Calibrating Camera Sensors – What You Need to Know

By Jacob Miller & Anas Al Badi

Cameras are one of the vital sensors for ADAS and automated driving systems. Most ADAS will require at least one forward-facing camera, but advanced safety and autonomous functions will require multi-camera systems to gain a larger field of view from various viewpoints and depth perception from stereo camera pairs. To enable the full functionality of camera systems, however, the cameras must first be calibrated.

Camera calibration is a process that will obtain parameters to describe a camera internally (intrinsically) and externally (extrinsically). These camera parameters will relate 2D image pixels to 3D points in the real world, remove distortion from images, and describe where the camera is mounted on the vehicle. Determining the exact location of cameras on a vehicle is vital to achieving stereo vision, enabling methods such as rectification and depth perception. As it becomes common for vehicles to come equipped with multiple cameras, camera calibration with be an important factor to consider during the automotive manufacturing and repair processes to ensure an accurate calibration for ADAS and automated vehicle functionality. This whitepaper provides foundational knowledge of camera calibration fundamentals, describes some commonly used calibration targets, and walks through the calibration process for mono or stereo cameras. The article also includes various tools that can be used for camera calibration and tips for a successful calibration based on VSI Labs' extensive experience with camera calibration.

Camera Calibration Fundamentals

Pinhole Camera Model

Camera models relate 3D world points to 2D projections in an image. The pinhole camera model is the most common model for cameras used today. It assumes that rays of light enter the camera through an infinitely small aperture, which is where the name pinhole comes from. The point where the light enters the camera is called the focal point, and the distance from the image plane at camera sensor to the focal point is called the focal length. Figure 1 shows an illustration of how light enters camera according to the pinhole camera model.

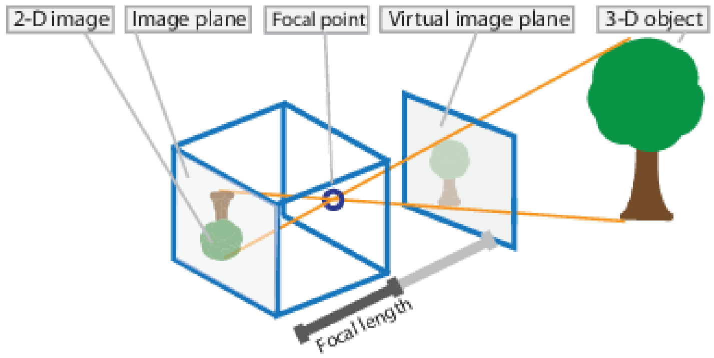

Figure 1: Rays of light, shown as orange lines, enter the camera through a small hole and project an inverted image on the image plane[1].

Lens Distortion

The pinhole camera model assumes an ideal pinhole camera and does not account for lens distortion. Distortion, however, is always present in cameras in the real world. There are three forms of radial distortion: barrel distortion, pincushion distortion, and mustache/handlebar distortion. Figure 2 shows what each type of distortion looks like on a uniform grid.

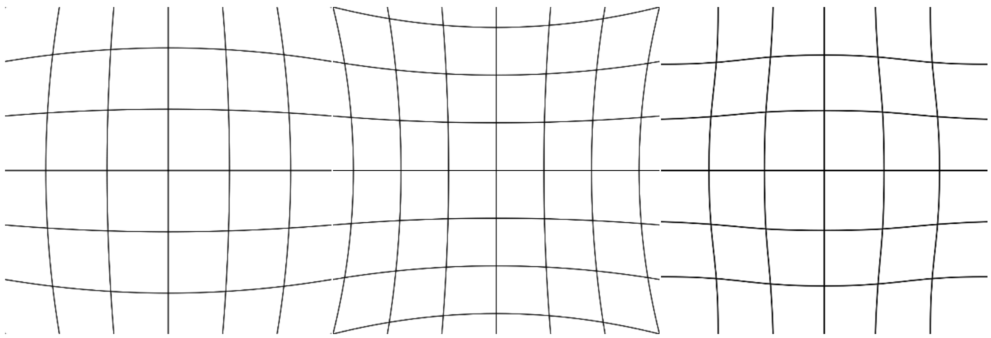

Figure 2: Barrel distortion (left), pincushion (middle), mustache/handlebar distortion (right)[2].

Cameras can also suffer from tangential distortion, which occurs when the image plane and the camera lens are not perfectly parallel. Figure 3 shows a diagram of a camera with tangential distortion. Tangential distortion is usually less significant than radial distortion and is often not necessary to account for with quality cameras.

[1] https://www.mathworks.com/help/vision/ug/camera-calibration.html
[2] https://en.wikipedia.org/wiki/Distortion_(optics)

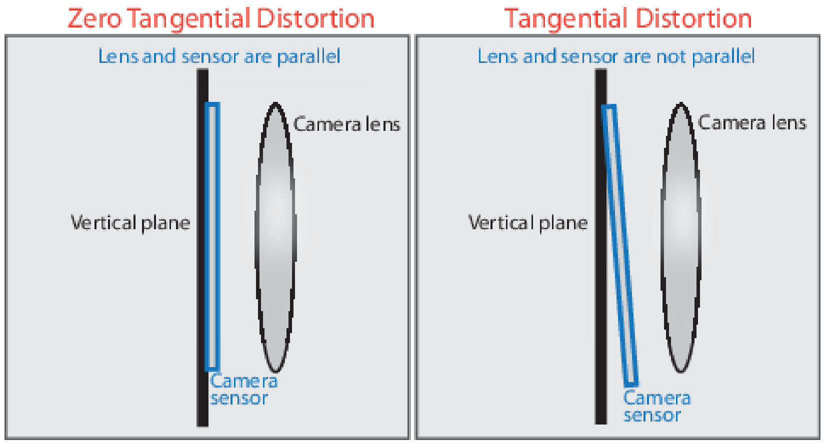

Figure 3: A depiction of a camera without tangential distortion (left) and with tangential distortion (right)[1].

Extrinsic vs. Intrinsic Calibration

Camera calibration is the process of using optimization techniques to find a set of parameters that relate the 2D image points to 3D world points. Extrinsic calibration obtains parameters to relate 3D world points to 3D camera points. These parameters include a rotation matrix and a translation vector. Intrinsic calibration obtains parameters to project 3D camera points to 2D image points. These parameters include the focal length (f_x, f_y), the principal point (c_x, c_y), and the skew coefficient (often times not included). Figure 4 shows the equation that relates an image point to a world point with the intrinsic and extrinsic parameters.

$$\underbrace{s\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}}_{\text{Image Point}} = \underbrace{\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Intrinsic Matrix}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}}_{\text{Extrinsic Matrix}} \underbrace{\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}}_{\text{World Point}}$$

Figure 4: An equation relating an image point to a world point with the intrinsic and extrinsic matrices.

Camera calibration also includes finding coefficients to model distortion. Once these coefficients are calculated, each pixel's location in the image can be adjusted to mitigate the distortion.

Calibration Targets

Calibration algorithms typically use a calibration target with known geometric shapes to optimize a camera's calibration parameters. Some commonly used calibration targets include checkerboard patterns, circle grids, and CharuCo targets. Each of these patterns will be described below.

Checkerboard Patterns

Checkerboard patterns are the most commonly used design when calibrating a camera. The checkboard pattern is easily found by binarizing the camera image and finding quadrilaterals. The corners of the checkerboard can be determined with high accuracy because they are small and neutral when the target is rotated, tilted, or distorted. When calibrating with tools such as MATLAB or OpenCV, the dimensions of the checkerboard and the size of each square are entered by the user so that the exact pattern is known by the calibration algorithm. Figure 5 shows and example of a checkerboard calibration target.

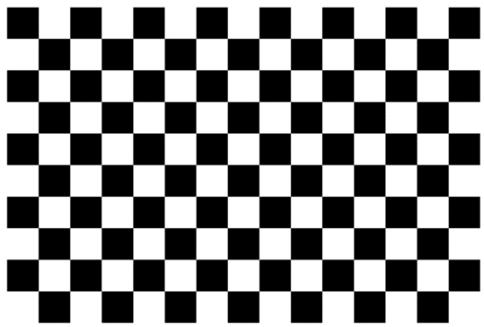

Figure 5: A checkerboard patten that can be printed and used for calibration.

The whole checkerboard must be visible in all calibration images so that all corners can be detected. It is important to include calibration images with the checkerboard near the edges of the image, where distortion is most prevalent.

Circle Grids

The circle grids are another common calibration pattern for camera calibration. The design is based on black circles with a white background or vice versa. Once the camera captures the pattern in its view, the process of extracting features from the pattern begins. It is easier to detect these circles accurately since there is no room for noise at the edges like in the checkerboard. However, the circles are imaged as ellipses in the camera point of view. These ellipses are not imaged perfectly and thus adds complexity to the already existing distortion from the camera lens. Figure 6 shows an example of a circle grid calibration target.

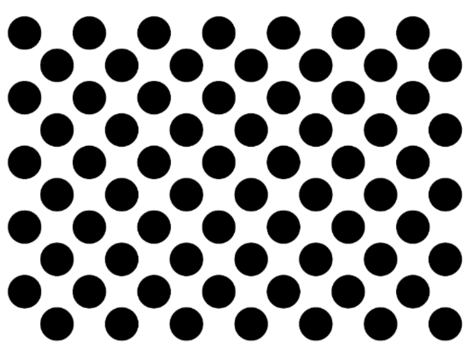

Figure 6: A circle grid pattern that can be printed and used for calibration.

CharuCo Targets

CharuCo targets require a complex algorithm for detection but they can outperform the classical checkboard for camera calibration. The white checker fields in the CharuCo targets have their own identifiable code which helps improve the calibration significantly. Because some of the light that hits the target might not hit every field equally, detection failures occur more frequently with the more complex patterns. However, the camera only needs to capture part of the CharuCo target since each square has its own identifiable code, making it easier to capture calibration images around the edges of the camera's field of view. Figure 7 shows an example of a CharuCo calibration target.

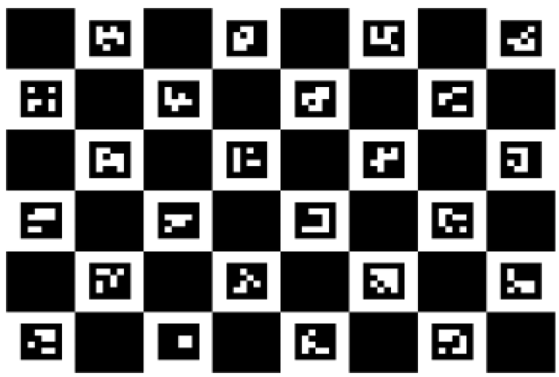

Figure 7: A CharuCo *pattern that can be printed and used for calibration.*

Calibration Process

Most camera calibration techniques follow the same general procedure:

1. Take pictures of the calibration target in various positions and tilts.
2. Detect or label the features of the calibration pattern in each image.
3. Compute the intrinsic, extrinsic, and distortion parameters using an optimization algorithm.

Calibration Tools/Libraries

There are several calibration tools and libraries that make camera calibration easy by automating most of the process. The most popular of which are included in OpenCV and MATLAB.

OpenCV is an open source code library for computer vision. It includes functions to detect calibration patterns, perform camera calibration, display images, among other image processing techniques. Using OpenCV for camera calibration requires some programming knowledge in C++ or Python to implement the calibration procedure.

MATLAB's Computer Vision Toolbox contains tools to calibrate a single camera or stereo cameras. Calibration can be done by calling functions in a script or by using their GUI. The GUI makes it simple to achieve an accurate calibration, even with no knowledge of programming. It shows the detected checkerboard pattern and the reprojected points on the image, allowing the user to make sure the detections are accurate. It also shows the reprojection errors of each calibration image, making it easy to remove outliers. MATLAB's Camera Calibrator is limited to detecting the checkerboard pattern, but custom MATLAB scripts could be written to detect other calibration patterns. Figure 8 shows the GUI for MATLAB's Camera Calibrator.

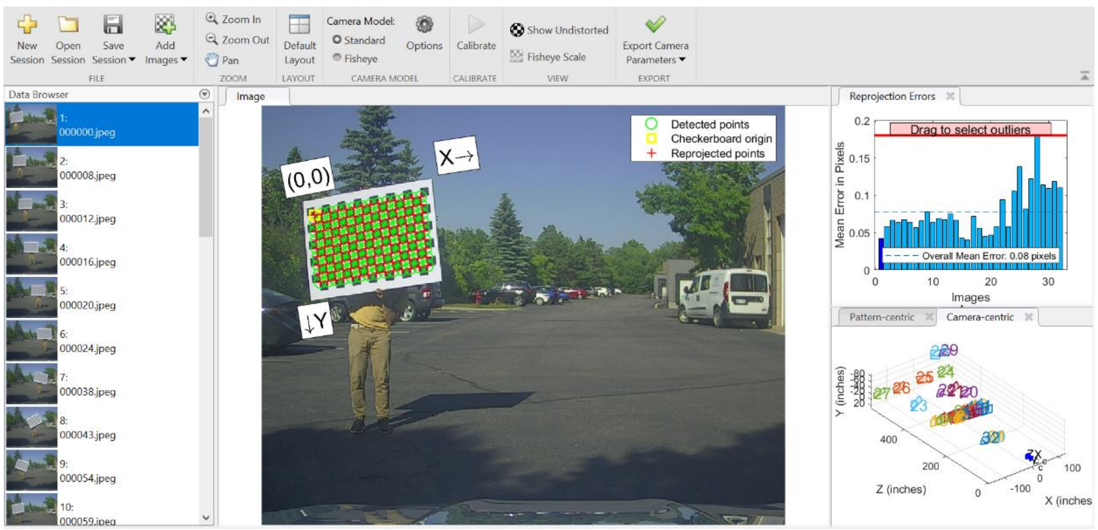

Figure 8: A screenshot of the MATLAB Camera Calibrator.

Thermal Cameras

Thermal cameras can be calibrated with the same general procedure as visible cameras but require a few additional steps and considerations. Listed below are some key observation from VSI Labs' extensive experience in calibrating thermal cameras:

- The calibration target should be made of a material with low thermal conductivity and high thermal capacity. This allows the calibration target to heat up some parts of the pattern while keeping other parts cool.
- Use the sun or a heat lamp to heat up black patterns on the calibration target while gathering calibration images.
- In the thermal image, the black patterns will appear white (indicating hot temperature) and the white patterns will appear black (indicating cold temperature). Functions included in OpenCV and MATLAB will look for a white border to detect the checkerboard pattern, which will appear black in the thermal image. Because of this, either the detection algorithm needs to be adjusted to accommodate thermal images, or the thermal images need to be

inverted. Figure 9 shows an inverted thermal image used by VSI Labs for calibration.

- Pick a calibration target with larger features. Thermal cameras usually have lower resolution, so small feature will be more difficult to detect in a thermal image.

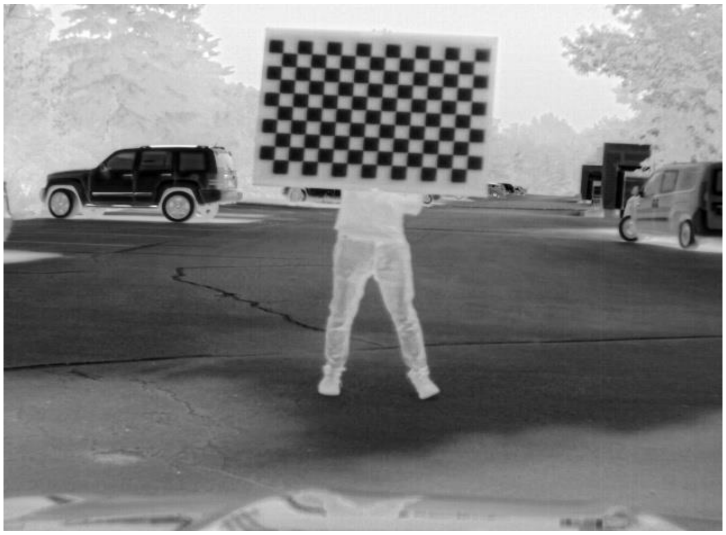

Figure 9: An example calibration image for a thermal camera.

Tips for a Successful Calibration

It oftentimes requires some trial-and-error to obtain an accurate calibration. Listed below are a number of tips for a successful calibration that VSI Labs compiled from research and experience with camera calibration:

- Calibration images should include the calibration target in various positions and tilts. Take enough calibration images to cover the full field-of-view of the camera, making sure to include the corners.
- The calibration target should cover close to half of the camera's field-of-view in each calibration image.
- The calibration target should be completely rigid and flat with sharp edges and corners for the features.
- Calibration images should have the calibration target placed at the working distance of the final application.
- Use a target with as high of a feature count as possible. More features provide more data for the optimization algorithm but can be more difficult to detect.
- Make sure the calibration target is properly illuminated. Shadows or poor lighting will make feature detection more difficult.
- Observe and remove bad calibration images. Calibration images that mis-detect the features or have a high reprojection error can hurt the calibration

results. Tools such as MATLAB's Camera Calibrator make this easy by showing the feature detections and reprojection errors on each image.
- If the camera is being extrinsically calibrated with other cameras or sensors, make sure the camera is properly mounted. If the camera moves from this position, it will need to be recalibrated.

Stereo Calibration

Multi-camera systems are common for ADAS and a requirement for higher level autonomy. Using multiple cameras offers several advantages over a single camera system, including a larger total field of view from different viewpoints, sensor redundancy, and better depth perception using triangulation. To take full advantage of multi-camera systems, stereo calibration is performed to obtain the intrinsic calibration of two cameras and the extrinsic calibration relating the positions of the two cameras.

The stereo calibration process is similar to that of a single camera, with some extra considerations. OpenCV and MATLAB have tools to automate most of the stereo calibration process, similar to the tools described above. The basic process is outlined below:

1. Mount the two cameras securely with an overlapping field of view. The separation distance for the two cameras will depend on the final application.
2. Take pictures of the calibration target in various positions and tilts. The pictures should be taken simultaneously for both cameras.
3. Detect or label the features of the calibration pattern in each image for each camera.
4. Compute the intrinsic and distortion parameters for each cameras and the extrinsic parameters relating the two cameras using an optimization algorithm.

Separation Distance

The ideal separation distance for a stereo camera system can be calculated with the following equation[3],
, where near point and far point describe the closest and farthest distances required for an accurate stereo correspondence:

$$Separation = depth\ factor * near\ point * \frac{far\ point}{far\ point - near\ point} * \frac{focal\ length\ of\ viewing\ lens}{focal\ length\ of\ camera\ lens}$$

In most stereo camera systems for autonomous vehicles, the following assumptions can be made to simplify the equation:

- Assume a depth factor of 1/30 (most commonly used).

[3] https://www.berezin.com/3d/Tech/lens_separation_in_stereo_photog.htm

- Assume the far point is much larger than the near point. This will cause the far point factor will converge to 1.
- Assume the two cameras have a similar focal length. This will mean the focal length factor is close to 1.

Using these assumptions, the ideal separation distance for a stereo camera system can be estimated by the following equation:

$$Separation = \frac{1}{30} * near\ point$$

Conclusion

Proper camera calibration is a must-have for ADAS and automated vehicle systems to ensure the correct functioning of camera perception systems. Camera calibration is especially important to take advantage of stereo images techniques for multi-camera systems, including rectification and triangulation for depth perception. Camera calibration can appear to be a daunting task, but the knowledge, tools, and tips from this Tech Brief make camera calibration a more manageable process.

About VSI Labs

Established in 2014 by Phil Magney, VSI Labs is one of the industry's top advisors on AV technologies, supporting major automotive companies and suppliers worldwide. VSI's research and lab activities have fostered a comprehensive breakdown of the AV ecosystem through hands-on development of its own automated vehicle platform. VSI also conducts functional validation of critical enablers including sensors, domain controllers, and AV software development kits. Learn more about VSI Labs at https://vsi-labs.com/.

How to Engage with VSI

VSI Labs offers subscription research packages to meet your needs:

- VSI Insights – High level technical analysis of CAV technologies and the future of automated driving.
- VSI CAV Technology Databases – Deep insights on the products and technologies that make up automated driving.

Learn more about our portal services or contact us to get started!

Conditions of this Report

While every effort has been made to ensure the quality and accuracy of the information provided, Vision Systems Intelligence LLC (VSI), its personnel, agents, or representatives, assume no responsibility as to the accuracy or completeness of and, to the extent permitted by law, shall not be liable for any errors or omissions or any loss, damage, or expense incurred by reliance on any information or statement contained herein. VSI makes no warranty, expressed or implied, as to the accuracy, completeness, or timeliness of any information in this document, and shall not in any way be liable to any recipient for any inaccuracies or omissions.