

# stereoParameters

Object for storing stereo camera system parameters

## Description

The stereoParameters object stores the intrinsic and extrinsic parameters of two cameras and their geometric relationship.

## Creation

You can create a stereoParameters object using the stereoParameters function described here. You can also create a stereoParameters object by using the estimateCameraParameters with an  $M$ -by-2-by- $numImages$ -by-2 array of input image points, where  $M$  is the number of keypoint coordinates in each pattern.

## Syntax

```
stereoParams = stereoParameters(cameraParameters1,cameraParameters2,rotationOfCamera2,translationOfCamera2)
stereoParams = stereoParameters(paramStruct)
```

## Description

stereoParams = stereoParameters(cameraParameters1,cameraParameters2,rotationOfCamera2,translationOfCamera2)  
creates a stereoParameters object that contains the parameters of a stereo camera system, and sets the CameraParameters1, CameraParameters2, RotationOfCamera2, and TranslationOfCamera2 properties.

example

stereoParams = stereoParameters(paramStruct) creates an identical stereoParameters object from an existing stereoParameters object with parameters stored in paramStruct.

## Input Arguments

collapse all

paramStruct — Stereo parameters struct

Stereo parameters, specified as a stereo parameters struct. To get a paramStruct from an existing stereoParameters object, use the toStruct function.

## Properties

collapse all

### Intrinsic and extrinsic parameters of the two cameras

CameraParameters1 — Parameters of camera 1  
cameraParameters object

Parameters of camera 1 , specified as a cameraParameters object. The object contains the intrinsic, extrinsic, and lens distortion parameters of a camera.

CameraParameters2 — Parameters of camera 2  
cameraParameters object

Parameters of camera 2 , specified as a cameraParameters object. The object contains the intrinsic, extrinsic, and lens distortion parameters of a camera.

### Geometric relationship between the two cameras

✓ **RotationOfCamera2 — Rotation of camera 2**  
3-by-3 matrix

Rotation of camera 2 relative to camera 1, specified as a 3-by-3 matrix.

✓ **TranslationOfCamera2 — Translation of camera 2**  
3-element vector

Translation of camera 2 relative to camera 1, specified as a 3-element vector.

✓ **FundamentalMatrix — Fundamental matrix**  
3-by-3 matrix

Fundamental matrix, stored as a 3-by-3 matrix. The fundamental matrix relates the two stereo cameras, such that the following equation must be true:

$$\begin{bmatrix} P_2 & 1 \end{bmatrix} * FundamentalMatrix * \begin{bmatrix} P_1 & 1 \end{bmatrix}' = 0$$

$P_1$ , the point in image 1 in pixels, corresponds to the point,  $P_2$ , in image 2.

✓ **EssentialMatrix — Essential matrix**  
3-by-3 matrix

Essential matrix, stored as a 3-by-3 matrix. The essential matrix relates the two stereo cameras, such that the following equation must be true:

$$\begin{bmatrix} P_2 & 1 \end{bmatrix} * EssentialMatrix * \begin{bmatrix} P_1 & 1 \end{bmatrix}' = 0$$

$P_1$ , the point in image 1, corresponds to  $P_2$ , the point in image 2. Both points are expressed in normalized image coordinates, where the origin is at the camera's optical center. The  $x$  and  $y$  pixel coordinates are normalized by the focal length  $f_x$  and  $f_y$ .

## Accuracy of estimated parameters

✓ **MeanReprojectionError — Average Euclidean distance**  
number of pixels

Average Euclidean distance between reprojected points and detected points over all image pairs, specified in pixels.

## Settings for camera parameter estimation

✓ **NumPatterns — Number of calibrated patterns**  
integer

Number of calibration patterns that estimate the extrinsics of the two cameras, stored as an integer.

✓ **WorldPoints — World coordinates**  
 $M$ -by-2 array

World coordinates of key points in the calibration pattern, specified as an  $M$ -by-2 array.  $M$  represents the number of key points in the pattern.

▼ **WorldUnits — World points units**  
'mm' (default) | character vector

World points units, specified as a character vector. The character vector describes the units of measure.

Object Functions

toStruct	Convert a stereo parameters object into a struct
----------	--

Examples collapse all

▼ **Stereo Camera Calibration**

Specify calibration images. Open Live Script

```
leftImages = imageDatastore(fullfile(toolboxdir('vision'),'visiondata', ...
    'calibration','stereo','left'));
rightImages = imageDatastore(fullfile(toolboxdir('vision'),'visiondata', ...
    'calibration','stereo','right'));
```

Detect the checkerboards.

```
[imagePoints,boardSize] = ...
    detectCheckerboardPoints(leftImages.Files,rightImages.Files);
```

Specify the world coordinates of the checkerboard keypoints. Square size is in millimeters.

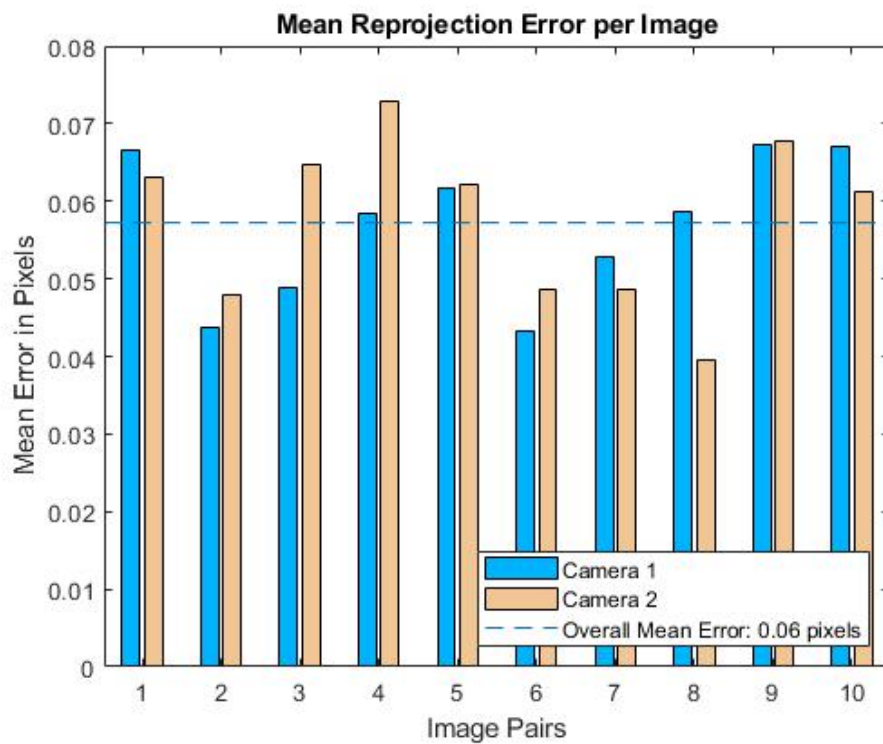
```
squareSize = 108;
worldPoints = generateCheckerboardPoints(boardSize,squareSize);
```

Calibrate the stereo camera system. Both cameras have the same resolution.

```
I = readimage(leftImages,1);
imageSize = [size(I,1),size(I,2)];
params = estimateCameraParameters(imagePoints,worldPoints, ...
    'ImageSize',imageSize);
```

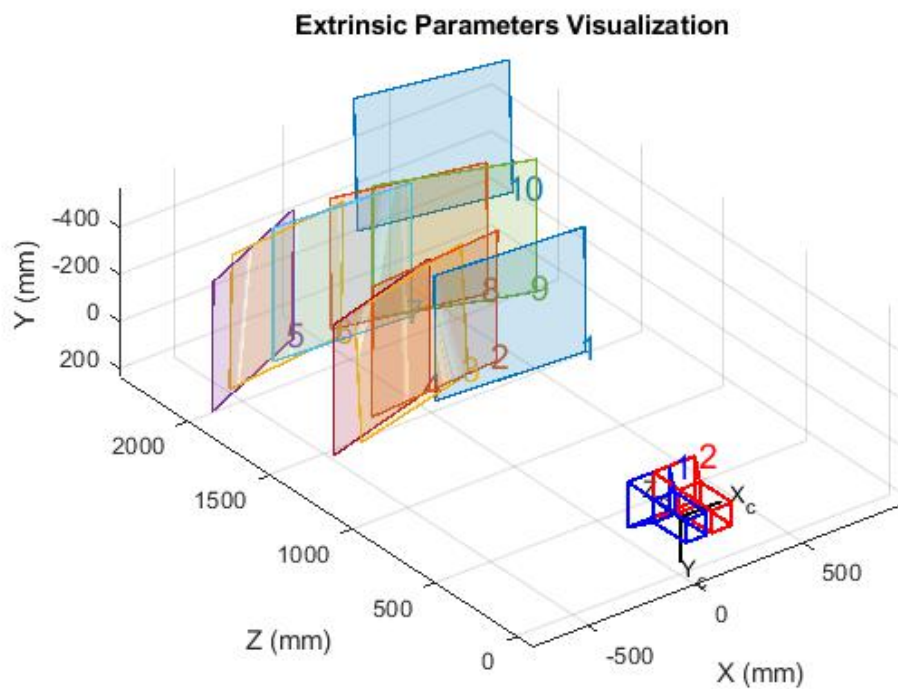
Visualize the calibration accuracy.

```
showReprojectionErrors(params);
```



Visualize camera extrinsics.

```
figure;  
showExtrinsics(params);
```



## References

- [1] Zhang, Z. "A Flexible New Technique for Camera Calibration". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 22, No. 11, 2000, pp. 1330–1334.
- [2] Heikkila, J, and O. Silven. "A Four-step Camera Calibration Procedure with Implicit Image Correction." *IEEE International Conference on Computer Vision and Pattern Recognition*. 1997.

## Extended Capabilities

## ▼ C/C++ Code Generation

Generate C and C++ code using MATLAB® Coder™.

Usage notes and limitations:

- Use in a MATLAB Function block is not supported.
- Use the [toStruct](#) method to pass a stereoParameters object into generated code. See the [Code Generation for Depth Estimation From Stereo Video](#) example.

## See Also

---

### Apps

[Camera Calibrator](#) | [Stereo Camera Calibrator](#)

### Classes

[cameraParameters](#) | [extrinsicsEstimationErrors](#) | [intrinsicsEstimationErrors](#) | [stereoCalibrationErrors](#)

### Functions

[detectCheckerboardPoints](#) | [estimateCameraParameters](#) | [estimateFundamentalMatrix](#) | [generateCheckerboardPoints](#) | [reconstructScene](#) | [rectifyStereoImages](#) | [showExtrinsics](#) | [showReprojectionErrors](#) | [undistortImage](#) | [undistortPoints](#)

### Topics

[Structure From Motion From Two Views](#)

[Structure From Motion From Multiple Views](#)

[Code Generation for Depth Estimation From Stereo Video](#)

[Single Camera Calibrator App](#)

[Stereo Camera Calibrator App](#)

---

**Introduced in R2014a**

---