis with the degree of bending increasing with the distance from the light to optical axis at the lens mid plane. Thus diffuse light emanating from a point M $[x, y, z]^T$ on an object will pass through the lens and the light rays will converge at a point called the ideal image point M' $[x', y', z']^T$. Thereafter the light rays diverge again to points M" $[x'', y'', z'']^T$ as shown in figure 2.1. If the sensor plane is coincident with the ideal image points of the light rays the image that forms on the sensor is inverted due to the way the lens bends the light. As a result an inverted coordinate system is used for the sensor plane.

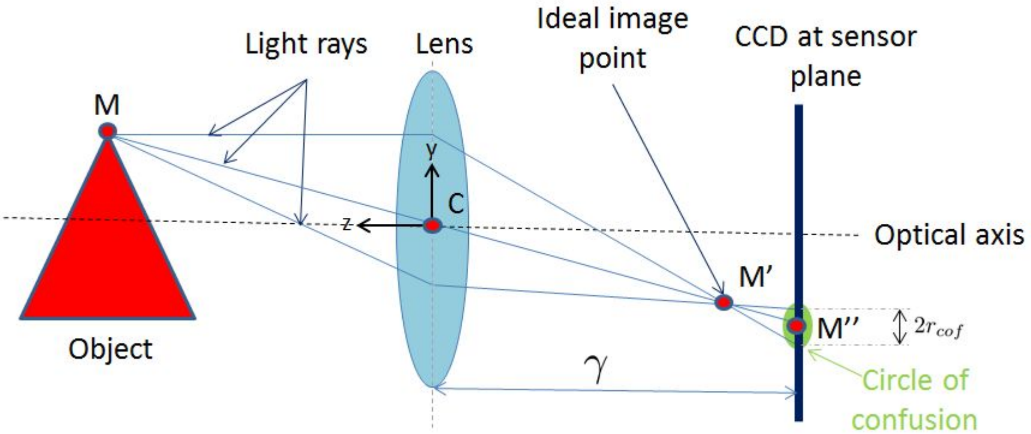


Figure 2.1: Illustration of how light rays are manipulated by the lens

The thin lens equation can be stated as

$$\frac{1}{|CM|} + \frac{1}{CM'} = \frac{1}{\bar{f}} \tag{2.2}$$

where $\bar{f}$ is the focal length; an inherent property of the lens. Additionally through similarity of triangles we have

$$\frac{y}{CM} = \frac{-y'}{CM'} \tag{2.3}$$
$$\frac{x}{CM} = \frac{-x'}{CM'} \tag{2.4}$$
$$\frac{z}{CM} = \frac{-z'}{CM'}. \tag{2.5}$$

Combining these a series of equations for the ideal image point can be obtained.

$$\begin{aligned} x' &= \frac{-\bar{f}x}{z-\bar{f}} \\ y' &= \frac{-\bar{f}y}{z-\bar{f}} \\ z' &= \frac{-\bar{f}z}{z-\bar{f}} \end{aligned} \tag{2.6}$$

2.2.1 Circle of confusion

It is impossible to align the sensor plane perfectly with the ideal image points and so the divergence of the light rays after the ideal image point causes the light rays to illuminate a circular area on the sensor plane. This blurred region is known as the circle of confusion. If this circular area is larger than the area spanned by an element of the sensor then this will result in blurring in the image as the light spills over multiple sensors. This blur could be eliminated by having the sensor plane the same distance from the lens as the the ideal image point but different points on the object will have different ideal image points.

The light rays that make up the outer perimeter of the circle of confusion are the rays that pass through the lens on the outer edges. Using these rays it is possible to determine the radius of the circle of confusion. Taking r to be the radius of the lens these outer light rays pass through the midplane of the lens at $[r\cos\beta, r\sin\beta, 0]^T$ where $0 < \beta < 2\pi$. Using trigonometry the y components of these light rays can be related.

$$\frac{y' - r\sin\beta}{z'} = \frac{y' - y''}{\gamma + z'} \tag{2.7}$$

Th distance between the lens and the sensor plane is given by γ. The same can be done for the x component. Using this and equation 2.6 an expression for the perimeter of the circle of confusion can be found.

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} r\cos\beta + r\cos\beta\left(1 + \frac{\gamma(\bar{f}-z)}{\bar{f}z}\right) \\ r\sin\beta + r\sin\beta\left(1 + \frac{\gamma(\bar{f}-z)}{\bar{f}z}\right) \\ 0 \end{bmatrix} + \begin{bmatrix} -\frac{\gamma x}{z} \\ -\frac{\gamma y}{z} \\ -\gamma \end{bmatrix} \tag{2.8}$$

The radius of the circle of confusion can be determine by taking the difference in the y components of y'' for $\beta = 90°$ and $\beta = 270°$.

$$2r_{cof} = \left[r\sin 90°\left(2 + \frac{\gamma\left(\bar{f}-z\right)}{\bar{f}z}\right)\right] - \left[r\sin 270°\left(2 + \frac{\gamma\left(\bar{f}-z\right)}{\bar{f}z}\right)\right] \tag{2.9}$$

$$r_{cof} = r\left(1 + \frac{\gamma\left(\bar{f}-z\right)}{\bar{f}z}\right) \tag{2.10}$$

Another way that the blur can be improved is by using an aperture. An aperture is at a basic level an opaque diaphragm with a hole in it which serves to reduce the amount of light that is incident upon the sensor. Thus by blocking off the light that passed through the outer edges of the lens the aperture reduces the size of the circle of confusion by effectively reducing the radius of the lens. The aperture can be place in front or behind the lens.

2.2.2 Depth of field

Depth of field is defined as the distance ahead and behind the object that is in focus. For a particular camera system the ideal image point of an object will fall upon the sensor plane if the object is at an ideal focal length from the lens. If the object is any closer or further from the lens it will cause a circle of confusion on the sensor plane as opposed to a point. If the circle of confusion is small enough it will not spill significant light over multiple sensors which will result in a clear image.

However if the circle of confusion is large enough it will cause blurring in the image. The largest circle of confusion which still results in a sharp image is referred to as the acceptable circle of confusion [6]. Thus the depth of field can be considered as the distance along the z-axis ahead or behind the ideal focus length which results in an acceptable circle of confusion.

The largest degree of freedom for a specific camera system occurs when the object is at the hyperfocal length, H, from the camera. The hyperfocal length is defined as the closest an object can be to the camera such that the depth of field extends to infinity behind the object. In this situation the depth of field starts at a distance of $\frac{H}{2}$. The hyperfocal length can be approximated as

$$H \simeq \frac{\bar{f}^2}{2Nr_{cof}} \tag{2.11}$$

Here N is given by $\frac{\bar{f}}{D_p}$ where D_p is the diameter of the entrance pupil for the camera system. Letting s be the distance from the camera to the object such that the camera is ideally focused at a distance s. The distance from the camera to the near limit of the depth of field, D_N, and the distance from the camera to the far limit of the depth of field, D_F, can be approximated.

$$D_N \simeq \frac{Hs}{H+s} \tag{2.12}$$

$$D_F \simeq \frac{Hs}{H-s} \tag{2.13}$$

These approximations assume that the object distance is large compared to the lens focal length. The depth of field can then be determined.

$$DOF = D_F - D_N = \frac{2Hs^2}{H^2 - s^2} \tag{2.14}$$

Combining equation 2.11 and 2.14

$$DOF = \frac{4Nr_{cof}\bar{f}^2 s^2}{\bar{f}^4 - 4N^2 r_{cof}^2 s^2}. \tag{2.15}$$

Thus it is clear that the depth of field can be controlled by altering the focal length of the lens, by altering N by changing the aperture or changing the distance between the camera and the object.

2.2.3 Field of view

The field of view is the extent of the world that the camera is capable of capturing in an image. It is quantified as the largest angle that a light ray, that is incident upon the sensor, makes with the optical axis. This angle is referred to as the angle-of-view.

viewing frustum

Using the pinhole camera model with a distance of L between the sensor and the lens and taking the lens height to be d a relation for the angle-of-view, α, can be derived using trigonometry.

$$\tan\left(\frac{\alpha}{2}\right) = \frac{d}{2L} \tag{2.16}$$

$$\alpha = 2\arctan\left(\frac{d}{2L}\right) \tag{2.17}$$

However for the best picture quality the distance between the sensor and the lens should be equal to the focal length f.

$$\alpha = 2\arctan\left(\frac{d}{2f}\right) \tag{2.18}$$

2.2.4 Transformation to image plane

The relation between an object point and the projection of the object point onto the sensor can be derived from equation 2.8 by eliminating the circle of confusion component. Additionally when the camera system is set up properly z' will be equal to γ so that the sensor plane is approximately coincident with the ideal image points.

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} -\frac{\gamma}{z} & 0 & 0 \\ 0 & -\frac{\gamma}{z} & 0 \\ 0 & 0 & -\gamma \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.19}$$

This equation presents two issues. Firstly the dependence of the sensor positions on z. Secondly all the terms in the equation have metric units whereas the image coordinate system has dimensions measured in pixels. These issues are fixed by using a homogeneous from of equation 2.19.

$$\alpha \begin{bmatrix} -x'' \\ -y'' \\ 1 \end{bmatrix} = \begin{bmatrix} \gamma & 0 & 0 & 0 \\ 0 & \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.20}$$

Here α is a scale factor which allows for the conversion from metric units for the world coordinate system to pixels used in the sensor coordinate system. This equation is referred to as perspective projection [6].

2.2.5 Front image plane model

A new imaging model that is often preferred for computer vision applications can be obtained by translating the sensor plane a distance of 2γ along the optical axis. In this case the sensor plane is in front of the lens. Treating the sensor coordinates M'' of an object as the intersection of the light ray with the sensor plane; equation 2.20 remains valid for this configuration.

This imaging model is advantageous in that the sensor plane coordinates are no longer inverted. This means that the scene being imaged is also not inverted.

2.3 Coordinate systems

Images are only capable of storing two dimensional information whereas we live in a three dimensional world. Thus cameras convert three dimensional information from the world coordinate system into two dimensional information in the sensor coordinate system when a picture is taken as shown in figure 2.2. The mathematical relationship between these two coordinate systems is discussed here.

describe how homogeneous coordinates allow n dim vectors to be manipulated in n+1 dim space

2.3.1 Homogeneous coordinates

It is common knowledge that any object's shape can be fully defined using distances and angles in 3D Euclidean space. However when an image is taken of this object these distances and angles become distorted. For example railway tracks consist of two beams that remain parallel to one another at a set distance apart in Euclidean space but in an image of the railway track (projective space) these beams appear to get closer and closer to one another as seen in figure 2.3. Therefore the parallelism between the beams in Euclidean space is

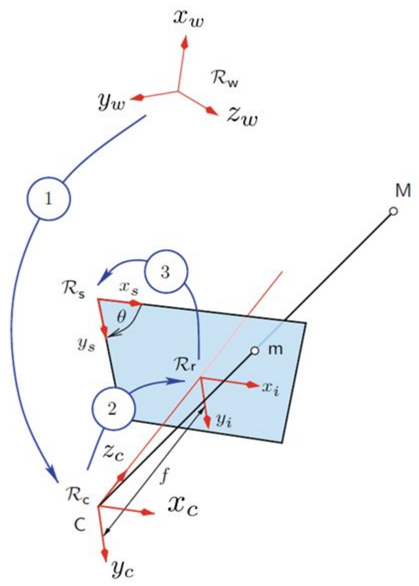


Figure 2.2: The conversion between coordinate systems that occurs when an image is taken

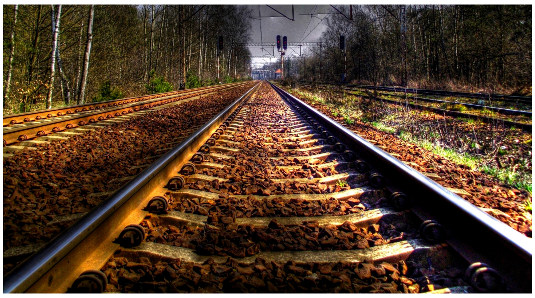


Figure 2.3: How projective space distorts parallelism present in Euclidean space

distorted in projective space. This occurs as a result of reducing 3D information to a 2D image.

add image

Homogeneous coordinates make it possible to take a 3D coordinate in Euclidean space into a 4D coordinate in projective space. This is necessary since a transformation matrix for 3D coordinates that applies both rotation and translation has 4 columns and so can only be applied to a 4D coordinate. Conversion to homogeneous coordinates involves adding an additional coordinate w to the coordinate vector and setting $w = 1$. To convert back from homogeneous coordinates requires dividing the x, y and z coordinates by w and then eliminating w.

do a 2D example

2.3.2 World to camera coordinate system

Converting from the world coordinate system to the camera coordinate system involves rigid transformations of translation T and rotation R. The world coordinate system is simply the coordinate system that would be used to classify the position and orientation of objects in the real world. The world coordinate systems orientation and origin is somewhat arbitrary and it is usually classified according to the object to be imaged.

The camera coordinate system is as the name suggests fixed according to the cameras position and orientation. The camera coordinate system's z axis is the optical axis of the camera. The conversion between the two coordinate systems can be represented as

$$\boldsymbol{X}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{T} \\ \boldsymbol{0} & 1 \end{bmatrix} \boldsymbol{X}_w. \tag{2.21}$$

Scaling of dimensions between the two coordinate systems also needs to be taken into account. The parameters in $\boldsymbol{R}$ and $\boldsymbol{T}$ are referred to as extrinsic parameters since they are not dependent on the camera system's hardware but rather the camera's position and orientation in the world coordinate system.

change T and R to lowercase

2.3.3 Camera and imaging plane coordinates

The 3D object defined in the camera coordinate system is converted to its projection on imaging plane. This conversion is from a 3D coordinate system to a 2D coordinate system as given below using homogeneous coordinates.

$$\boldsymbol{X}_p = \alpha \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} \gamma & 0 & 0 & 0 \\ 0 & \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \tag{2.22}$$

The variable α is an arbitrary scale factor.

2.3.4 Image plane and sensor coordinates

At this point the locations of where the light rays originating from the object will intersect the imaging plane are known. Now it is necessary to mathematically represent how the sensor would interpret these light rays incident upon the sensor into the form of pixels. In order to do this a relation between the position of a point within a coordinate system and the pixel within an image is necessary. Additionally the sensor array is not guaranteed to the orthogonal and so a skewed coordinate system must be taken into account.

The transformation from the image plane coordinate system to a temporary skewed coordinate system with an angle of ϕ between the two axes can be represented by

$$\begin{bmatrix} x_{temp} \\ y_{temp} \end{bmatrix} = \begin{bmatrix} 1 & -\cot\phi \\ 0 & \frac{1}{\sin\phi} \end{bmatrix} \begin{bmatrix} x_p \\ y_p \end{bmatrix}. \tag{2.23}$$

It is assumed that the two principle directions in the sensor coordinate system have different scale factors, S_x and S_y,which have units of pixels per unit length. Applying these to the coordinates calculated in equation 2.23 and accounting for the translations $\hat{c}_x$ and $\hat{c}_y$ to convert to the origin of the sensor coordinate system results in the sensor coordinates below.

$$\begin{bmatrix} x_s \\ y_s \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x_{temp} \\ y_{temp} \end{bmatrix} - \begin{bmatrix} S_x\hat{c}_x - S_x\hat{c}_y\cot\phi \\ \frac{S_y\hat{c}_y}{\sin\phi} \end{bmatrix} = \begin{bmatrix} S_x & -S_x\cot\phi \\ 0 & \frac{S_y}{\sin\phi} \end{bmatrix} \begin{bmatrix} x_p \\ y_p \end{bmatrix} - \begin{bmatrix} S_x\hat{c}_x - S_x\hat{c}_y\cot\phi \\ \frac{S_y\hat{c}_y}{\sin\phi} \end{bmatrix} \tag{2.24}$$

This can be rewritten in homogeneous form so that it is consistent with equation 2.22.

$$\boldsymbol{X}_s = \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & -S_x\cot\phi & -S_x\left(\hat{c}_x - \hat{c}_y\cot\phi\right) \\ 0 & \frac{S_y}{\sin\phi} & -\frac{S_y\hat{c}_y}{\sin\phi} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \boldsymbol{A}\boldsymbol{X}_i \tag{2.25}$$

talk about these parameters being intrinsic and combine them better

2.3.5 World to sensor coordinates

The conversions between coordinate systems described in the previous sections can be combined into one conversion from the world coordinate system to the sensor coordinate system.

$$\boldsymbol{X}_s = \alpha \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & -S_x \cot\phi & -S_x\left(\hat{c}_x - \hat{c}_y \cot\phi\right) \\ 0 & \frac{S_y}{\sin\phi} & -\frac{S_y \hat{c}_y}{\sin\phi} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \gamma & 0 & 0 & 0 \\ 0 & \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{2.26}$$

This can be simplified by combining the first two matrices.

$$\boldsymbol{X}_s = \alpha \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \begin{bmatrix} \gamma S_x & -\gamma S_x \cot\phi & -S_x\left(\hat{c}_x - \hat{c}_y \cot\phi\right) & 0 \\ 0 & \frac{\gamma S_y}{\sin\phi} & -\frac{S_y \hat{c}_y}{\sin\phi} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{2.27}$$

Replacing the elements of the first matrix in equation 2.27 with single variables for the purposes of simplicity the equation can be rewritten as

$$\alpha \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & f_s & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{2.28}$$

$$= \boldsymbol{K}\boldsymbol{V}\boldsymbol{X}_w. \tag{2.29}$$

Here the parameters relating the world coordinate system to the sensor coordinate system are separated into two matrices. The first matrix $\boldsymbol{K}$ contains the intrinsic parameters which are fixed for a specific camera. The second matrix $\boldsymbol{V}$ contains the extrinsic parameters. The extrinsic parameters change when the camera's position and orientation relative to the world coordinate system changes.

explain this better

2.4 Distortion

Distortion refers to a collection of phenomena that cause the actual image to differ from that of the idealised image expected from the pinhole camera model. This happens because lenses cannot be manufactured and assembled perfectly and so some misalignment and defects exist in the imaging system. Therefore in order to calculate accurate displacement information the distortions must be accounted for and corrected prior to the correlation process.

Distortion can be separated into many types as done below. The overall distortion of the image can then be mathematically represented as the linear combination of the mathematical expressions for the individual types. A brief explanation of each distortion type and the equation that accounts for the distortion it creates is given below.

2.4.1 Spherical distortion

Spherical distortion is when light rays originating from a point on an object intersect at different points on the sensor plane. This is as a result of the lens not having the perfect curvature required for all the light rays to intersect at the same point on the sensor plane.

It is assumed to be axis symmetric relative to the axis passing through the centre of the sensor and to be a function of the radial distance.

$$\boldsymbol{D} = \kappa_1 \rho^4 \boldsymbol{e}_r \tag{2.30}$$

Here $\boldsymbol{e}_r$ is the radial unit vector and ρ is the distance from the origin of the sensor coordinate system to the point under consideration.

2.4.2 Coma distortion

Coma distortion affects light rays that travel towards the lens at an angle to the optical axis. Light rays that go through the centre portion of lens refocus to a point on the sensor plane, whereas light rays that pass through the outer portion of the lens don't refocus fully and intersect the sensor plane further (positive coma) or closer (negative coma) to the optical axis. This results in light from a point on an object creating a comet-like shape on the sensor plane. It is corrected with the following equation.

$$\boldsymbol{D} = \kappa_2 \rho^3 \cos\left(\bar{\theta} - \bar{\theta}_c\right) \boldsymbol{e}_r \tag{2.31}$$

Here $\bar{\theta}_c$ is the orientation of the projected lens tilt angle in the sensor plane.

2.4.3 Astigmatism

Astigmatism is caused by a lens having curvature that varies when measured along perpendicular planes. For instance the vertical plane of the lens has a different curvature to the horizontal plane of the lens. The different curvatures have different focal lengths and so light passing through the vertical portion of the lens will focus at a different distance from the lens than light that passes through the horizontal portion of the lens. This results in line like blurs on the sensor plane for the curvature that is not in focus. The degree to which the astigmatism affects the image increases further away from the centre of the image.

$$\boldsymbol{D} = \kappa_3 \rho^2 \cos\left(\bar{\theta} - \bar{\theta}_A\right) \boldsymbol{e}_r \tag{2.32}$$

Where $\bar{\theta}_A$ is the orientation of the projected astigmatic plane within the sensor plane.

2.4.4 Curvature of field

Curvature of field is a distortion that results due to the curved nature of the optical elements such as the lens. As light rays pass through the lens with a small angle to the optical axis they refocus at the sensor plane. However light rays that pass through the lens at a larger angle to the optical axis refocus at a point that is closer to the lens. This results in the light rays refocusing on a curved plane much like a shallow dome. Since the sensor is planar (flat) this causes the centre of the image to be in focus while the edges of the image are not in focus.

Curvature of field is assumed to be symmetric with respect to the optical axis and to be a quadratic function of the radial position [6].

$$\boldsymbol{D} = \kappa_4 \rho^2 \boldsymbol{e}_r \tag{2.33}$$

Here κ_4 is the amplitude of the curvature of distortion in the sensor plane measured in pixels^{-1}.

2.4.5 Linear

This non-symmetric distortion component is assumed to be a linear function of radial position and is dependent on the angular position [6].

$$\boldsymbol{D} = \kappa_5 \rho \cos\left(\bar{\theta} - \bar{\theta_L}\right) \boldsymbol{e}_r \tag{2.34}$$

Here $\bar{\theta_L}$ is the angular orientation of the linear distortion axis.

2.4.6 Radial

Radial distortion is caused by the lens having different magnification levels based on the angle of the light rays to the optical axis. As a result the image can experience a decrease in magnification with increasing distance from the optical axis (barrel distortion) or the image can experience increasing magnification with increasing distance from the optical axis (pincushion distortion). This distortion is symmetric with respect to the optical axis.

$$\boldsymbol{D} = \kappa_6 \rho^3 \boldsymbol{e}_r + \kappa_7 \rho^5 \boldsymbol{e}_r + \kappa_8 \rho^7 \boldsymbol{e}_r \tag{2.35}$$

2.4.7 De-centering

De-centering distortion is caused by the lens not being in perfect alignment with the rest of the camera system. Usually this type of distortion is less severe than radial or spherical distortions.

$$\boldsymbol{D} = \kappa_9 \rho^2 \left[3 \sin\left(\bar{\theta} - \bar{\theta_d}\right) \boldsymbol{e}_r + \cos\left(\bar{\theta} - \bar{\theta_d}\right) \boldsymbol{e}_t\right] \tag{2.36}$$

Here $\boldsymbol{e}_t$ is the tangential unit vector and $\bar{\theta_d}$ is the orientation of the axis for maximum tangential distortion.

Chapter 3

Calibration

Calibration is a necessary process that must be completed in order to extract metric information from images. The calibration process solves for parameters that define the optical characteristics of the camera, parameters that define the orientation and position of the camera coordinate system to the world coordinate system and parameters that define the distortions that must be corrected for in the images. By determining all of these parameters the calibration process will have solved the camera model that predicts how an object in three dimensional space is related to the pixel, which containing the light reflected by the object, in the image captured of the object.

The parameters that define the optical characteristics and the distortions are referred to as intrinsic parameters since they are fixed for a specific camera. The parameters that describe the orientation and position of the camera within the world coordinate system are extrinsic parameters because they change if the camera is moved.

explain intrinsic and extrinsic better

Multiple methods of calibration exist however calibration using a calibration plate is used in this project since it is one of the most popular methods and manufacturing a calibration plate is relatively simple and inexpensive.

3.1 Inverse problem

Calibration is essentially a method of finding the parameters that allow 3D coordinates in the world coordinate system to be accurately related to 2D coordinates in the image coordinate system. Thus the inputs, 3D world coordinates, and outputs, 2D image coordinates, needed to be used to solve for the parameters which describe the relationship between the two which constitutes an inverse problem.

Inverse problems are typlically hard to solve and calibration is no exception. In order to solve for the calibration parameters more reliably the camera model that relates world points to sensor points is broken down to the simple pinhole camera model initially in order to solve for as few parameters as possible in the beginning. These parameters are solved for using a closed form solution which gives good estimates to the parameters. Thereafter once these parameters have been estimated the camera model is made more complex by introducing radial distortion in order to account for imperfections in the lens system of the camera. These radial parameters are first estimated and now with each parameter having a corresponding estimate all the parameters are optimized simultaneously in an iterative manner.

This inverse problem is sensitive to errors in the 3D world coordinates (inputs) and 2D image coordinates (outputs) used. Thus these need to be known to a high degree of accuracy in order to solve for the intrinsic and extrinsic camera parameters reliably. This is achieved by using a calibration plate.

3.2 Calibration plate

A calibration plate is an object with a flat surface containing a high-contrast, regular pattern. The pattern is such that it contains definitive, point-like features which can be located to a high degree of accuracy within images taken of it. For example a checker board pattern allows for accurate calculation of the points at the corners of the squares. Thus the coordinates of these point-like features on the calibration plate will be known (inputs) and the coordinates of these point-like features in the image can be determined to a high degree of accuracy (outputs).

Since images inherently contain some level of noise it is best to have an overdetermined system of equations. This is accomplished by taking multiple images of the calibration plate and changing the relative position and orientation between the calibration plate and the camera for each image. This effectively reduces the effects of noise; making the system more robust.

3.3 Homography

Homography is a transformation that can be applied to points on a plane to bring it into alignment with another plane. It is used to bring the points on the calibration plate in the world coordinate system into alignment with their location in the image in the sensor coordinate system. The transformation from world coordinates to sensor coordinates in equation 2.29 is a type of homography.

Treating the calibration plate such that it lies in the x-y plane of the world coordinate system; the homography, $\boldsymbol{H}$, for calibration is given by the following equation.

$$\begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \alpha \begin{bmatrix} f_x & f_s & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 0 \\ 1 \end{bmatrix} \tag{3.1}$$

This can be reduced to

$$\begin{aligned} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} &= \alpha \begin{bmatrix} f_x & f_s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} && (3.2) \\ &= \alpha \boldsymbol{K} \begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{t} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} && (3.3) \\ &= \alpha \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} = \alpha \boldsymbol{H} \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} && (3.4) \end{aligned}$$

Here $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are the first and second columns of the rotation matrix. It is clear that the homography matrix contains both intrinsic and extrinsic parameters and thus it is different for each image taken of the calibration plate. Additionally note that the homography matrix is defined up to a scale factor α.

3.4 Estimating homography with direct linear transformation

The homographies that relate the world coordinates of the calibration plate targets to the targets in the image can be estimated using direct linear transformation (DLT) [2]. Equation 3.4 can be written out as

$$x_s = \alpha\left(h_1 x_w + h_2 y_w + h_3\right) \tag{3.5}$$

$$y_s = \alpha\left(h_4 x_w + h_5 y_w + h_6\right) \tag{3.6}$$

$$1 = \alpha\left(h_7 x_w + h_8 y_w + h_9\right). \tag{3.7}$$

The scale factor, α, can be eliminated by dividing equations 3.5 and 3.6 by 3.7 to get

$$x_s\left(h_7 x_w + h_8 y_w + h_9\right) = \left(h_1 x_w + h_2 y_w + h_3\right) \tag{3.8}$$

$$y_s\left(h_7 x_w + h_8 y_w + h_9\right) = \left(h_4 x_w + h_5 y_w + h_6\right) \tag{3.9}$$

These equations then reduce to

$$-h_y x_w x_s - h_8 y_w x_s + h_1 x_w + h_2 y_w + h_3 = h_9 x_s \tag{3.10}$$

$$-h_7 x_w y_s - h_8 y_w y_s + h_4 x_w + h_5 y_w + h_6 = h_9 y_s. \tag{3.11}$$

In order to avoid the trivial solution where every element in the homogrpahy matrix is equal to zero; constraints need to be placed on the elements of the homography matrix. In this case the element h_9 is set equal to 1 however other constraints are also possible such as $h_7^2 + h_8^2 + h_9^2 = 1$. Note that if the true value of h_9 is close to zero then this assumption will introduce a singluarity [9].

Each target, having points x_{w_i} and y_{w_i}, on the calibration plate that is captured within an image of the calibration plate, having points x_{s_i} and y_{s_i}, provides two equations of the form of equations 3.10 and 3.11. These are then combined into an equation of the form

$$\begin{bmatrix} x_{w_1} & y_{w_1} & 1 & 0 & 0 & 0 & -x_{w_1}x_{s_1} & -y_{w_1}x_{s_1} \\ 0 & 0 & 0 & x_{w_1} & y_{w_1} & 1 & -x_{w_1}y_{s_1} & -y_{w_1}y_{s_1} \\ x_{w_2} & y_{w_2} & 1 & 0 & 0 & 0 & -x_{w_2}x_{s_2} & -y_{w_2}x_{s_2} \\ 0 & 0 & 0 & x_{w_2} & y_{w_2} & 1 & -x_{w_2}y_{s_2} & -y_{w_2}y_{s_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \end{bmatrix} = \begin{bmatrix} x_{s_1} \\ y_{s_1} \\ x_{s_2} \\ y_{s_2} \\ \vdots \end{bmatrix} \tag{3.12}$$

This overdetermined system of equations then can be used to solve for the homography matrices of each calibration image using least squares. This is the first step involved in calibration.

speak about normalisation

3.4.1 Direct linear transformation normalisation

In practice the used of the DLT method to find estimates to the homographies was found to be sensitive to noise in the images. It was discovered by R. Hartley [5] that the DLT method can be made much more robust to noise by applying translation and scaling to the image and world coordinate points prior to performing DLT. Specifically the image and world coordinate points are first translated such that the origin of each coordinate system is located at the centroid of the points defined in that coordinate system. Then the coordinates are scaled such that the average distance of the coordinates from the origin is equal to $\sqrt{2}$.

This scaling and translation has the effect of reducing the condition number of the matrix in equation 3.12.

The translation and scaling of the coordinates is done separately for the image and world coordinates as shown with $\boldsymbol{X'_w}$ and $\boldsymbol{X'_s}$ representing the transformed and scaled versions of the world and image coordinate points respectively.

$$\boldsymbol{X'_w} = \boldsymbol{T_w X_w} \tag{3.13}$$

$$\boldsymbol{X'_s} = \boldsymbol{T_s X_s} \tag{3.14}$$

Since different coordinate points are used for DLT the homography computed, $\boldsymbol{H'}$, is not the homography of the original coordinate pairs, $\boldsymbol{H}$. However the homography calculated can be transformed back in order to obtain the desired homography matrix.

$$\boldsymbol{H} = \boldsymbol{T_s^{-1} H' T_w} \tag{3.15}$$

3.5 Absolute conic

As mentioned before two objects that are parallel in Euclidean space appear to intersect each other in projective space. The intersection of these two lines in projective space occurs at a point that lies on the plane at infinity. If a point lies upon the plane at infinity its w is equal to zero in homogeneous coordinates.

The absolute conic lies on the plane at infinity and is defined by the set of points, $\tilde{\boldsymbol{x}}_{ac} = [x, y, z, w]^T$, that satisfies the following.

$$w = 0 \tag{3.16}$$

$$\boldsymbol{x}_{ac}^T \boldsymbol{x}_{ac} = x^2 + y^2 + z^2 = 0 \tag{3.17}$$

Here tilde is used to indicate homogeneous coordinates whereas $\boldsymbol{x}_{ac} = [x, y, z]^T$ would be the point in Euclidean space.

Thus it is this conic of purely imaginary points that lies upon the plane at infinity. The importance of the absolute conic is that it is invariant under any Euclidean transformations. In other words the relative position of the absolute conic to a moving camera is unaffected by extrinsic parameters. Consider a point $\boldsymbol{x}_{ac}$ in Euclidean space in the world coordinate system that lies on the absolute conic. It has homogeneous coordinates $\tilde{\boldsymbol{x}}_{ac} = [\boldsymbol{x}_{ac}^T, 0]^T$ in projective space. Applying a different rotation and translation to this point to obtain $\tilde{\boldsymbol{x}}'_{ac}$, which is equivalent to moving the camera in the world coordinate system, a corresponding point is obtained in the camera coordinate system.

$$\tilde{\boldsymbol{x}}'_{ac} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{T} \\ \boldsymbol{0} & 1 \end{bmatrix} \tilde{\boldsymbol{x}}_{ac} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{ac} \\ y_{ac} \\ z_{ac} \\ 0 \end{bmatrix} \tag{3.18}$$

$$= \begin{bmatrix} \boldsymbol{R x}_{ac} \\ 0 \end{bmatrix} \tag{3.19}$$

It is clear that this point still lies on the plane at infinity since its w is equal to zero. More importantly it can be proven that this point $\boldsymbol{x}'_{ac}$ is on the same absolute conic.

$$\boldsymbol{x}_{ac}^{\prime T} \boldsymbol{x}'_{ac} = (\boldsymbol{R x}_{ac})^T (\boldsymbol{R x}_{ac}) = \boldsymbol{x}_{ac}^T \boldsymbol{R}^T \boldsymbol{R x}_{ac} = \boldsymbol{x}_{ac}^T \boldsymbol{x}_{ac} = 0 \tag{3.20}$$

Thus the absolute conic is invariant to Euclidean transformations. Now consider again a point, $\boldsymbol{x}_{ac}$, lying on the absolute conic. The corresponding point, $\boldsymbol{m}_s$, in the sensor plane is given by

$$\tilde{\boldsymbol{m}}_s = \frac{1}{\alpha}\boldsymbol{K}\begin{bmatrix}\boldsymbol{R} & \boldsymbol{T}\\ \boldsymbol{0} & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_{ac}\\ 0\end{bmatrix} = \frac{1}{\alpha}\boldsymbol{K}\begin{bmatrix}\boldsymbol{R}\boldsymbol{x}_{ac}\\ 0\end{bmatrix} \tag{3.21}$$

$$\therefore \boldsymbol{m}_s = \frac{1}{\alpha}\boldsymbol{K}\boldsymbol{R}\boldsymbol{x}_{ac}. \tag{3.22}$$

Checking whether this point satisfies equation 3.17 results in

$$\boldsymbol{m}_s^T\boldsymbol{K}^{-T}\boldsymbol{K}^{-1}\boldsymbol{m}_s = \frac{1}{\alpha^2}\boldsymbol{x}_{ac}^T\boldsymbol{R}^T\boldsymbol{R}\boldsymbol{x}_{ac} = \frac{1}{\alpha^2}\boldsymbol{x}_{ac}^T\boldsymbol{x}_{ac} = 0 \tag{3.23}$$

Thus the image of the absolute conic is itself an absolute conic. The image of the absolute conic is defined by $\boldsymbol{K}^{-T}\boldsymbol{K}^{-1}$ [3]. Thus since the image of the absolute conic is dependent only on intrinsic camera parameters it can be used to solve for the intrinsic camera parameters.

absolute or imaginary conic?

3.6 Constraints on intrinsic parameters

According to Zhang [9] there are two constraints placed upon the intrinsic parameters of the camera. These are important later on in the solving for these intrinsic parameters. The plane of the calibration plate in the camera coordinate system is given by [9]

$$\begin{bmatrix}\boldsymbol{r}_3\\ \boldsymbol{r}_3^T\boldsymbol{t}\end{bmatrix}^T\begin{bmatrix}x\\ y\\ z\\ w\end{bmatrix} = 0. \tag{3.24}$$

Here w is zero for points on the plane at infinity and one for those that are not. This plane intersects the plane at infinity on a line and it happens that $[\boldsymbol{r}_1^T, 0]^T$ and $[\boldsymbol{r}_2^T, 0]^T$ are points on this line [9]. Thus it is known that any point on this line, x_c^∞, is a linear combination of these two points.

$$x_c^\infty = a\begin{bmatrix}\boldsymbol{r}_1\\ 0\end{bmatrix} + b\begin{bmatrix}\boldsymbol{r}_2\\ 0\end{bmatrix} = \begin{bmatrix}a\boldsymbol{r}_1 + b\boldsymbol{r}_2\\ 0\end{bmatrix} \tag{3.25}$$

Now assume that this point, x_c^∞, lies on the absolute conic. Then this point must satisfy equation 3.17. This would require that $a^2 + b^2 = 0$ which results in the solution $b = \pm ai$, where $i = \sqrt{-1}$. As a result it can be seen that the two points along this line intersect the absolute conic at

$$x_c^\infty = a\begin{bmatrix}\boldsymbol{r}_1 \pm i\boldsymbol{r}_{\boldsymbol{2}}\\ 0\end{bmatrix}. \tag{3.26}$$

Since these points lie on the absolute conic they are invariant under Euclidean transformations. Their projection, up to a scale factor, in the sensor coordinate system is given by

$$x_s^\infty = \boldsymbol{K}\left(\boldsymbol{r}_1 \pm \boldsymbol{r}_2\right) = \boldsymbol{h}_1 \pm i\boldsymbol{h}_2. \tag{3.27}$$

Substituting these points into equation 3.17 results in

$$\left(\boldsymbol{h}_1 \pm i\boldsymbol{h}_2\right)^T\boldsymbol{K}^{-T}\boldsymbol{K}^{-1}\left(\boldsymbol{h}_1 \pm i\boldsymbol{h}_2\right) = 0. \tag{3.28}$$

Requiring both the imaginary and real parts of this equation to equal zero results in two constraints on the intrinsic camera parameters.

$$\boldsymbol{h}_1^T\boldsymbol{K}^{-T}\boldsymbol{K}^{-1}\boldsymbol{h}_2 = 0 \tag{3.29}$$

$$\boldsymbol{h}_1^T\boldsymbol{K}^{-T}\boldsymbol{K}^{-1}\boldsymbol{h}_1 = \boldsymbol{h}_2^T\boldsymbol{K}^{-T}\boldsymbol{K}^{-1}\boldsymbol{h}_2 \tag{3.30}$$

Where $\boldsymbol{h}_i = [h_{i1}\, h_{i2}\, h_{i3}]^T$ is the i$^{\text{th}}$ column of the homography matrix.

3.7 Intrinsic parameters and the absolute conic

An homography, $\boldsymbol{H}$, between the calibration plate and the image of the calibration plate can be estimated. This homography can then be used to solve for the image of the absolute conic. Once the image of the absolute conic is known it can be used to solve for the intrinsic camera parameters. The image of the absolute conic can be represented as

$$\boldsymbol{B} = \boldsymbol{K}^{-T}\boldsymbol{K}^{-1} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{12} & b_{22} & b_{23} \\ b_{13} & b_{23} & b_{33} \end{bmatrix} \tag{3.31}$$

Since $\boldsymbol{B}$ is symmetric its contents can be represented by a 6D vector $\boldsymbol{b} = [b_{11}\, b_{12}\, b_{22}\, b_{13}\, b_{23}\, b_{33}]^T$. Then

$$\boldsymbol{h}_I^T \boldsymbol{B} \boldsymbol{h}_j = \boldsymbol{v}_{ij}\boldsymbol{b} \tag{3.32}$$

where

$$\boldsymbol{v}_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T. \tag{3.33}$$

Then equations 3.29 and 3.30 can be rewritten as

$$\begin{bmatrix} \boldsymbol{v}_{12}^T \\ (\boldsymbol{v}_{11} - \boldsymbol{v}_{22})^T \end{bmatrix} \boldsymbol{b} = \boldsymbol{0} \tag{3.34}$$

A separate version of this equation exist for each image taken of the calibration plate and these equations can be stacked, in $\boldsymbol{V}$, to give

$$\boldsymbol{V}\boldsymbol{b} = \boldsymbol{0}. \tag{3.35}$$

The solution for $\boldsymbol{b}$, up to a scale factor λ, is known to be the eigenvector of $\boldsymbol{V}^T\boldsymbol{V}$ associated with the smallest eigenvalue [9]. Once b has been determined it can be used to determine the intrinsic parameters of matrix $\boldsymbol{K}$. The relation between $\boldsymbol{B}$ and $\boldsymbol{K}$ is $\boldsymbol{B} = \lambda\boldsymbol{K}^{-T}\boldsymbol{K}^{-1}$. The intrinsic parameters are determined as follows.

$$\begin{aligned}
c_y &= (b_{12}b_{13} - b_{11}b_{23})/(b_{11}b_{22} - b_{12}^2) && (3.36)\\
\lambda &= b_{33} - (b_{13}^2 + c_y(b_{12}b_{13} - b_{11}b_{23}))/b_{11} && (3.37)\\
f_x &= \sqrt{\frac{\lambda}{b_{11}}} && (3.38)\\
f_y &= \sqrt{\frac{\lambda b_{11}}{b_{11}b_{22} - b_{12}^2}} && (3.39)\\
f_s &= \frac{-b_{12}f_x^2 f_y}{\lambda} && (3.40)\\
c_x &= \frac{f_s c_y}{f_y} - \frac{b_{13}f_x^2}{\lambda} && (3.41)
\end{aligned}$$

Thereafter the extrinsic parameters can be determined.

$$\begin{aligned}
\boldsymbol{r}_1 &= \lambda\boldsymbol{K}^{-1}\boldsymbol{h}_1 && (3.42)\\
\boldsymbol{r}_2 &= \lambda\boldsymbol{K}^{-1}\boldsymbol{h}_2 && (3.43)\\
\boldsymbol{r}_3 &= \boldsymbol{r}_1 \times \boldsymbol{r}_2 && (3.44)\\
\boldsymbol{T} &= \lambda\boldsymbol{K}^{-1}\boldsymbol{h}_3 && (3.45)
\end{aligned}$$

At this point it is necessary to incorporate distortions and use them to find better approximations for the intrinsic and extrinsic parameters.

this section is too close to the paper

3.8 Distortion in Calibration

Sections 3.3 through 3.7 outline the framework for the closed form solution of the calibration problem when distortion is not considered. It is only once the closed form solution to the extrinsic and intrinsic parameters is optimised that distortion can be accounted for during the calibration process. As already presented, there are many types of distortion that occur when an image is taken. However it is seldom possible to take all of these distortion types into account since the more distortion parameters introduced into the camera model; the more likely the optimization process becomes unstable.

It has been found that good calibration results can be achieved by taking only radial distortion into account during calibration [7,8]. This is beneficial since it is possible to solve for initial guesses to the radial distortion parameters which helps the optimisation process to avoid local minima as a result of the distortion parameters used.

add reference for this

The distortion applied to the points on the image plane is of the form

$$\hat{\boldsymbol{X}_{\boldsymbol{p_i}}} = \begin{bmatrix}\hat{x_{p_i}}\\ \hat{y_{p_i}}\end{bmatrix} = \left(1+k_1r^2+k_2r^4\right)\begin{bmatrix}x_{p_i}\\ y_{p_i}\end{bmatrix} = D\begin{bmatrix}x_{p_i}\\ y_{p_i}\end{bmatrix} \tag{3.46}$$

$$\text{where} \qquad r_i = \sqrt{x_{p_i}^2 + y_{p_i}^2} \quad \text{and} \quad D = \left(1+k_1r^2+k_2r^4\right). \tag{3.47}$$

Here $\hat{\boldsymbol{X}_{\boldsymbol{p_i}}} = \begin{bmatrix}\hat{x_{p_i}} & \hat{y_{p_i}}\end{bmatrix}^T$ represent the distorted image coordinates.

At this point the intrinsic and extrinsic parameters have been determined using the pinhole camera model and the error between the predicted calibration plate targets and the actual location of these in the images is attributed to radial distortion [2]. This error is represented as the observed distortion vector $\boldsymbol{d_{1_i}}$.

$$\boldsymbol{d_{1_i}} = \boldsymbol{X}_{\boldsymbol{s_i}}^a - \boldsymbol{X}_{\boldsymbol{s_i}}^c \tag{3.48}$$

Here $\boldsymbol{X}_{\boldsymbol{s_i}}^a$ is the actual coordinates observed and $\boldsymbol{X}_{\boldsymbol{s_i}}^c$ is the predicted coordinates that are calculated according to the camera model.

An undistorted point, $\boldsymbol{X_{p_i}}$ is distorted to point $\hat{\boldsymbol{X}_{\boldsymbol{p_i}}}$ according to

$$\begin{aligned}
\hat{\boldsymbol{X}_{\boldsymbol{p_i}}} &= \boldsymbol{U_c} + (\boldsymbol{X_{p_i}} - \boldsymbol{U_c}) \times (1 + k_1r_i^2 + k_2r_i^4) && (3.49)\\
&= \boldsymbol{U_c} + \boldsymbol{X_{p_i}} - \boldsymbol{U_c} + (\boldsymbol{X_{p_i}} - \boldsymbol{U_c}) \times (k_1r_i^2 + k_2r_i^4) && (3.50)\\
&= \boldsymbol{X_{p_i}} + (\boldsymbol{X_{p_i}} - \boldsymbol{U_c}) \times (k_1r_i^2 + k_2r_i^4) && (3.51)\\
&= \boldsymbol{X_{p_i}} + \boldsymbol{d_{2_i}} && (3.52)
\end{aligned}$$

Here $\boldsymbol{U_c} = \begin{bmatrix}u_c & v_c\end{bmatrix}^T$ is the projection centre or principal point (the intersection of the optical axis with the image plane) and $\boldsymbol{d_{2_i}}$ is referred to as the model distortion vector.

The distortion parameters can be estimated by minimizing the difference between the model and observer distortion vectors. Thus the least squares solution to the overdetermined system of equations of the form $\boldsymbol{d_{2_i}} = \boldsymbol{d_{1_i}}$ is what is needed. For each point $\boldsymbol{X_{p_i}}$ considered there are two equations of the form

$$\begin{aligned}
(x_{p_i} - u_c) \times (k_1r_i^2 + k_2r_i^4) &= (x_{s_i}^a - x_{s_i}^c) && (3.53)\\
(y_{p_i} - v_c) \times (k_1r_i^2 + k_2r_i^4) &= (y_{s_i}^a - y_{s_i}^c). && (3.54)
\end{aligned}$$

Two equations of this form are created for each target of the calibration plate and these equations are then stacked into a system of equations which is used to solved for the least squares solution to the distortion parameters.

$$\begin{bmatrix} (x_{p_1} - u_c) \times r_1^2 & (x_{p_1} - u_c) \times r_1^4 \\ (y_{p_1} - v_c) \times r_1^2 & (y_{p_1} - v_c) \times r_1^4 \\ (x_{p_2} - u_c) \times r_2^2 & (x_{p_2} - u_c) \times r_2^4 \\ (y_{p_2} - v_c) \times r_2^2 & (y_{p_2} - v_c) \times r_2^4 \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} x_{s_1}^a - x_{s_1}^c \\ y_{s_1}^a - y_{s_1}^c \\ x_{s_2}^a - x_{s_2}^c \\ y_{s_2}^a - y_{s_2}^c \\ \vdots \end{bmatrix} \tag{3.55}$$

3.9 Non-linear optimisation

At this point good approximations have been calculated for all the calibration parameters using closed form solution methods. However, since the correlation process relies upon the calibration process to link optical flow in images to displacement in the 3D world, these calibration parameters need to be refined so that they don't negatively affect the correlation results. This is done by combining the camera model and distortion into one mathematical relation that related 3D coordinates in the physical world to 2D coordinates in the images. It is important to note that the distortion model is inserted into the camera model prior to converting the image plane coordinates to the sensor plane coordinates. Thus the camera model becomes

$$\boldsymbol{X_s} = \boldsymbol{K}D\boldsymbol{V}\boldsymbol{X_w}. \tag{3.56}$$

This camera model and the approximations of the calibration parameters are employed in a non-linear optimization scheme in order to improve all of the calibration parameters simultaneously. Matlab's built in Levenberg-Marquart non-linear least squares optimisation algorithm was used for this. The sum of the absolute of the difference between the calculated and observed calibration plate targets was used as a measure of the error for the algorithm.

Chapter 4

Correlation

The Lucas-Kanade algorithm was used in combination with the zero-mean normalised sum of squared differences correlation criteria. This correlation criteria was chosen since it takes into account offset and scaling in the light intensities of the images making the overall algorithm more robust. The Lucas-Kanade algorithm assumes that the deformation of the image is constant over a area of the image referred to as the subset. As such this algorithm tries to find the deformation field that best corrects a subset of the deformed image so that it appears to be the same as the reference image.

4.1 Correspondence problem and speckle patterns

Although subsets of pixels are usually easier to track than individual pixels, using subsets leads to a new issue. The correspondence problem is to do with the inability to track displacements as a result of the grayscale patterns within the image. To understand this consider the aperture problem which is a special case of the correspondence problem.

The aperture problem involves the ambiguity in attempting to track the motion of a one-dimensional spatial structure (such as a line) when viewed through an aperture (such as when considering a subset of pixels) that cuts off the view of the ends of the one-dimensional spatial structure [1]. This can be illustrated by considering the difficulty in tracking the motion of a line in the direction along the line when the end points of the line are not visible.

image?

Similarly if the pattern in the image to be tracked is a repeating pattern with a constant grid spacing then the displacement cannot be uniquely determined since the image will contain the same pattern repeated. Thus the algorithm will determine the displacement to be within a multiple of the grid spacing from the actual displacement unless the subset contains a unique feature such as being at the edge of the specimen.

deformation and difficulty in tracking edges?

Thus it is clear that the patterns to be tracked in the images can have a significant impact on the performance of the correlation process. To avoid the aperture problem is is clear that the pattern should be isotropic such that it can be tracked equally well in all directions. Additionally to solve the correspondence problem the pattern should also be highly random and non-periodic so that each subset of an image contains a unique pattern.

This is achieved by applying a random speckle pattern to the surface of the specimen to be tested since these provide very unique patterns which are isotropic in nature. Additionally the high information density enables smaller subsets to be used than would otherwise have been possible.

4.2 Correlation criteria

A correlation criteria is a mathematical way of quantifying how well a subset in one image matches a subset in another image. It is used to determine whether the warp function parameters that has been solved for are sufficiently accurate to explain how the deformed image is related to the reference image. Furthermore the correlation criteria is usually used as the basis for the iterative solve that performs the correlation process. As such it is important that the correlation criteria reliably indicates the fit between the two subsets.

This reliability refers to dealing with the changes in lighting between images. As images are taken, as the specimen is deformed, the intensity of the images can change as a result of changes in lighting, changes in reflectivity as the material strains and so on. Although this does not cause the patterns in the subsets to change it does cause the grayscale values of individual pixels, that make up the patterns, to change which can negatively affect the correlation process. The changes in lighting that takes place between images has been classified into two types; offset and scaling.

The two most commonly used correlation criteria are the Cross-Correlation and Sum of Squared difference criteria. Each of these has four versions for dealing with various combinations of light intensity changes.

4.2.1 Cross-Correlation

The cross-correlation criteria obtains a coefficient by multiplying each element of the reference subset with its corresponding element in the investigated subset and then summing all of these values together as shown below. Thus a higher value means better correlation.

$$C_{CC} = \sum_{i}^{n} F_i G_i \tag{4.1}$$

This correlation criteria can be modified to be capable of dealing with offsets in light intensity by using zero-mean normalisation. Zero-mean normalisation determines the mean intensity value of a subset and then subtracts that mean intensity from every pixel in that subset. This is done for both subsets to obtain the Zero-Mean Cross-Correlation as shown below.

$$C_{ZCC} = \sum_{i}^{n} \left(F_i - \bar{F}\right)\left(G_i - \bar{G}\right) \tag{4.2}$$

$$\text{where} \quad \bar{F} = \frac{\sum_i^n F_i}{n} \tag{4.3}$$

$$\text{and} \quad \bar{G} = \frac{\sum_i^n G_i}{n} \tag{4.4}$$

The correlation criteria can also be modified to deal with scaling of light intensities by dividing it by the root-sum-square of the subset's intensities. This is referred to as Normalised Cross-Correlation.

$$C_{NCC} = \frac{\sum_i^n F_i G_i}{\sqrt{\sum_i^n F_i^2 \sum_i^n G_i^2}} \tag{4.5}$$

Both of these methods can be combined together to obtain Zero-Mean Normalised Cross-Correlation which is insensitive to both scaling and offset.

$$C_{ZNCC} = \frac{\sum_i^n (F_i - \bar{F})(G_i - \bar{G})}{\sqrt{\sum_i^n (F_i - \bar{F})^2 \sum_i^n (G_i - \bar{G})^2}} \tag{4.6}$$

4.2.2 Sum of Squared Difference

As the name suggests this correlation criteria is based on summing the square of the differences between the reference subset and the investigated subset.

$$C_{SSD} = \sum_{i}^{n} F_i G_i \tag{4.7}$$

Since this correlation criteria measures differences; a smaller value means better correlation between the subsets. This correlation criteria can be modified in the same way to become insensitive to both offset (Zero-Mean Sum of Squared Difference) and scaling (Normalised Sum of Squared Difference).

$$C_{ZSSD} = \sum_{i}^{n} \left[\left(F_i - \bar{F}\right) \left(G_i - \bar{G}\right) \right]^2 \tag{4.8}$$

$$C_{NSSD} = \sum_{i}^{n} \left[\frac{F_i}{\sqrt{\sum_i^n F_i^2}} - \frac{G_i}{\sqrt{\sum_i^n G_i^2}} \right] \tag{4.9}$$

These can then be combined to obtain the Zero-Mean Normalised Sum of Squared Difference correlation criteria which is insensitive to both scaling and offset in light intensities.

$$C_{ZNSSD} = \sum_{i}^{n} \left[\frac{F_i - \bar{F}}{\sqrt{\sum_i^n (F_i - \bar{F})^2}} - \frac{G_i - \bar{G}}{\sqrt{\sum_i^n (G_i - \bar{G})^2}} \right] \tag{4.10}$$

4.2.3 Relation between ZNCC and ZNSSD

It can be shown that the Zero-Mean Normalised Cross-Correlation and Zero-Mean Normalised Sum of Squared Difference correlation criteria are related.

$$C_{ZNSSD} = \sum_{i}^{n} \left[\frac{F_i - \bar{F}}{\sqrt{\sum_i^n (F_i - \bar{F})^2}} - \frac{G_i - \bar{G}}{\sqrt{\sum_i^n (G_i - \bar{G})^2}} \right] \tag{4.11}$$

$$= \sum_{i}^{n} \left(\frac{(F_i - \bar{F})^2}{\sum_i^n (F_i - \bar{F})^2} - 2 \frac{(F_i - \bar{F})(G_i - \bar{G})}{\sqrt{\sum_i^n (F_i - \bar{F})^2} \sqrt{\sum_i^n (G_i - \bar{G})^2}} + \frac{(G_i - \bar{G})^2}{\sum_i^n (G_i - \bar{G})^2} \right) \tag{4.12}$$

$$= 2 - 2 \frac{(F_i - \bar{F})(G_i - \bar{G})}{\sqrt{\sum_i^n (F_i - \bar{F})^2} \sqrt{\sum_i^n (G_i - \bar{G})^2}} \tag{4.13}$$

$$= 2\left(1 - C_{ZNCC}\right) \tag{4.14}$$

This is significant since the ZNSSD criteria is easier to calculate while the ZNCC criteria value is easier to interpret. Thus it is common practice to calculate the ZNSSD criteria coefficient and convert it to the ZNCC value for easier interpretation.

4.3 Warp function

When applying DIC to to material science the deformation behaviour of materials under load must be taken into account. This is because the deformation of the material causes the speckle pattern on its surface to deform in the same way. This is an issue since the pattern contained within a subset of the reference image will be deformed in the second image which can cause difficulties in matching the two subsets.

This is solved by allowing for the subsets to deform in a similar way to that of the material by defining the allowable deformations in a warp function. The purpose of these warp functions is to transform the pixel coordinates of the reference subset so that the resulting pattern is closer to the pattern in the deformed image. For this analysis the common warp function which accounts for affine transformations is used since it is consistent with the strains that the material is expected to experience. Thus by determining the warp function parameters for a specific subset; the strains for that subset are also determined.

$$W(\boldsymbol{\zeta}, \boldsymbol{p}) = \begin{bmatrix} \Delta x_{warp} \\ \Delta y_{warp} \end{bmatrix} = \begin{bmatrix} 1 + \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & u \\ \frac{\partial v}{\partial x} & 1 + \frac{\partial v}{\partial y} & v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ 1 \end{bmatrix}. \tag{4.15}$$

Here u and v represent the displacements in the x and y directions respectively, $\frac{\partial u}{\partial x}$ and $\frac{\partial v}{\partial y}$ represent the elongation in the x and y directions respectively whereas $\frac{\partial v}{\partial x}$ and $\frac{\partial u}{\partial y}$ represent the shear deformation of the subset. All of these variables that explain how the subset is warped and translated are stored within the vector $\boldsymbol{p} = \begin{bmatrix} u & \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & v & \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix}$ which are referred to as the warp function parameters. Additionally Δx and Δy are the distances from the centre of the undeformed subset to the pixel under consideration and are contained within the variable $\boldsymbol{\zeta} = \begin{bmatrix} \Delta x & \Delta y & 1 \end{bmatrix}^T$. The outputs Δx_{warp} and Δy_{warp} are the modified distances from the centre of the subset to the pixel under consideration. Thus it is the pixel locations within the subset that are modified in order to induce warp and not the subset itself which is warped by this function. In other words the Δx and Δy are warped by the function which has the effect of warping the subset itself since it changes the locations within the subset that are to be evaluated. Thus the warp function applied to a subset is mathematically expressed as

$$F_{warped} = F(\boldsymbol{x_0} + W(\boldsymbol{\zeta}, \boldsymbol{p})) \tag{4.16}$$

where $\boldsymbol{x_0} = \begin{bmatrix} x_0 & y_0 \end{bmatrix}^T$ are the x and y coordinates, in units of pixels, of the centre of the subset.

With most subset based DIC algorithms the more complex the warp function becomes the more computationally expensive each iteration is. This is because the correlation criteria must be optimized in terms of more variables leading to more work. keep this?

4.4 Interpolation

It is clear that warping a subset according to equation 4.16 requires that the light intensity values between pixel locations can be determined. This is an issue because images store this information in a discrete manner. This is solved by using interpolation to determine the light intensity values between pixels based on the light intensity values of neighbouring pixels. For the purposes of this project Matlab's built in interp2 fucntion was used to perform linear interpolation to determine the light intensity values for non-standard pixel locations where needed.

4.5 Subset matching

The purpose of DIC is to find the warp function parameters which optimise the correlation criteria between a specific subset, F, in the reference image and the investigated subset, G, in the deformed image.. Multiple methods of doing this have been proposed; each with their own advantages and disadvantages.

check

Typically the method used to solve the subset matching problem has been to make use of a gradient based optimization algorithm that is applied to the correlation criteria equation in order to iteratively improve upon an initial guess to the warp function parameters. The Lucas-Kanade method is used to illustrate this. It makes use of intensity gradients to guide the search direction to optimise the correlation criteria. A limitation of the method is that it can only achieve good correlation if the initial guess provided for the warp function parameters is reasonably close to the true value otherwise it will converge to a local minimum for the correlation criteria.

change to displacements need to be small?

The Lucas-Kanade method can be implemented in two distinct ways that differ in how the correlation criteria function is implemented in the Guass-Newton solver. The first, the original way, is referred to as the forward additive method which applies the current estimate of the warp function parameters, $\boldsymbol{p}$, and the iterative improvement of the warp function parameters, $\Delta\boldsymbol{p}$, to the investigated subset. This deformed subset is then compared with the unchanged reference subset and the difference between the two is used to incrementally improve the warp function parameter estimates. In this method the Hessian matrix is dependent on the investigated subset's light intensity values, G, and as such it needs to be computed for each iteration of the solver.

The second method, the inverse compositional method, is a modification to the forward additive method in an effort to decrease the computational cost per iteration. For this method the current estimate for the warp function parameters, $\boldsymbol{p}$, is applied to the investigated subset while the iterative improvement to the warp function parameters, $\Delta\boldsymbol{p}$, is applied to the reference subset. This is done so that the Hessian that needs to be determined depends only on the reference subset and is independent of the warp function parameters which allows it to be pre-computed outside of the iterations. As a result this method is more efficient which means it takes less time to converge to the solution.

The algorithm explained here is based on the one proposed by B. Pan [4]. It is the inverse compositional Lucas-Kanade method using the Guass-Newton approximation to the Hessian matrix with the Zero-Mean Normalised Sum of Squared Difference used as the correlation criteria. The current guess of the warp function parameters is applied to the investigated subset while the iterative improvement of the warp function parameters is applied to the reference subset as shown below.

$$C^2_{ZNSSD} = \sum_{\zeta}\left[\frac{F(\boldsymbol{x_0}+W(\zeta,\Delta\boldsymbol{p}))-\bar{F}}{\Delta F}-\frac{G(\boldsymbol{x_0}+W(\zeta,\boldsymbol{p}))-\bar{G}}{\Delta G}\right]^2 \tag{4.17}$$

Taking the first-order Taylor expansion of equation 4.17 in terms of $\Delta\boldsymbol{p}$ gives

$$C^2_{ZNSSD} = \sum_{\zeta}\left[\frac{F(\boldsymbol{x_0}+\zeta)+\triangledown F\frac{\partial W}{\partial \boldsymbol{p}}\Delta\boldsymbol{p}-\bar{F}}{\Delta F}-\frac{G(\boldsymbol{x_0}+W(\zeta,\boldsymbol{p}))-\bar{G}}{\Delta G}\right]^2 \tag{4.18}$$

where $\triangledown F = \begin{bmatrix}\frac{\partial F(\boldsymbol{x_0}+\boldsymbol{\zeta})}{\partial x} & \frac{\partial F(\boldsymbol{x_0}+\boldsymbol{\zeta})}{\partial y}\end{bmatrix}$ is the gradients of the reference subset in the x and y directions and $\frac{\partial W}{\partial \boldsymbol{p}} = \begin{bmatrix}1 & \Delta x & \Delta y & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & \Delta x & \Delta y\end{bmatrix}$ is the Jacobian of the warp function.

Taking the derivative of equation 4.18 with respect to $\Delta\boldsymbol{p}$ and setting it to zero gives the least squares solution for finding $\Delta\boldsymbol{p}$.

$$\frac{\partial \chi^2_{ZNSSD}}{\partial \Delta \boldsymbol{p}} = 0 = 2\sum_{\zeta} \left[\frac{F(\boldsymbol{x_0}+\zeta) + \nabla F \frac{\partial W}{\partial \boldsymbol{p}} \Delta\boldsymbol{p} - \bar{F}}{\Delta F} - \frac{G(\boldsymbol{x_0} + W(\zeta,\boldsymbol{p})) - \bar{G}}{\Delta G}\right] \frac{\nabla F \frac{\partial W}{\partial \boldsymbol{p}}}{\Delta F} \tag{4.19}$$

$$\Delta\boldsymbol{p} = -H^{-1} \times \sum_{\zeta} \left[(\nabla F \frac{\partial W}{\partial \boldsymbol{p}})^T \times \left[(F(\boldsymbol{x_0} + \boldsymbol{\zeta}) - \bar{F}) - \frac{\Delta F}{\Delta G}(G(\boldsymbol{x_0} + W(\boldsymbol{\zeta},\boldsymbol{p})) - \bar{G}) \right]\right] \tag{4.20}$$

where $$H = \sum_{\zeta} \left[(\nabla F \frac{\partial W}{\partial \boldsymbol{p}})^T (\nabla F \frac{\partial W}{\partial \boldsymbol{p}}) \right] \tag{4.21}$$

Here H is the Guass-Newton approximation to the Hessian matrix which remains constant since it depends only on the reference subset. Thus the Hessian matrix can be computed before the algorithm starts iterating in order to reduce computational costs. However in order to achieve this reduced computational cost the iterative improvement to the warp function parameters was applied to the reference subset. This means that the iterative improvement cannot simply be added to the old warp function parameters to obtain the updated warp function parameters as in the forward additive method. Instead the updated warp function parameters must be determined using the following equation.

$$W(\boldsymbol{\zeta}, \boldsymbol{p_{new}}) = W(\boldsymbol{\zeta}, \boldsymbol{p_{current}}) \times W(\boldsymbol{\zeta}, \Delta\boldsymbol{p})^{-1} \tag{4.22}$$

$$= \begin{bmatrix} 1+\frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & u \\ \frac{\partial v}{\partial x} & 1+\frac{\partial v}{\partial y} & v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1+\Delta\frac{\partial u}{\partial x} & \Delta\frac{\partial u}{\partial y} & \Delta u \\ \Delta\frac{\partial v}{\partial x} & 1+\Delta\frac{\partial v}{\partial y} & \Delta v \\ 0 & 0 & 1 \end{bmatrix}^{-1} \tag{4.23}$$

The new approximation to the warp function parameters can then be extracted from this warp function matrix, $W(\boldsymbol{\zeta}, \boldsymbol{p_{new}})$, and used in the next iteration of the solver.

4.6 Stopping criteria

The subset matching algorithm will continue to improve the warp function parameters until stopped. Thus a means of determining when the iterations should stop is needed. Ideally the iterations should stop once the correlation between the subsets is sufficiently accurate. Thus correlation could be halted once the correlation criteria coefficient has reached the desired value. However determining the correlation coefficient within each iteration is computationally expensive. Therefore the stopping criteria suggested by B. Pan [4] is used since it is much more efficient to compute. This criteria halts the iterations once the norm of the warp function parameter update is less than 0.001.

$$||\Delta\boldsymbol{p}|| = \sqrt{(\Delta u)^2 + (\Delta\tilde{x} \times \Delta\frac{\partial u}{\partial x}) + (\Delta\tilde{y} \times \Delta\frac{\partial u}{\partial y}) + (\Delta v)^2 + (\Delta\tilde{x} \times \Delta\frac{\partial v}{\partial x}) + (\Delta\tilde{y} \times \Delta\frac{\partial v}{\partial y})}$$
$$\text{where} \quad \Delta\tilde{x} = max(\Delta x) \quad \text{and} \quad \Delta\tilde{y} = max(\Delta y) \tag{4.24}$$

4.7 Algorithm outline

The steps performed within the algorithm in order to achieve correlation are listed here in order to illustrate how the theory presented here is related to the code. It should be noted that the initial guesses for the warp function parameters were determined by the user selecting seed points in the reference and investigated images.

1. First $\bar{F}$ is determined for the reference subset.
2. Then ∇F is calculated using Matlab's built in image gradient function (imgradientxy) and the Jacobian of the warp function is calculated. These are then used to determine the Hessian matrix.
3. Thereafter ΔF is determined since it too remains constant.
4. The initial guess for the warp function parameters is then used to calculate the Δx_{warp} and Δy_{warp} for the investigated subset G using equation 4.15.
5. Δx_{warp} and Δy_{warp} are used in equation 4.16 to obtain the warped version of the investigated subset.
6. $\bar{G}$ and ΔG are then determined.
7. With all the necessary values determined; equation 4.20 can be used to calculate $\Delta \boldsymbol{p}$.
8. Then equation 4.23 is used to determine the updated approximation to the warp function parameters.
9. Step 4 until step 8 are repeated until the stopping criteria is satisfied.

Chapter 5

Results

This section provides the results of the Matlab codes in order to prove that it works. The results will be presented here in accordance to the layout of the project sections.

5.1 Part A

Part A dealt with calibration only. The intrinsic parameters determined were

$$\boldsymbol{K} = \begin{bmatrix} 657.6284016509245 & 0.3033509221103 & 303.7286777984020 & 0 \\ 0 & 658.4782284119660 & 245.6917686350662 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{5.1}$$

The radial distortion parameters were determined to be $k_1 = -0.2488456293258$ and $k_2 = 0.1101356622771$. Using these calibration parameters the image ”img01.tiff” was undistorted and is provided in figure 5.2. Comparing this to the reference image in figure 5.1 it is clear that the calibration parameters determined are sufficiently accurate.

5.2 Part B

Part B deals with the correlation process only. The warp function parameters determined by the Matlab code are presented below.

$$u = 5.03754 \tag{5.2}$$
$$\frac{\partial u}{\partial x} = 0.04073 \tag{5.3}$$
$$\frac{\partial u}{\partial y} = -0.02176 \tag{5.4}$$
$$v = -1.03562 \tag{5.5}$$
$$\frac{\partial v}{\partial x} = -0.00915 \tag{5.6}$$
$$\frac{\partial v}{\partial y} = 0.03048 \tag{5.7}$$

The corresponding correlation coefficients are

$$C_{ZNSSD} = 0.005872084419686 \tag{5.8}$$
$$C_{ZNCC} = 0.997063957790157. \tag{5.9}$$

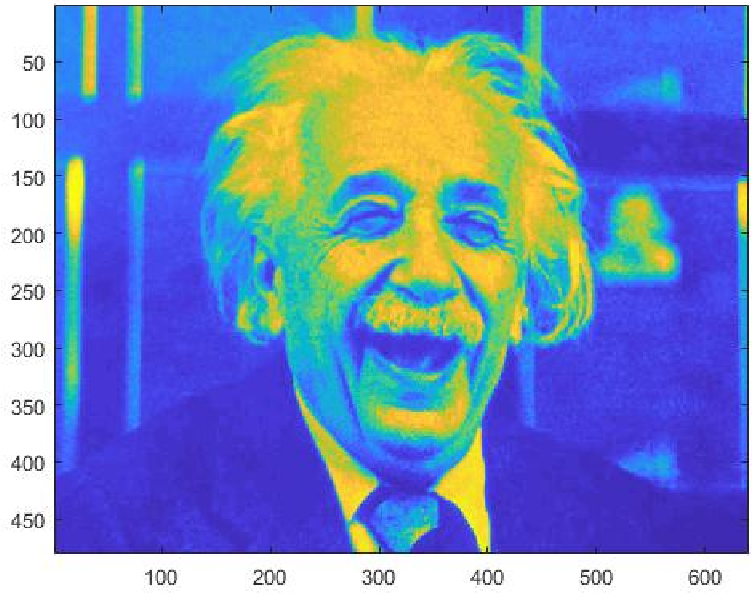


Figure 5.1: Reference image for comparison purposes

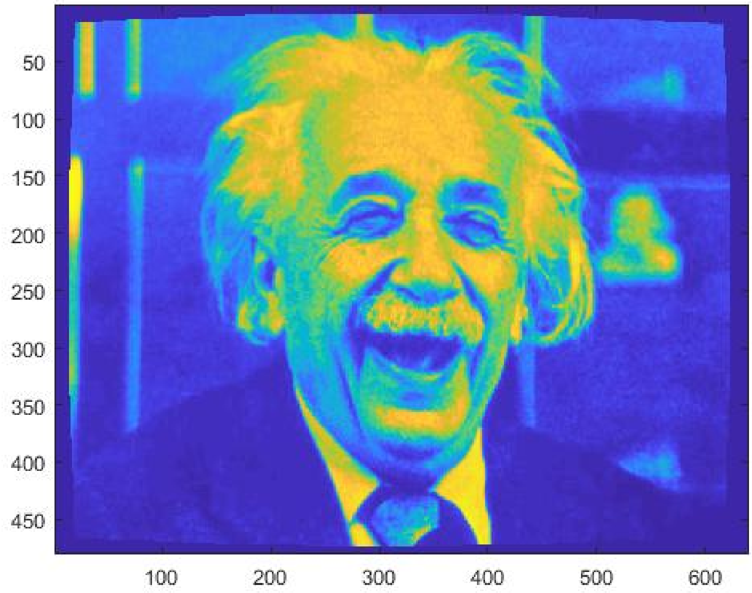


Figure 5.2: The distorted image "img01.tiff" which has been corrected for distortions

Both correlation coefficients have been presented here in order to illustrate that the C_{ZNCC} coefficient is much easier to interpret. The determined warp function parameters were used to undeform the image "img02.tiff" to obtain figure 5.4. Comparing this to the reference figure in figure 5.3 it can be seen that the determined warp function parameters are sufficiently accurate.

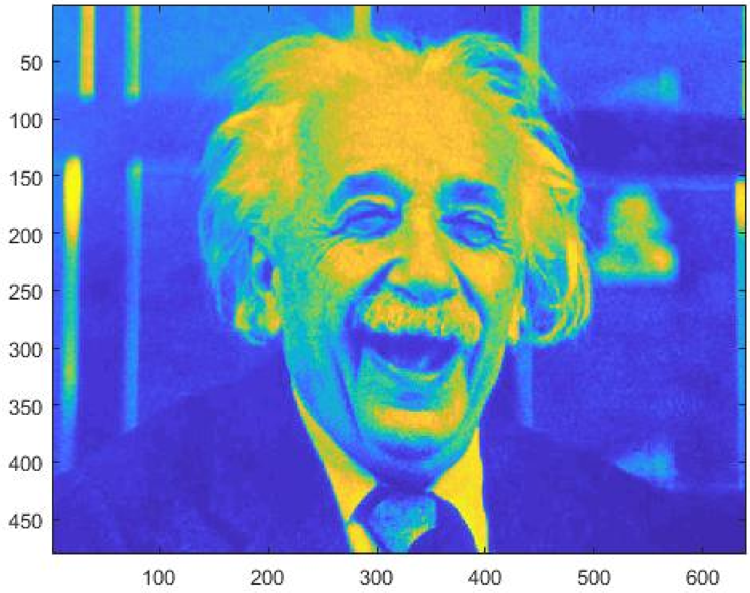


Figure 5.3: Reference image for comparison purposes

5.3 Part C

Part C combine both Part A and Part B to determine the calibration parameters, use these calibration parameters to undistort the investigated image and then perform correlation on this undistorted image.

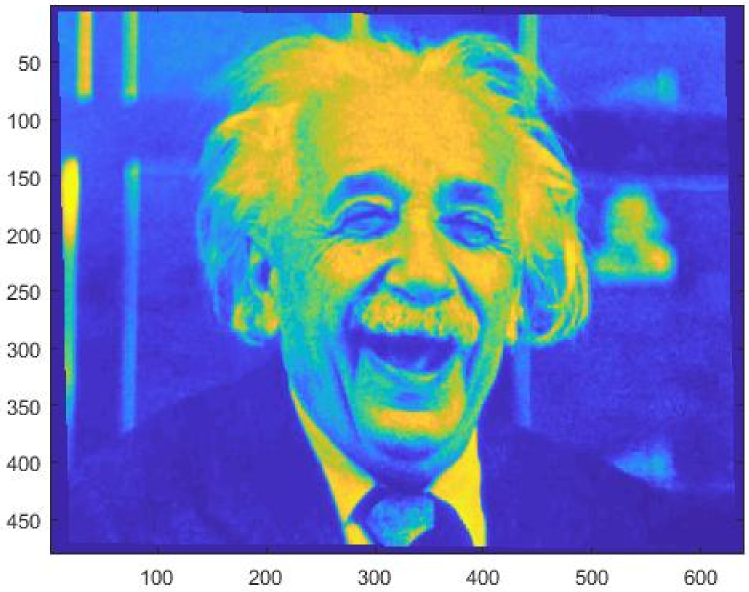


Figure 5.4: The distorted image "img01.tiff" which has been corrected for distortions

Chapter 6

Conclusion

This document proposes a research project aimed at creating a program for Matlab that is capable of performing DIC or DVC on a given set of images. The proposed program is aimed at providing the user with full control over the correlation process so that users no longer need to treat DIC and DVC as a black box. This project is not intended to have any direct application in industry but rather to provide researchers with a comprehensive tool to perform correlation to determine material deformation.

The motivation of the project has been outlined along with literature review on DIC. The objectives of the project have also been outlined with a research plan which is aimed at achieving these objectives. A time-line for the activities of the research plan is also provided. It is evident that sufficient knowledge and resources are available for the successful completion of this project.

Bibliography

[1] Marc D. Binder, Nobutaka Hirokawa, and Uwe Windhorst, editors. *Aperture Problem.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[2] Wilhelm Burger. Zhangs camera calibration algorithm: In-depth tutorial and implementation. *month*, 2016.

[3] Q-T Luong and Olivier D Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *International Journal of computer vision*, 22(3):261–289, 1997.

[4] B Pan, K Li, and W Tong. Fast, robust and accurate digital image correlation calculation without redundant computations. *Experimental Mechanics*, 53(7):1277–1289, 2013.

[5] HI Richard. In defence of the 8-point algorithm. In *Proceedings of the 5th International Conference on Computer Vision*, pages 1064–1070. Cambridge: IEEE Computer Science Press, 1995.

[6] M.A. Sutton, J.J. Orteu, and H. Schreier. *Image Correlation for Shape, Motion and Deformation Measurements: Basic Concepts,Theory and Applications.* Springer US, 2009.

[7] Roger Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344, 1987.

[8] Guo-Qing Wei and Song De Ma. Implicit and explicit camera calibration: Theory and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):469–480, 1994.

[9] Zhengyou Zhang. *Camera Calibration.* Prentice Hall PTR, 2004.