Manual for easyCamera Camera Configuration Planning Tool

Diane H. Theriault

Feb 14, 2014

Contents

1 Installation

Place the zipped easyCamera1.zip file in any directory. Unzip the compressed file using any decompression software. When unzipped successfully, you will see several files and directories, including:

```
easyCamera.m                          +util
constructCameraPlacement.m            +plot_util
estimateReconstructionUncertainty.m  +plot_tools
plotCameraSetup.m                     +geometry_algorithm
plotReconstructionError.m             +reconstruction
...
```

The directories that start with a + sign are Matlab packages and their contents are used by the easyCamera software.

Once the files have been unzipped successfuly, make sure that Matlab has access to this directory (by changing your working directory, or adding the directory to the Matlab path). Run the easyCamera tool by typing "easyCamera" at the command prompt.

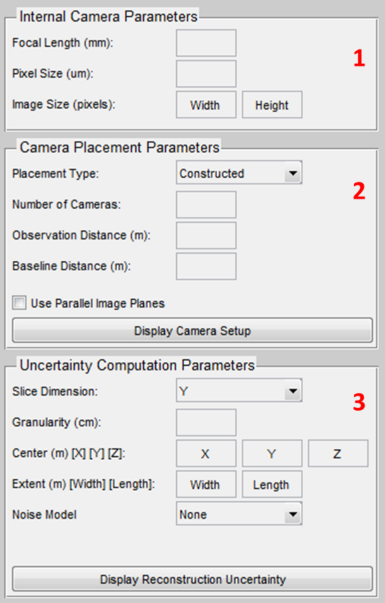
2 Description

The easyCamera tool allows users to design idealized camera configurations, visualize their overlapping fields of view, and visualize estimated reconstruction uncertainty due to image quantization and image localization ambiguity as discussed in Theriault et al (2014), "A protocol and calibration method for accurate multi-camera field videography", Journal of Experimental Biology. Camera configurations, as estimated by the easyWand software, can also be visualized. Reconstruction uncertainty will vary spatially throughout the volume of view; the easyCamera tool will estimate and display the reconstruction uncertainty for points in a rectilinear plane cutting through the space. Multiple planes in different orientations can be visualized on a single plot. It assumes that all cameras are identical and uses a pinhole camera model. The actual reconstruction uncertainty may be higher if the actual cameras experience significant lens distortion.

3 User Interface

3.1 Overview

The easyCamera interface contains three sub-panels for providing parameters. The first panel is for entering the intrinsic parameters of the camera (1). The second planel is for providing the pose of the cameras (2). The third panel provides parameters needed to estimate the reconstruction uncertainty (3).



Placement Type: Constructed

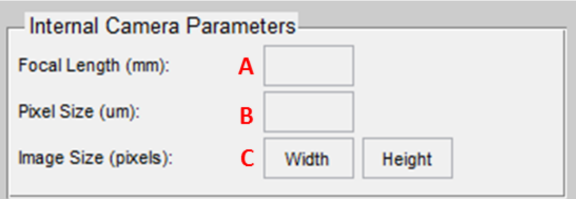
□ Use Parallel Image Planes

Slice Dimension: Y

Noise Model: None

3.2 Intrinsic Parameters

In order to properly determine the field of view of the cameras, and subsequently estimate the reconstruction uncertainty, the software needs the internal parameters of the cameras, in particular, the focal length, physical pixel size, and image resolution. This information is provided on the first sub-panel of the interface (1).



(A) **Focal Length** (in millimeters (mm)). The focal length can typically be found printed on the lens. A shorter focal length will give a wider field of view, and a longer focal length will give a narrower field of view.

(B) **Pixel Size** (in micrometers (μm)). The physical size of the pixels in the camera can typically be found in the camera specification documentation or on the manufacturer's website. If only the sensor size is given, the pixel size can be found by dividing the size of the sensor by the image resolution. Only square pixels are supported.

(C) **Image Size** (pixels). The resolution of the image, in pixels. The width and height are entered separately.

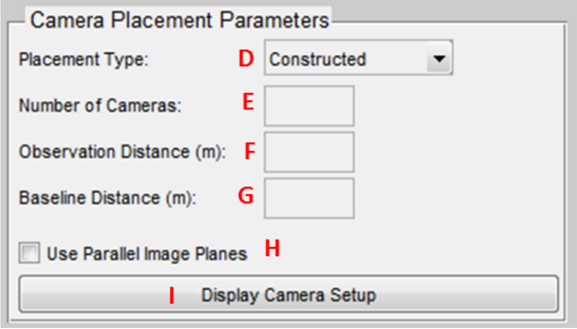
3.3 Extrinsic Parameters

Users can visualize camera placements in two ways: by constructing an idealized configuration by providing simple geometric parameters, or by using the estimated position of the cameras, as computed by easyWand. The estimated pose of the cameras are encoded in the DLT coefficients that easyWand produces. The user provides the necessary information via the second sub-panel of the interface (2).

When using a "constructed" camera placement (Section 3.3.2), the user must first specify the number of cameras in the configuration. All of the cameras will be placed on an arc of a circle, oriented so that they face a common fixation point. The distance between the cameras and the fixation point, the "observation distance," is the radius of the circle. The distance between the outermost cameras, the "baseline distance," determines the angle of the arc, θ via the formula $\theta = sin^{-1}(\frac{\text{baselineDistance}}{2}/\text{observationDistance})$. The angle between subsequent cameras on the arc cameras is given by $2\theta/(\text{cameraNumber} - 1)$. The length of the displayed view frustums will be twice the given observation distance.

When using an "estimated" camera placement (Section 3.3.1), the camera position and orientation will be determined based on the provided DLT coefficients, which can be computed with the easyWand software. The user must specify the length of the displayed view frustums.

3.3.1 Constructed Camera Placement



Camera Placement Parameters

Placement Type: D Constructed ▾

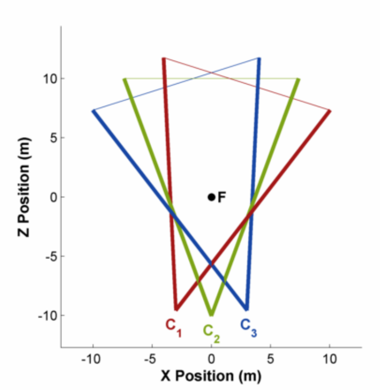
Number of Cameras: E

Observation Distance (m): F

Baseline Distance (m): G

☐ Use Parallel Image Planes H

I Display Camera Setup



(D) **Placement Type: Constructed** This drop-down menu will allow users to switch between using either an idealized "constructed" camera placement (D) or the camera configuration calculated from the DLT coefficients as estimated by easyWand (D2).

(E) **Number of cameras**. The number of cameras to place on the arc.

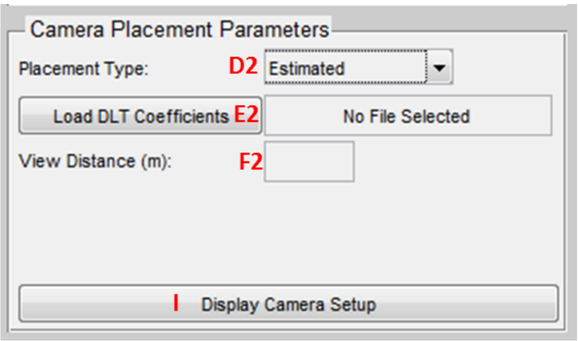
(F) **Observation Distance** (meters (m)). The anticipated distance between the cameras and the animals of study is used as the radius of the circle where the cameras are placed. The plotted view distance of the camera frustums will be twice the observation distance.

(G) **Baseline Distance** (meters (m)). The maximum distance between the outermost cameras.

(H) **Use Parallel Image Planes** (check box). If desired, the cameras can be constructed so that the image planes are parallel; the camera positions will still be on an arc with a radius of baselineDistance)

(I) **Display Camera Setup** Will create a new plot depicting the view frustums of the cameras.

3.3.2 Estimated Camera Placement

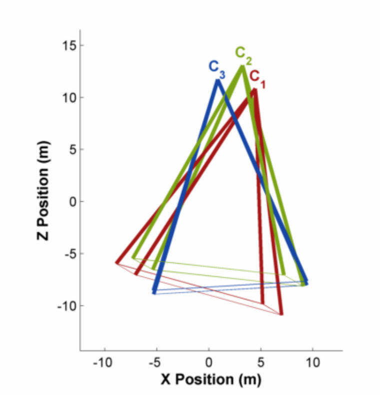


Camera Placement Parameters

Placement Type: D2 Estimated

Load DLT Coefficients E2 No File Selected

View Distance (m): F2



(D2) **Placement Type: Estimated** This drop-down menu will allow users to switch between using either an idealized "constructed" camera placement (D) or the camera configuration calculated from the DLT coefficients as estimated by easyWand. The numbers in the file should contain one column per camera, with 11 rows for the coefficients.

(E2) **Load DLT Coefficients** This button allows the user to select the file containing the desired DLT coefficients. The name of the file will be displayed in the text box to the right. By placing the cursor in the box and using the arrow keys, users can see the entire filename, even if the filename is long.

(F2) **View Distance** The length of the plotted view frustums.

(D) **Display Camera Setup** Will create a new plot depicting the view frustums of the cameras. It may be necessary to adjust the rotation of the figure to display the view frustums as expected. This can be done using the Matlab rotation tool. Unless axis points were defined in easyWand, the X-Z view will probably be the most intuitive.

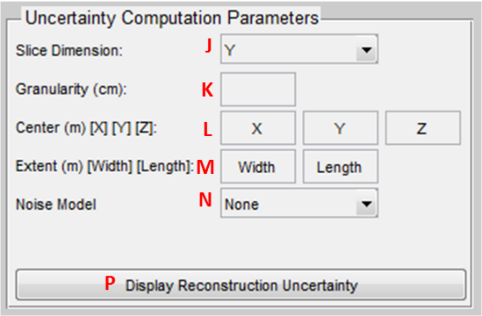
3.4 Uncertainty Estimation

Once the camera configuration has been designed or loaded from a file via the DLT coefficients, it is possible to estimate the reconstruction uncertainty. In the case of estimated camera configurations, it is assumed that the calibration is good and the DLT coefficients are correct (see easyWand for measures of the quality of the calibration). All of the information needed to display the reconstruction uncertainty is entered in the third sub-panel of the interface (3).

Reconstruction uncertainty varies spatially. It is possible to calculate the uncertainty throughout the entire volume of space, but this is difficult to visualize, so easyCamera displays the reconstruction uncertainty for points in a slice through the space. To display the spatially varying uncertainty, 3D points are generated on a lattice and then jittered to prevent a visible interference pattern. The uncertainty is estimated for each point and then it is displayed.

Uncertainty comes from two sources: image quantization and image localization ambiguity. To estimate uncertainty due to image quantization, 3D points are projected onto the cameras, then their coordinates are quantized, and lastly the quantized points are used to reconstruct the 3D position. Finally, the reconstructed 3D position is compared to the original 3D position. This source of uncertainty is geometric and does not have a stochastic element. In order to include image localization ambiguity, it is possible to add noise to the points before performing the reconstruction. Image localization ambiguity arises both when the exact image "location" of an animal is ambiguous and from annotation error, where the exact location, if it exists, may not always be annotated precisely due to differences between annotators.

3.4.1 Uncertainty Computation Geometry



Uncertainty Computation Parameters

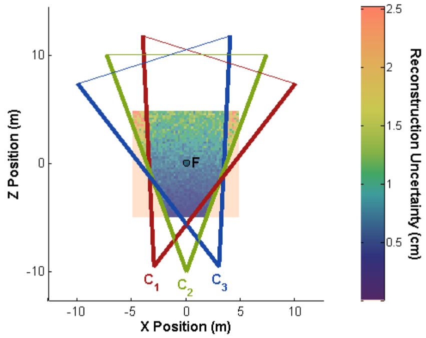
Slice Dimension: J Y

Granularity (cm): K

Center (m) [X] [Y] [Z]: L X Y Z

Extent (m) [Width] [Length]: M Width Length

Noise Model N None



(K) **Slice Dimension** This drop-down determines the cutting direction of the plane. The selected direction will be normal to the plane. If the value is set to the Y plane, for example, the uncertainty will be computed for a particular value of Y over the entire extent for X and Z. If the plane does not appear to be displayed after the program finishes computing, try rotating the plot.

(L) **Granularity** (centimeters (cm)) The size of the voxels used to compute the uncertainty. Approximately nine points will be generated for each voxel (there may be some variation due to jitter added to the 3D points to prevent an interference pattern). The displayed uncertainty will be an average of the uncertainty estimated for all points in the voxel. A recommended value is 1/100 of the largest dimension. (Note that granularity is given in centimeters whereas the extent is given in meters).

(M) **Center** (meters (m)) The center of the plane. An appropriate value may be chosen by examing the plot of the camera configuration generated by pressing the "Display Camera Setup" (I) button.

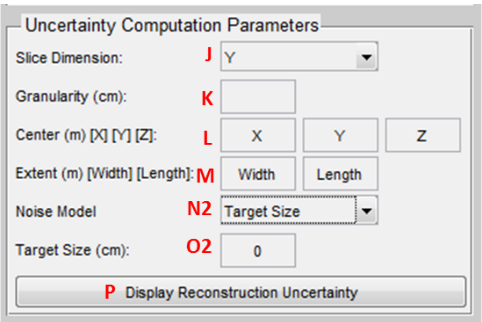
(N) **Extent** (meters (m)) The width and length of the plane. If the normal direction is Y, the width will be the extent in the X dimension and the length will be the extent in the Z dimension. If the normal direction is X, the width will be the extent in the Y dimension and the length will be the extent in the Z dimension. If the normal direction is Z, the width will be the extent in the X dimension and the length will be the extent in the Y dimension.

(N) **Noise Model: None** To visualize uncertainty only due to image quantization, choose "None" as the noise model.

(P) **Display Reconstruction Uncertainty** Compute and display the reconstruction uncertainty on the last camera configuration plot that was produced. (Will create a new plot if the previous plot had been closed.)

(Not shown) **Save** After the reconstruction uncertainty has been computed, an additional button will appear to allow the user to save the reconstruction uncertainty data. This button will prompt for a filename that will be used to create a Matlab .mat file that will contain two variables: `reconstructionUncertainty` and `histAxis`. The `reconstructionUncertainty` variable is a 3D matrix containing the average reconstruction uncertainty in each voxel. The `histAxis` variable is a cell array containg 3 cells, containing the coordinates for each dimension. These variables can then be imported into Matlab using the `load` command. (Example usage below)

3.4.2 Uncertainty Computation Noise Models



Uncertainty Computation Parameters

Slice Dimension: Y

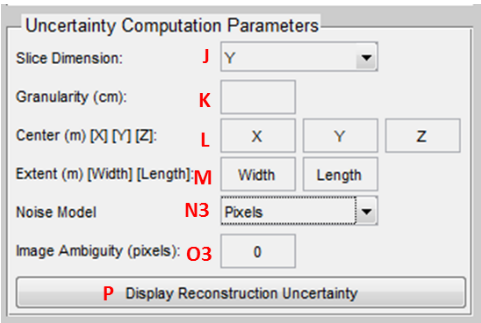
Granularity (cm):

Center (m) [X] [Y] [Z]: X Y Z

Extent (m) [Width] [Length]: Width Length

Noise Model: Target Size

Target Size (cm): 0



Uncertainty Computation Parameters

Slice Dimension: Y

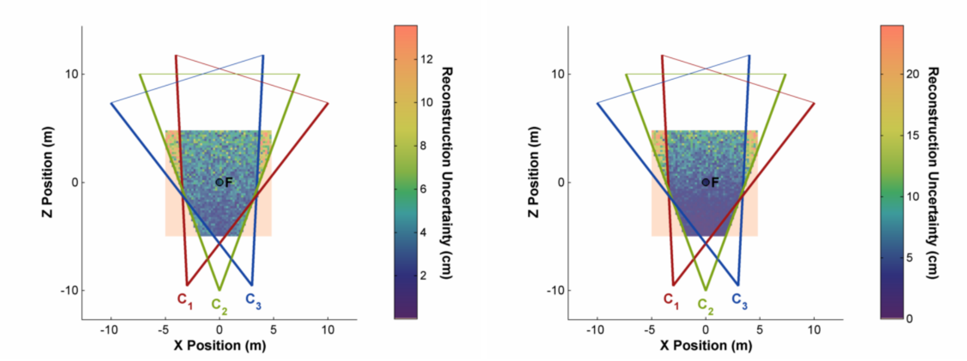
Granularity (cm):

Center (m) [X] [Y] [Z]: X Y Z

Extent (m) [Width] [Length]: Width Length

Noise Model: Pixels

Image Ambiguity (pixels): 0



(N2) **Noise Model: Target Size** To visualize reconstruction uncertainty that arises when the exact image "location" of an animal is ambiguous, choose "Target Size" as the noise model. In this case, isotropic Gaussian noise will be added to the image points before they are used to reconstruct the 3D points. The standard deviation of the distribution used to generate the noise will be different for each 3D point and will be 1/6 of the calculated apparent size of an object at the location of the point.

(O2) **Target Size** (centimeters (cm)) The physical size of the objects of study, which will be used to calculate the apparent size of the objects, in order to determine the standard deviation to use to generate noise to add to the image coordinates corresponding to each 3D point. Active only when the "Target Size" noise model is used.

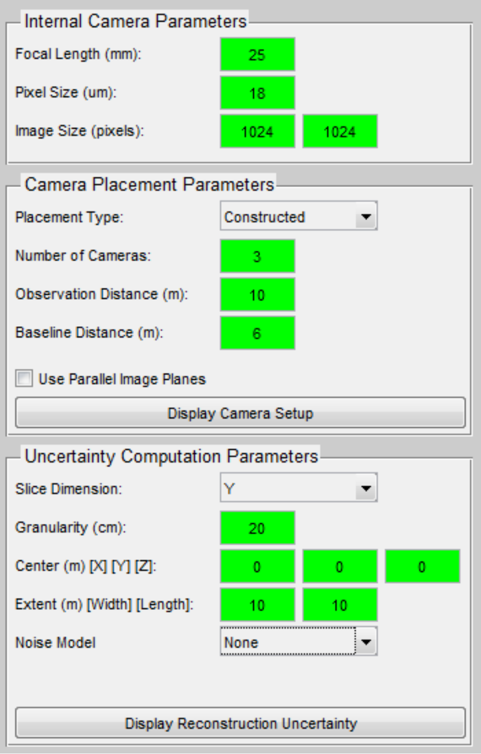
(N3) **Noise Model: Pixels** To visualize reconstruction uncertainty that may arise due to image localization error from some fixed source, such as inconsistency between human annotators, choose "Pixels" as the noise model. In this case, isotropic Gaussian noise will be added to the image points where the standard deviation of the distribution used to generate the noise will be the same for each 3D point.

(O3) **Image Ambiguity** (pixels) The standard deviation of the noise to add to the image coordinates corresponding to each 3D point. Active only when the "Pixels" noise model is used.

4 Example

This example shows how to create the figure with the "constructed" camera placement used as an example throughout this manual. We used the following parameters shown in the following table. After pressing the "Display Reconstruction Uncertainty" the button turns yellow while the software computes the uncertainty. This can take a little time, especially if the granularity chosen is small with respect to the extents. With the example parameters, it took a few seconds.

Focal Length (mm)	25
Pixel Size (μm)	18
Image Size (pixels)	1024, 1024
Number of Cameras	3
Observation Distance (m)	10
BaselineDistance (m)	6
Granularity (cm)	20
Center (m)	0, 0, 0
Extent (m)	10, 10
Noise Model	None



Internal Camera Parameters

Focal Length (mm): 25

Pixel Size (um): 18

Image Size (pixels): 1024 1024

Camera Placement Parameters

Placement Type: Constructed

Number of Cameras: 3

Observation Distance (m): 10

Baseline Distance (m): 6

□ Use Parallel Image Planes

Uncertainty Computation Parameters

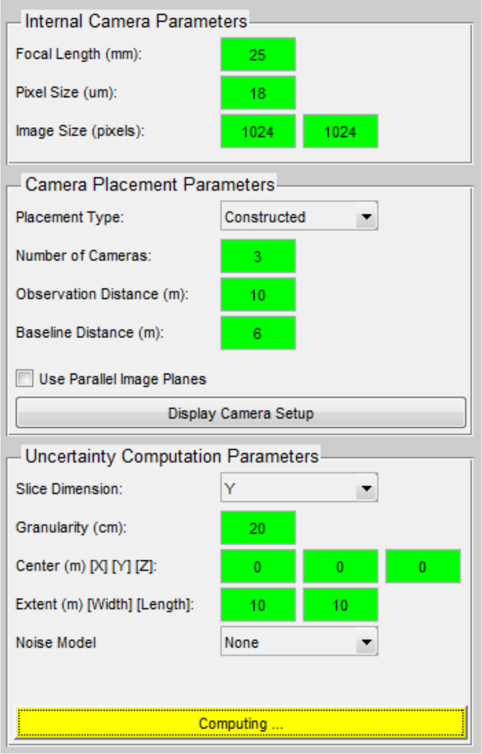
Slice Dimension: Y

Granularity (cm): 20

Center (m) [X] [Y] [Z]: 0 0 0

Extent (m) [Width] [Length]: 10 10

Noise Model None



Internal Camera Parameters

Focal Length (mm): 25

Pixel Size (um): 18

Image Size (pixels): 1024 1024

Camera Placement Parameters

Placement Type: Constructed

Number of Cameras: 3

Observation Distance (m): 10

Baseline Distance (m): 6

□ Use Parallel Image Planes

Uncertainty Computation Parameters

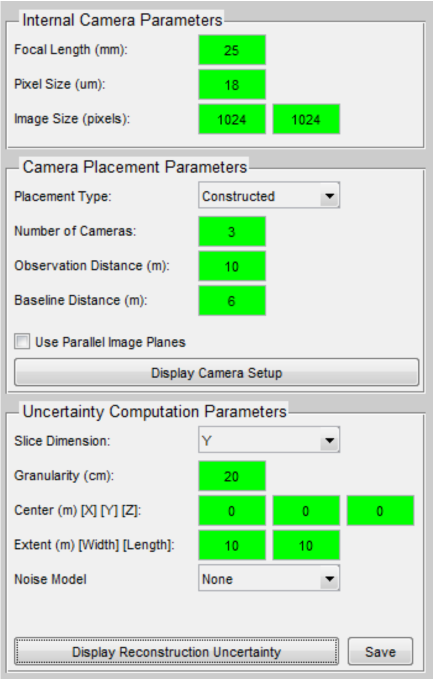
Slice Dimension: Y

Granularity (cm): 20

Center (m) [X] [Y] [Z]: 0 0 0

Extent (m) [Width] [Length]: 10 10

Noise Model None



Internal Camera Parameters

Focal Length (mm): 25

Pixel Size (um): 18

Image Size (pixels): 1024 1024

Camera Placement Parameters

Placement Type: Constructed

Number of Cameras: 3

Observation Distance (m): 10

Baseline Distance (m): 6

☐ Use Parallel Image Planes

Uncertainty Computation Parameters

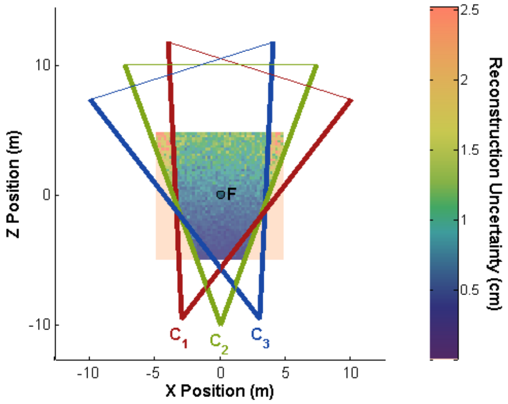
Slice Dimension: Y

Granularity (cm): 20

Center (m) [X] [Y] [Z]: 0 0 0

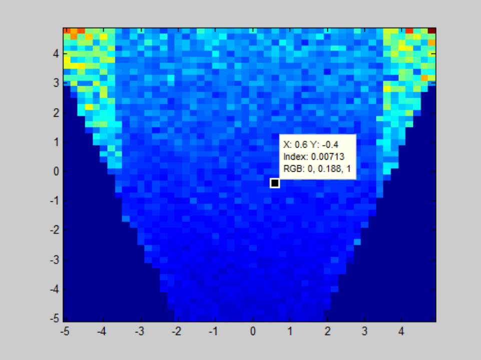
Extent (m) [Width] [Length]: 10 10

Noise Model None

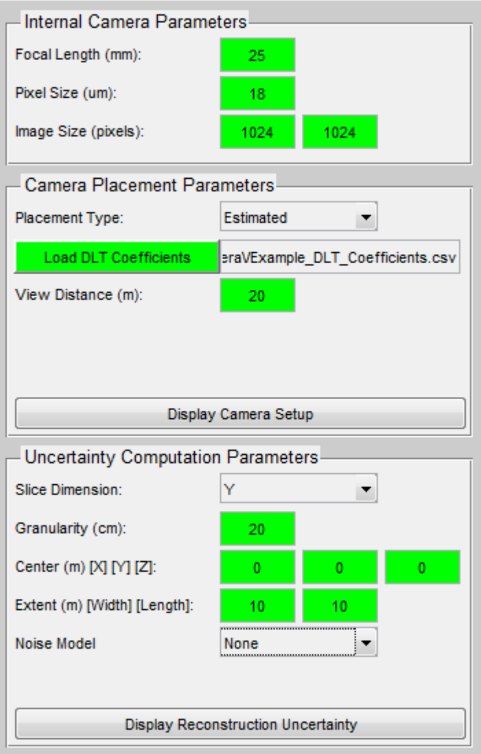


After the computation is done, the cammera configuration and associated uncertainty is displayed, and the “Save” button on the main panel appears. This button allows the user to choose a Matlab .mat file where the results can be saved. The variables containing the reconstruction uncertainty can be imported into Matlab and displayed with the following commands, and then plotted in a figure. It should be noted that the X values will be stored in the first dimension (columns) of the matrix, since Matlab stores its data in column-major order. When plotting an image, Matlab expects that the X values are in the rows. Hence, it is necessary to transpose the matrix, otherwise the image will be transposed.

```
>> load exampleOutput.mat
>> figure;
>> imagesc(histAxis{1}, histAxis{2}, squeeze(reconstructionUncertainty)');
>> set(gca, 'ydir', 'normal');
```



To use the estimated position of the cameras by loading the DLT coefficients computed by easyWand, change the “Camera Placement Type” to “Estimated.” Then, the DLT coefficients can be loaded by clicking on the button. The selected filename will appear in the text box.



Internal Camera Parameters

Focal Length (mm): 25

Pixel Size (um): 18

Image Size (pixels): 1024 1024

Camera Placement Parameters

Placement Type: Estimated

Load DLT Coefficients eraVExample_DLT_Coefficients.csv

View Distance (m): 20

Display Camera Setup

Uncertainty Computation Parameters

Slice Dimension: Y

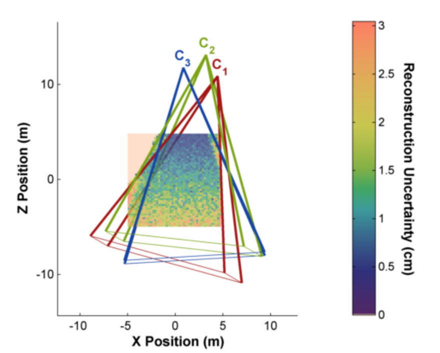
Granularity (cm): 20

Center (m) [X] [Y] [Z]: 0 0 0

Extent (m) [Width] [Length]: 10 10

Noise Model None

Display Reconstruction Uncertainty



5 References

Hartley, R. and Zisserman, A (2004). *Multiple view geometry in computer vision.* Cambridge University Press, 2nd edition.

Kwon3D: Y. Kwon, Camera Calibration - DLT Method, 1998, http://www.kwon3d.com/theory/dlt/dlt.html

6 Citing easyCamera

Users of easyWand and easyCamera are encouraged to cite Theriault et al (2014), “A protocol and calibration method for accurate multi-camera field videography”, Journal of Experimental Biology.

7 Getting easyCamera

Software packages for easyCamera, easyWand and easySBA and documentation can be downloaded from `http://www.cs.bu.edu/fac/betke/jeb/Theriault_et_al_2014.zip`

8 Software Sources and License

EasyCamera is licensed under the GNU Public License version 3 (GPLv3). The software is usable for all current, popular MATLAB platforms later than 2009a. It may work with earlier versions of Matlab. There are no MEX files to build.