Applications of Camera Calibration: Stereo Vision

Camera calibration has many applications in the field of machine vision. This article will focus on how camera calibration is used for stereo vision. The PixelTraq calibration approach is an excellent way to obtain the necessary calibrations for implementing a custom stereo vision application. Please see our other articles for more background on camera models, camera calibration, or for more details on the PixelTraq process.

Stereoscopic Camera Depth

Camera calibration is essential for stereo depth or stereoscopic vision. Stereo vision, sometimes referred to as binocular vision, refers to the use of two cameras to reconstruct 3D information. Because projection into the image plane of the camera reduces the dimensionality from 3D to 2D, we lose the ability to reason 3D information about an unstructured scene observed by a camera with only a single view. By adding an additional view of the same object from a different perspective, we can reconstruct the 3D information.

One simple way to think of this is that a single point in 3D can be represented as three unknowns, the X,Y, and Z coordinates of the point. With a single camera view of the point we get two equations for those three points, the equations for the u and v pixel coordinates. We have two equations and three unknowns, so at best we can only solve for a one parameter family of solutions with these two equations. This is why an inverse camera model, or back projection, produces a ray which represents a one parameter family of points along the ray direction. By adding the second camera view, we now have four equations and three unknowns. So we can now solve for the 3D coordinates by inverting the non-linear camera equations.

In a geometric sense, this is the same concept as triangulation of the point. Both rays intersect at a point in 3D space. Back-projection using the camera model allows us to get the angle of each ray and the baseline, or pinhole-pinhole distance of the two cameras allows us to find the depth at which the ray intersect.

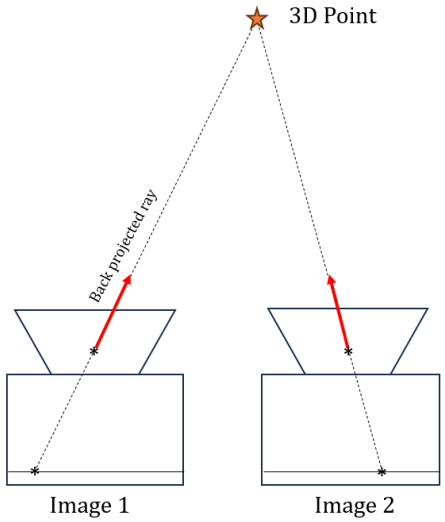


Figure 1Triangulation of a 3d point from image points

There are two common types of stereo vision, point (feature) based stereo which operates on individual points in the image (much like the point in the previous example) and area (dense) based stereo which operates on the entire image [1].

Point Based Stereo Vision

Many applications do not require a full depth image reconstruction like the ones created by area based stereo methods. In some applications, much higher accuracy can be achieved by computing point based stereo. Image processing algorithms can identify the same point across two camera views by detecting "features". Features are points that can be accurately identified by image processing algorithms. When we have identified the same feature in two images, we call this a "point correspondence". We can use the camera models of each camera to reconstruct the 3D point from point correspondence.

The first step in this process is to use the inverse model of each camera to get the two 3D rays.

$$\widehat{X_L} = f^{-1}(u_{*L}, v_{*L})$$

Here, $\widehat{X_L}$ is the ray projected from the left camera, f^{-1} is the inverse camera model function, and u_{*L} and v_{*L} are the x and y pixel coordinates respectively in the left image frame. The same is possible for the right camera.

Next, we need to convert the rays to be represented in the same reference frame. Using the PixelTraq process we get the extrinsics of each camera relative to a fixed reference frame. Let's call this frame "d" for the datum reference frame. The extrinsic parameters of each camera relative to d are $(R_{dL}.p_{dL})$ and $(R_{dR}.p_{dR})$. For simplicity, we will convert each ray into the d reference frame from their respective camera reference frame.

$$\widehat{X_{L_d}} = R_{dL}\widehat{X_{L_L}}, \qquad \widehat{X_{R_d}} = R_{dR}\widehat{X_{R_L}}$$

These rays, combined with the extrinsic positions of each camera can be represented using the 3D parametric equation of a line.

$$\overrightarrow{r_L(t)} = \widehat{X_{L_d}}t + p_{dR}$$

$$\overrightarrow{r_L(t)} = \widehat{X_{L_d}}t + p_{dR}$$

In 2D, finding the intersection of two lines is simple. In 3D however, lines rarely intersect. Due to image processing noise, these two lines will almost certainly not intersect. Even so, there are ways to find the best approximation of the intersection point of these two lines. One such way is to find the shortest line segment between the two lines and take its midpoint. This line segement has the unique property that is is the only segment that is mutually orthogonal to both lines [2]. There are many common algorithms for computing the intersection point of two rays. Taking this point we have now found the best approximation of the 3D point using point based stereo vision.

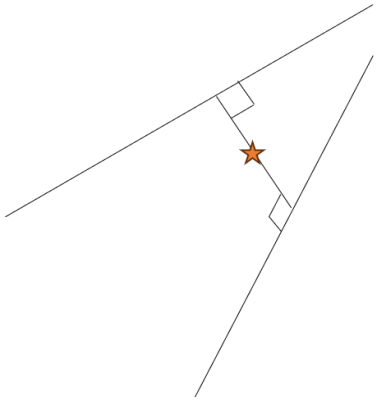


Figure 2 Triangulation in 3D

There are many applications for point based stereo vision. One common use case is point cloud registration. 3D point clouds from methods like area based stereo are often very challenging to overlay

across subsequent views. By extracting point features from the images used to generate the point clouds and their correspondences in subsequent views, the point clouds can be registered to high accuracy without the need to use complex algorithms that operate on the point cloud itself. Another area where point based stereo is used is in dynamic motion capture. A common approach to motion capture is to instrument a device or person with spherical reflectors and capture stereo images of the reflectors. Point features can be extracted from images of the reflectors and the motion of the object can be computed.

Area Based Stereo Vision

In some applications, rather than computing the 3D point associated with individual image points, we would like to compute a depth map for the entire image, or a "depth image". This is a much more challenging problem than computing depth from point features. The typical process of computing area based depth from stereo image pairs consists of the following steps:

- Image undistortion
- Image rectification
- Stereo matching (disparity computation)
- Depth computation

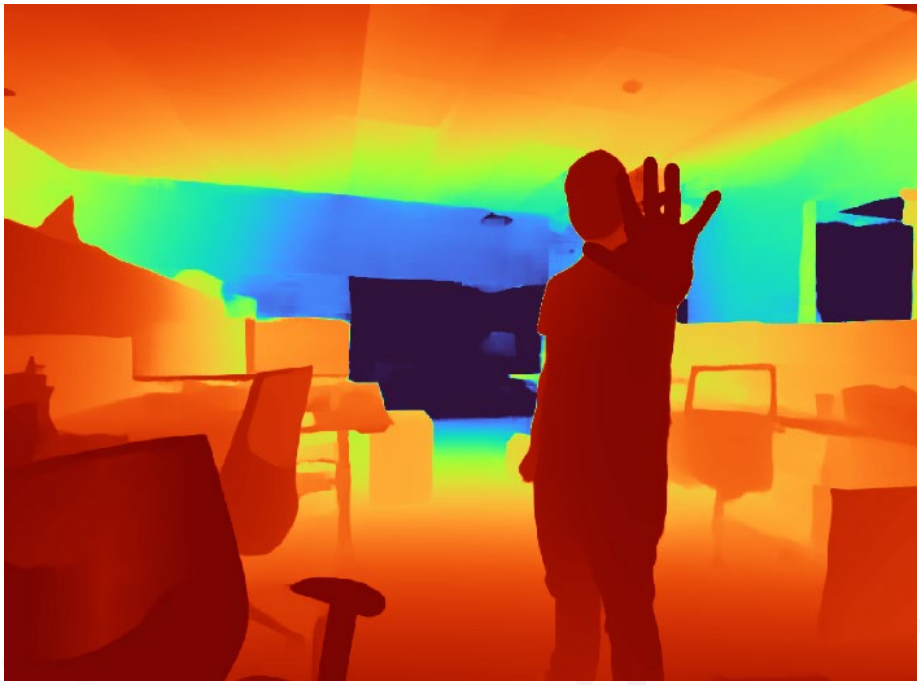


Figure 3 Depth image computed from area based stereo

Step 1: Image Undistortion

The first step of area based stereo depth is to remove distortion from the images. This can be done using a well calibrated camera model such as camera calibrated using the PixelTraq system. Image undistortion typically maps a camera from a distorted camera model such as a Kannala or Brown Conrady style model to a Pinhole model. The pinhole model is desirable for stereo depth computation because it follows a simple geometric projection. The structure of the pinhole model is exploited for the depth computation step of area based stereo.

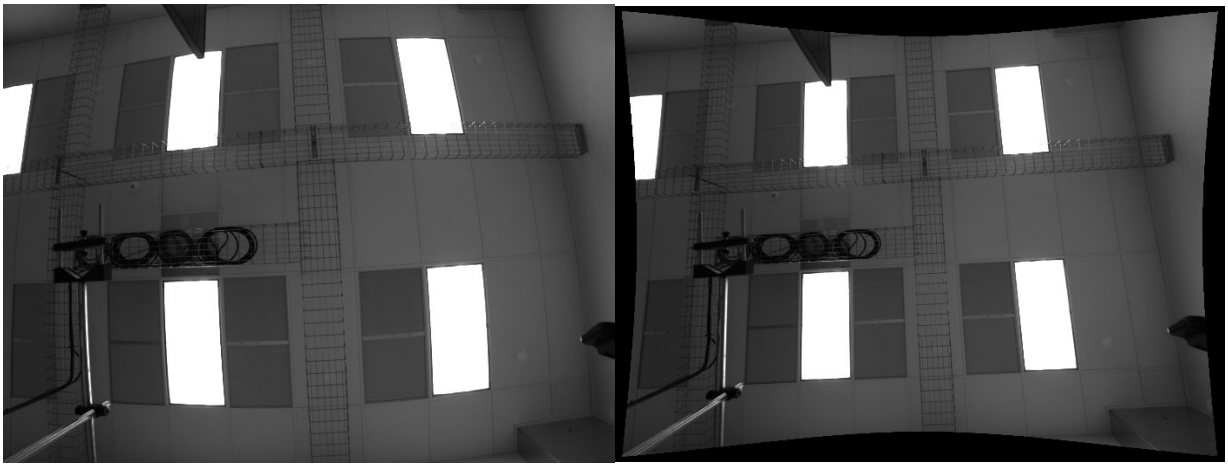


Figure 4 Distorted image (left) undistorted image (right)

Step 2: Image Rectification

The next step of the area based stereo depth is to perform image rectification. Image rectification is necessary to compute disparity. The idea behind disparity is that for two pinhole cameras with perfectly aligned image planes, the depth is inversely related to the difference in pixel position that the feature appears in the two images. With simple geometry, we can derive the relationship shown below.

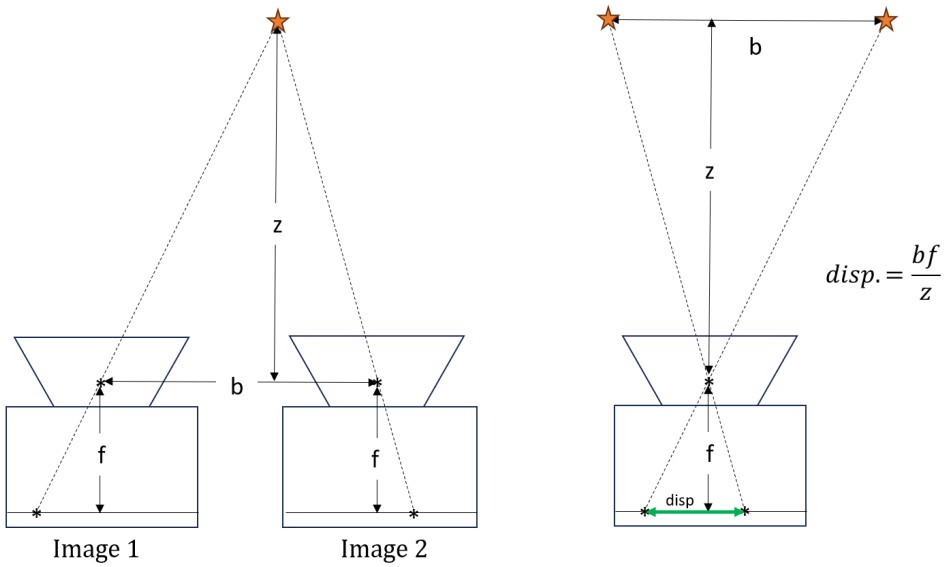


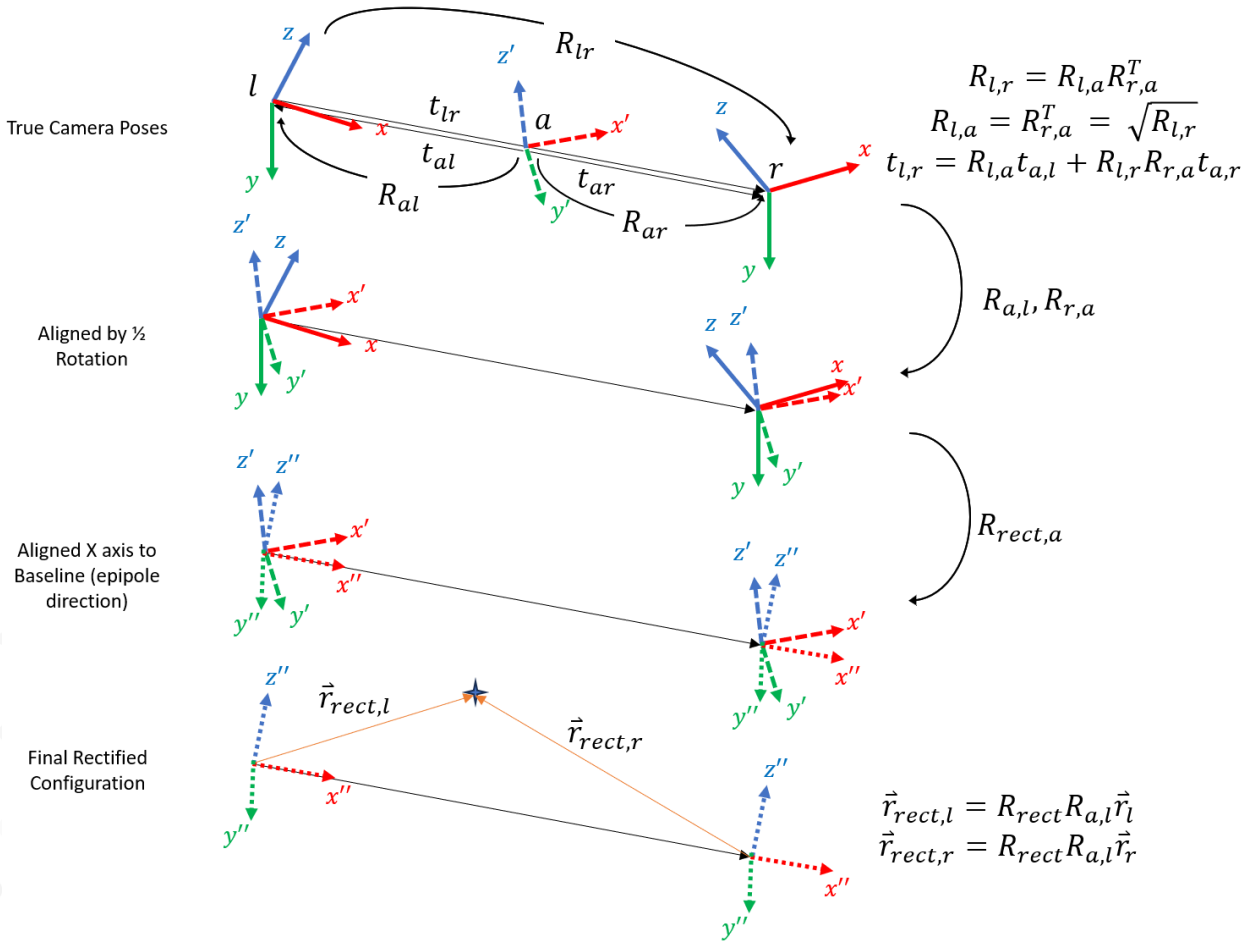
Figure 5 Derivation of disparity by similar triangles

Even with two calibrated and undistorted camera images, we do not have two perfectly aligned image planes though. This is where image rectification is used. Through rectification, we can realign the cameras to ensure that disparity can be calculated along the same rows of pixels between the two cameras.

A common process for computing the rectified stereo images for calibrated cameras is known as Bouguet's Algorithm [3].

This process computes the rotation matrix for each camera in the stereo pair that can rotate a ray from the true camera into a corrected camera view that satisfies the disparity depth relationship. The steps to perform this process are:

1. Compute the rotation that rotates each camera to half of the original rotation from the extrinsic parameters, $R_{a,l}$ and $R_{a,r}$. This allow an incoming ray to be rotated into an aligned frame from either camera. This is not enough alone though, since the x axis of the new frame is not necessarily aligned to the baseline of the cameras so the epipolar lines of the images are not parallel to the image x axes which is required for simple disparity calculation.
2. Compute the rotation that rotates the frames such that the epipolar lines are parallel and aligned to the baseline and image x-axes, R_{rect}.



Obtaining the matrix R_{rect} is the most complex part of this process. Taking $t = R_{a,l}t_{lr}$, the baseline in the aligned camera frame

$$e_1 = \frac{t}{\|t\|}$$

$$e_2 = \frac{[-t_y t_x 0]^T}{\sqrt{t_x^2 + t_y^2}}$$

$$e_3 = e_1 \times e_3$$

$$R_{rect} = \begin{bmatrix} e_1^T \\ e_2^T \\ e_3^T \end{bmatrix}$$

This matrix can be thought of as the rotation that transforms rays in either aligned camera frame into the rectified camera frame.

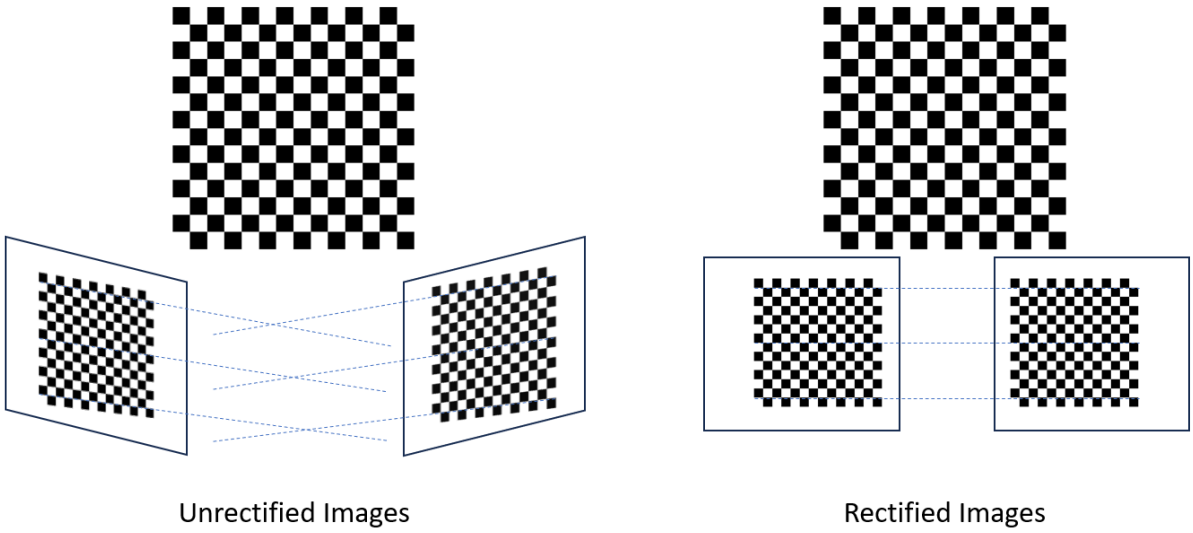


Figure 6 Illustration of unrectified vs rectified images

One thing to note is that it is possible (and often more efficient) to perform rectification at the same time as undistortion is performed by rotating the rays before projecting them through the target undistorted model.

Step 3: Stereo Matching

Following rectification, the next step is to match points of objects in the left frame with the same points on objects in the right frame. In many respects, this is the most challenging aspect of the area based stereo depth computation. This is commonly referred to as the "stereo matching" problem. Many academic papers have been written on this subject with algorithms ranging from simple to highly complex [1] [4] [5]**.** The state of the art methods for stereo matching use artificial neural networks to generalize over complex scenes where more traditional methods often fail. At the date of this writing, this remains a topic of much research interest.

Detailed description of these algorithms is outside of the scope of this document, but many of their implementations are available as open source algorithms with detailed documentation. Often, the specifics of the stereo matching algorithm deployed depend on the speed, resolution, computational power, and accuracy requirements of the application. As such, the choice of algorithm can vary significantly across applications.

Regardless of the implementation of stereo matching, the output is a disparity image that can be readily converted to a depth image as shown in the following section.

Step 4: Depth Computation

Finally, the disparity image is converted to a depth image using the relationship

$$z = \frac{bf}{disp.}$$

for each pixel in the disparity image. It is important to note that this equation is only valid if both camera models used as the undistortion target camera models have the same focal length. Because of this, it is typical to undistort incoming images to a target model that has the same focal length for both cameras.

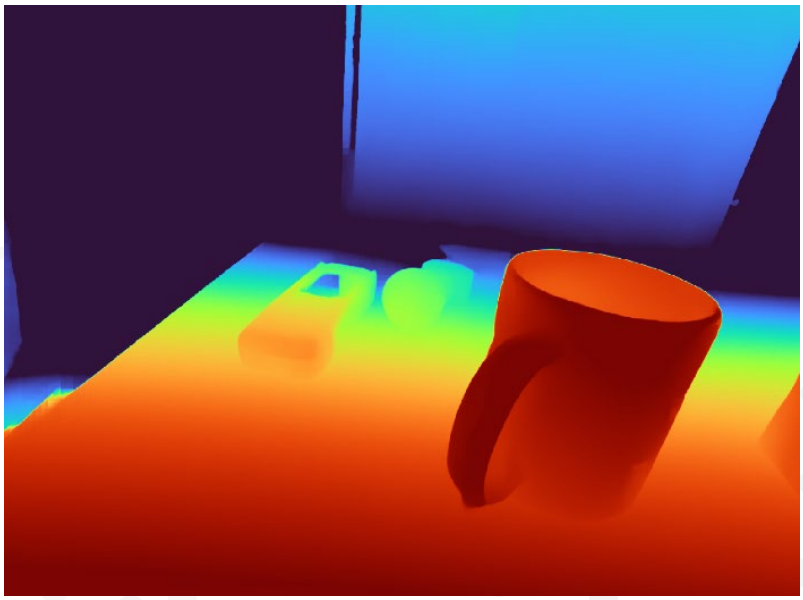


Figure 7 Resulting depth image for area based depth

Generating Point Clouds using Stereo Vision

A common use case for the area based stereo vision workflow described in the previous section is to generate a 3D point cloud. With a 3D point cloud, it is possible to make 3D measurements using the two camera views.

Once the area based depth image is obtained, the final step is to propagate the rays back into 3D space.

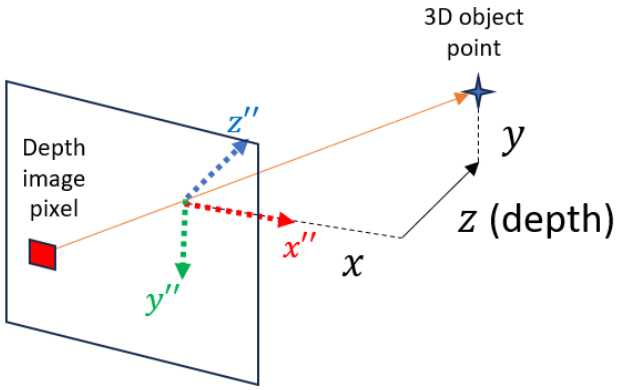


Figure 8 Mapping from Depth image pixel to a 3D object point

Taking the pinhole model used to undistort the images, the rays for each pixel in the depth image can be scaled such that the z coordinate is equal to the determined depth. The resulting coordinates $\vec{p}$ are the coordinates of this point with respect to the frame of the camera that the depth image is measured in.

$$\vec{p} = \frac{\vec{r}}{r_z} depth$$

Figure 9 Example of a 3D point cloud reconstruction generated using PixelTraq calibrated cameras

Common Challenges in Stereo Vision

There are certain aspects of the stereo depth computation that have a high likelihood of introducing inaccuracies into the results. Some of the most common ones are:

- Feature correspondence (stereo matching) errors
  - Inaccuracies in stereo matching will introduce disparity errors that affect the computed depth map accuracy. These can typically be addressed by using an algorithm that is known for performing well for the type of subject matter to be observed.
- Inaccurate camera calibration: impacts image undistortion and rectification
  - Incorrect knowledge of the stereo baseline, relative camera orientation, or image distortion will result in errors that degrade the quality of the depth image results especially for 3D measurement applications where the depth image will be used to compute a 3D point cloud. One way to address this is to use a high quality calibration process such as the PixelTraq process.

References

[1] K. Konolige, "Small Vision Systems: Hardware and Implementation," 1998.

[2] P. Bourke, "Points, lines, and planes," 1998. [Online]. Available: https://paulbourke.net/geometry/pointlineplane/.

[3] G. Bradski and A. Kaehler, Learning OpenCV, O'Reilly Media, Inc, 2008.

[4] H. Hirschmuller, "Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual," *IEEE Conference on Computer Vision and Pattern Recognition,* 2005.

[5] L. Lipson, Z. Teed and J. Deng, "RAFT-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching," in *2021 International Conference on 3D Vision (3DV)*, London, United Kingdom, 2021.

Glossary

Epipolar lines – For a stereo system, a point viewed along a ray from the perspective of one camera traces a theoretical line in the image of the other camera for perfectly aligned image planes, these lines become parallel

Features – aspects of an image that can be extracted to identify points uniquely across images

Point correspondence – a pair of image points, one from each camera, that represent projections of the same feature in the scene

Rectification – a realignment of a camera's perspective for stereo vision that reorients the incoming rays such that both cameras are aligned to the each other and the camera's baseline allowing the computation of disparity

Stereo matching – the process of matching points between two stereo images across the entire image for the computation of disparity

Stereo vision – using a pair of calibrated cameras to reason 3D information such as depth

Undistortion – a remapping of an image's coordinates to an alternative projection model that removes undesirable distortion from the image