EE 5450 Project 01: Simultaneous Monocular Calibration and Pose Estimation

David R. Mohler

February 20, 2018

1 Introduction

Determination of the pose of rigid objects within images is a common problem within many practical application of computer vision, robotics, computer graphics, etc. The following set of experiments are focused on the application of the simultaneous monocular calibration and pose estimation algorithm to a series of images of rigid objects with known dimensions and correspondence points. From this known data, the algorithm is capable of calibrating a single monocular camera with a fixed focal length. The calibration enables simultaneous discovery of the pose of the viewed object, and from this generate a three dimensional representation of that same object.

2 Methods and Results

The foundation of this project is based on the known dimensions of a given rigid object, a box for example, which is manually assigned a number of points on the object corresponding to its corners (i.e. the coordinates in the object frame represent the dimensions of the object). A model of the initial box and its correspondence points can be seen in Figure 1, additionally, the measured dimensions of the object are shown in Table 1. Given this data we are able to proceed with the implementation of the monocular pose algorithm.

Using the object coordinates in their homogeneous form (i.e. $X_o^i = [x_o^i \quad y_o^i \quad z_o^i \quad 1]^T$) we begin with calculating an approximation of the projection matrix, Π_{est}. The projection matrix is such that $\Pi = [KR \quad KT] \in \Re^{3\times4}$, where K is the camera calibration matrix, R is a rotation matrix, and T is the translation vector. Given that we know the location of the correspondence points, χ^{pj}, and their matching locations in the object frame, X_0^j, we are able to .This approximation can be expressed as a least squares minimization of the

Parameter	Value (cm)
Length	45.6
Height	32.5
Width	10.1

Table 1: Object Dimensions

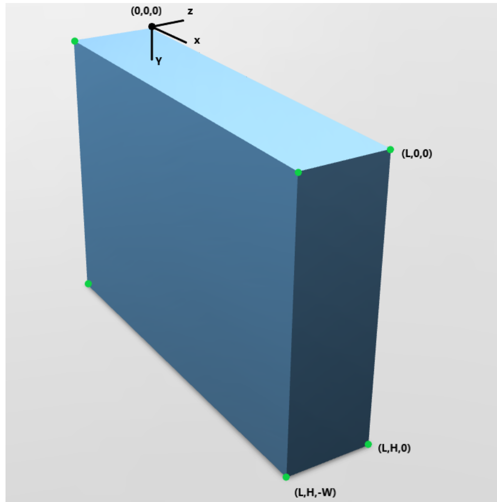

Figure 1: Rigid object model.

$$\chi^{pj} = \frac{\Pi_{est} X_0^j}{\lambda^j} \tag{1}$$

2.1 Corrupted Correspondence Points

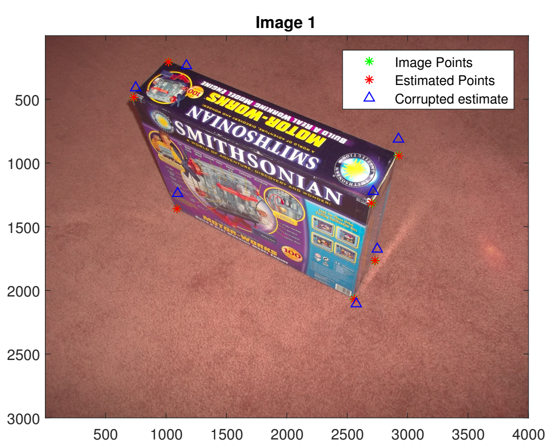

Figure 2: Estimated image coordinates

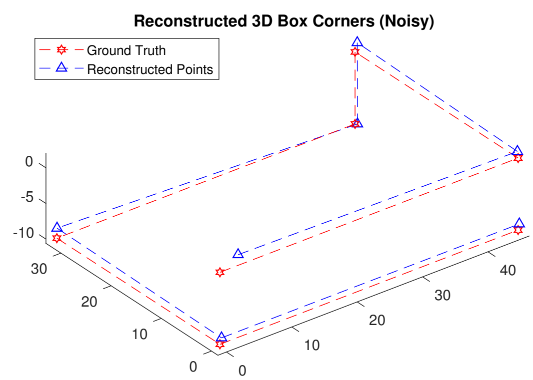

Figure 3: Reconstructed box relative to ground truth

2.2 Improper Dimensions

3 Conclusions

In this project we have demonstrated two major methods critical to estimation and uncertainty quantification. The first being a method to sample from any arbitrary closed

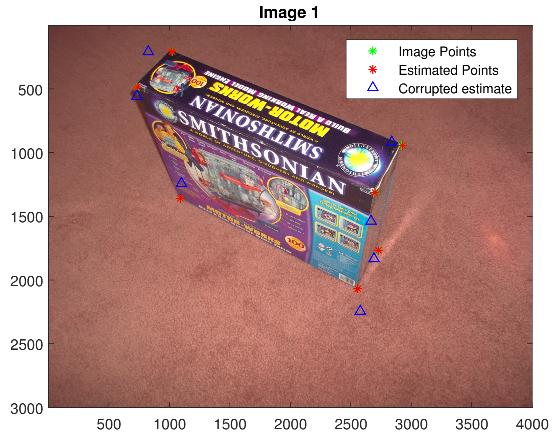

Figure 4: Estimated image coordinates (unique coordinate component noise)

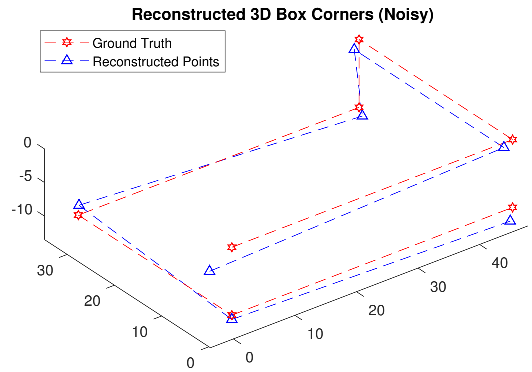

Figure 5: Reconstructed box relative to ground truth

functional form of a continuous probability distribution, and the second being the application of the Expectation Maximization algorithm to a probability mixture model. For our implementation the rejection sampling method is essential to the proper operation of the EM algorithm. Rejection sampling provides a simple method for artificial data generation. By varying the number of data points that we wish to sample from our known functional form of the distribution we are able to view the accuracy of the sampled PDF relative to the true curve. It is noted that the number of data points required to generate an "accurate" curve is application and computational burden dependent. For the purposes of this project we found that samples ranging in the low hundreds did not provide enough data to create a decent approximation, however, beyond 1000 samples we were able to obtain good results with low margins of error. From this, we note that balancing between excessive computation and accuracy requirements lacks a singular hard threshold and should be approached with the specific design parameters in mind.

The Expectation Maximization algorithm has proven itself to be a fairly robust tool for the approximation of distribution parameters. When provided with a sufficient number of samples and reasonable initializations of the estimated parameters EM is,on average, capable of producing results with single digit error precision (percentage). However, it is not entirely immune to failure. The algorithm diverges in cases where it is initially assumed that the weight of any given component is zero, or the variance of any given Gaussian component is initialized with a zero value. Similarly, while not entirely failing, the margin of error that occurs in the impoverished case of too few samples provided to the algorithm is vastly larger on average. This error margin was observed to reach a mean of as high as nearly 41% error, roughly 8 times as large as that of the trials performed with a sufficient number of samples. We have also demonstrated that the algorithm tends to provide better performance when faced with distributions that are, for the most part distinct and separate. Whereas when tasked with a distribution that is heavily mixed and the components are largely indistinguishable from one another, the performance is much poorer, even with strong initialization parameters.

A Code Listings

References