Camera calibration and Image rectification

J.C Pérez Muñoz

October 2011

Contents

List of Figures

Chapter 1

Introduction

The purpose of this short manual is to give an explanation of the Pinhole model parameters and provide a clear example of the use of the Matlab scripts coded for camera calibration and Image rectification. One of the main goals of the camera calibration is to obtain a reliable relation between the coordinates on the space and the coordinates on the image such that it is possible to rectify a desired portion of the image as showed on the Figure 1.1.

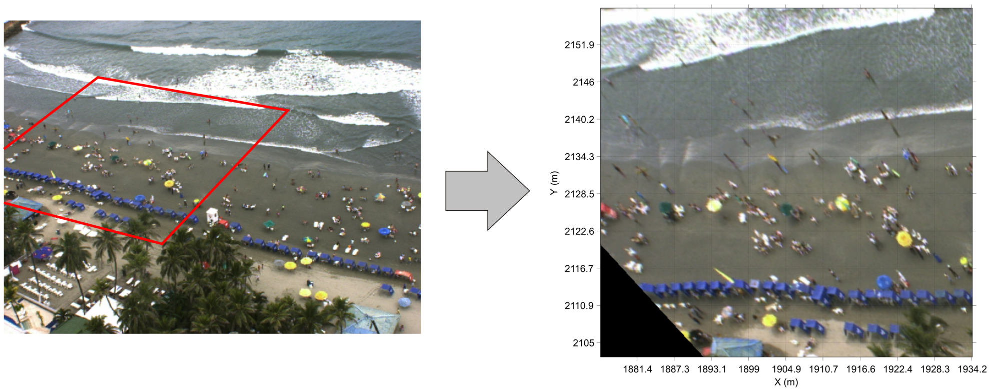

Figure 1.1: Example of the rectification of a desired area in the image

The data used in the example of this manual are attached with the pdf file.

Chapter 2

Pinhole Model

The Pinhole model is one of the most used camera models [1, 2, 5, 6, 7, 8] and there are different ways to obtain values for its parameters. All those methods for camera calibration have in common the usage of Ground Control Points (GCP) which are points easy to identify in the image and whose coordinates in the space frame are known.

2.1 Ideal Pinhole model

The *ideal* Pinhole model does not take into account the distortion effects caused by the lenses and it is based on the collinearity principle, which means that it assumes all the projections of points in the space on the image to cross the same point known as optical center [5]. A scheme of the Pinhole model is presented in Figure 2.1 where it is possible to identify some of its parameters. The Pinhole model has two sets of parameters: Intrinsic and extrinsic, the first are parameters related to the camera configuration such size of the sensor, local length, resolution, etc., they are:

- f Local length of the lenses.
- D_u Scale parameter in the direction u of the image.
- D_v Scale parameter in the direction v of the image.
- u_0 Coordinate in the direction u of the center of the image.
- v_0 Coordinate in the direction v of the center of the image.

The extrinsic parameters are related to the position an orientation of the camera measured in the spatial reference frame where the GCP were measured, they are:

- x_c Coordinate in the x direction of the Optical center of the image.
- y_c Coordinate in the y direction of the Optical center of the image.

- z_c Coordinate in the z direction of the Optical center of the image.
- τ Tilt of the camera.
- σ Roll of the camera.
- φ Azimuth of the camera.

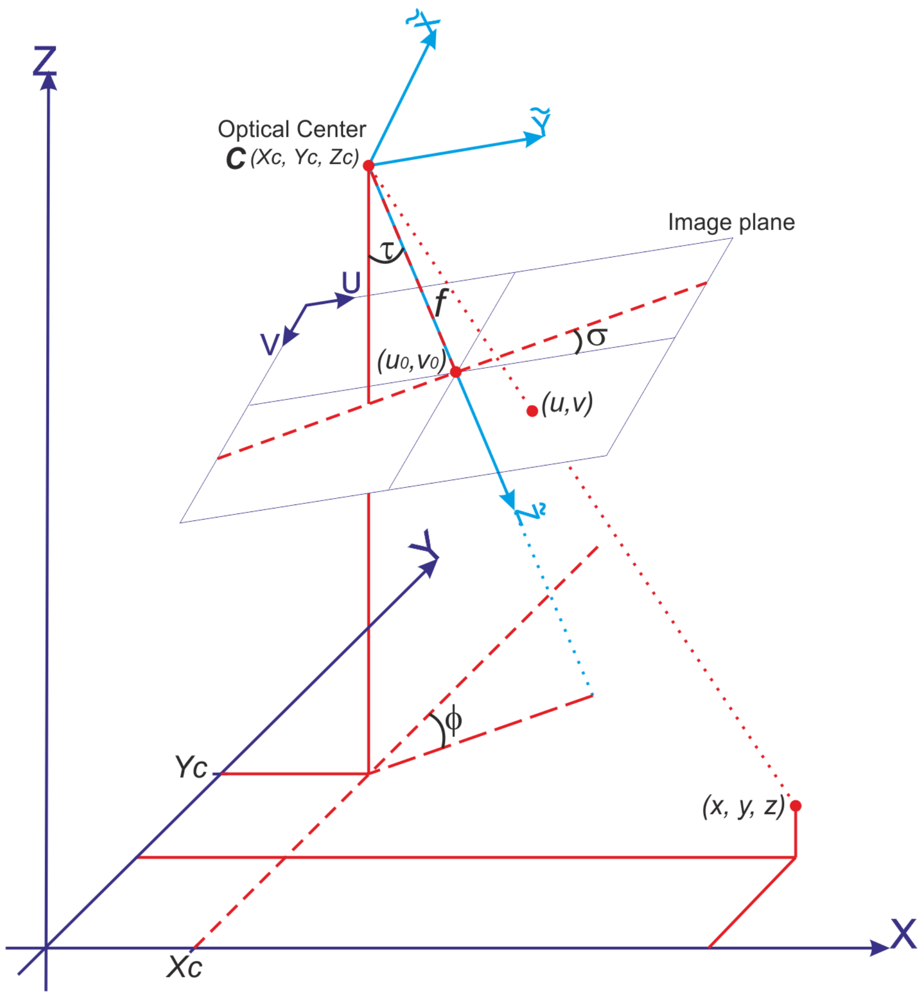

Figure 2.1: Scheme of the Pinhole Model.

Then the *ideal* Pinhole model has 11 parameters. The image coordinates are usually represented as showed in Figure 2.2 and thus, the Pinhole model is represented as a transformation function from the space to the image as presented in equation 2.1.

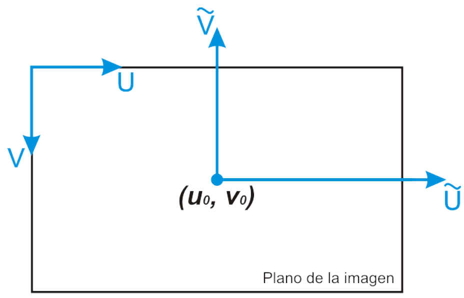

Figure 2.2: Image Coordinates.

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \underbrace{\begin{bmatrix} fD_u & 0 & u_0 \\ 0 & -fD_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{K} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}}_{R} \begin{bmatrix} 1 & 0 & 0 & -x_c \\ 0 & 1 & 0 & -y_c \\ 0 & 0 & 1 & -z_c \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.1}$$

The matrix K is know as the internal calibration matrix and the matrix R is the rotation matrix and depends of the rotation angles τ, σ, ϕ, as presented in equation (2.2).

$$\begin{aligned}
r_{11} &= \cos\phi\cos\sigma + \sin\phi\cos\tau\sin\sigma \\
r_{12} &= -\sin\phi\cos\sigma + \cos\phi\cos\tau\sin\sigma \\
r_{13} &= \sin\tau\sin\sigma \\
r_{21} &= -\cos\phi\sin\sigma + \sin\phi\cos\tau\cos\sigma \\
r_{22} &= \sin\phi\sin\sigma + \cos\phi\cos\tau\cos\sigma \\
r_{23} &= \sin\tau\cos\sigma \\
r_{31} &= \sin\phi\sin\tau \\
r_{32} &= \cos\phi\sin\tau \\
r_{33} &= -\cos\tau
\end{aligned} \tag{2.2}$$

Note that the internal calibration matrix compresses the parameters f, D_u and D_v in just 2 parameters reducing the total number of Pinhole parameters to 10. Some authors [1, 3] propose to use a linear approximation of the Pinhole Model called the Direct Linear Transform (DLT) that compresses the model of equation 2.1 in just a 3×4 matrix such that 2.3 is satisfied:

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = H \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \approx K\,[R \mid -RC] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.3}$$

Where $C = [x_c y_c z_c]^T$.

2.2 Pinhole model with distortion

The ideal Pinhole model does not take into account the effects of the distortion caused by the lenses but, in most of the real world applications, those effects are important and need to be considered. There are several sources of lens distortion but several authors [1, 4, 5, 9] recommend to focus just in the radial distortion as the other types of distortion can be minimized in the lens construction process. The radial distortion can enlarge or diminish the resolution in the image as it is showed in the Figure 2.3.

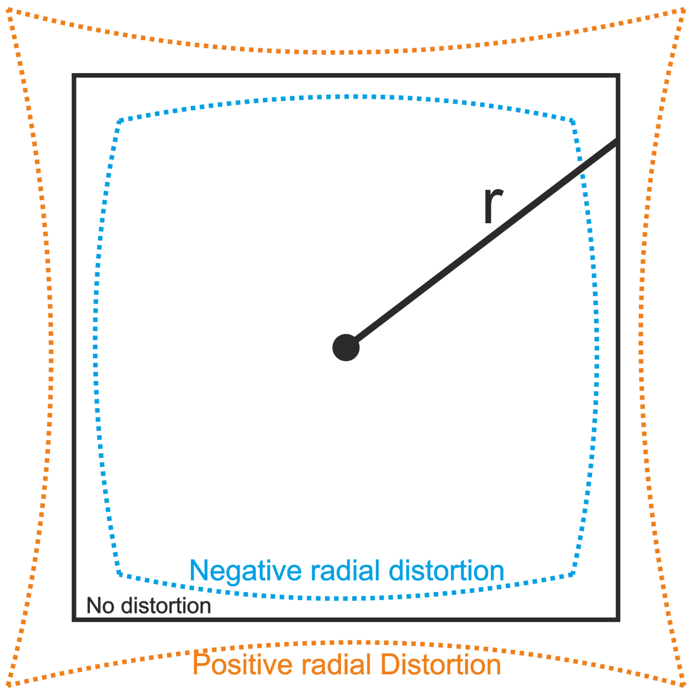

Figure 2.3: Radial distortion effect in an image.

In the case that the radial distortion is modeled with 2 parameters, the complete Pinhole model can be written as equation 2.2.

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = K\left[R \mid -RC\right]\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} + K\begin{bmatrix} \widetilde{w}\delta_u\left(\widetilde{u},\widetilde{v}\right) \\ \widetilde{w}\delta_v\left(\widetilde{u},\widetilde{v}\right) \\ 0 \end{bmatrix}$$

Where

$$\begin{bmatrix} \widetilde{w}\widetilde{u} \\ \widetilde{w}\widetilde{v} \\ \widetilde{w} \end{bmatrix} = [R| - RC]\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.4}$$

$$\begin{bmatrix} \delta_u\left(\widetilde{u},\widetilde{v}\right) \\ \delta_v\left(\widetilde{u},\widetilde{v}\right) \end{bmatrix} = \begin{bmatrix} \widetilde{u} \\ \widetilde{v} \end{bmatrix}\left[k_1\left[\sqrt{\widetilde{u}^2+\widetilde{v}^2}\right]^2 + k_2\left[\sqrt{\widetilde{u}^2+\widetilde{v}^2}\right]^4\right] \tag{2.5}$$

Note that the only new parameters are k_1, k_2 increasing the total number of Pinhole parameters to 12. Given that each non coplanar GCP adds two degrees of freedom for the solution of the model, we need at least 6 GCP in order to find good estimates for the Pinhole parameters. It is also possible to use less GCP in the case that some parameters are known and initial estimates for all the unknown parameters are provided.

Chapter 3

Camera Calibration

3.1 Example data

We will calibrate a camera using just one image and a set of 8 GCP. The image used is named TestImage.jpg as is presented in Figure 3.1. The GCP data is saved in the file GCP.mat, in the case the user needs to obtain the (u, v) coordinates of the GCP directly from the image, he can use the function ginput from Matlab®. The GCP data is presented in Table 3.1.

Figure 3.1: Image used in the calibration process

GCP	u	v	x	y	z
a	282	368	287.434998768731	488.902405441972	-2.36150000000000
b	316	379	295.005853720475	491.828358442523	-2.44370000000000
c	281	405	298.183500600397	483.830748810899	-2.32680000000000
d	574	421	333.839192084095	514.136837212835	-1.39360000000000
e	745	520	364.188881038164	517.130134872394	-2.39730000000000
f	721	562	368.185621592042	509.514491064707	-1.06750000000000
g	788	584	375.776590871392	512.661294844002	-2.35510000000000
h	811	538	371.857618456590	520.423063220922	-2.34250000000000

Table 3.1: GCP data used in the example.

3.2 DLT calibration

First we will show how to obtain the DLT of the camera using 6 or more GCP. For this, use the function DLT.m which finds the DLT H of the camera under the condition

$$h_{31}^2 + h_{32}^2 + h_{33}^2 = 1 \tag{3.1}$$

After loading the GCP data in matlab using the command load('GCP.mat'), use the next command to obtain the DLT:

H=DLT(UV(:,1),UV(:,2),XYZ(:,1),XYZ(:,2),XYZ(:,3))

Also it is possible to find the DLT under the condition $h_{34} = 1$, in this case use the command:

H=DLT_non_homogeneus(UV(:,1),UV(:,2),XYZ(:,1),XYZ(:,2),XYZ(:,3))

For the case when the Pinhole parameters should be estimated, there are some consideration to keep in mind:

- The angles τ, σ, ϕ are replaced by the vector elements a, b, c, where $\begin{bmatrix} a & b & c \end{bmatrix}^T$ is the rotation vector obtained from the rotation matrix using the Rodrigues formula [5, 8].
- The coordinates of the optical center $C = \begin{bmatrix} x_c & y_c & z_c \end{bmatrix}^T$ are replaced by the rotated coordinates $\begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T = RC$.

Then, the complete set of parameters used in the functions is

$$fD_u, fD_v, u_0, v_0, k_1, k_2, a, b, c, t_x, t_y, t_z$$

The Rodrigues formula is implemented in the function rodrigues.m in such way that if R is a rotation matrix ($\det(R) = 1, R^T R = I$) then:

[c]=rodrigues(R)

Produces a 3 elements vector with the values of a, b, c and $R =$rodrigues(rodrigues(R)).//

3.3 Pinhole parameters estimation

The function camera_cal.m allows the user to calibrate the Pinhole parameters including or not distortion, if there is no initial estimates of the parameters then the user needs to provide at least 6 GCP no matter how many parameters are going to be estimated. For example, after loading the GCP data in Matlab, the user can obtain the ideal Pinhole model parameters (Without distortion) using the command:

```
[H K D R t P MSEuv MSExy NCError]=...
camera_cal(UV(:,1),UV(:,2),XYZ(:,1),XYZ(:,2),XYZ(:,3)),[],[1 1 1 1 0 0 1 1 1 1 1 1])
```

The last vector specifies which Pinhole parameters are going to be estimated with a 1 and the ones that remain constant with 0. The order is

$$[fD_u, fD_v, u_0, v_0, k_1, k_2, a, b, c, t_x, t_y, t_z].$$

The MSEuv error is the Mean Square Error of the projections of the (x, y, z) coordinates to the image using the estimated parameters. The MSExy error is the mean square error of the back-projections of the coordinates (u, v) from the image to the space using the coordinate z as parameter. The NCE error is the Normalized Calibration Error [5, 6] and its interpretation is:

- $NCE \sim 1$ means that the error produced by the model is close in magnitude to the error caused by the digitalization process in the camera.
- $NCE \ll 1$ means that the error produced by the model is despicable compared with the digitalization error.
- $NCE \gg 1$ means that the model is not accurate enough.

In the case all the parameters need to be estimated use the command:

```
[H K D R t P MSEuv MSExy NCError]=...
camera_cal(UV(:,1),UV(:,2),XYZ(:,1),XYZ(:,2),XYZ(:,3)),[],[1 1 1 1 1 1 1 1 1 1 1 1])
```

Now, the model can be refined if after obtaining all the parameters the intial conditions for (u_0, v_0) are changed to the center of the image plane, for example the center go the test image in this case is $(512, 384)$, then using the next commands it is possible to obtain a model whose parameters are closer to the expected values.

```
est=[K(1,1) -K(2,2) 512 384 0 0 rodrigues(R)' t'];
[H K D R t P MSEuv MSExy NCError]=...
camera_cal(UV(:,1),UV(:,2),XYZ(:,1),XYZ(:,2),XYZ(:,3),est,[1 1 1 1 1 1 1 1 1 1 1 1])
```

Chapter 4

Coordinates transformation and rectification

4.1 Coordinates transformation

The real image generally has some radial distortion and, hence, its coordinates need to be corrected before they can be transformed using the ideal Pinhole model, Figure 4.1 presents the relationship between coordinates is a schematic way meanwhile the Figure 4.2 presents the relationship between the coded functions to transform coordinates.

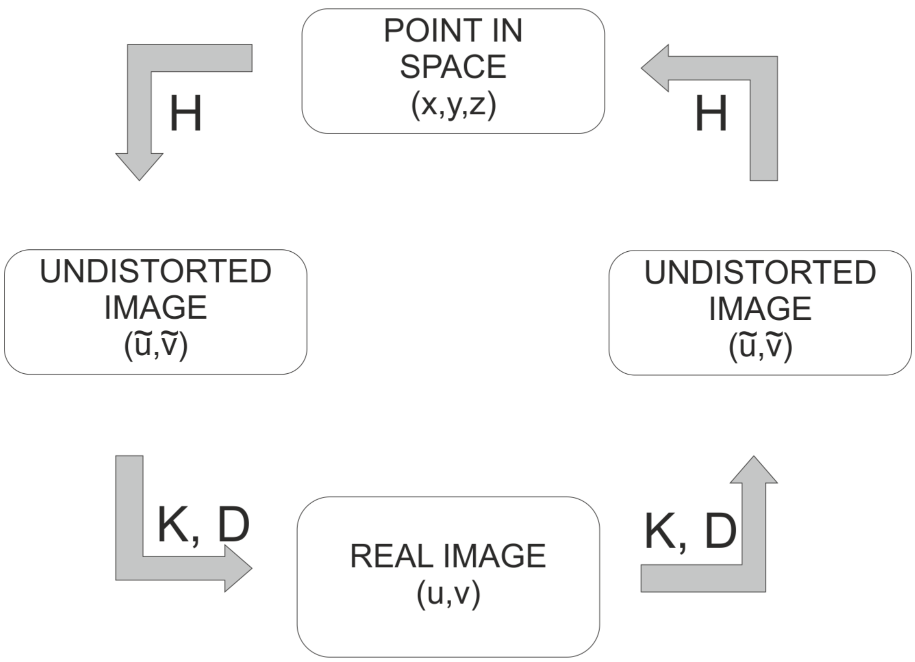

Figure 4.1: Relationship between coordinates spaces.

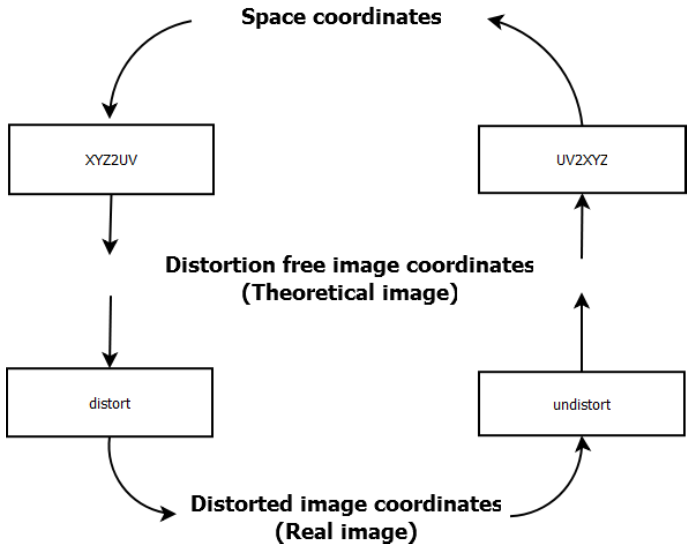

Figure 4.2: Relationship between coordinate transformation functions.

For example, if the point $(500, 600)$ over the image needs to be transformed to the space coordinates using $z = -3$ (Because the GCP have z values around -2), the commands to be used are:

```
[u v]=undistort(K,D,[500 600])
[x y z] = UV2XYZ(H,u,v,-3)
```

And to transform back those points, use the commands:

```
[u,v]=XYZ2UV(H,[x y z])
[u v]=distort(K,D,[u v])
```

In the case the user is interested in correcting the distortion in all the image and obtain a new distortion-free image, he can use the commands:

```
In=undistort_Image(K,D,'TestImage.jpg',1,1);
```

4.2 Image Rectification

In order to rectify an image, the user needs to define a polygon in it that encircle the interest area. Such polygon can be defined using the matlab function ginput after loading and plotting the image in Matlab, then the function rectify.m can be used. In this case we use again $z = -3$ and we define four points defining the area of interest using ginput:

```
I=imread('TestImage');
figure(1), imshow(I)
[u1 v1]=ginput(4);
[u2 v2 rectimg]=rectify(I,H,K,D,[u1 v1],-3,1/10,1);
```

Note that the output resolution is $1/10m/pixel$. The output image is presented in Figure 4.3

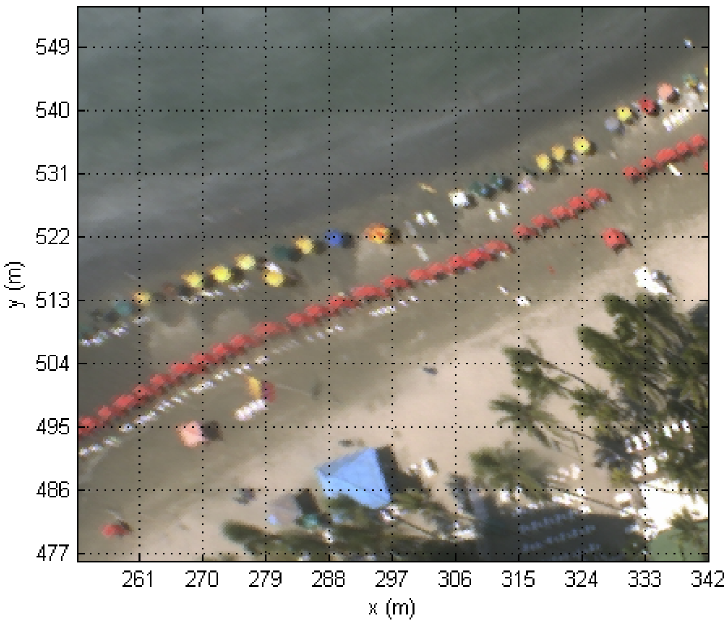

Figure 4.3: Rectified image of a portion of the original test image.

Some users may be interested in obtaining a resolution plot of the image as, after all, the resolution in the original image is not constant. In this case the user just needs to rectify an image and pass the results to the function plot_resolution:

```
plot_resolution(H,K,D,u2,v2,rectimg,-3)
```

In this case, the result image is presented in Figure 4.4

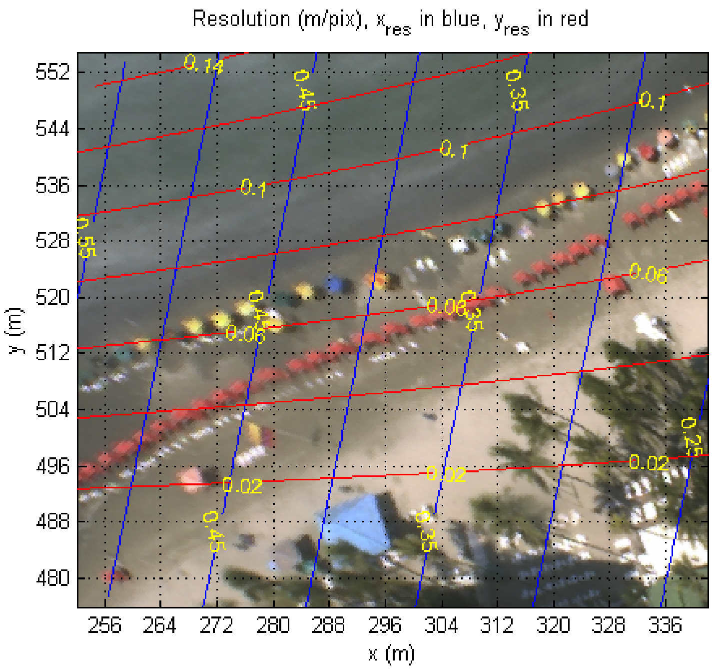

Figure 4.4: Rectified image of a portion of the original test image and the resolution over it.

Bibliography

[1] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1106–1112.

[2] K.T. Holland, R.a. Holman, T.C. Lippmann, J. Stanley, and N. Plant. Practical use of video imagery in nearshore oceanographic field studies. *IEEE Journal of Oceanic Engineering*, 22(1):81–92, 1997.

[3] G Hommen, M de Baar, P Nuij, G McArdle, R Akers, and M Steinbuch. Optical boundary reconstruction of tokamak plasmas for feedback control of plasma position and shape. *The Review of scientific instruments*, 81(11):113504, November 2010.

[4] A.F. Osorio, J.C. Pérez, C.A. Ortíz, and R. Medina. Técnicas basadas en imágenes de video para cuantificar variables ambientales en zonas costeras. *Avances en Recursos Hidráulicos*, (16):51–64, 2007.

[5] J.C. Pérez Muñoz. Optimización no lineal y calibración de cámaras fotográficas. Master's thesis, National University of Colombia, 2009.

[6] Joaquim Salvi, Xavier Armangué, and Joan Batlle. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition*, 35(7):1617–1635, July 2002.

[7] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):965–980, 1992.

[8] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[9] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.