# Annotation of Calibration Patterns for RGB-LiDAR Evaluations using Segmentation Models

**BRUNO SILVA[1],PhD. Student**
**GONÇALO RIBEIRO[1],PhD. Student**
[1]Department of Mechanical Engineering, University of Aveiro

**ABSTRACT** Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

**INDEX TERMS** Extrinsic Calibration, Robot Calibration, Calibration Pattern, Machine Learning

## I. INTRODUCTION

Extrinsic calibration is a fundamental process in robotics vision that involves determining the relative pose (position and orientation) between different sensors, known as *sensor to sensor calibration*, or between a sensor and a known reference frame, which is known as *sensor to coordinate frame*. This process is crucial because it allows for the accurate integration of data from multiple sensors, enabling sensor fusion. For instance, in an autonomous vehicle, the visual information of the camera needs to be accurately aligned with the distance measurements of the LiDAR to build a coherent understanding of the surroundings. Similarly, in robot arms, the position of the camera position relative to the end-effector must be precisely known to perform tasks like object manipulation.

Usually, iterative approaches are used. These rely on a cost function specific to a sensor modality but usually suffer from ambiguities in some way or another. An typical cost function for an RGB camera relies on computing the difference between the projection of the detection in the 2D image of some key points into a coordinate frame where these key points are precisely known. The objects that contain these precisely known points are called *Calibration patterns*. The most common types are chessboards and *ChArUcos*. *ChArUcos* are chessboards with little unique identifiable symbols that allow computer vision algorithms to decipher if the pattern is upside down or if the framing of the image cuts them off. The issue with the aforementioned RGB cost function is that they have multiple local minima and not always converge. A simple example is to picture a RGB camera pixed on the end of a prismatic joint with the calibration pattern in front of the sensor, perfectly perpendicular to it. Both moving the offset of the joint or moving the RGB sensor on the mount can lead to the same relative distance between the sensor and the pattern, thus leading to ambiguity.

Despite the shortcomings mentioned, these cost functions have the advantage of generally performing more accurately in larger systems, as there is much more variety of data, and can still be used effectively in most simpler systems, as their requirements are only the existence of a sensor and a pattern. However, this creates the need for true evaluation procedures to assess the quality of the calibration results and to make them comparable in order to be publishable.

This project tackles a improvement to a previously fully manual and cumbersome evaluation method between RGB and LiDAR sensors, integrated on ATOM, a well established multi-sensor multi-modal calibration framework. The working principle is that by knowing the physical outer limits of the pattern in the 2D image, these points can be projected into the coordinate frame of LiDAR sensor, provided that the camera intrinsics are known. Afterward, the 3D points resulting from the projection can be directly compared with the 3D points of the outer border of the pattern. This comparison is not ambiguous as the projected points only line up in the 3D frame if the geometric transformations required for the projection are indeed correct. In ATOM, the 3D border points are already priorly labeled as they are a requirement for the cost function that optimizes the pose of these sensors. However, the 2D points of the border are not required by the RGB cost function. The current solution is a manual corner labeling method, as an automatic method struggles with detecting orientation and deal with edge cases. This project aims to develop an automatic method to simplify the evaluation pipeline to the user.

## II. PROPOSED APPROACH

### A. DATA PREPARATION

The data we had: - Images with corners labeled

Captured more images to have more dataset diversity with other types of calibration patterns

Developed a annotation program to label corners of the patterns. Developed a program to convert labeled corners into segmentation masks that served later to train the models.

### B. WHY USE SEGMENTATION MODELS

Using a more simple CNN/FC combo to find just the coordinates of the corners of the pattern was not feasible as the size of the output is not constant and a neural network with dynamic outputs is much more complex. More often than not, at least one corner of the pattern is clipped in the image. If one corner is clipped, 5 points are instead needed to draw lines on the 4 sides of the pattern. If 2 adjacent corners are clipped, only 3 sides are visible. The solution we found to answer this problem is to find the segmentation mask of the pattern instead and compute the edges afterward with classical computer vision.

### C. TRANSFER LEARNING WITH DEEPLABV3 AND RESNET50 BACKBONE

Using transfer learning with a pretrained DeepLabv3 model was the first option considered. DeepLabV3 is a complex model designed for image segmentation. Including the Resnet50 backbone, which is a CNN, the model has in total around 41M parameters. Upon freezing the backbone, we were left with 17M trainable parameters. The pretrained model came from PyTorch built-in models and was originally trained on a subset of known COCO dataset, using only the 20 categories that are present in the Pascal VOC dataset, mainly consisting of big every objects like cars, planes, sofas, bicycle...

The train was conducted using the Adam optimizer and cross entropy loss, batch size of 4 and 10 epochs. Our dataset had 80 images. We found the loss to stay constant and high, meaning our model was not learning anything from the training. We figured that the dataset that the model was originally trained on was too different from ours and the model could not learn within a feasible number of epochs with only 80 images. Our next attempt was to try a simpler segmentation model and train it from the ground up.

### D. U-NET TRAINED FROM SCRATCH

Training the previous model from scratch would require a lot of compute resources and probably be overkill for the desired task. As such, we decided to try a simpler model and train it from scratch. Unet is blablabla

Good results

### E. U-NET WITH PRETRAINED RESNET50 BACKBONE

Better results

## III. CONCLUSION

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.