

ENTER YOUR TITLE

by

ENTER YOUR NAME

Enter Your Previous Degrees

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

ENTER YOUR DEGREE NAME

Enter Your Department Name

Enter Your College Name

KANSAS STATE UNIVERSITY

Manhattan, Kansas

Graduation Year

Approved by:

Major Professor

Enter Your Major Professor's Name

Copyright

Enter Your Name

Graduation Year

Abstract

Enter the text of your abstract in the `abstract.tex` file. Be sure to delete the text below before you submit your ETDR.

This template uses a separate file for each section of your ETDR: title page, abstract, preface, chapters, reference, etc. This makes it easier to organize and work with a lengthy document. The template is configured with page margins required by the Graduate School and will automatically create a table of contents, lists of tables and figures, and PDF bookmarks.

The file `etdrtemplate.tex` is the "master" file for the ETDR template. This is the file you need to process with PDFLaTeX in order to produce a PDF version of your ETDR. See the comments in the `etdrtemplate.tex` and other files for details on using the template. You are not required to use the template, but it can save time and effort in making sure your ETDR meets the Graduate School formatting requirements.

Although the template gives you a foundation for creating your ETDR, you will need a working knowledge of LaTeX in order to produce a final document. You should be familiar with LaTeX commands for formatting text, equations, tables, and other elements you will need to include in your ETDR.

Table of Contents

| | |
|---|------|
| List of Figures | viii |
| List of Tables | ix |
| Acknowledgements | ix |
| Dedication | x |
| Preface | xi |
| 1 Background | 1 |
| 1.1 Similar Research | 1 |
| 1.2 Coordinate Systems | 1 |
| 1.2.1 Latitude and Longitude | 1 |
| 1.2.2 Universal Transverse Mercator (UTM) | 2 |
| 1.2.3 Field Coordinate System | 3 |
| 1.2.4 Platform Coordinate System | 4 |
| 1.2.5 Camera Coordinate System | 4 |
| 1.3 Coordinate Projections | 5 |
| 1.3.1 Projection Model | 6 |
| 1.3.2 Projection Methods | 7 |
| Bibliography | 15 |
| A Title for This Appendix | 16 |

| | |
|-------------------------------------|----|
| B Title for This Appendix | 17 |
|-------------------------------------|----|

List of Figures

| | | |
|-----|-------------------------------------|---|
| 1.1 | Latitude and longitude | 2 |
| 1.2 | UTM zones | 3 |
| 1.3 | Field coordinates | 3 |
| 1.4 | Platform coordinate frame | 4 |
| 1.5 | Camera coordinate frame | 5 |
| 1.6 | Projection model | 6 |
| 1.7 | Focal points | 7 |

List of Tables

Acknowledgments

Enter the text for your Acknowledgements page in the `acknowledge.tex` file. The Acknowledgements page is optional. If you wish to remove it, see the comments in the `et-drtemplate.tex` file.

Dedication

Enter the text for your Dedication page in the `dedication.tex` file. The Dedication page is optional. If you wish to remove it, see the comments in the `etdrtemplate.tex` file.

Preface

Enter the text for your Preface page in the `preface.tex` file. The Preface page is optional. If you wish to remove it, see the comments in the `etdrtemplate.tex` file.

Chapter 1

Background

address at transplanter? One-time shot if things mess up. Difficult to identify plants. Constant stopping, too much motion and difficult shading make its not ideal for images.

1.1 Similar Research

1.2 Coordinate Systems

A key step in any mapping process is the correct choice of coordinate frames. In the context of mapping a coordinate system defines the location of an item with respect to another item. These coordinate systems, or frames of reference, are typically fixed to items that are significant in the mapping process which include the Earth, field, platform and cameras. The specific coordinate systems used in this research are discussed in the following sections.

1.2.1 Latitude and Longitude

A common frame that is fixed to the Earth is a spherical coordinate frame, with the angles referred to as latitude and longitude. However, since the Earth is not a sphere these angles are projected onto an ellipsoid called a datum. The datum that is used in this research is the World Geodetic System (WGS-84) which is the default datum used by the Global

Positioning System (GPS) [TODO cite?]. The third dimension is altitude and is measured relative to the ellipsoid. Coordinates in the frame are denoted (Φ, λ, A) .

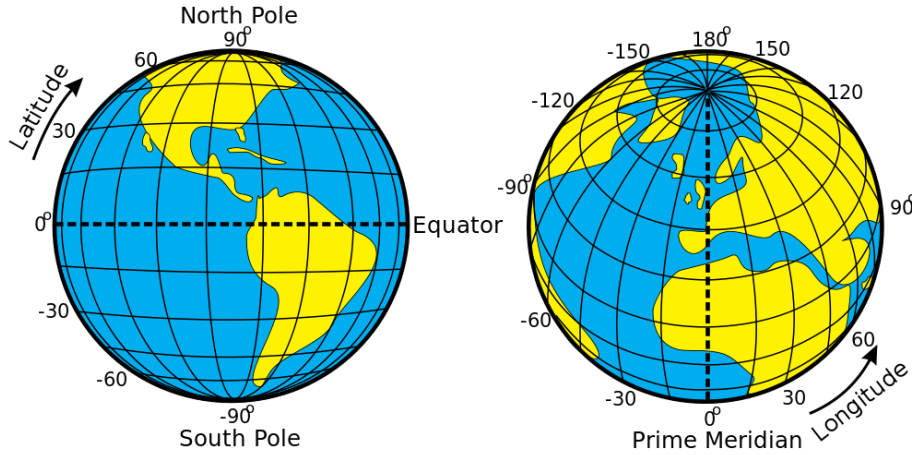


Figure 1.1: *Illustration of Latitude (Φ) and Longitude (λ). Obtained under the Creative Commons License from the Wikimedia Commons.*

1.2.2 Universal Transverse Mercator (UTM)

In order to describe the points on the Earth's surface in a two-dimension plane, as is desired for mapping, a projection model must be used on the angular latitude and longitude coordinates. The projection model used in this research is known as the Universal Transverse Mercator (UTM) which splits the Earth into 60 lateral bands, examples of which are shown in figure 1.2. In each zone an item is described by its easting, northing and altitude coordinates. However, due to how the platform coordinate system is defined it's beneficial to have the Z axis in the downward direction. This requires the easting and northing be switched to maintain a right-handed coordinate system. Therefore, UTM coordinates are described by (N, E, D) . Another important modification is that the Z, or down, component is measured relative to the ground, not the WGS-84 ellipsoid.

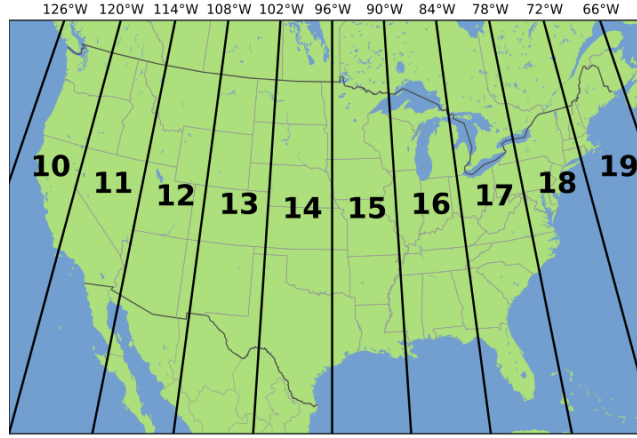


Figure 1.2: Visualization of UTM zones over the United States. Obtained under the Creative Commons License from the Wikimedia Commons.

1.2.3 Field Coordinate System

One issue with the easting and northing described by UTM is that rows in a field are not always planted north and south or east and west. Many of the post-processing steps benefit from removing this relative field orientation. A new field coordinate system is defined where the y-axis runs parallel to the rows and increases in the planting direction of the first row, and the x-axis increases in the direction of increasing row numbers. The origin is selected so that all items in the field have positive x and y coordinates. Similar to UTM the units of this frame are meters. Coordinates in this frame are denoted (x_f, y_f, z_f) .

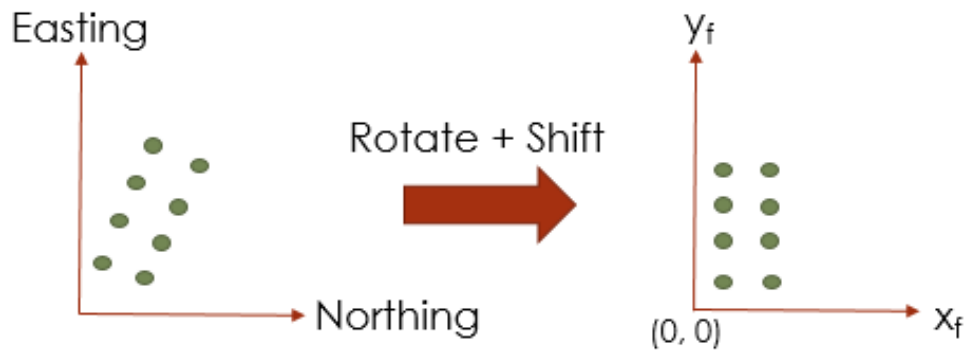


Figure 1.3: Conversion from modified UTM coordinate frame (left) to field coordinate frame (right)

1.2.4 Platform Coordinate System

Another useful coordinate system is one fixed to the platform holding the cameras. This coordinate system allows the relative spacing and orientation between the cameras and the GNSS [Todo already defined?] antenna to be specified. Also this coordinate system defines how the platform's orientation is defined in terms of Euler angles, which is useful for accounting for non-level camera orientation. The axes of this coordinate system are shown in the figure 1.5, which defines the x-axis out of the front of the platform, the y-axis out the right hand side and the z-axis orthogonal in the downward direction.

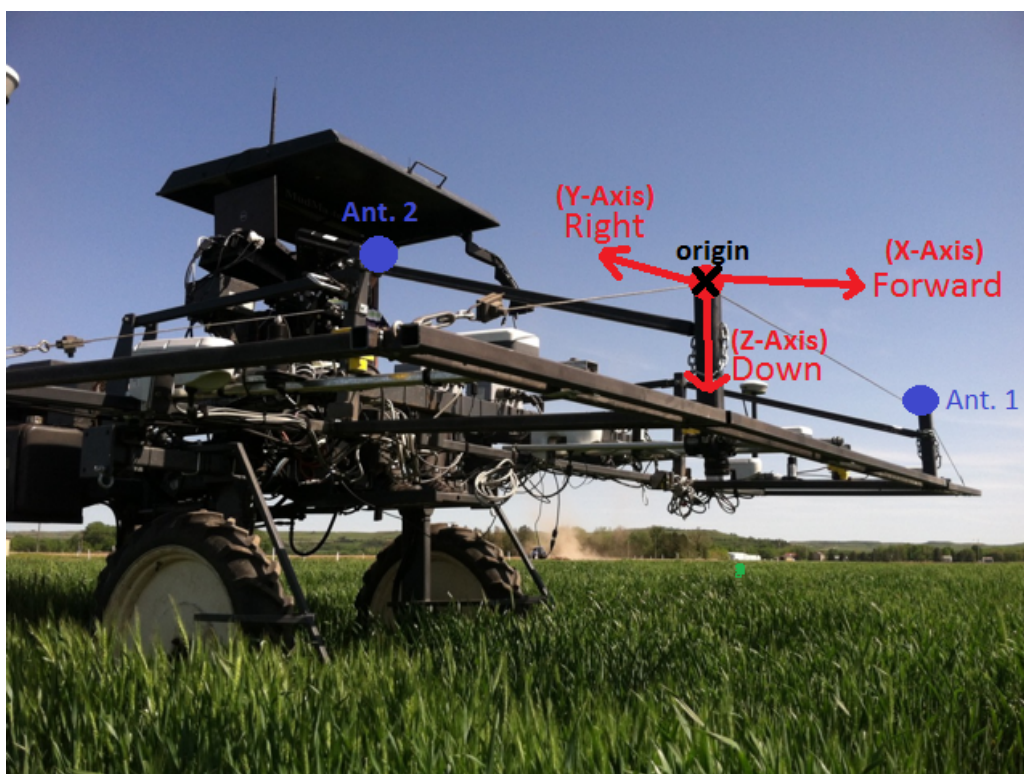


Figure 1.4: *Example of platform coordinate frame on a tractor systems with two GNSS antennas shown in blue. The origin is the average of these antenna locations and is shown as a black X.*

1.2.5 Camera Coordinate System

The camera coordinate system describes the locations of objects in the world relative to the camera. Typically a camera coordinate system is defined by the x-axis out of the right

hand side of the camera, the y-axis out of the bottom and the z-axis along the optical axis. [TODO cite] However, for this application an alternative camera coordinate system is defined which has the x-axis out of the top of camera, the y-axis out of the right side and the z-axis along the optical axis. This is so when the camera is mounted on the platform facing the ground with the top forward, the camera axes will align with the forward-right-down platform coordinate system. This makes the meaning of the Euler angles consistent, and is also enforced by the data collection program. The origin is defined to be located at a distance f in front of the imaging sensor along the optical axis, where f is the focal length of the lens. This is discussed more in the next section.



Figure 1.5: Image showing camera coordinate axes.

1.3 Coordinate Projections

An important step in the mapping process is converting between pixel and world coordinates, which is also the subject of many computer vision algorithms. Since pixels are described in \mathbb{R}^2 and world points in \mathbb{R}^3 , this conversion requires a projection.

However, before this can be defined a model must be chosen which describes how points in the world are projected onto the imaging sensor. This section first presents the model used in this research and then discusses two equivalent methods that can be used for converting coordinates.

1.3.1 Projection Model

A common choice for a thin, convex lens is the central perspective imaging model shown in figure 1.6.

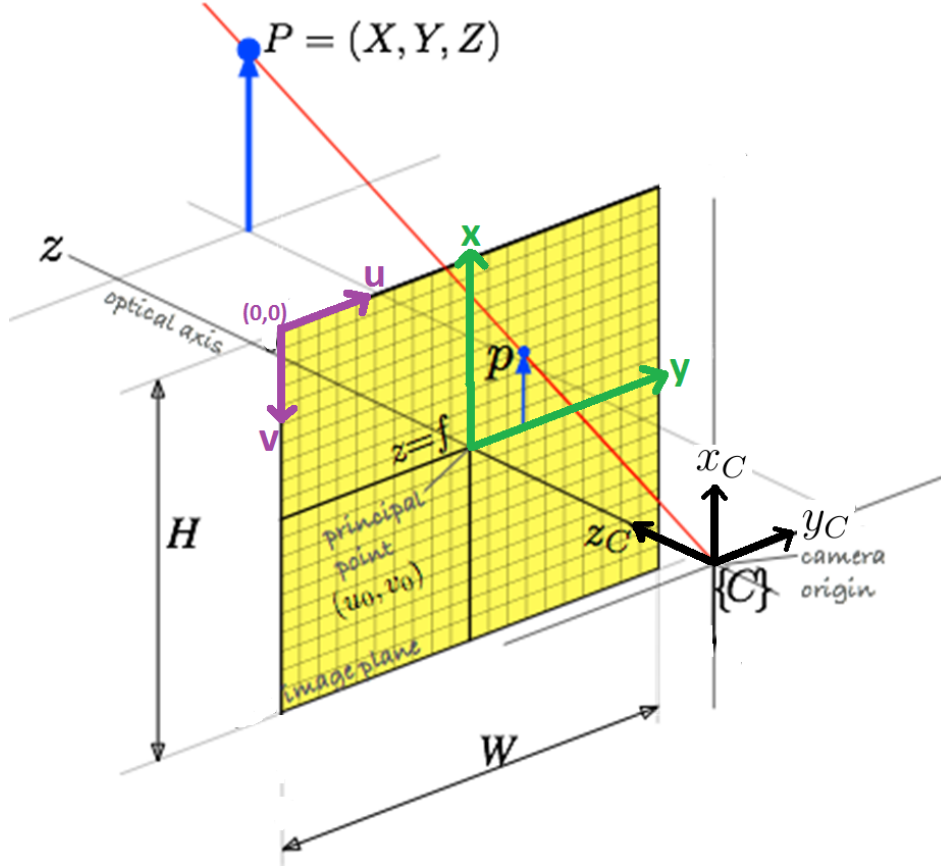


Figure 1.6: Central perspective imaging model showing the image plane in yellow. Adapted from "Robotics, Vision and Control" 1st edition, by Peter Coorke, 2013, p.252. Modified to show pixel frame in purple, the sensor frame in green and switched the ordering of the camera axes.

This model is based on the concept of an ideal lens that is treated as an equivalent pin-hole, which implies the lens is perfectly focused and has no geometrical distortions. The ideal lens will have two focal points along the optical axis which can be seen in figure 1.7. The central perspective model defines the image plane to be at the focal point in front of the lens, which is the left focal point shown in the figure.

Points in the image plane can be described using two different coordinate frames. The first is the sensor frame and the second is the pixel frame. The origin of the sensor frame

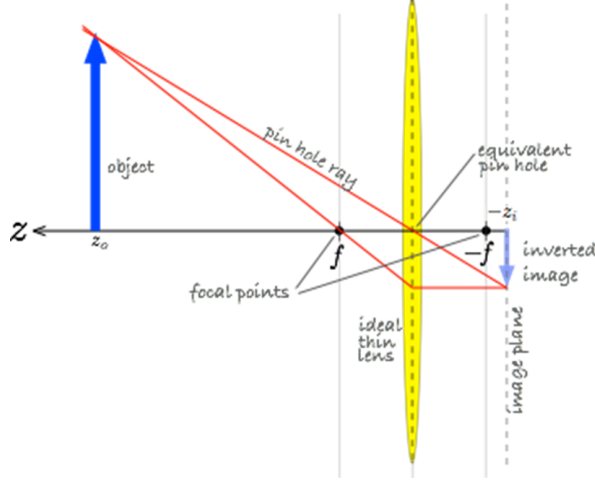


Figure 1.7: *Equivalent pin hole lens showing both focal points. Adapted from "Robotics, Vision and Control" 1st edition, by Peter Coorke, 2013, p.252*

is located at the point where the optical axis intersects the image plane. A point in this frame is described by (x, y) , where the axes are in the same direction as camera frame and are shown in blue in figure 1.6.

On the other hand, the origin of the pixel frame is located at the top left of the image plane and the x-axis increases to the right and the y-axis increases downwards. Rather than (x, y) a pixel's location is denoted as (u, v) . The center of the image sensor is referred to as the principal point and is denoted (u_0, v_0) . Another key difference between these frames is the units. The sensor frame has units of millimeters while the pixel frame has units of pixels.

1.3.2 Projection Methods

The conversion between pixel and world coordinates can go both ways, and both are used in the post-processing pipeline. However, going from pixel to world coordinates is used more often and is slightly more challenging, so that is what is presented in this section. The conversion involves three frame transformations

$$\text{Pixel } (u, v) \rightarrow \text{Sensor } (x, y) \rightarrow \text{Camera } (X, Y, Z) \rightarrow \text{World } (N, E, DF)$$

This is referred to as a backwards projection, since it is projecting two-dimensional points back out into the world. As mentioned above this is more challenging because depth informa-

tion is lost during the forward projection as the image is captured. There are many ways of recovering this depth information, which typically use multiple images or cameras. However, the height of objects being mapped are relatively small and is not useful for this mapping application. For that reason an assumption is made that every point in the world frame has a height of zero. This reduces the world to a plane and results in a \mathbb{R}^2 to \mathbb{R}^2 mapping which is possible with a single image.

There are two different methods for performing this backwards projection. The first method discussed operates in Euclidean space and is more intuitive to understand. The second method builds on this method by instead working in Projective space which produces a more efficient, but equivalent, conversion. Both of these methods use the central perspective model and both make the same assumptions about the camera and lens. These are:

- The lens does not cause any distortion.
- There is no skew or displacement in how the image sensor is positioned within the camera.
- The focal length exactly matches the manufacturer’s specification.

All of these assumptions are false to some extent, and it’s possible to correct these assumptions using certain camera calibration procedures [TODO cite]. However, for the system presented in this research these effects are considered negligible. Another assumption is the camera’s orientation is relative to the ground. However, typically orientation angles are measured with respect to the gravity vector, which on hills will not be orthogonal to the ground. If the mapping is done on a hill then the simplest solution is to make sure the camera’s orientation about the platform’s X and Y axes is zero.

Euclidean Space

The first transformation is to convert pixel coordinates to sensor coordinates. Using the frame definitions shown in figure 1.6, this can be represented in matrix form as

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 & -\rho_h \\ \rho_w & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} \rho_h v_0 \\ \rho_w u_0 \end{bmatrix}$$

where ρ_w and ρ_h are the width and height of each pixel in millimeters, and u_0 and v_0 are the coordinates of the principal point in pixels. The next transformation is to convert from sensor coordinates to camera coordinates. It's clear from the projection model that if the sensor coordinates are combined with the focal length f as

$$(x, y, f)$$

then the result is a vector in the camera frame, but with units of millimeters rather than meters. The last frame transformation is to describe this vector in terms of the world frame. This can be accomplished by a rotation matrix R

$$\begin{bmatrix} n \\ e \\ d \end{bmatrix} = R * \begin{bmatrix} x \\ y \\ f \end{bmatrix} \tag{1.1}$$

where this rotation matrix is composed from the active, extrinsic rotations about the worlds north, east and down axes in that order, by angles ϕ (roll), θ (pitch) and ψ (yaw). These are the Tait-Bryan angles of the camera frame with respect to the world frame. Technically this is a passive rotation by negative angles since the vector is changing frames, but using active rotations avoids many double negatives.

$$\begin{aligned}
R &= R_z(\psi) * R_y(\theta) * R_x(\phi) \tag{1.2} \\
&= \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \\
&= \begin{bmatrix} \cos(\theta) \cos(\psi) & \sin(\phi) \sin(\theta) \cos(\psi) - \cos(\phi) \sin(\psi) & \sin(\phi) \sin(\psi) + \cos(\phi) \sin(\theta) \cos(\psi) \\ \cos(\theta) \sin(\psi) & \cos(\phi) \cos(\psi) + \sin(\phi) \sin(\theta) \sin(\psi) & \cos(\phi) \sin(\theta) \sin(\psi) - \sin(\phi) \cos(\psi) \\ -\sin(\theta) & \sin(\phi) \cos(\theta) & \cos(\phi) \cos(\theta) \end{bmatrix}
\end{aligned}$$

Lowercase coordinates (n, e, d) in equation 1.1 are used to designate that these components are in the north, east and down directions, but the units are still in millimeters and this does not represent the world point yet. The position of this vector in the world frame is given by the camera's world position

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} N_{cam} \\ E_{cam} \\ D_{cam} \end{bmatrix} \tag{1.3}$$

The actual world point can be found by finding the intersection of this vector, (n, e, d) , with the flat plane of the Earth given by the equation $D = 0$.

This can be done by parameterizing the vector with S as

$$\begin{bmatrix} N \\ E \\ D \end{bmatrix} = S \begin{bmatrix} n \\ e \\ d \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{1.4}$$

plugging in the third component, D , into the plane equation and solving for

$$S = -T_z/d,$$

which can be plugged back into equation 1.4. If d is zero then the position vector is parallel

to the Earth's surface and will never intersect it.

While this method is easy to visualize, it requires multiple steps for each pixel that must be converted to world coordinates. A more efficient method is presented in the next section.

Projective Space

An ideal solution would be to describe the backwards projection from the pixel plane to the world plane as a single matrix that can be applied in one step. In order to do this, coordinates in each frame must be converted into the Projective space which adds an extra coordinate. This coordinate represents scale, and a set of coordinates in this space is referred to as homogeneous coordinates.

There are many benefits to using homogeneous coordinates which is why they are commonly used in computer vision. These benefits include

1. performing translations and rotations in a single matrix.
2. avoiding unnecessary division in intermediate calculations which is typically slower than other operations.
3. representing a coordinate at infinity with real numbers which is true when the extra coordinate is zero.

In order to differentiate between regular Euclidean coordinates, homogeneous coordinates are followed by an apostrophe. For example coordinates in the pixel frame can be described in Projective space as (u', v', w') where

$$u' = u * w' \text{ and } v' = v * w'.$$

In backwards projection the pixel coordinates (u, v) are what is known so w' is always set to 1 so it does not change the scale. In order to convert from pixel to sensor coordinates the same relationship using the pixel sizes and principal point are used, however when homogeneous coordinates are substituted this relationship can be described in a single matrix

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 0 & -\rho_h & \rho_h v_0 \\ \rho_w & 0 & \rho_w u_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$$

This is typically referred to as the inverse parameter matrix, or K^{-1} . The second transformation is to convert sensor coordinates to camera coordinates. A consequence of defining the origin of the camera frame to be the location of the equivalent pin-hole is that all incoming rays of light converge to the origin of the camera frame. This leads to the simple relationship using the focal length f of

$$x = fX/Z \text{ and } y = fY/Z$$

which can easily be derived by similar triangles. When described as homogeneous coordinates this relationship becomes

$$x' = fX, y' = fY \text{ and } z' = Z$$

or in matrix form

$$\begin{bmatrix} X_u \\ Y_u \\ Z_u \end{bmatrix} = \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}.$$

The subscript u denotes that these coordinates in the camera frame are unscaled, similar to how (x, y, f) is not correctly scaled in the Euclidean method. Also note that (X_u, Y_u, Z_u) are not homogeneous coordinates. This matrix is referred to as the inverse camera matrix, or C^{-1} . The last frame transformation is to go from the camera to world frame and correctly scale the position vector. This can be done in one step if homogeneous world coordinates (N', E', D', S') are used. In terms of the rotation matrix R and translation vector T defined in equations ?? and ??

$$\begin{bmatrix} R^{-1} & -T \end{bmatrix} \begin{bmatrix} N' \\ E' \\ D' \\ S' \end{bmatrix} = \begin{bmatrix} X_u \\ Y_u \\ Z_u \end{bmatrix}.$$

If each element in R is denoted by its row, i , and column, j , as R_{ij} , and since the inverse of an orthogonal matrix is equal to its transpose then this is expanded to

$$\begin{bmatrix} r_{11} & r_{21} & r_{31} & -T_x \\ r_{12} & r_{22} & r_{32} & -T_y \\ r_{13} & r_{23} & r_{33} & -T_z \end{bmatrix} \begin{bmatrix} N' \\ E' \\ D' \\ S' \end{bmatrix} = \begin{bmatrix} X_u \\ Y_u \\ Z_u \end{bmatrix}.$$

However, this matrix is clearly not invertible and the assumption that $D = 0$ must be enforced. Since $D' = D/S'$ then D' is also zero, and it can be removed along with the third column of the matrix. After inverting this results in

$$\tilde{p} = \begin{bmatrix} N' \\ E' \\ S' \end{bmatrix} = \begin{bmatrix} r_{11} & r_{21} & -T_x \\ r_{12} & r_{22} & -T_y \\ r_{13} & r_{23} & -T_z \end{bmatrix}^{-1} \begin{bmatrix} X_u \\ Y_u \\ Z_u \end{bmatrix}.$$

Even though an entire column of the rotation matrix is discarded no information is lost as this column can be described as the cross product of the first and second columns. This inverted matrix is denoted ξ^{-1} , so that the entire projection can be represented as a single homography matrix

$$H = \xi^{-1}C^{-1}K^{-1}$$

Once (N', E', S') is computed the actual northing and easting is simply the inhomogeneous coordinates

$$N = N'/S' \text{ and } E = E'/S'$$

While it's not obvious, this result is indeed equivalent to the one derived in the first method. This is verified symbolically in Appendix TODO ref.

Bibliography

[numbered,framed]matlab-prettifier

Consolas

Appendix A

Appendix A

[style=Matlab-editor,caption=Sample code from Matlab]matlab/projection_compare.m

Appendix B

Title for This Appendix

Enter the content for Appendix B in the appendixB.tex file. If you do not have an Appendix B, see comments in the etdrtemplate.tex file for instructions on how to remove this page.