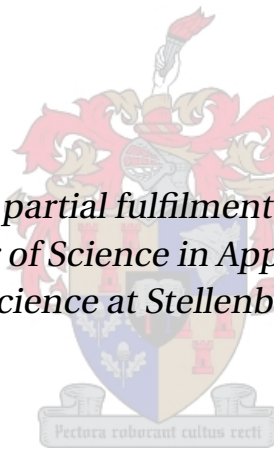


Accurate Camera Position Determination by Means of Moiré Pattern Analysis

by

Gideon Joubert Zuurmond

*Thesis presented in partial fulfilment of the requirements for
the degree of Master of Science in Applied Mathematics in the
Faculty of Science at Stellenbosch University*



Department of Mathematical Sciences,
Stellenbosch University,
Private Bag X1, Matieland 7602, South Africa.

Supervisors: Dr W. Brink, Prof. B. Herbst

March 2015

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: March 2015

Copyright © 2015 Stellenbosch University
All rights reserved.

Abstract

We introduce a method for determining the position of a camera with accuracy beyond that which is obtainable through conventional methods, using a single image of a specially constructed calibration object.

This is achieved by analysing the moiré pattern that emerges when two high spatial frequency patterns are superimposed, such that one pattern on a plane is observed through another pattern on a second, semi-transparent parallel plane, with the geometry of both the patterns and the planes known. Such an object can be created by suspending printed glass over printed paper or by suspending printed glass over a high resolution video display such as an OLED display or LCD.

We show how the camera's coordinate along the axis perpendicular to the planes can be estimated directly from frequency analysis of the moiré pattern relative to a set of guide points in one of the planes. This method does not require any prior camera knowledge. We further show how the choice of the patterns allows, within limits, arbitrary accuracy of this coordinate estimate at the cost of a stricter limit on the span along that coordinate for which the technique is usable. This improved accuracy is illustrated in simulation.

With a sufficiently accurate estimate of the camera's full set of 3D coordinates, obtained by conventional methods, we show how phase analysis of the moiré pattern in relation to the guides allows calculation of a new estimate of position in the two axes parallel to the planes. This new estimate is shown in simulation to offer significant improvement in accuracy.

Uittreksel

Ons stel 'n metode bekend om die posisie van 'n kamera te bepaal met akkuraatheid bo dit wat verkrygbaar is uit konvensionele metodes, deur gebruik te maak van 'n spesiaal gekonstrueerde kalibrasievoorwerp.

Die metode berus op analise van moiré patrone wat ontstaan waneer twee hoë ruimtelike frekwensie patrone oor mekaar neergelê word deurdat een patroon op 'n plat vlak waargeneem word deur 'n tweede patroon op 'n parallelle, gedeeltelik deursigtige vlak, met die geometrie van beide patrone en vlakke bekend. So 'n voorwerp kan geskep word deur 'n spesiaal gedrukte glas vlak te monteer oor gedrukte papier of oor 'n hoë resoluksie skerm, soos 'n OLED skerm of LCD.

Ons wys hoe die kamera se koördinaat langs die as loodreg op die vlakke direk uit frekwensie analise van die moiré patroon relatief tot 'n stel gidspunte in een van die vlakke bepaal kan word. Hierdie metode vereis geen vooraf kennis oor die kamera nie. Ons wys verder hoe die keuse van patrone, binne perke, arbitrêre akkuraatheid in die bepaling van hierdie koördinaat kan verkry ten koste van 'n nouer bruikbare versterk in hierdie koördinaat. Die verbeterde akkuraatheid is geïllustreer in simulasie.

Indien 'n benadering van die kamera se volle stel 3D koördinate met voldoende akkuraatheid ook beskikbaar is, wys ons ook hoe fase analise van die moiré patroon relatief tot die gidspunte ons in staat stel om 'n nuwe benadering te kan maak vir die koördinate in die twee asse parallel aan die vlakke. Daar word in simulasie getoon hoe hierdie nuwe benadering beduidend beter akkuraatheid kan bied in vergelyking met konvensionele metodes van benadering.

Contents

1	Introduction	1
2	Camera Model	4
2.1	Perspective	4
2.2	Camera Rotation	6
2.3	Camera Translation	7
2.4	General Camera Matrix	7
2.5	Decomposing the Camera Matrix	8
2.6	Nonlinear Lens Distortion	9
2.7	Solving for the Camera Matrix	10
2.8	Single Set of Non-Planar Correspondences	10
2.9	OpenCV	11
3	Plane to Plane Mapping	13
3.1	Removing Perspective Distortion	14
3.2	Planes Parallel to the Target Plane	15
4	Single Frequency Analysis Using the DFT	17
4.1	Analysis in One Dimension	17
4.2	Analysis in Two Dimensions	22
5	Analysis of Moiré Patterns in Flat Geometry	25
5.1	Product of Sinusoids in One Dimension	26
5.2	Product of Sinusoids in Two Dimensions	27
5.3	Sensitivity of Frequency Estimation from Moiré Analysis	29
5.4	Sensitivity of Phase Estimation from Moiré Analysis	29
6	Analysis of Moiré Patterns in Parallel Planes	31
6.1	Frequency Analysis	32
6.2	Condition Number of Height Analysis from Frequency	35
6.3	Phase Analysis	36
7	Moiré-based Camera Calibration Object	38
7.1	Extracting Moiré Parameters from Image of Calibration Object	41

<i>CONTENTS</i>	v
8 Analysis of Moiré-based Calibration Object	46
8.1 Camera <i>Z</i> -coordinate	48
8.2 Camera <i>X</i> - and <i>Y</i> -coordinate	50
8.3 Pinhole Model Parameters	52
9 Conclusions and Future Research	54
9.1 Conclusions	54
9.2 Imperfect Object	55
9.3 Alternative Configurations of the Proposed Calibration Object . . .	57
Bibliography	58

Chapter 1

Introduction

The term camera calibration as is used in the context of this thesis refers to the process of estimating the pose of a camera (its position and orientation) and the parameters that describe its internal lens geometry as they appear in a particular camera model. The camera pose is known as the extrinsic camera parameters and those that describe the camera internally are known as the intrinsic camera parameters.

With the camera model and the set of parameters known (the camera is said to be calibrated), we can obtain a theoretical mapping between points in 3D world coordinates and the corresponding 2D image coordinates at which these points appear on an image taken with the camera. This ability allows us to derive information about objects imaged by one or more cameras, such as estimating their size, position or orientation. A solved camera model can also be used for correct alignment of artificially inserted objects in a video sequence, such as for augmented reality or as is widely done in the film industry [8].

A simple camera model often used is the basic pinhole camera, generalisable to the general projective camera model [8]. This model is a projective transformation represented by a homogeneous 3×4 matrix, where multiplication with this matrix maps homogeneous world coordinates to homogeneous image coordinates.

The homography in the general projective camera model can be solved from a set of world-image point correspondences – points for which both world and corresponding image coordinates are known – using the well-known direct linear transform method [8, 11]. With 11 unknowns to be solved, this requires at least $5\frac{1}{2}$ correspondences and are subject to certain constraints to avoid degenerative cases. In particular, the points may not all be coplanar.

Expanding the camera model to include nonlinear lens distortion (part of the intrinsic parameter set) introduces more unknowns to be estimated and more point correspondences are needed. A direct solution is no longer available and nonlinear optimisation methods such as Marquardt-Levenberg are often used [11, 8, 10].

The method for camera calibration introduced by Zhang [13] allows accurate camera calibration using at least two images of an easily constructable calibration object. A variation on this method has been adopted by OpenCV [10] as the preferred technique for intrinsic camera calibration. With intrinsic camera parameters known, the camera pose can be estimated from at least three additional correspondences by solving the well-known perspective n -point problem [7, 9].

Provided that the correspondences and intrinsic parameters are sufficiently accurate, the methods used in OpenCV produce results with accuracy that is sufficient for most computer vision applications. For applications where even higher accuracy is required, specifically for the camera position, this thesis suggests a method that may offer the required improvement.

The need for higher accuracy in camera position, also referred to as the camera centre, can be motivated for uses where the cost of any error in this 3D parameter is large. Higher accuracy relative to the quality of input data would also be of benefit in certain situations, such as when having low image resolution, high image noise or large error in assumed intrinsic camera parameters. Augmented reality applications may also require extremely accurate camera position estimates when virtual objects are superimposed near the camera and inaccuracies would appear exaggerated due to parallax.

Using moiré pattern analysis on an image of a specially constructed calibration object as a means of camera calibration, we show in simulation how improvements in accuracy of the camera position can be obtained when compared to the conventional methods used in OpenCV.

In chapter 2 we will derive the camera model assumed in this thesis. We will also discuss commonly used techniques for solving the parameters associated with the model. Chapter 3 discusses a technique for resampling an image of a planar region in order to remove perspective distortion. We then show that removing perspective distortion for one such plane in an image also correctly removes perspective distortion for parallel planes in the same image. This property forms a key component of the analysis of the proposed object.

Another important step required in the analysis of moiré patterns on the proposed object is accurate frequency and phase estimation. In chapter 4 we will show how the 2D discrete Fourier transform, combined with an appropriate window function and analysis method, can be used for this purpose when given a 2D sampling of a sinusoidal function.

In chapters 5 and 6 we show how traditional and 3D moiré patterns resulting from patterns on parallel planes can be analysed. In the 3D case we also show how the pattern relates to camera position. Some analysis is done on how camera position estimation by means of moiré analysis may allow for improved accuracy. A customisable condition number in the calculation of the camera coordinate perpendicular to the planes may offer accuracy improvement in this dimension. A phenomenon called moiré speedup, that is rapid apparent changes in the pat-

tern for small camera movements, offers the possibility of accuracy improvement in estimation of the camera coordinates that are parallel to the planes.

Chapter 7 describes how an object can be constructed for camera position estimation from moiré analysis, with one particular instantiation of such an object described in detail. We discuss the process of analysing an image of such an object, using the example of a physically constructed implementation.

Chapter 8 discusses the parameter choices for such a calibration object and shows some simulated results for good choices, including significant improvement in accuracy of camera position estimation over the conventional method in certain instances. Some advantages and drawbacks of the proposed methods are discussed.

Finally, chapter 9 summarises the findings of this study and concludes with suggestions for future research and extensions on the object and analysis methods.

Chapter 2

Camera Model

In order to analyse images from a camera and extract meaningful information about the 3D environment that was imaged, we need a model that describes how any point in 3D world coordinates $\vec{A} = (A_X, A_Y, A_Z)$ is mapped to a point in 2D image coordinates $\vec{a} = (a_x, a_y)$, measured in pixels on an image taken by the camera.

We initially limit this camera model to a homography, that is a projective transformation which maps lines to lines: any straight line in the 3D world will map to some straight line in 2D in the image when imaged. This simplification, which does not account for any nonlinear lens distortion, is generally referred to as the pinhole camera model [8].

The pinhole camera model does take into account any external translation and rotation of the camera, meaning that the camera may be centred at an arbitrary position in the world and at an arbitrary orientation relative to the world coordinate system. Internally the camera may have any arbitrary focal length and reference origin in the 2D image plane.

2.1 Perspective

Given a pinhole camera with its centre at the world origin, its image plane perpendicular to the Z -axis and no roll on the orientation of the camera (the image x - and y -axis are parallel to the world X - and Y -axis, respectively), we can calculate the mapping from world coordinates to image coordinates using

$$a_x = f \frac{A_X}{A_Z} \quad \text{and} \quad a_y = f \frac{A_Y}{A_Z},$$

as derived from simple triangle geometry in figure 2.1. The parameter f is the focal length, which is the distance of the image plane from the origin. In this case the unique line passing through the camera centre and which is perpendicular to the image plane, called the principal axis, coincides with the Z -axis. The

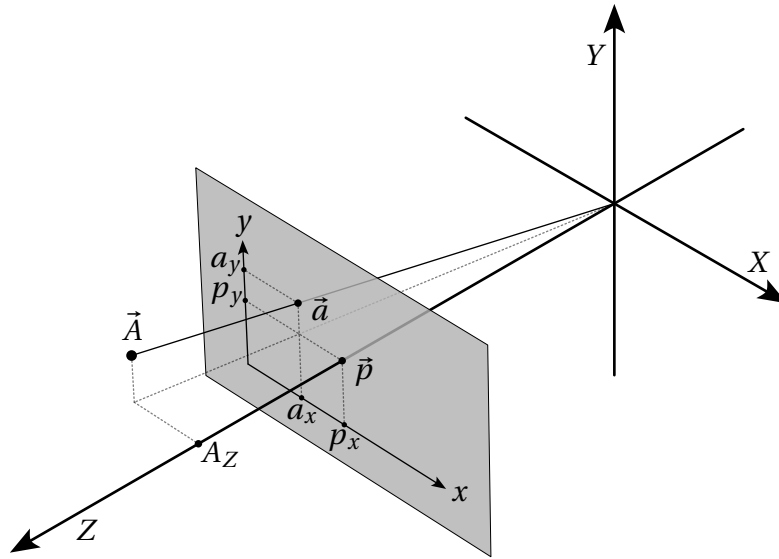


Figure 2.1: Perspective projection with the camera at the origin, aligned with the world coordinate system.

point in image coordinates where this line intersects the image plane is called the principal point.

If we further introduce an arbitrary reference origin in the image plane, we get

$$a_x = f \frac{A_X}{A_Z} + p_x \quad \text{and} \quad a_y = f \frac{A_Y}{A_Z} + p_y,$$

where (p_x, p_y) are the image coordinates of the principal point, expressed relative to the reference origin.

A homogeneous coordinates representation of the point \vec{a} , which we express as $[h_1 \ h_2 \ h_3]^T$ can be calculated from \vec{A} conveniently using a matrix multiplication:

$$\begin{aligned} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} &= \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A_X \\ A_Y \\ A_Z \end{bmatrix} \\ &= \mathbf{K} \begin{bmatrix} A_X \\ A_Y \\ A_Z \end{bmatrix}, \end{aligned} \tag{2.1}$$

such that

$$a_x = \frac{h_1}{h_3} \quad \text{and} \quad a_y = \frac{h_2}{h_3}.$$

Note that, depending on the units that a_x and a_y are measured in (e.g. mm or pixels), f implicitly also contains any scaling parameters between the world units and the final units of a_x and a_y . If the scaling factors to these individual components are not equal, it is possible to have different focal length parameters for the two axes. We can also introduce s , a so-called skew parameter, to obtain a general expression for the camera's calibration matrix:

$$\mathbf{K} = \begin{bmatrix} f_x & s & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}.$$

Matrix \mathbf{K} contains what is known as the intrinsic camera parameters. Multiplication by \mathbf{K} transforms points from a coordinate system whose axes are aligned with those of the camera in both position and orientation.

2.2 Camera Rotation

Given a pinhole camera with its centre at the world origin, but with arbitrary orientation relative to the world coordinate system, we want to transform coordinates of points from the world coordinate system to a new coordinate system that is aligned with the camera so that it can be provided as input to the perspective projection step discussed in the previous section.

This can be achieved using a matrix multiplication:

$$\begin{bmatrix} A'_X \\ A'_Y \\ A'_Z \end{bmatrix} = \mathbf{R} \begin{bmatrix} A_X \\ A_Y \\ A_Z \end{bmatrix},$$

where \mathbf{R} is a 3×3 rotation matrix, which is an orthogonal matrix. If the orientation of the camera is known then \mathbf{R} can be calculated directly, noting that \mathbf{R} is not the rotation that was applied to the camera, but its inverse. \mathbf{R} forms part of the extrinsic parameter set for the camera.

Rotation can also be expressed using several other forms such as the axis-angle representation, Euler angles or quaternions. Each of these can be converted to and from this matrix form using the appropriate conversion.

By applying first the appropriate rotation and then the perspective projection, we now have described a means of mapping 3D points in any coordinate system with the camera at its origin to 2D points in the image plane.

2.3 Camera Translation

Given a pinhole camera at an arbitrary position and orientation in the world, we want to transform coordinates of points from the world coordinate system to a new coordinate system which has the centre of the camera at the origin. This is done so that it can be provided as input to the combined rotation and perspective projection steps discussed in the previous section.

We can transform to the new coordinate system, centred around the camera, by subtracting the world coordinates of the camera centre $\vec{C} = (C_X, C_Y, C_Z)$ for any point \vec{A} :

$$\begin{aligned} A'_X &= A_X - C_X, \\ A'_Y &= A_Y - C_Y, \\ A'_Z &= A_Z - C_Z. \end{aligned}$$

As with the perspective projection and rotation, this translation can also be expressed as a matrix multiplication. We do this by expressing the point in homogeneous world coordinates as $[A_X \ A_Y \ A_Z \ 1]^T$. A translation can now be applied as a multiplication by a 3×4 matrix as follows:

$$\begin{bmatrix} A'_X \\ A'_Y \\ A'_Z \end{bmatrix} = [\mathbf{I} \quad -\vec{C}] \begin{bmatrix} A_X \\ A_Y \\ A_Z \\ 1 \end{bmatrix},$$

where \mathbf{I} is the 3×3 identity matrix. \vec{C} forms part of the extrinsic parameter set of the camera.

2.4 General Camera Matrix

With the transformations discussed in the previous sections, each expressed as a matrix multiplication, the general camera matrix \mathbf{P} is now simply the product of these matrices, taken in reverse so that the assumptions about the camera's position and orientation relative to the input coordinate systems are met during the rotation and perspective projection steps. Thus a point \vec{A} with homogenous coordinates $[A_X \ A_Y \ A_Z \ 1]^T$ projects to homogeneous image coordinates $\vec{h} = [h_1 \ h_2 \ h_3]^T$ according to

$$\vec{h} = \mathbf{KR}[\mathbf{I} \quad -\vec{C}] \vec{A} \quad (2.2)$$

$$\begin{aligned} &= \mathbf{K}[\mathbf{R} \quad \vec{T}] \vec{A} \\ &= \mathbf{P}\vec{A}, \end{aligned} \quad (2.3)$$

where $\vec{T} = -\mathbf{R}\vec{C}$.

In summary, to calculate the mapping of a point \vec{A} with world coordinates (A_X, A_Y, A_Z) to a point in 2D image coordinates $\vec{a} = (a_x, a_y)$, we make use of a linear transformation in homogeneous coordinates. We first write \vec{A} in homogeneous coordinates as $[A_X \ A_Y \ A_Z \ 1]^T$ and then multiply this vector by the camera matrix \mathbf{P} to obtain $\vec{h} = [h_1 \ h_2 \ h_3]^T$, the homogeneous image coordinates. We then transform back from these homogeneous coordinates to obtain the final image coordinates as

$$a_x = \frac{h_1}{h_3} \quad \text{and} \quad a_y = \frac{h_2}{h_3}.$$

The camera matrix \mathbf{P} is 3×4 , and therefore has 12 entries so that it can be interpreted as a vector in 12-dimensional space. Because \mathbf{P} is homogeneous, it can be solved for a particular camera up to one degree of freedom: multiplication by any non-zero scalar. Conversely, it can vary by 11 degrees of freedom depending on the particular camera configuration it represents.

Expressing (solving) \mathbf{P} for a particular camera therefore involves supplying (finding) a 12-dimensional basis for the 1-dimensional space that applies to that camera, using equation 2.3.

2.5 Decomposing the Camera Matrix

Given a camera matrix \mathbf{P} , we can decompose it into three different parts: matrices \mathbf{K} and \mathbf{R} and vector \vec{C} .

\mathbf{P} can be written as

$$\mathbf{P} = [\mathbf{M} \quad -\mathbf{M}\vec{C}]$$

where \vec{C} is the camera centre and \mathbf{M} is a 3×3 non-singular matrix. The camera centre can be solved by left-multiplying the last column of \mathbf{P} by $-\mathbf{M}^{-1}$.

Because \mathbf{K} is upper triangular and \mathbf{R} is orthogonal, \mathbf{M} can be decomposed into the product of \mathbf{K} and \mathbf{R} using RQ-decomposition. As with the better known QR-decomposition, RQ-decomposition can be calculated using the Gram-Schmidt process. However, as with QR-decomposition, since the RQ-decomposition is not unique, we may find a \mathbf{K} that is different from the form we expressed in equation 2.1 and \mathbf{R} may not be a rotation matrix. To remedy this we note that we can negate any column of \mathbf{K} , provided we negate the corresponding row of \mathbf{R} to ensure that the two matrices still multiply to give the original \mathbf{M} from which they were RQ-decomposed. We choose for f_x and f_y to be positive so that \mathbf{K} has positive diagonal entries. To ensure this, we multiply each column that violates this property by -1 , along with the corresponding row of \mathbf{R} .

We also note that due to the homogeneous coordinates in the output we can multiply both \mathbf{K} or \mathbf{R} with any non-zero scalar without affecting the mapping.

We can thus scale \mathbf{K} so that the bottom right entry is 1. The decomposed \mathbf{R} is an orthogonal matrix and therefore has a determinant of either -1 or 1 . In case of -1 , we multiply \mathbf{R} by -1 to ensure that it has a determinant of 1 : a proper rotation matrix.

2.6 Nonlinear Lens Distortion

If the camera used is not a pinhole camera, but uses a lens to project points from the world onto the image plane, then some nonlinear lens distortion may be introduced.

Where applicable, lens distortion is typically modelled as a mapping between uncorrected image coordinates (as resulted from the projective transformation) and final image coordinates. This still allows us to use the convenience of a homography to map between world coordinates and the uncorrected image coordinates as in the case of the pinhole camera model, but requires an additional model to compensate for the lens distortion in 2D.

The parameters required by the lens distortion model are a function of the lens itself and thus form part of the intrinsic camera parameter set. Figure 2.2 shows how such a model can be used to generate an image that is free from lens distortion from one that is not.



Figure 2.2: An example where a lens distortion model was used to resample an image with the effects of lens distortion present (left), to remove those effects (right). Straight lines in 3D get mapped to straight lines in 2D on the corrected image, but often appear curved on the uncorrected image.

One such model calculates undistorted coordinates (a'_x, a'_y) using

$$\begin{bmatrix} a'_x \\ a'_y \end{bmatrix} = \begin{bmatrix} c_x \\ c_y \end{bmatrix} + L(r) \begin{bmatrix} a_x - c_x \\ a_y - c_y \end{bmatrix}, \quad (2.4)$$

where $r = \sqrt{(a_x - c_x)^2 + (a_y - c_y)^2}$ and $L(r)$ is some unknown function of r . We can approximate $L(r)$ with a truncation of its Taylor series expansion

$$1 + k_1 r + k_2 r^2 + k_3 r^3 + k_4 r^4 + \dots$$

The model in equation 2.4 only compensates for radial distortion, so the assumption was that the distortion is concentric about the point (c_x, c_y) in the image plane. But the model allows that this point, called the centre of distortion, not necessarily correspond with the principal point.

The computer vision library OpenCV uses a distortion model that is centred around the principal point [10], but allows for both radial and tangential components¹.

Somewhat inconsistent with the ordering of operations discussed above, the distortion model in OpenCV is applied after converting back from homogeneous coordinates but before compensating for the focal length. This has the advantage that the distortion parameter values are independent of the final image scale in pixels. If an image taken with a calibrated camera was digitally resized, a new principal point and focal length(s) would need to be calculated to compensate for the rescaling, but the same distortion parameters as per the original image still apply.

Simulations done as research for this thesis cover only cameras where non-linear lens distortion is absent, but where specified the techniques discussed in later chapters include the cases where lens distortion is present.

2.7 Solving for the Camera Matrix

If the focal length, image coordinates of the principal point and the pose of the camera are known then \mathbf{K} , \mathbf{R} and \vec{C} in equation 2.2 can be combined directly in order to calculate a basis vector for the general camera matrix. In practice this information is not always known and the camera matrix is typically estimated from analysis done on an imaged scene or from an image of a calibration object of known dimensions.

Since the proposed technique in this thesis focuses specifically on extracting the camera position (\vec{C}), we discuss two commonly used techniques to solve the camera matrix and thus the camera position.

2.8 Single Set of Non-Planar Correspondences

In the absence of nonlinear lens distortion, we may use the direct linear transform (DLT) method to find a basis for \mathbf{P} from a set of point correspondences be-

¹OpenCV 3.0 also makes provision for thin prism distortion coefficients. At the time of writing this version was in beta.

tween homogeneous world coordinates and homogeneous image coordinates, provided that the points are not in a degenerate configuration. A planar arrangement of all the points is one such degenerate case from which \mathbf{P} cannot be solved [8].

For any point from the set of correspondences with homogeneous world coordinates $\vec{A} = [A_X \ A_Y \ A_Z \ 1]^T$ and homogeneous image coordinates $\vec{a} = [a_x \ a_y \ 1]^T$, the vectors $\mathbf{P}\vec{A}$ and \vec{a} differ only up to multiplication with some non-zero scalar. This can be expressed in terms of the cross product as

$$\vec{a} \times \mathbf{P}\vec{A} = \vec{0},$$

which we can rewrite as

$$\begin{bmatrix} \vec{0}^T & -\vec{A}^T & a_y \vec{A}^T \\ A^T & \vec{0}^T & -a_x \vec{A}^T \\ -a_y \vec{A}^T & a_x \vec{A}^T & \vec{0}^T \end{bmatrix} \begin{bmatrix} \vec{p}_1 \\ \vec{p}_2 \\ \vec{p}_3 \end{bmatrix} = \vec{0},$$

or

$$\mathbf{H}\vec{p} = \vec{0},$$

where \vec{p}_1^T , \vec{p}_2^T and \vec{p}_3^T are the rows of \mathbf{P} and are together expressed as \vec{p} . Simple inspection shows that \mathbf{H} only has rank 2, so that only two similarity equations are constructed from a single point correspondence. From n point correspondences $2n$ similarity relations can be constructed.

Basis vectors for \mathbf{P} are the basis vectors of the null space of \mathbf{H} , which can be computed using singular value decomposition (SVD). If \mathbf{H} has a nullity of 1 then a single basis vector for \mathbf{P} can be found and we have a solution. Because \mathbf{P} has 11 camera-dependent degrees of freedom, at least 11 equations are required to ensure that \mathbf{H} has a nullity of 1. This would require at least $5\frac{1}{2}$ point correspondences, where the $\frac{1}{2}$ denotes that in one of the correspondences only one of the two inhomogeneous image coordinates need to be known.

Using the DLT method on overdetermined systems, Hartley and Zisserman [8] argues that the quantity which is minimised in a least squares sense may not be ideal. The Levenberg-Marquardt minimisation algorithm is proposed to instead minimise the reprojection error. This means minimising the sum of the squared distances between the image coordinates of the reference points and the projected world coordinates of the reference points. The solution from the DLT method may serve as an initial estimate for the minimisation.

2.9 OpenCV

The routines in the computer vision library OpenCV encourages a two stage process [10]. First the intrinsic parameters of the camera (including any lens distortion if this is assumed present) are solved with data from multiple images using

a method based largely on work by Zhang [13]. With the intrinsic parameters then assumed known, the extrinsic parameters (the six unknown parameters of the camera pose), can be solved from a different set of world-image point correspondences [11, 9, 7]. This last step, known as the perspective n -point problem has solutions that require only three point correspondences [7] which implies that a set of planar correspondences can be used.

Image processing routine `findChessboardCorners` from OpenCV is called to detect the inner corners of a chequerboard pattern of known dimensions. Assuming that the chequerboard is aligned with the world coordinate system, this provides a single set of planar 3D world to 2D image coordinate correspondences.

A routine `calibrateCamera` is provided which takes as input multiple such sets of correspondences, one set per image, with the plane being in a different position and orientation for each set. On success, it calculates the lens distortion and intrinsic camera parameters with Zhang's method.

Once the intrinsic parameters (the distortion coefficients and matrix \mathbf{K}) are known, this can be assumed fixed as long as the camera lens is not adjusted. If the camera is now placed in an arbitrary position and orientation then only six degrees of freedom remain to be solved: C_X , C_Y , C_Z and the three degrees of freedom of an arbitrary rotation.

OpenCV provides a routine `solvePnP` which takes as input the previously solved intrinsic camera parameters and a new set of at least four correspondences. From this set, which may be planar in world coordinates, the remaining parameters can be solved and a basis for \mathbf{P} computed.

This method will be the benchmark against which the technique proposed further on will be measured for camera position estimation accuracy.

Chapter 3

Plane to Plane Mapping

Given an image of a plane in 3D space, we may wish to find a projective transformation or homography that maps between 2D image coordinates measured in pixels and 2D coordinates in world units on that plane for which it provides a mapping. Writing both as homogeneous coordinates, we are thus looking for a 3×3 matrix \mathbf{A} so that

$$\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = \mathbf{A} \begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix},$$

where $\vec{H} = [H_1 \ H_2 \ H_3]^T$ is the homogeneous representation of a point (U, V) and $\vec{h} = [h_1 \ h_2 \ h_3]^T$ is a homogeneous representation of the corresponding point (x, y) in the image plane. That is:

$$U = \frac{H_1}{H_3} \quad \text{and} \quad V = \frac{H_2}{H_3},$$

and

$$x = \frac{h_1}{h_3} \quad \text{and} \quad y = \frac{h_2}{h_3}.$$

Because \mathbf{A} is a homogeneous matrix, it is only specified up to scale, so that there are at most eight degrees of freedom that can vary because of the particular camera and target world plane that it provides a mapping for. Hartley and Zisserman [8] show that in the case where both the sets of inhomogeneous coordinates before and after the mapping are in Cartesian coordinate systems, there are in fact only six degrees of freedom. For simplicity we will not apply this limitation and instead find a more general solution.

A known basis for \mathbf{A} allows us to map arbitrary 2D points between the image plane and the world plane, but since a full camera matrix is not being solved for, which can map 3D world coordinates to 2D image coordinates, this mapping is only applicable to points that lie in the target world plane.

Each point correspondence can be written as a pair of linear equations [8]:

$$\begin{bmatrix} \vec{0}^T & -h_3 \vec{H}^T & h_2 \vec{H}^T \\ h_3 H^T & \vec{0}^T & -h_1 \vec{H}^T \end{bmatrix} \begin{bmatrix} \vec{a}_1 \\ \vec{a}_2 \\ \vec{a}_3 \end{bmatrix} = \vec{0},$$

where \vec{a}_1^T , \vec{a}_2^T and \vec{a}_3^T are the rows of \mathbf{A} . This means that each point correspondence can be used to construct two equations, so that n correspondences will yield $2n$ equations. Since there are eight unknowns, at least four point correspondences are required to find a basis for \mathbf{A} .

3.1 Removing Perspective Distortion

Given a homography that maps between image coordinates and world coordinates in a target plane, it is possible to remove perspective distortion from an image of the plane. This is done by iterating over a regular grid of world coordinates within the plane and calculating the corresponding image coordinates. Pixel values from the image can then be read at these coordinates to construct a new image which has the perspective effects removed for the texture. An example is shown in figure 3.1. We will use this technique in later chapters as a preprocessing step in order to perform analysis on textures.

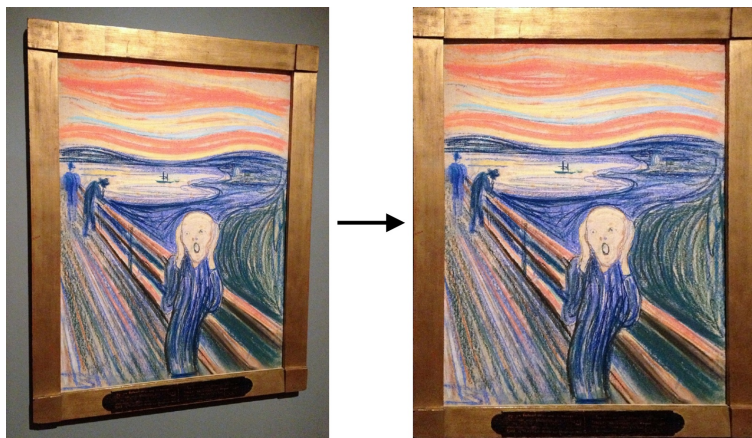


Figure 3.1: Perspective free version of a famous drawing, generated using a homography that was estimated from its publicly known size of 74 cm \times 91 cm.

In the remainder of this thesis we will consider only the special case of the targeted world plane being the XY -plane of the scene, with the perspective free version generated to have the image origin coinciding with the world origin and to have the X - and Y -axis of the scene coinciding with the x - and y -axis of the image respectively. In this case (U, V) equals (X, Y) .

3.2 Planes Parallel to the Target Plane

Because the computations to remove perspective for a plane relies on resampling the source image, the process can be viewed in terms of ray tracing in the source scene, with all rays emanating from the camera centre and being directed towards the target plane. The points where these rays intersect the target plane form a regular grid.

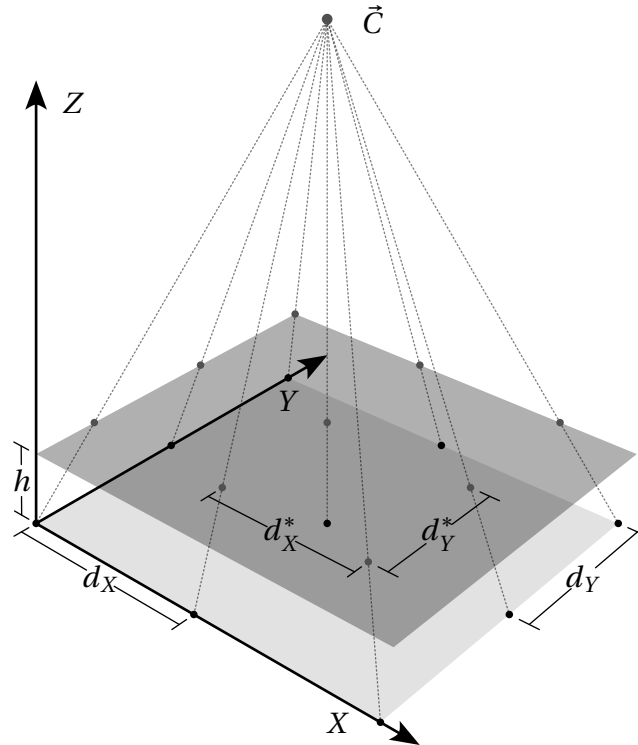


Figure 3.2: Planes parallel to the target plane also have perspective distortion removed.

Visualising the camera as above the plane, we now consider a plane parallel to and slightly above the target plane as illustrated in figure 3.2. We will refer to this plane as the secondary plane. Keeping in mind that the target plane is the XY -plane, we can write the 3D sample points at the ray-plane intersections in the bottom (target) plane as

$$\vec{P}_{i,j} = j \begin{bmatrix} d_X \\ 0 \\ 0 \end{bmatrix} + i \begin{bmatrix} 0 \\ d_Y \\ 0 \end{bmatrix},$$

where d_X and d_Y are the spacing of the sample points in the X - and Y -axis directions respectively. The integer pixel coordinates in the destination image are

given by (j, i) , with $(0,0)$ corresponding with the ray passing through the world origin.

If we define the secondary plane as $Z = h$ and assume the camera position to be $\vec{C} = [C_X \ C_Y \ C_Z]^T$, then the ray-plane intersections in the secondary plane can be calculated as

$$\begin{aligned} \vec{P}_{i,j}^* &= \vec{C} + \left(1 - \frac{h}{C_Z}\right) (\vec{P}_{i,j} - \vec{C}) \\ &= \frac{h}{C_Z} \vec{C} + \left(1 - \frac{h}{C_Z}\right) \vec{P}_{i,j} \\ &= \begin{bmatrix} \frac{h}{C_Z} C_X \\ \frac{h}{C_Z} C_Y \\ h \end{bmatrix} + j \left(1 - \frac{h}{C_Z}\right) \begin{bmatrix} d_X \\ 0 \\ 0 \end{bmatrix} + i \left(1 - \frac{h}{C_Z}\right) \begin{bmatrix} 0 \\ d_Y \\ 0 \end{bmatrix}. \end{aligned} \quad (3.1)$$

Equation 3.1 shows that the coordinates to which the origin on the target plane correspond on the secondary plane are shifted in the X and Y directions by $\frac{h}{C_Z} C_X$ and $\frac{h}{C_Z} C_Y$ respectively. Furthermore, by sampling the target plane in a regular grid with spacings of d_X and d_Y along the world axis directions, the secondary plane will also be sampled in a regular grid with spacings of d_X^* and d_Y^* where

$$d_X^* = \left(1 - \frac{h}{C_Z}\right) d_X \quad \text{and} \quad d_Y^* = \left(1 - \frac{h}{C_Z}\right) d_Y.$$

This shows that removing the perspective distortion for the target plane $Z = 0$ also correctly removes the distortion for any parallel secondary plane $Z = h$, with only an additional translation and scaling factor in the sampling grid.

In chapter 6 we will use this property of perspective removal in parallel planes to analyse the moiré patterns that result when high frequency patterns in such planes are superimposed.

Chapter 4

Single Frequency Analysis Using the DFT

In this chapter we analyse sinusoidal functions of distance. The techniques that are described for frequency and phase estimation will be used for moiré pattern analysis in the following chapters.

In the spatial domain we denote position by x when working in one dimension and by $[x \ y]^T$ when working in two dimensions. We use ξ and $\vec{\xi} = [\xi_x \ \xi_y]^T$ for the corresponding axes in frequency domain in one and two dimensions respectively.

4.1 Analysis in One Dimension

We consider a sinusoidal function of x :

$$g(x) = \cos(2\pi f x - \phi), \quad (4.1)$$

where f represents the frequency, measured in cycles per unit of distance, x represents distance and ϕ represents the phase delay in radians.

We can think of $\frac{\phi}{2\pi}$ as the fraction of a wavelength by which a cosine function at the specified frequency is shifted in the direction of increasing x . The wavelength in this context is defined as

$$\lambda = \frac{1}{f}.$$

Given a set of n uniformly spaced samples y_0, y_1, \dots, y_{n-1} taken of a function of the form given in equation 4.1, we may wish to calculate the frequency and phase parameters f and ϕ . In order to be able to calculate f we must know the sampling density, which we call the sampling frequency, and in order to calculate ϕ we also need a reference zero point. We assume a sampling frequency of f_s ,

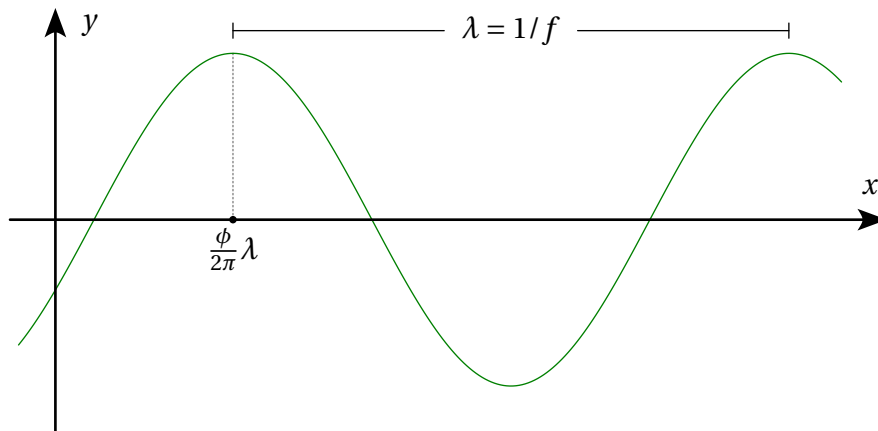


Figure 4.1: Sinusoidal function of x with frequency f and phase delay ϕ .

with $f_s > 2f$ for the Nyquist sampling criterion to be satisfied. For now we further assume that the first sample in the set is taken at $x = 0$.

Because $g(x)$ is nonlinear in f and ϕ , linear algebra will not offer a direct solution. However, we can apply a linear transform in order to rewrite the sample set in a form that allows for f and ϕ to be calculated conveniently. We turn to spectral analysis, first applying the discrete Fourier transform (DFT). The DFT will transform the sample set of real input values to a set of complex coefficients $\hat{y}_0, \hat{y}_1, \dots, \hat{y}_{n-1}$ such that

$$\begin{aligned} y_p &= \sum_{q=0}^{n-1} \hat{y}_q \cdot e^{2\pi i p q / n} \\ &= \sum_{q=0}^{n-1} \hat{y}_q \cdot (\cos(\omega_q p) + i \sin(\omega_q p)), \quad p \in \{0, 1, \dots, n-1\}, \end{aligned} \quad (4.2)$$

where $\omega_q = \frac{2\pi q}{n}$. The index q iterates over different frequency components and the index p iterates over the original samples.

After applying the transform we determine an amplitude spectrum from the set of magnitudes of the complex coefficients \hat{y}_q for spectral analysis. As a rough first estimate, we can now approximate the frequency f as $\frac{q}{n} f_s$ for the particular q where \hat{y}_q is of largest magnitude in the spectrum. This allows us to estimate f with accuracy up to the coarse sampling that the DFT provides in the frequency domain.

It is important to note here that because our input values are real-valued (our cosine is not a complete complex exponential function), we can expect two equally large maxima, which we refer to as peaks, in both the positive and negative spectral components of the DFT. This can be visualised by writing the cosine

as the sum of complex exponentials:

$$\begin{aligned}\cos t &= \frac{1}{2}(\cos t + i \sin t) + \frac{1}{2}(\cos t - i \sin t) \\ &= \frac{1}{2}e^{it} + \frac{1}{2}e^{-it}.\end{aligned}$$

Because the sinusoidal functions of interest to this thesis are real-valued, we can expect the positive and negative spectral peaks to appear symmetrical about the zero frequency when the input data is transformed to the frequency domain. We analyse only the positive half of the spectrum.

For an improvement in the accuracy with which f can be estimated as compared to the coarse method suggested above, we begin by applying a Gaussian window function before application of the DFT:

$$y'_p = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{p-\frac{n}{2}}{2\sigma^2}\right)} y_p, \quad p \in \{0, 1, \dots, n-1\}, \quad (4.3)$$

where y'_0, \dots, y'_{n-1} are the samples on which we then perform the DFT. This window function is symmetrical around $\frac{n}{2}$. We express it in a form that corresponds to the probability density function of a normal distribution with mean $\frac{n}{2}$ and standard deviation σ .

We note that the Fourier transform of a Gaussian function is again a (different) Gaussian function [3]. By multiplying by a Gaussian window function in the spatial domain, we are effectively convolving the frequency spectrum with a Gaussian function, resulting in an amplitude spectrum which is the sum of a pair of Gaussian functions: one centred on the positive and one on the negative representation of f .

Since the Gaussian function is infinitely wide and we have a finite sample set, we can only use an approximation by truncating it. The Gaussian function drops off rapidly in magnitude as one moves away from the mean, so we assume that such an approximation is acceptable if a Gaussian function of sufficiently narrow standard deviation is chosen in the spatial domain. We typically choose a value for σ in the range $\frac{n}{8}$ to $\frac{n}{4}$. We also assume that this provides a pair of Gaussian functions in the frequency domain that are still sufficiently narrow to not interfere with one another when analysed.

When analysing only the positive half of the amplitude spectrum, it should be approximately a sampled version of a Gaussian function of the form

$$\hat{g}(\xi) = h e^{-\left(\frac{w(\xi-f)n}{f_s}\right)^2}, \quad h > 0,$$

for some parameters h and w . This can be rewritten in the form

$$\hat{g}(\xi) = e^{a\xi^2 + b\xi + c}, \quad a < 0, \quad (4.4)$$

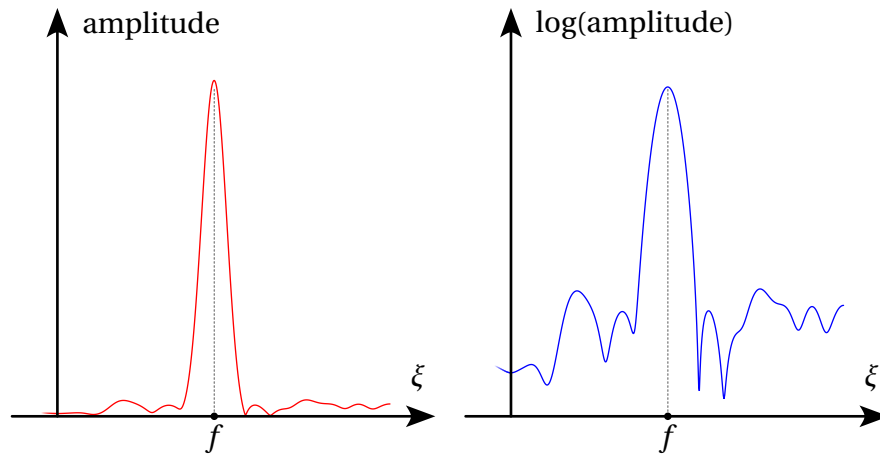


Figure 4.2: The amplitude spectrum of a sinusoidal function with some random noise added. On a linear scale the spectral peak is approximately a Gaussian function (left) while on a logarithmic scale it is approximately parabolic (right).

for some a , b and c .

Equation 4.4 implies that the natural logarithm of the amplitude spectrum is simply a quadratic function of frequency: $a\xi^2 + b\xi + c$. We can therefore use three samples in the amplitude spectrum to fit a parabola and so calculate the values of a , b and c [2]. We can then use the formula for the global maxima of a parabola, $f = \frac{-b}{2a}$, to obtain with sub-sample accuracy the frequency f where the Gaussian function is centred. Figure 4.2 illustrates by example this amplitude peak shape in the frequency domain on linear and logarithmic scales, while figure 4.3 illustrates the method of accurately finding the location of a peak maximum in frequency domain.

In practice, when there is also some noise present in the sample set, we want to fit the parabola through frequency samples where the log values are least affected by noise. We would typically choose the largest amplitude sample and its two neighbouring samples for this purpose. If we assume roughly equal noise levels across the spectrum then it follows that those samples that contain the targeted signal with largest amplitude have the smallest relative error in its representation. The logs of these values therefore have the smallest absolute error and applying the quadratic fit there will yield a good approximation.

To calculate ϕ we inspect equation 4.2 for a particular component associated with coefficient \hat{y}_q . Assuming that this coefficient is expressed in the form $a_q + b_q i$, the component becomes

$$\begin{aligned} \hat{y}_q \cdot e^{i2\pi q/n} &= (a_q + b_q i) \cdot (\cos(\omega_q p) + i \sin(\omega_q p)) \\ &= a_q \cos(\omega_q p) - b_q \sin(\omega_q p) + i (b_q \cos(\omega_q p) + a_q \sin(\omega_q p)). \end{aligned}$$

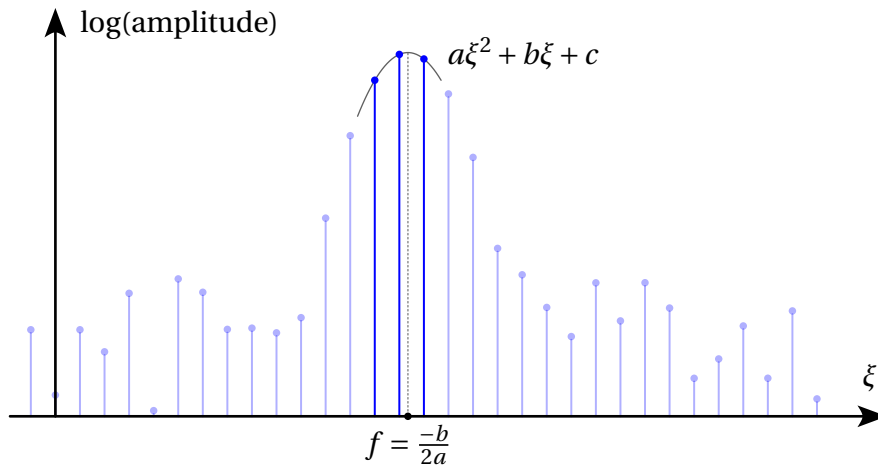


Figure 4.3: Parabola fitted through three points on the log(amplitude) spectrum.

Since our original samples y_0, y_1, \dots, y_{n-1} are all real, we need only look at the real part of the expression. We can see that for a particular coefficient \hat{y}_q and its associated frequency, a_q represents the amplitude of the cosine and $-b_q$ the amplitude of sine present at that frequency. A linear combination of a cosine and sine of equal frequency can be expressed in the form of equation 4.1, so we calculate

$$\phi_q = \tan^{-1}(-b_q/a_q). \quad (4.5)$$

By inspecting this value for the coefficient associated with the frequency of maximum amplitude, we find ϕ .

In practice we would use the function `atan2`, available in a variety of computer languages, which takes $-b_q$ and a_q as separate inputs and allows the phase to be calculated to the full range $(-\pi, \pi]$, expressed in the correct quadrant. Otherwise, by taking the fraction as in equation 4.5 the individual signs of b_q and a_q are combined and some information is lost.

Dropping one of our initial assumptions, if the phase needs to be expressed relative to a different origin than the first sample in the set then we can compensate by adding a multiple of 2π . The multiple of 2π added equals the number of wavelengths by which the origin is offset to the right of the first sample point:

$$\phi = \phi' + (0 - s)f,$$

where ϕ' is the phase expressed relative to the first sample y_0 , and s is the x -coordinate of the first sample.

4.2 Analysis in Two Dimensions

We now consider a 2D sinusoidal function of the form

$$g(x, y) = \cos\left(2\pi \vec{f}^T \begin{bmatrix} x \\ y \end{bmatrix} - \phi\right),$$

where \vec{f} represents the 2D frequency, x and y represent two spatial dimensions and ϕ represents the phase delay in the direction of \vec{f} in radians. Such a function is shown in figure 4.4.

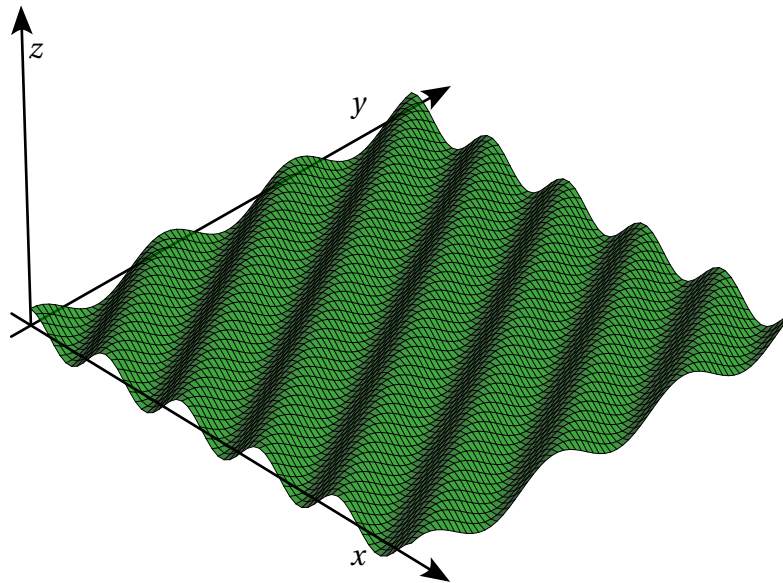


Figure 4.4: A sinusoidal function in 2D.

As in the 1D case above, we may wish to calculate \vec{f} and ϕ from a square grid of samples:

$$\begin{array}{cccc} z_{0,0} & z_{0,1} & \dots & z_{0,n-1} \\ z_{1,0} & z_{1,1} & \dots & z_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n-1,0} & z_{n-1,1} & \dots & z_{n-1,n-1} \end{array}.$$

We use a similar approach, using a Gaussian window function and the DFT, but this time in two dimensions. This results in the amplitude being expressible as a Gaussian function such as shown in figure 4.5. Its natural logarithm being expressible as a bivariate quadratic function

$$G(\xi_x, \xi_y) = e^{A\xi_x^2 + B\xi_y^2 + C\xi_x + D\xi_y + E\xi_x\xi_y + F}.$$

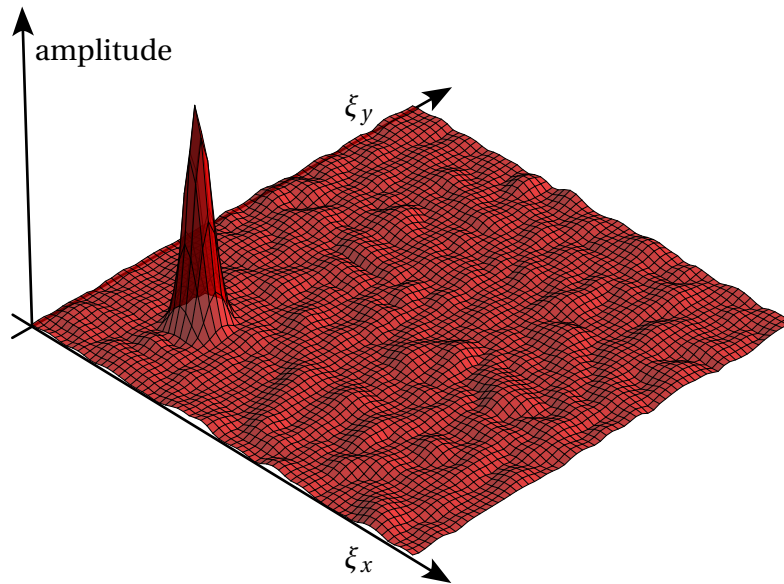


Figure 4.5: Gaussian peak visible in the amplitude spectrum of a simulated 2D sinusoidal with some random noise added.

As before, we wish to find a local maximum for the Gaussian function, which is where the bivariate quadratic has its maximum. Assuming that some noise is present on the measurement, we want to fit the quadratic function through the values least affected by the noise. We therefore choose in the frequency domain the largest amplitude value and its five largest neighbours (from the eight neighbouring samples available) to apply the fit on.

With values A, \dots, F solved we calculate \vec{f} as

$$\vec{f} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$

where

$$f_x = -\frac{2BC - DE}{4AB - E^2} \quad \text{and} \quad f_y = -\frac{2AD - CE}{4AB - E^2}.$$

The phase ϕ is estimated as in the 1D case using the function at an2 from the complex value associated with the highest amplitude. An adjustment is made for a reference origin other than sample $z_{0,0}$ by adding a multiple of 2ϕ for how far the reference origin is shifted relative to this sample point in the direction of \vec{f} , measured in wavelengths:

$$\phi = \phi' + (\vec{0} - \vec{s}) \cdot \vec{f},$$

where ϕ' is the phase referenced to the first sample $z_{0,0}$, $\vec{0}$ is the zero vector and \vec{s} is the position of the first sample.

We have now described a method to estimate frequency and phase of a sampled sinusoidal function in 2D with some noise present. In the case where the sampled function is the sum of multiple 2D sinusoidal functions of which one must be analysed, we need prior knowledge about the particular component we wish to analyse.

Given limits in the range of frequency that we expect our targeted component to have, we could search for the sample of maximum amplitude in the frequency domain merely in the region that adheres to these limits. This strategy would work, provided that the component of maximum amplitude within this region is the targeted component. We use this strategy for moiré analysis in chapter 5.

Chapter 5

Analysis of Moiré Patterns in Flat Geometry

A moiré pattern is a visual artefact that appears when two high spatial frequency patterns with slight differences in scale, orientation or composition are superimposed, resulting in the emergence of an apparent low spatial frequency pattern.

Line moiré, like the example shown in figure 5.1, is a special case where the individual patterns consist of closely spaced lines or curves. The effect can be illustrated by placing a transparency printed with fine, uniformly spaced strips alternating between black and clear on top of a sheet of white paper printed with similar strips at a slightly different frequency. The result is the appearance of an apparent large wavelength variation between light and dark.

With this method of constructing moiré patterns we refer to the paper as the base layer and the transparency as the revealing layer, in accordance with the terminology used by Gabrielyan [6].

When imaging the pattern, spatial bandwidth limitation in the optical imaging device used may enhance the effect if the variation in the longer wavelength (lower frequency) resultant variation is preserved whereas the variation in the original shorter wavelength (higher frequency) patterns is partially or completely filtered. This can be illustrated for the case of the human eye by ensuring that the subject is standing sufficiently far from the patterns for the alternating strips to be indiscernible on both layers, leaving only the moiré pattern visible.

In the line moiré example above, the variation in each layer alternates between solid black and clear when travelling perpendicular to the lines, forming a square wave of opacity. In this thesis we consider the special case where the base and revealing patterns vary sinusoidally in opacity along a single direction. Specifically, we will look at cosine functions in two dimensions where moiré pattern analysis can easily be done using the efficient 2D fast Fourier transform (FFT) algorithm.

Since we can rewrite all periodic functions of opacity, such as the square wave of line moiré, as the sum of a series of basic sinusoidal functions [3], we could

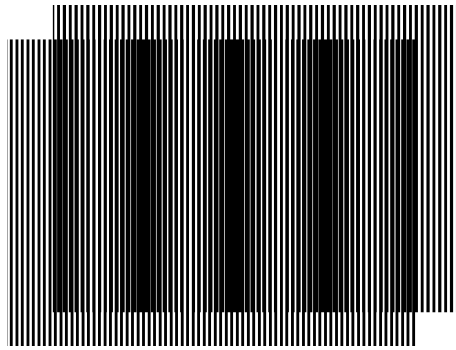


Figure 5.1: Example of line moiré.

describe these other cases in terms of our special case. Keeping this in mind, we consider only this special case of basic sinusoidal functions. The description of other patterns falls outside of the requirements of the thesis.

5.1 Product of Sinusoids in One Dimension

In the case of sinusoidal patterns in one dimension being superimposed (multiplied), a low frequency pattern is a result of the trigonometric identity

$$\cos(a)\cos(b) = \frac{1}{2}\cos(a+b) + \frac{1}{2}\cos(a-b). \quad (5.1)$$

We consider two sinusoidal wave functions:

$$\begin{aligned} g_1(x) &= \cos(2\pi f_1 x), \\ g_2(x) &= \cos(2\pi f_2 x). \end{aligned}$$

The parameters f_1 and f_2 represent frequency expressed as wavelengths per unit of distance. Applying equation 5.1 we obtain

$$g_1(x)g_2(x) = \frac{1}{2}\cos(2\pi(f_1 + f_2)x) + \frac{1}{2}\cos(2\pi(f_1 - f_2)x). \quad (5.2)$$

Stated simply, when two sinusoidal functions of different frequencies are multiplied, the resulting product can be expressed as the sum of two new sinusoidal functions, one at the sum of the frequencies and one at the difference when viewed in the frequency domain. Figure 5.2 shows an illustration of this.

If the original two frequencies in equation 5.2, f_1 and f_2 , are sufficiently high then they would be strongly attenuated during observation when imaged or sampled by a limited bandwidth sensor. The sum frequency term of even higher frequency can then be assumed to also be filtered, while the difference frequency term becomes prominent.

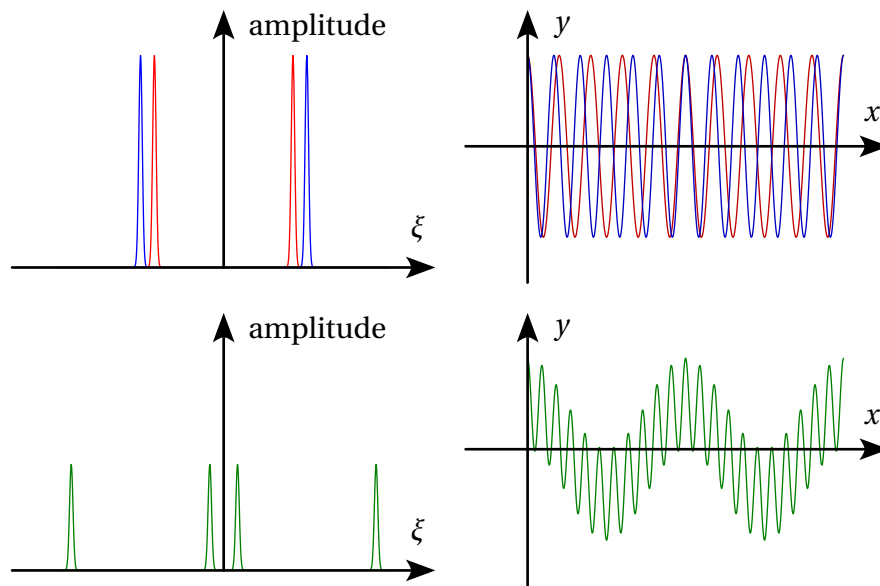


Figure 5.2: Difference and sum frequencies (green) resulting from multiplication of two sinusoidal functions (red and blue). The frequency domain is shown on the left.

We denote the frequency of the difference term as

$$f_{\Delta} = \pm(f_2 - f_1). \quad (5.3)$$

The sign ambiguity here comes from being able to express the frequency as either positive or negative. The only consequence of working with the negative frequency is that the phase is reversed. Changing the sign of the phase and the frequency simultaneously has no effect on a sinusoidal as expressed in equation 4.1 because of the symmetric nature of the cosine function.

Initially we will consider only the positive case of equation 5.3. Where appropriate, we will include the sign ambiguity or where convenient, switch to the negative case. For the remainder of this thesis we will consider f_1 and f_2 to be both significantly larger in magnitude than f_{Δ} .

5.2 Product of Sinusoids in Two Dimensions

If \vec{f}_1 and \vec{f}_2 are 2D vectors then they describe 2D frequencies and the difference term, when visualised on a 2D image, becomes the moiré pattern. Figure 5.3

illustrates. This can be expressed as

$$g_1(x, y) = \cos\left(2\pi \vec{f}_1^T \begin{bmatrix} x \\ y \end{bmatrix}\right),$$

$$g_2(x, y) = \cos\left(2\pi \vec{f}_2^T \begin{bmatrix} x \\ y \end{bmatrix}\right),$$

so that

$$g_1(x, y)g_2(x, y) = \frac{1}{2} \cos\left(2\pi(\vec{f}_1 + \vec{f}_2)^T \begin{bmatrix} x \\ y \end{bmatrix}\right) + \frac{1}{2} \cos\left(2\pi(\vec{f}_1 - \vec{f}_2)^T \begin{bmatrix} x \\ y \end{bmatrix}\right).$$

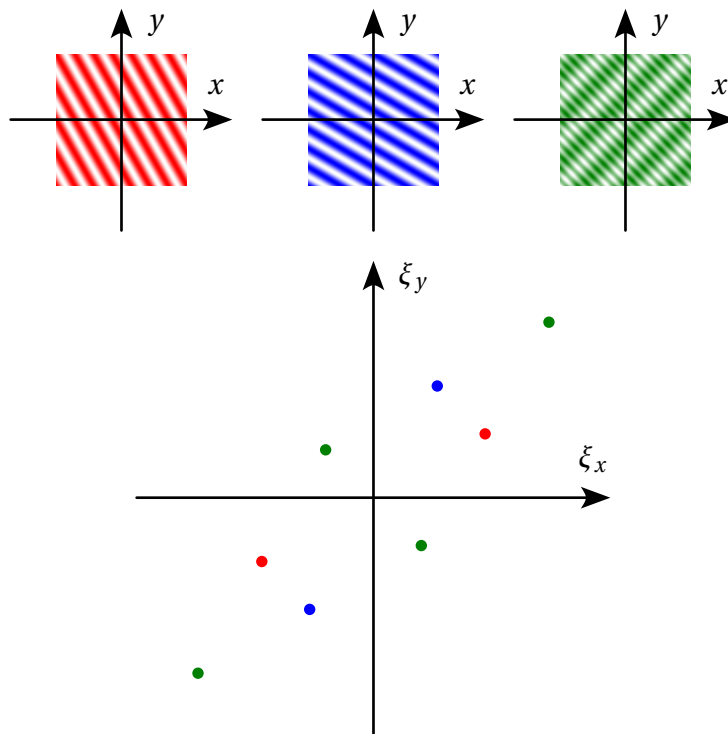


Figure 5.3: 2D spatial difference and sum frequencies (green) resulting from multiplication of two sinusoidal functions (red and blue). The bottom graph shows the location of the spectral elements in the frequency domain.

Since we are considering the case where \vec{f}_Δ is of significantly lower magnitude than \vec{f}_1 and \vec{f}_2 , we can use the frequency analysis techniques from chapter 4 to estimate the frequency and phase of the frequency component that forms the moiré pattern. We can place limits on our search for the sample of maximum amplitude in the frequency domain to ensure that the correct component responsible for the moiré pattern is selected.

5.3 Sensitivity of Frequency Estimation from Moiré Analysis

If the frequency vector \vec{f}_1 is known, then performing analysis on the frequency of the resulting moiré pattern (the difference between \vec{f}_1 and \vec{f}_2) would allow the frequency vector \vec{f}_2 to be calculated as the sum of \vec{f}_1 and that difference:

$$\vec{f}_2 = \vec{f}_1 + \vec{f}_\Delta.$$

If the frequency of the moiré pattern can, through analysis, be measured to within some unknown absolute error vector $\vec{e} = \begin{bmatrix} e_x \\ e_y \end{bmatrix}$, then we can express the measured frequency difference as

$$\vec{f}_\Delta^* = \vec{f}_\Delta + \begin{bmatrix} e_x \\ e_y \end{bmatrix}. \quad (5.4)$$

The vector \vec{f}_2 can now be approximated as

$$\begin{aligned} \vec{f}_2^* &= \vec{f}_1 + \vec{f}_\Delta^* \\ &= \vec{f}_2 + \begin{bmatrix} e_x \\ e_y \end{bmatrix}. \end{aligned} \quad (5.5)$$

We notice that the same absolute error term occurs in equations 5.4 and 5.5, but we know \vec{f}_2 to be of significantly larger magnitude than \vec{f}_Δ . Therefore, if this error term represents a moderate relative error in estimation of \vec{f}_Δ , we can expect a significantly smaller relative error in the estimation of \vec{f}_2 .

This property of moiré analysis is very useful for measuring fine variations in \vec{f}_2 even when the high frequency pattern associated with \vec{f}_2 cannot be observed directly and even if the frequency analysis method produces a moderate relative error in direct frequency estimation.

5.4 Sensitivity of Phase Estimation from Moiré Analysis

Expanding the 2D sinusoidal functions to also include a phase, we write:

$$\begin{aligned} g_1(x, y) &= \cos(2\pi(\vec{f}_1)^T \begin{bmatrix} x \\ y \end{bmatrix} - \phi_1), \\ g_2(x, y) &= \cos(2\pi(\vec{f}_2)^T \begin{bmatrix} x \\ y \end{bmatrix} - \phi_2). \end{aligned}$$

The parameters ϕ_1 and ϕ_2 represent the phase delays in radians and can be thought of as a shift of the waveform by $\phi/2\pi$ wavelengths in the direction of \vec{f} in each case (added delay). Applying equation 5.2 we obtain

$$\begin{aligned} g_1(x, y)g_2(x, y) &= \frac{1}{2} \cos \left(2\pi(\vec{f}_1 + \vec{f}_2)^T \begin{bmatrix} x \\ y \end{bmatrix} - (\phi_1 + \phi_2) \right) \\ &+ \frac{1}{2} \cos \left(2\pi(\vec{f}_1 - \vec{f}_2)^T \begin{bmatrix} x \\ y \end{bmatrix} - (\phi_1 - \phi_2) \right). \end{aligned}$$

Inspecting only the low frequency term (the moiré pattern), it is apparent that any phase delays in the original two sinusoidal functions manifest as a phase delay in the lower frequency sinusoidal that is equal to the difference of the phase delays at the original frequencies, as expressed in radians.

Similar to the case with frequency, even if the original two sinusoidal functions cannot be observed directly, if the phase of one of the two is known then the phase of the other can be calculated from the measured phase of the moiré pattern as the difference.

An additional benefit comes from a phenomenon known as moiré speedup [6]. When changing the phase of the pattern in one of the layers by moving that layer by some fraction of the short wavelength of the pattern on that layer, the same fraction of a wavelength move is observed in the moiré pattern. However, the moiré pattern has a much longer wavelength and thus appears to be moving much more rapidly.

During analysis, we make use of this effect. The lower the frequency of a sampled 2D sinusoidal pattern is, the more pixels a wavelength corresponds to. Assuming that the error in phase analysis has a fixed magnitude expressed in units of pixels, this means that the absolute error in phase analysis, expressed in radians, is inversely proportional to wavelength of the sinusoidal pattern being analysed. Since the moiré pattern is assumed to be of much longer wavelength than the original sinusoidal functions, its phase can be measured, and the phase of one of the layer patterns can be calculated from it, with much greater accuracy.

In the next chapter we will show how the discussed benefits of frequency and phase estimation by means of moiré pattern analysis allow us to detect subtle changes in apparent scale of the base and revealing layers. We will use this technique to derive information about the camera's position in relation to the layers.

Chapter 6

Analysis of Moiré Patterns in Parallel Planes

We now expand on the moiré patterns discussed in chapter 5 by replacing the transparency (revealing layer) with a thin, rigid glass sheet. We can suspend it at some small distance parallel over the patterned paper (base layer) to observe the resulting superimposition. In this case, however, we must also account for the perspective that causes the pattern on the glass sheet (nearer to the viewer) to appear to have a slightly lower spatial frequency. In this chapter we show how the slight change in frequency will affect the moiré pattern and, given an image of the moiré pattern, we show how we can estimate the camera position relative to the base layer through frequency and phase analysis.

Considering the case where the same true spatial frequency is represented on both the glass sheet and the paper, we can expect a slight change in the apparent frequency on the glass sheet to cause a moiré pattern with a low frequency component, corresponding to the small difference in observed frequencies. A simulation of this is shown in figure 6.1.

In chapter 5 we assumed that all analysis is performed on a sampled version of the moiré pattern. We assumed that scale and rotation of measurements were relative to the sampling grid used, with no need for an external reference.

In order to perform meaningful analysis on the moiré pattern resulting from parallel planes in 3D, we must acknowledge the existence of a world coordinate system which is not necessarily aligned with our camera coordinate system. We will opt to rely on a set of guides of known world and image coordinates to find the appropriate mapping between image coordinates and world coordinates.

We assume a pinhole camera model with no lens distortion. A complete mapping between the two coordinate systems would now require at least $5\frac{1}{2}$ non-planar point correspondences, as shown in chapter 2. But, recalling the properties of perspective removal mappings on parallel planes in chapter 3, we require only four correspondences of points within one of the planes to calculate a mapping for both.

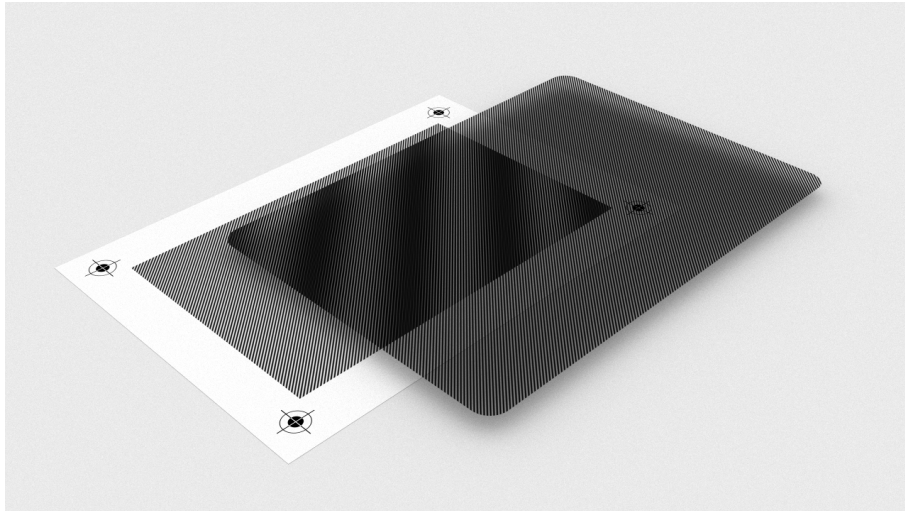


Figure 6.1: Moiré pattern resulting from a specially printed glass sheet suspended over printed paper.

Including at least four guides of known world coordinates and for which corresponding image coordinates can be found on the base layer (such as those printed on the corners of the paper in figure 6.1), a mapping between world coordinates and image coordinates can be estimated for that plane and any plane of known distance parallel to it. Image coordinates are calculated for an axis-aligned regular grid of world coordinates in this plane and the image resampled to obtain a perspective-free version such as the one shown in figure 6.2. Noting that we have now also sampled the parallel revealing layer in an axis-aligned regular grid, we will be analysing the moiré pattern in this resampled image.

In the following sections we develop a theory for the expected moiré pattern frequencies based on the expected range of camera position and the composition of the patterned layers. We choose our grid of samples with a sampling density that meets the Nyquist sampling criterion for any moiré patterns we expect to encounter. We also benefit by choosing the sampling grid with a size in each dimension that is an integer power of 2, allowing us to make use of the efficient FFT implementation of the DFT. While there is no benefit to sampling in intervals smaller than one pixel in the original image, resampling too dense is preferred to sampling too sparse as aliasing effects that result when subsampling may be detrimental to the frequency and phase analysis steps that follow.

6.1 Frequency Analysis

We assume that the base plane is the XY -plane. Using world axis-aligned spacings of d_X and d_Y for the grid in the base layer, we recall from equation 3.1 that

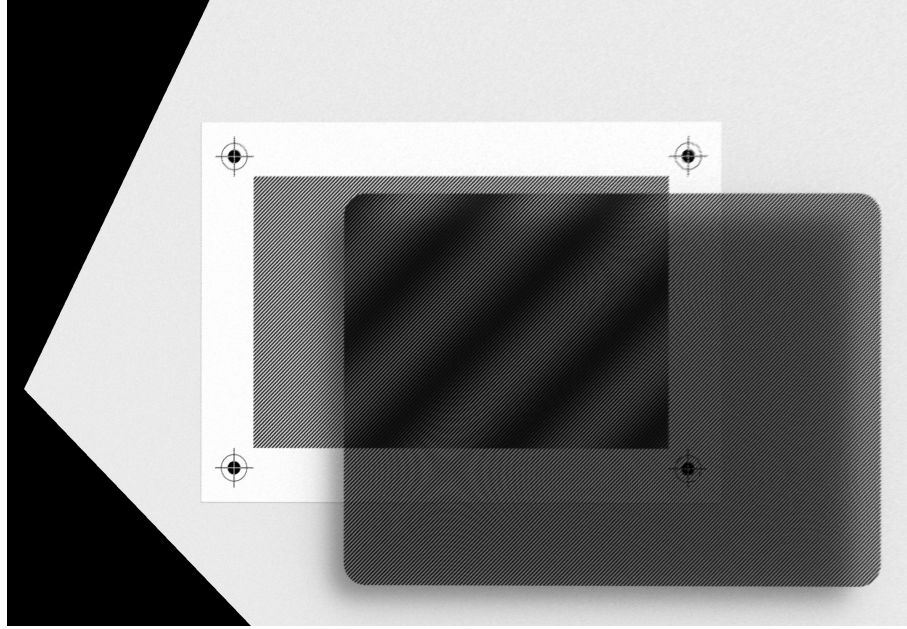


Figure 6.2: Perspective corrected version of figure 6.1 using the guides as a set of point correspondences.

we thereby sample the transparent plane at axis-aligned spacings of d_X^* and d_Y^* , with

$$\begin{aligned} d_X^* &= \left(1 - \frac{h}{C_Z}\right) d_X, \\ d_Y^* &= \left(1 - \frac{h}{C_Z}\right) d_Y, \end{aligned}$$

where the parameter h is the perpendicular distance between the planes.

Compared to the base layer, both axes are sampled finer by a factor of $(1 - \frac{h}{C_Z})$, so we can assume the frequency of any sinusoidal measured on the resampled image to be lowered by this factor when expressed in any form which is proportional to cycles-per-sample along each axis.

In the case of sinusoidal opacity functions of equal true spatial frequency and alignment (rotation) being present on the planes, the resulting moiré pattern in the resampled image is purely as a result of the difference in observed frequency. We first consider the effect in one dimension along a single axis:

$$\begin{aligned} f_\Delta &= \pm(f_r - f_b) \\ &= \pm\left(1 - \frac{h}{C_Z}\right)f_b - f_b \\ &= \mp \frac{h}{C_Z} f_b, \end{aligned} \tag{6.1}$$

where f_b is the frequency on the base plane (expressed as cycles per unit length), f_r is the frequency observed in the revealing plane, expressed in cycles per unit length as if it had been observed in the base plane, thus of lower apparent frequency. Parameter f_Δ is the resulting frequency directly measured on the mapping.

From equation 6.1 it is clear that the absolute value of f_Δ will decrease as the absolute value of C_Z increases. The further the camera is along the Z -axis, the lower the frequency f_Δ will be calculated to be. Equation 6.1 can now be rewritten to calculate the height (Z -coordinate) of the camera from the frequency measured through analysis:

$$C_Z = \pm \frac{h}{f_\Delta} f_b.$$

Apart from the assumption about the absence of lens distortion, the above formula can be applied to calculate the Z -coordinate of the camera without any prior knowledge of the camera used. The sign ambiguity is easily resolved by noting that the camera is viewing the layers from above, allowing us to write

$$C_Z = \parallel \frac{h}{f_\Delta} f_b \parallel.$$

Since we are not doing any analysis on phase yet, it is unnecessary to keep track of whether the positive or negative value of f_Δ was used.

For the more general case where the frequencies represented on the two layers are not equal, we have

$$\begin{aligned} f_\Delta &= \pm(f_r - f_b) \\ &= \pm \left(\left(1 - \frac{h}{C_Z}\right) f_t - f_b \right), \end{aligned} \quad (6.2)$$

where f_t is the true frequency in the revealing layer (as opposed to f_r , the observed frequency).

Again, we can rewrite this formula to calculate the height of the camera:

$$C_Z = \frac{h}{1 - \frac{f_b \pm f_\Delta}{f_t}}. \quad (6.3)$$

In this case it is not so clear how the sign ambiguity of f_Δ should be interpreted. We can evaluate by setting f_Δ to zero:

$$C_Z = \frac{h}{1 - \frac{f_b}{f_t}}.$$

Here we can see that the height where $f_\Delta = 0$ depends on the ratio of f_b and f_t .

For example, when f_b is $\frac{1}{2}f_t$, f_Δ will be zero when $C_Z = 2$. In general, a positively or negatively interpreted f_Δ will result in different possible values for C_Z . We therefore need to keep track of which interpretation of f_Δ is chosen.

The ratio of f_b and f_t is also a useful quantity to keep track of, as will become apparent in the following chapter. We therefore define ρ as

$$\rho = \frac{f_b}{f_t}.$$

We use this definition to simplify equation 6.2 to

$$f_\Delta = \pm f_t \left(\rho - 1 + \frac{h}{C_Z} \right). \quad (6.4)$$

Equation 6.3 now simplifies to

$$C_Z = \frac{h}{1 - \rho \pm \frac{f_\Delta}{f_t}}. \quad (6.5)$$

6.2 Condition Number of Height Analysis from Frequency

The condition number of a function $f(x)$ of one variable can be calculated as a new function of that variable using

$$\kappa_f(x) = \frac{x f'(x)}{f(x)},$$

where $f'(x)$ denotes the first derivative of $f(x)$ with respect to x . Here the condition number is a measure of how a small relative error in x would translate to relative error in f when compared to the value of f associated with the correct value of x . The condition number is thus a ratio of relative errors.

Applying this to the positive case of equation 6.4, we can calculate the condition number of calculating f_Δ with respect to C_Z :

$$\begin{aligned} \kappa_{f_\Delta}(C_Z) &= \frac{C_Z f'_\Delta(C_Z)}{f_\Delta(C_Z)} \\ &= -\frac{h}{C_Z(\rho - 1) + h}. \end{aligned} \quad (6.6)$$

For the case where $\rho = 1$ (the frequencies used in the two layers are equal) we can see that the condition number reduces to -1 : a small relative increase

in C_Z leads to a small relative decrease in f . For other values of ρ , the condition number becomes dependent on C_Z , h and ρ . Assuming we can construct our apparatus with a custom ρ , we can choose ρ based on a desired condition number for a particular h and in the vicinity of a particular camera Z -coordinate:

$$\rho = 1 - \frac{h}{C_Z} \left(1 + \frac{1}{\kappa_{f_\Delta}} \right). \quad (6.7)$$

This ability to tune the condition number by which f_Δ is calculated from C_Z (or conversely, C_Z is calculated from f_Δ) forms one of the central advantages of camera position estimation using the method proposed in this thesis.

6.3 Phase Analysis

In addition to frequency, the 2D DFT output which is analysed as described in chapter 4 provides a phase value relative to the first sample along each axis. As we have shown, we can adjust this phase value to be made relative to the origin of the world coordinate system.

We will assume that the sinusoidal patterns on the two individual layers are aligned in rotation, so that $\vec{f}_b = u\vec{f}_t$ for some scalar u . Also assuming that the patterns both have zero phase offset relative to the origin in the XY -plane, we could expect zero phase offset when viewing them from above along the Z -axis.

Any movement of the camera in a direction perpendicular to that of \vec{f}_r would introduce no additional phase offset, whereas any movement parallel to it would be related to the observed phase as follows:

$$C_R = \frac{C_Z}{h} \lambda \left(\frac{\phi}{2\pi} + n \right), \quad (6.8)$$

where C_R is the component of the displacement from the Z -axis in the direction of \vec{f}_r , expressible as $\vec{C} \cdot \vec{f}_r / \|\vec{f}_r\|$, λ is the wavelength of \vec{f}_t , expressible as $1/\|\vec{f}_t\|$, and n is some integer. Figure 6.3 illustrates this.

The presence of n means there is an ambiguity. This comes from only being able to resolve the phase to the range $(-\pi, \pi]$ from observation, whereas any additional integer multiple of the top layer wavelength that it is displaced by gets discarded.

If n is known then C_R can be calculated directly. If n is not known, but an estimate for C_R is available, then a possible value for n can be calculated as

$$n = \text{round} \left(\frac{C_R^* h}{C_Z \lambda} - \frac{\phi}{2\pi} \right), \quad (6.9)$$

where C_R^* is our estimate of C_R and $\text{round}()$ denotes rounding to the nearest integer. This value of n would be correct if

$$|C_R^* - C_R| < 0.5 \frac{C_Z}{h} \lambda. \quad (6.10)$$

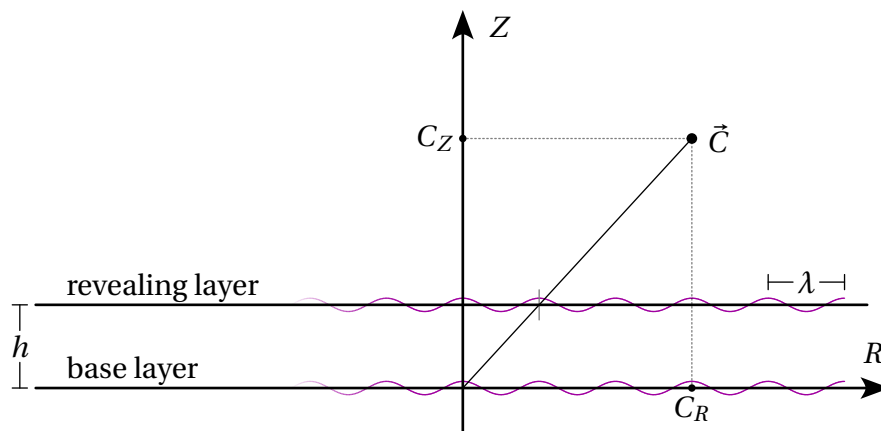


Figure 6.3: Moving in the direction of R , observed phase repeats every $\frac{C_Z}{h} \lambda$ units of length, introducing ambiguity.

In the next chapter we show how a sufficiently accurate estimate for the camera position allows us to calculate a more accurate C_X and C_Y by means of phase analysis. The definition of sufficient accuracy is related to equation 6.10.

Chapter 7

Moiré-based Camera Calibration Object

Based on the formulas derived in the previous chapters, we propose an object and method by which a subset of camera parameters can be calculated for a colour camera, given an image taken of the object.

The object consists of a specially printed glass sheet suspended over a modern tablet computer with a high resolution colour video display such as an OLED display or LCD. The display serves as the base layer to a 3D moiré pattern and the glass sheet serves as the revealing layer. The two layers are perfectly parallel and the glass is flat and rigid enough to be modelled successfully as a plane. The sheet is also thin enough that the refraction within the glass does not have to be taken into account and it can be modelled as infinitesimally thin.

The print on the glass sheet can be the sum of the outputs of one or more sinusoidal functions, scaled and shifted in opacity to fill the range of black to transparent. The tablet can display one or more sinusoidal functions, scaled and shifted to fill the range of black to white, as well as any guides that are required to find the mapping between world and image coordinates in the base plane, as required for moiré analysis in 3D.

In the case of the tablet we note that we have the option of colour multiplexing the patterns. By placing different patterns and/or guides in the red, green and blue channels of the display, we can rely on the colour filter on the camera's sensor to separate the resulting moiré patterns. This allows three analysable patterns to be captured in a single image of the object.

For the physical experiments performed, a 0.5 mm thick printed sheet of Corning[®] Gorilla[®] glass was suspended over a third generation iPad[®]. The glass was placed with the printed side down. All other experiments were performed in simulation based on ray-traced renderings of this combination to ensure known geometry and perfect alignment.

One particular arrangement of the above described object was used for research and is described in this chapter. This arrangement was found useful in

that it was realisable as a physical implementation with relative ease and the information extractable from the moiré patterns could be used in simulation to estimate the camera position.

In this arrangement the world X - and Y -axis are chosen to align with the pixel grid on the display, with the world origin in the centre of the display and the display coinciding with the $Z = 0$ plane. Two sinusoidal functions of brightness are displayed at perpendicular frequency vectors of equal magnitude and at $\pm 45^\circ$ angles with respect to the Y -axis in the red and blue channels. The green channel displays a 6×8 chequerboard.

The print on the glass sheet used is the sum of two sinusoidal functions of opacity at perpendicular frequency vectors of equal magnitude. The frequency vectors are also at $\pm 45^\circ$ angles relative to the Y -axis. The glass sheet is placed so that the directions of \vec{f}_t and \vec{f}_b are perfectly aligned.

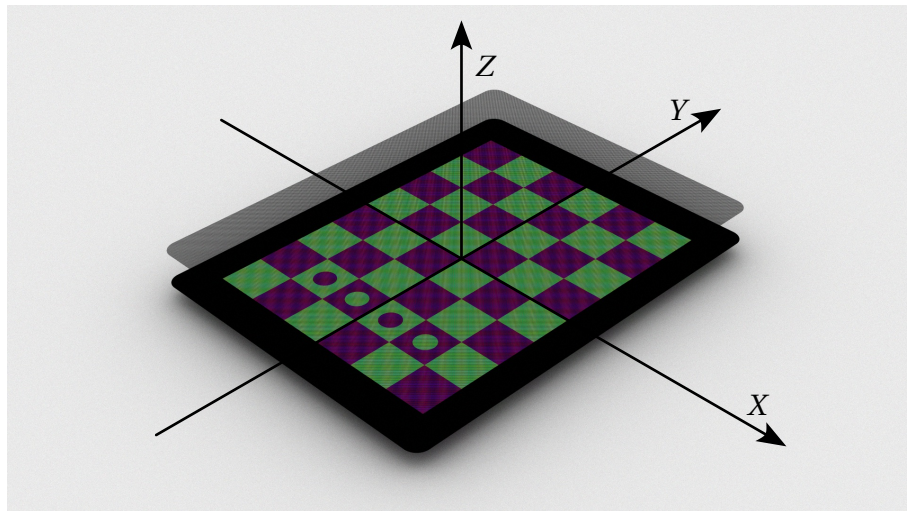


Figure 7.1: 3D rendering of proposed calibration object.

The $\pm 45^\circ$ angles are chosen rather than an axis-aligned arrangement because this allows higher spatial frequencies to be displayed on the pixel lattice of the tablet display. The benefit of this configuration is important. In chapter 8 we show how some useful arrangements require very high frequencies to be displayed.

The chequerboard pattern is chosen because `findChessboardCorners`, an available routine in OpenCV, can conveniently be used to automatically detect the inner corners of such a layout, where squares of similar colour meet. The focus of this thesis is on moiré pattern analysis and we rather rely on an industry standard technique for detecting guides as a necessary step leading up to the analysis. The green channel is chosen for this to benefit from the Bayer filter

pattern that is used in many modern colour cameras, favouring the green colour channel. After demosaicing is applied, the images produced offer highest effective resolution in the green channel. This is beneficial for accurate corner detection.

The chequerboard pattern is modified to have four disks of alternate colour at the centres of four of the squares at the bottom of the tablet. These disks are used to solve an ambiguity that arises from the symmetry of the chequerboard pattern. With these disks made small enough, they are found not to interfere with the corner detection in `findChessboardCorners`.

The red and blue channels are chosen to display two of the base layers of the moiré patterns. Unlike the case of the green colour channel where corner detection is required, the lower spatial resolution resulting from a Bayer pattern in the red and blue channels can be tolerated because low frequency sinusoidal waves that vary smoothly are measured in those channels.

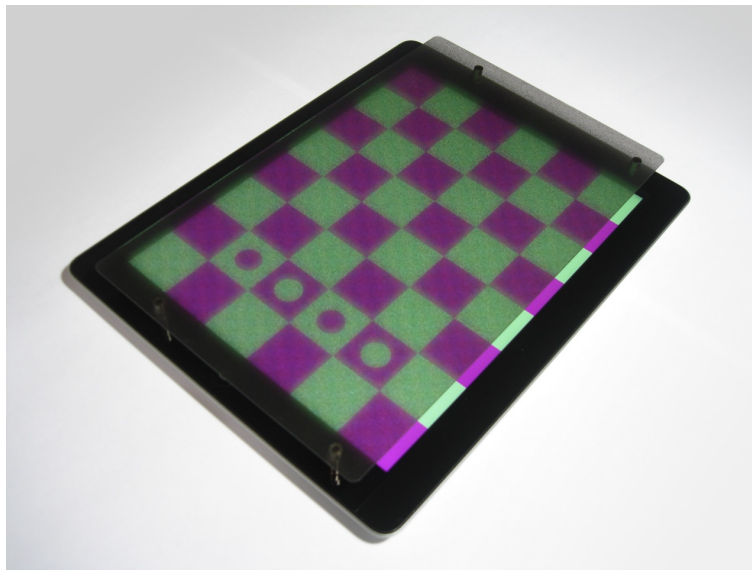


Figure 7.2: Image of a physical implementation of the proposed moiré-based calibration object.

The following parameters are yet to be defined:

- $\|\vec{f}_t\|$: the magnitude of the spatial frequency of the revealing layer pattern, expressed as wavelengths per unit length;
- $\rho = \|\vec{f}_b\|/\|\vec{f}_t\|$: the ratio of the magnitudes of the base layer frequency and the revealing layer frequency;
- h : the Z -displacement between the revealing and base layers.

We can vary the above parameters to achieve a particular objective. First, the theoretical moiré pattern that would be observed must be analysable at the desired camera position relative to the object. Provided this is the case, we can tune the parameters for optimal accuracy in phase detection as well as a frequency to C_Z conversion function with a favourable condition number. It is possible, even desirable, to choose the individual frequencies high enough to not be directly observable at the resolution of the camera used.

Note that changing $\|\vec{f}_i\|$ requires a differently printed glass sheet whereas changing ρ can be achieved easily with a different image on the display. The parameter h can be varied with a different set of supports for the glass sheet.

7.1 Extracting Moiré Parameters from Image of Calibration Object

To analyse an image of the described object, we first split the image into three separate grey scale images from the red, green and blue colour components. An example is shown in figure 7.1.

Some crosstalk between channels have been observed during analysis. This may be a result of an imperfect Bayer filter, the demosaicing process or possible colour mixing as a result of a software process such as colour space conversion or compression.

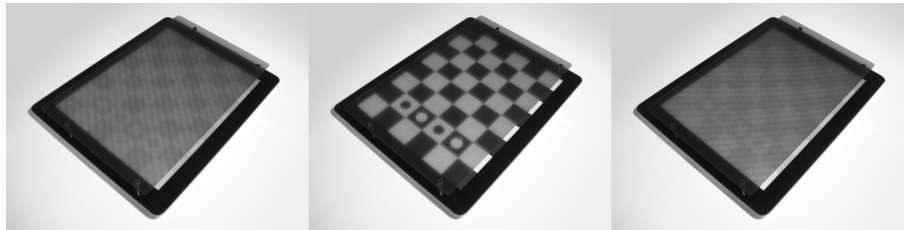


Figure 7.3: Red, green and blue channels (left to right) of an image of a physical implementation of the moiré-based calibration object. Although some crosstalk between channels can be observed, the moiré patterns are visible in the red and blue channels.

While it has been found that the moiré patterns are difficult to observe with the naked eye, inspection of the colour separated images makes these clearly visible in the red and blue channels if an appropriate exposure value was used during imaging.

The frame of the tablet used was black in our case. To ensure reliable detection of chequerboard corners by OpenCV, we invert the image of the green channel to create an image of a chequerboard with a white border. This arrangement is recommended in the OpenCV documentation [10] and was found to give

the most reliable results. The image is passed to `findChessboardCorners` and upon success we obtain image coordinates of the inner corners. An example of such an image and the detected corners is shown in figure 7.4.

Because of the symmetry of the chequerboard pattern we have an ambiguity in the correspondences of world coordinates of the squares to the detected points. We may need to reverse the order to ensure that the first point detected is at the top left of the tablet when held in an upright position. We determine if the points have been detected in an upside-down arrangement by detecting the presence of the alternately coloured disks which we know are positioned at the bottom of the tablet. If the disks appear at the top then we reverse the order of our array of image points.

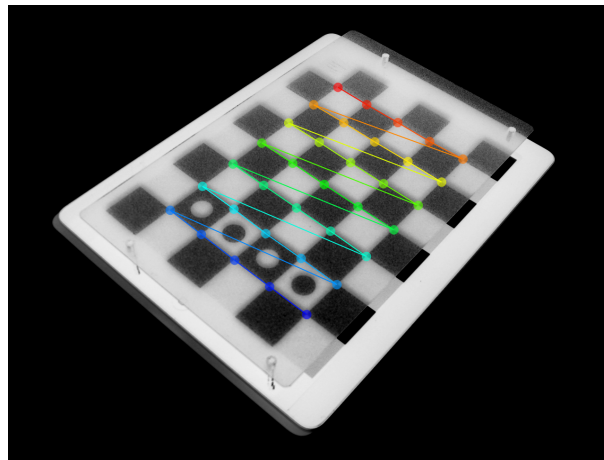


Figure 7.4: Inverted green channel from calibration object image as is passed as input to `findChessboardCorners`. The coordinates and corrected ordering (from red to blue) of the detected chequerboard corners are shown.

Given the set of corrected world to image point correspondences, we now wish to resample the images of the red and blue channels in order to remove perspective distortion for moiré analysis using the technique described in chapter 3. If we assume that no lens distortion is present then we use the entire set of correspondences to find the appropriate projective transformation from the overdetermined system for this purpose. If we assume for there to be some lens distortion present then we will opt to calculate a projective transformation for each square using its four corner points. This yields a piecewise linear solution. For the squares at the edges we extrapolate using the homography calculated for the nearest squares for which coordinates on all four corners are available.

Once a mapping has been found between world and image coordinates, we resample the images of the red and blue channels. We choose the middle square region of the tablet display and resample using a square grid of $n \times n$ samples $z_{0,0}, \dots, z_{n-1,n-1}$, where $n = 2^m$ for some integer m . By choosing a power of 2

for the grid size we ensure that we can make use of the efficient FFT algorithm for moiré analysis. We use simple bilinear interpolation to calculate pixel values from the sample points on the input images with perspective distortion removed, as shown in figure 7.5. At this point note is taken of the sample spacing and we choose the central sample $z_{n/2,n/2}$ to correspond with the origin of the world axis system.

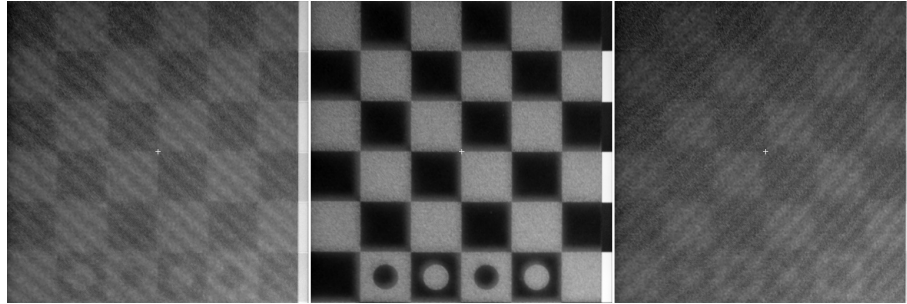


Figure 7.5: Resampled red, green and blue channels from calibration object image for removal of perspective distortion. The moiré patterns in the red and blue channels are analysed.

The FFT is now taken of the sample grid, first along the rows and then the columns to give a 2D FFT output. Hereafter we create an amplitude spectrum from the magnitudes of the complex output values and search for the maximum as per the technique in chapter 4. Given knowledge of the general position of the camera relative to the object, the search may only need to be conducted in a region of feasibly expected frequency values. Furthermore, given the construction of our object, we expect these maximum values to occur only at or near $\pm 45^\circ$ lines projected out from the zero frequency coefficient.

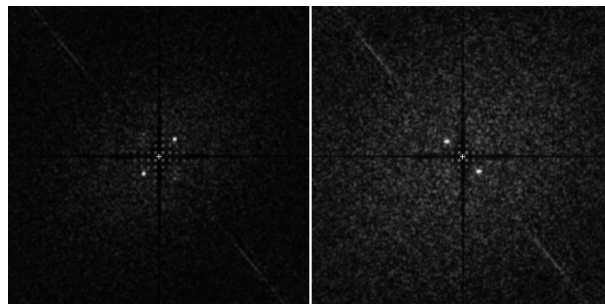


Figure 7.6: Frequency domain amplitude spectrum of perspective-corrected red and blue channels of the calibration object image, after some filtering has been applied to suppress the zero frequency component. The dual peaks associated with the moiré patterns in each case can be observed in the $\pm 45^\circ$ directions respectively.

Finally we use the quadratic interpolation method outlined in chapter 4 to calculate the detected frequency f_{Δ} and the phase of the sinusoidal function, shifted to be relative to sample $z_{n/2, n/2}$ which corresponds to the world origin.

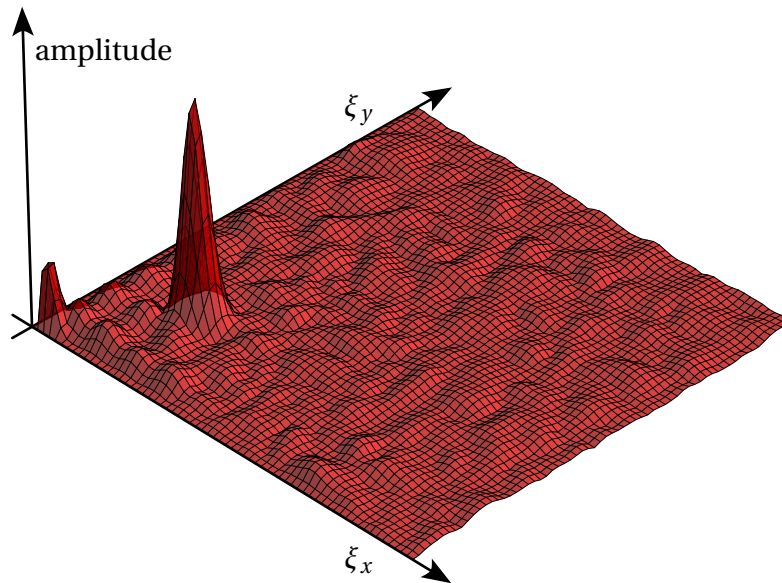


Figure 7.7: 3D visualisation of frequency domain amplitude spectrum of perspective-corrected red channel from calibration object image. Only the first quadrant (positive ξ_x and ξ_y) is shown. Some filtering has been applied to suppress the zero frequency component.

Applying the steps above on various images of a physical implementation of the calibration object, it was found that `findChessboardCorners` was the most likely step to fail due to noise or poor visibility in the inverted green channel of the image. The glass printing process used was found to be responsible for a slight blurring effect on the display below, which made corner detection less reliable. Beyond this step, peak detection in the frequency analysis step was a second vulnerability. Reasons for this step failing include:

- over- or underexposed moiré pattern in the image, possibly due to the automatic exposure in the camera used;
- poor colour channel separation (the chequerboard pattern remains visible in the red and blue channels);
- poorly constructed calibration object causing uneven moiré patterns and thus not a strong, single frequency signal. An example of this is when the

glass sheet is not perfectly parallel to the display. This effect is slightly visible in the blue channel moiré pattern of the image used in this chapter.

To avoid some of the above challenges, further research discussed in this thesis was done on simulations of the calibration object. Given a high quality printing process and a means of ensuring perfect alignment in the calibration object, further work can be done to analyse physical implementations for potential use in certain camera calibration scenarios. Constraints on time and budget prevented this research from being completed as part of the thesis.

Chapter 8

Analysis of Moiré-based Calibration Object

This chapter discusses a series of experiments that were performed in simulation. Based on knowledge of the chosen object parameters, we used a combination of the equations derived in chapter 6 and conventional camera calibration techniques from OpenCV to estimate a subset of the camera parameters.

Firstly, without the need for any prior camera knowledge, we can estimate the Z -coordinate of the camera from moiré frequency analysis. We show in this chapter how, by choosing the object parameters appropriately, we have the option of improved accuracy over conventional methods at the cost of a narrower usable span in C_Z .

Secondly, we show how accuracy in camera X - and Y -coordinates can be improved by the addition of moiré phase analysis, assuming that an estimate of sufficient accuracy is available for the complete set of camera parameters. This may include lens distortion.

Finally, we briefly discuss some experiments that have been performed with limited success to solve all pinhole model camera parameters from a single image of the proposed calibration object. The assumptions are made that there is no skew, that the effective focal length parameters are equal and that there is no nonlinear lens distortion present.

Conclusions made about improved accuracy over conventional methods are made based on ray-traced simulations of the proposed calibration object, generated with various camera and object parameters at 1280×720 resolution. The ray-trace was performed with 8 times oversampling so that 64 values were averaged per pixel. The display part of the object was simulated with colour depth spanning between 5% and 95% brightness, appearing darker in the final image due to partial masking when imaged through the printed glass sheet. The floating point red, green and blue channels of each image were scaled to the range $[0, 256)$, whereafter random Gaussian noise with a standard deviation of 2 was added to all pixels on all colour channels. The values were then truncated down

to the nearest integer, clipped to the range from 0 to 255 and together saved as 24-bit colour images. The pixel values were encoded with a gamma value of 1.0, which means linear encoding.

As a benchmark case for each image, a ray-traced simulation of an opaque chequerboard of the same size as the calibration object was generated in the same orientation and position relative to the camera as the base layer in the moiré object, with the black and white squares represented as 5% and 95% brightness respectively. Hereafter the same scaling, addition of Gaussian noise, truncation, clipping and saving to 24-bit images were applied as in the case of the proposed object.

A pair of sample images of the proposed object and the benchmark chequerboard is shown in figure 8.1. In both cases the accuracy of the camera position estimation is compared against the exact parameters used in the ray-traced rendering.

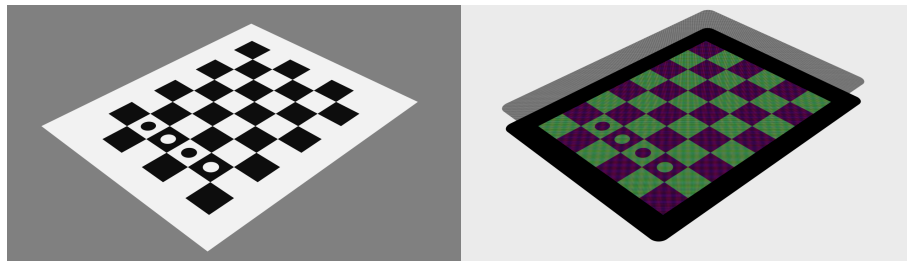


Figure 8.1: Sample images of ray-traced objects used for research.

For the benchmark case the OpenCV routine `findChessboardCorners` was used to detect the inner corners of the chequerboard. The detected image coordinates, along with their known world coordinates and the correct intrinsic camera parameters used during ray tracing (including no lens distortion) was then passed to `solvePnP` to calculate the camera position. These positions are the benchmarks against which the proposed method is compared.

For the simulated size of the calibration object with a square analysable region with a side length of 14.8 cm it was found that an f_{Δ} of 200 cycles per metre can be detected reliably. Below 80 cycles per metre any constant frequency components present in the analysed region begin to interfere with the analysis. Above 500 cycles per metre there is the risk that the Nyquist sampling criterion might not be met at the 1280×720 resolution of the images, especially when the object is viewed at large angles with respect to the plane normal, causing the pattern to occupy less area in the image.

Because of the relationship between f_{Δ} and C_Z , these limitations on f_{Δ} place an inherent limit on the range of C_Z for which viable analysis can be done. For this experiment we thus limited the simulations to the range of C_Z roughly associated with the bounds on f_{Δ} outlined above. The camera position \vec{C} was further

limited to angles less than 60° with the respect to the plane normal through the origin of the world axis system.

Images were generated simulating a camera with random orientation, provided that all four outer corners of the calibration object mapped to points in the image bounds and that the total area enclosed by a polygon through the image coordinates of these points occupied at least 15% of the image area.

8.1 Camera Z -coordinate

Given \vec{f}_t , $\rho = \|\vec{f}_b\|/\|\vec{f}_t\|$ and h , we can calculate the Z -coordinate of the camera directly from f_Δ using equation 6.5 from chapter 6. The frequency of the moiré $f_\Delta = \|\vec{f}_\Delta\|$ as calculated from either the red or the blue channel analysis is used:

$$C_Z = \frac{h}{1 - \rho \pm \frac{f_\Delta}{f_t}}.$$

The accuracy with which one can expect f_Δ to be estimated is at most as good as the accuracy with which the reference markers, that is the chequerboard corners on the green layer, can be detected. Any errors introduced in the corner detection step would bring about an error in translation, scale or rotation during the mapping step which would alter the observed value for f_Δ . It may also alter the mapping itself, causing perspective distortion removal to be inexact and thus causing the observed moiré pattern to not be a constant frequency sinusoidal function over the analysed region. Any noise, reflections or nonlinearities present in the red or blue channels of the analysed region would introduce additional error to the extracted estimate of f_Δ .

Given these potential errors, we could expect the accuracy of C_Z determined from equation 6.5 to be of worse accuracy than that of the benchmark C_Z . Figure 8.2 shows an illustration of this, with the proposed method showing on average 1.29 times the error over 50 random simulations from the C_Z range 0.2 m to 1.5 m.

Because we are free to choose the parameters of our calibration object, we may choose an arbitrary condition number for equation 6.4. A large negative condition number implies that a small relative change in C_Z translates to a large relative change in f_Δ . Conversely, any relative error in calculation of f_Δ translates to a much smaller relative error in the calculation on C_Z . Using equation 6.6, we evaluated the effect of choosing κ_{f_Δ} as -10 .

The cost of this change is that we are left with a much narrower range of C_Z for which f_Δ is feasibly analysable. From the value of κ_{f_Δ} and the resulting value of ρ we can calculate the C_Z range associated with a predicted f_Δ between 80 and 500 cycles per meter as between 0.53 m and 0.44 m, a mere 9 cm region compared to the previous case of over a metre.

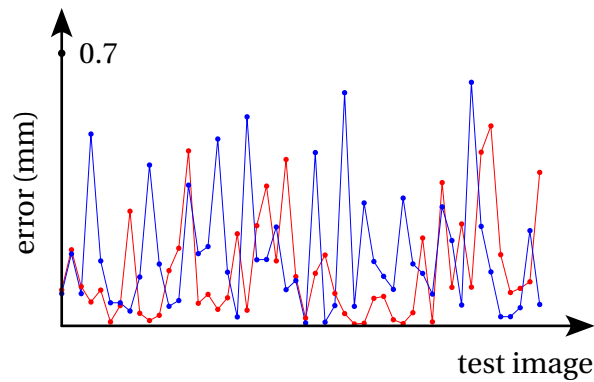


Figure 8.2: Absolute errors of C_Z of proposed method (blue) vs. conventional method (red) with object parameters $\rho = 1$ ($\kappa = -1$), $h = 0.04$ and $f_t = 2500$. Over this set the proposed method gives an average absolute error 1.29 times larger.

However, the benefit of choosing κ_{f_Δ} as -10 is shown in figure 8.3. Taken over 50 random simulations, we have an average error 8.39 times smaller than that of the conventional method.

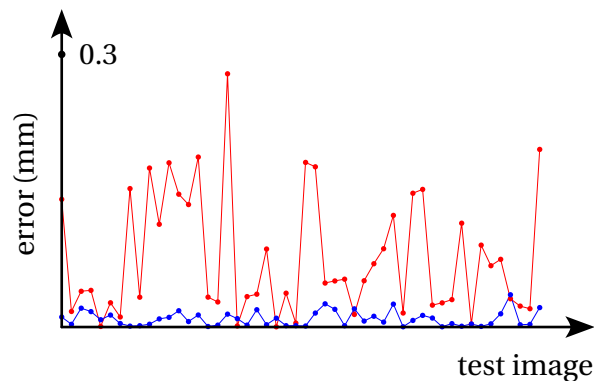


Figure 8.3: Absolute errors in C_Z of proposed method (blue) vs conventional method (red) with object parameters $\rho = 0.82$ ($\kappa = -10$), $h = 0.1$ and $f_t = 10000$. Over this set the proposed method gives an average absolute error 8.39 times smaller.

A dual benefit is offered here in the calculation of C_Z over the conventional method. Firstly we did not have to make any assumption about the camera. We did not need to know any of the intrinsic camera parameters to estimate C_Z . Secondly, within a limited range we have the option to tune the parameters of our object to ensure that camera position can be determined with much better accuracy.

The proposed sequence to go about when choosing the object parameters is to first choose a target C_Z and κ_{f_Δ} based on the region of interest and the required accuracy. We also choose the largest value for h for which we can still easily construct the object. We then calculate the appropriate ρ using equation 6.7. Next we choose f_t to ensure that for C_Z in the middle of our target span we get an ideal f_Δ . This is done by rewriting equation 6.4:

$$f_t = \frac{f_\Delta}{\rho - 1 + \frac{h}{C_Z}}.$$

Finally we calculate the bounds of the feasible span of C_Z for which analysis can be done by inputting the bounds of feasible f_Δ in equation 6.5.

The main drawback of the proposed method is that when not exercising the option of choosing a value for κ_{f_Δ} other than -1 , the accuracy is generally worse than with the conventional method. When making use of a large negative κ_{f_Δ} for improved C_Z accuracy, the usable span is limited for fixed object parameters. When choosing object parameters for an ideal placement of this limited range, and this region is far from the calibration object then either h or $\|\vec{f}_b\|$ must be large, which may make it difficult or expensive to construct the object accurately.

A major advantage of the proposed method is that, provided the assumed object parameters are accurate, a favourable condition number can suppress errors regardless of where they arise. Larger errors can be tolerated, not only in the perspective removal and moiré analysis steps, but also those steps that are shared with the benchmark method such as the corner detection.

8.2 Camera X - and Y -coordinate

With the addition of knowledge of the intrinsic camera parameters, and using phase analysis as discussed in chapter 6, we can also attempt to find improved estimates for the X and Y components of \vec{C} .

As is done for the benchmark method, we can easily calculate an initial estimate for \vec{C} using `solvePnP` in OpenCV. If a sufficient condition number is chosen then we can expect our estimate of C_Z , calculated using the frequency analysis method discussed above, to be better than the conventionally calculated coordinate. We therefore use this improved C_Z estimate.

We now calculate two additional coordinates for \vec{C} in the directions of \vec{f}_t in the red and blue directions respectively:

$$\begin{aligned} C_U^* &= \frac{C_X^* + C_Y^*}{\sqrt{2}}, \\ C_V^* &= \frac{-C_X^* + C_Y^*}{\sqrt{2}}. \end{aligned}$$

C_U^* and C_V^* can now play the role of C_R^* in equation 6.9, in the red and blue channels respectively in order to calculate for each channel an appropriate n to be used in equation 6.8. Provided that C_U^* and C_V^* as indirectly calculated from the conventional method are accurate enough to satisfy the condition in equation 6.10, we may be able to improve the accuracy and calculate new C_X and C_Y estimates:

$$C_X = \frac{C_U - C_V}{\sqrt{2}},$$

$$C_Y = \frac{C_U + C_V}{\sqrt{2}}.$$

Experiments were performed using a κ_{f_Δ} of -10 , but the large values for h and f_t that were required meant that condition 6.10 was not reliably met; it required an estimate for C_X and C_Y using conventional methods that had too strict an accuracy requirement. Using a κ_{f_Δ} of -4 , some favourable results were obtained as shown in figure 8.4.

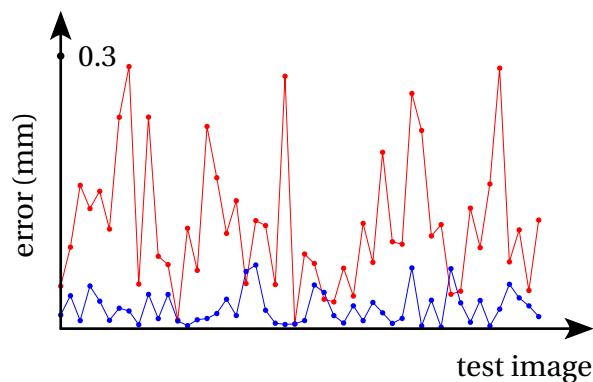


Figure 8.4: Distance between (C_X, C_Y) and estimations from proposed method (blue) vs conventional method (red) with object parameters $\rho = 0.85$ ($\kappa = -4$), $h = 0.1$ and $f_t = 4000$. Over this set the proposed method gave an average distance error 5.05 times smaller.

It is important to note here that the improvement in accuracy of C_X and C_Y would not have been possible without the improvement in accuracy of C_Z since, by equation 6.8, the values C_U and C_V were calculated proportionally to the assumed correct C_Z . Any error in C_Z translates directly to a proportional error in C_R (substituted for C_U and C_V) and hence C_X and C_Y . This improvement would thus not be possible without the use of a condition number with magnitude larger than 1, or by obtaining a more accurate estimate for C_Z by some other means.

If the estimate for the intrinsic parameters used to create an initial estimate for \vec{C} contained some error, it would only affect whether or not the correct n can be obtained. Provided that the C_Z from the frequency analysis method is used and that the correct n can be calculated, the improved estimate for \vec{C} would be further unaffected by any erroneous intrinsic parameter assumptions.

If the technique is applied to a series of sequential images, such as from a video of the object taken with a moving camera, an alternative way in which n could be resolved would be to use an initial value and keep track of the phase with time. Provided that the movement between successive images remains below $\frac{1}{2}C_Z\frac{\lambda}{h}$ across the sequence, the resolved value for C_R from any frame can be used as C_R^* , i.e. a sufficient estimate for the next frame. The condition in equation 6.10 would be satisfied.

This method of keeping track of n in video might be especially useful in augmented reality applications. Even if an incorrect initial value was used for n , this merely means that the estimated C_R would have a fixed offset error. Subtle relative movements along the R -axis would still be captured by the phase analysis, which is useful for superimposing virtual objects that are close to the camera that would require this extra accuracy for a realistic parallax effect.

8.3 Pinhole Model Parameters

Experiments were performed to analyse whether, assuming that no nonlinear lens distortion is present, a complete set of pinhole camera model parameters could be extracted from an image of the calibration object.

As noted in chapter 2, the camera matrix \mathbf{P} in equation 2.3 cannot be solved from a set of world-image correspondences that all lie in the same plane alone. Writing the correspondences as a set of linear equations, we obtain a homogeneous system

$$\mathbf{H}\vec{p} = \vec{0},$$

where \vec{p} contains the entries of \mathbf{P} rewritten as a column vector.

Simulated experiments were performed using the inner chequerboard corners of the base plane of the proposed calibration object as point correspondences with various object and camera arrangements. Decomposing the matrix \mathbf{H} constructed from these point sets using the SVD, results consistently showed \mathbf{H} to have a nullity of 4 for these sets of planar points. In order to find a basis for \mathbf{P} , a single basis vector needs to be found. Assuming without proof that this nullity of 4 can be relied upon, we therefore assume that in general four linearly independent basis vectors can be found for the null space of \mathbf{H} . Since the correct single basis vector lies in this null space it can be expressed as a linear combination of these four basis vectors:

$$\vec{p} = a\vec{p}_1 + b\vec{p}_2 + c\vec{p}_3 + d\vec{p}_4.$$

In order to find coefficients a , b , c and d , additional information is required. We showed in chapter 7 that C_Z can be calculated from frequency analysis without any prior knowledge of camera parameters. Also assuming no skew parameter (the second entry in the top row of \mathbf{K} in equation 2.2 is zero) and square pixels (the first and second diagonal entries of \mathbf{P} are equal), we should in theory have eliminated three degrees of freedom. Because we only need to find a basis for \mathbf{P} , we can finally assume that $\|\vec{p}\| = 1$. We are therefore looking for values a , b , c and d that meet all these requirements.

Various tests were performed using the Nelder-Mead minimisation technique to determine if these values could be solved and a complete set of pinhole model parameters calculated. Experiments were done with various objective functions to be minimised, based on the sums of measures of error in C_Z , skewness and inequality of focal length parameters while ensuring that coefficients a , b , c and d are scaled together so that $\|\vec{p}\| = 1$. A decomposition of \mathbf{P} is calculated at each step.

Simulations were run with various sets of camera parameters and convergence to the correct solution was frequently, but inconsistently observed. If the technique proves useful, more research may be required to analyse precise conditions that lead to non-convergence. For this particular solution, it may be that no benefit may be offered beyond a conventional calibration object that uses point correspondences in a non-planar (and hence non-degenerate) configuration.

Chapter 9

Conclusions and Future Research

As part of the work for this thesis we started with a study of existing techniques for camera calibration and the methods used for (amongst other parameters) camera position estimation. Focus was placed on the most widely used techniques such as those adopted by OpenCV. Moiré patterns that result from 3D arrangements of periodic patterns have been found to produce rapid apparent changes for small changes in vantage point. We attempted to construct a calibration object that makes use of this phenomenon with the goal of achieving high accuracy camera position estimation. Various configurations were explored.

We then formalised the mathematics that describe the phenomenon in order to analyse images of such an object and perform simulations. Results were compared against one of the most widely used methods of camera calibration.

9.1 Conclusions

We showed in simulation how moiré frequency and phase analysis on an image of the proposed calibration object allows for significant improvement in the accuracy of camera position estimation over a benchmark method based on the industry standard algorithms in OpenCV.

The novel construction of the proposed calibration object provides for a tunable condition number in the final calculation of C_Z from frequency analysis. This allows for a trade-off between C_Z accuracy and the width of the C_Z range at which the analysis can be performed. A favourable condition number allows us to mitigate errors from earlier steps of the analysis process, potentially leading to more robustness when dealing with poor image quality due to high noise or low resolution. This holds true provided a spectral peak can be successfully detected for frequency analysis.

Moiré speedup on the object allows for accurate phase analysis. We showed how this can be combined with an accurate C_Z estimate and rough initial estimates for C_X and C_Y to obtain more accurate C_X and C_Y estimates. Bounds were

calculated for the errors that are allowed on the initial estimates for the technique to be successful.

Even though the proposed calibration object can be constructed with relative ease at a moderate budget, limited resources during work on the thesis meant that we could not successfully estimate camera position directly from a physical implementation. The printing process used in an attempt to create a high frequency pattern on the glass caused a blurring effect which made the detection of guides in the base layer unreliable. The implementation was sufficient, however, to confirm by visual inspection that moiré patterns behave in a physical implementation as simulated. It would be worth exploring the technique with a better printing process.

Another reason why testing was preferred on simulations of the calibration object rather than the physical implementation was that object parameters could be finely controlled. While the technique is very tolerant of errors in image analysis, the object parameters have to be known precisely. Because object construction is not exact, it is therefore required that a technique be developed for calibration of the moiré object itself as discussed in the following section.

Compared to the conventional methods of camera position estimation, the proposed method is more sensitive to under- or overexposure. This is because of the need for frequency analysis rather than simple edge or feature detection. Features may still be locatable if the image is under- or overexposed, but subtle periodic patterns quickly suffer loss of fidelity.

The need for a glass sheet means that unwanted reflections can appear on the object. Unlike patterns that can be printed on matte paper, anti-reflective coatings on the glass would be the best measure that can be taken.

We finally conclude that the significant improvements in the accuracy of camera position estimation that the method can offer would likely be of some use in practical situations. The technique seems best suited for controlled environments where positional knowledge of the camera is of the highest importance. For most standard applications, requiring only moderate accuracy in camera position, existing techniques such as those in OpenCV are probably easier to set up and offers sufficient accuracy.

9.2 Imperfect Object

In the simulations of chapter 8 we assumed that the construction of the proposed calibration object was flawless. In practice this is not a realistic assumption.

Because of the sensitivity that extracted values have to some of the object parameters, it is important that these parameters be known exactly. It may also be necessary to modify the object model to handle certain construction offsets. Such imperfections may include:

- the true value of $\|\vec{f}_b\|$ is not as intended;

- the directions of \vec{f}_b in the red and blue channels are not perpendicular;
- the true value of h is not as intended;
- the glass sheet has been placed with non-zero rotation about the Z -axis;
- the cosine pattern on the glass sheet has an unknown phase delay relative to the origin due to inexact placement;
- the glass sheet is not parallel with the display;
- the exact value of $\|\vec{f}_t\|$ is not as intended;
- the directions of \vec{f}_t corresponding to the red and blue channels are not perpendicular;
- higher order errors exist, such as bend in the glass sheet or refraction inside the glass sheet or display that is not negligible.

Note that the scenarios with non-perpendicular frequency directions could arise because of a printing process or display manufacturing process where the aspect ratio is not exactly 1.0.

Assuming a correct display resolution (pixels per inch), and assuming an aspect ratio in the display of 1.0, the unknown magnitude and directions of \vec{f}_b can be solved by simply choosing the pixel array as the definition of scale along the world axes. We already chose our axis system to lie within this plane and be aligned with the pixel grid. We could now simply choose display pixels as our world units in order to solve the first two listed items.

Calculating the rotation of the glass sheet about the Z -axis, phase relative to world origin and the exact values of \vec{f}_t and h may be possible again with moiré analysis. Even if these object parameters are not exactly as intended, the more important goal is to know the exact values for use in the model. It would be worth investigating whether some of the techniques used to calculate a high accuracy camera position could be used to determine one or more of these values with high accuracy. This may include leveraging moiré speedup or condition number adjustment by choosing \vec{f}_b (or ρ) optimally. By choosing the display pixels as the world units, this single value may be changed without consequence, first for optimal object calibration and thereafter for optimal camera calibration as may be required.

Finally, some higher order errors may require that the model be expanded upon for some of the imperfections that are not covered by the currently used equations.

9.3 Alternative Configurations of the Proposed Calibration Object

The use of a tunable condition number and moiré speedup may be used in different ways than have thus far been described. We list a few alternative configurations in the following paragraphs.

First we note that we limited our object to simultaneously show two moiré patterns by making use of colour multiplexing. If we find alternative guides to those in the green channel then we may use a third moiré pattern here, and beyond this there is an option to apply frequency multiplexing. By adding more patterns of different frequencies and/or directions we may come up with alternative ways of resolving the ambiguity in the phase to distance conversion. The drawback of adding more patterns is that the amplitude per pattern decreases as more patterns need to be accommodated within the same range of values. The lower signal-to-noise ratio that results leads to less accurate frequency and phase estimation per pattern.

The information derived from resolved phase allows a ratio C_R/C_Z to be estimated as per equation 6.8. Two such ratios measured for C_R projections along perpendicular axes allow a direction to be estimated for the line connecting the camera centre and the origin. Previously we have combined this with an estimate of C_Z to estimate C_X and C_Y . An alternative may be to use two of the proposed calibration objects and instead attempt to triangulate the position of the camera.

One of the drawbacks of adjusting the condition number of the function to calculate C_Z from f_Δ is that higher accuracy limits the span of C_Z for which moiré patterns of analysable frequencies emerge. It may be worth investigating the feasibility of constructing a system where moiré analysis is done in real-time and \vec{f}_b (or ρ) as is displayed on the tablet is adjusted based on feedback from a smart camera in order to ensure that the camera remains within such a region. This will require that information about the chosen \vec{f}_b be transferred back to the camera in real-time as well, possibly encoded on the display itself.

Bibliography

- [1] Abdel-Aziz, Y.I. and Karara, H.M., “Direct Linear Transformation into Object Space Coordinates in Close-Range Photogrammetry.” *Proceedings of the Symposium on Close-Range Photogrammetry*, pp. 1–18, 1971.
- [2] Abe, M. and Smith III, J. O. “Design Criteria for Simple Sinusoidal Parameter Estimation Based on Quadratic Interpolation of FFT Magnitude Peaks.” *Audio Engineering Society Convention*, pp. 6256–6268, 2004.
- [3] Bracewell, R.N. *The Fourier Transform and Its Applications*, McGraw-Hill, 1986.
- [4] Bradski, G. and Kaehler, A. *Learning OpenCV: Computer vision with the OpenCV library*, O’Reilly Media, Inc., 2008.
- [5] Datta, A., Kim, J.S. and Kanade, T. “Accurate Camera Calibration Using Iterative Refinement of Control Points.” *IEEE International Conference on Computer Vision Workshops*, pp. 1201–1208, 2009.
- [6] Gabrielyan, E. “The Basics of Line Moiré Patterns and Optical Speedup.” *arXiv preprint physics/0703098*, 2007.
- [7] Gao, X.S., Hou, X.R., Tang, J., and Cheng, H.F. “Complete Solution Classification for the Perspective-Three-Point Problem.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, 2003.
- [8] Hartley, R. and Zisserman, A. *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2nd edition, 2003.
- [9] Lepetit, V., Moreno-Noguer, F. and Fua, P. “EPnP: An Accurate $O(n)$ Solution to the PnP Problem.” *International Journal of Computer Vision*, 81(2):155–166, 2009.
- [10] “OpenCV API Reference.” Retrieved 29 October 2014, from <http://docs.opencv.org/modules/refman.html>

- [11] Tsai, R. Y. "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using off-the-Shelf TV Cameras and Lenses." *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987.
- [12] Zhang, Z. "Flexible Camera Calibration by Viewing a Plane from Unknown Orientations." *Proceedings of the IEEE International Conference on Computer Vision*, pp. 666–673, 1999.
- [13] Zhang, Z. "A Flexible New Technique for Camera Calibration." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.