Maximum Likelihood Data Association for Monocular SLAM

Julian Straub - JulianStraub@gatech.edu

Problem Definition

All tracking applications require robust data association over time in order to work properly. This means after identifying objects in a new sensor measurement it is necessary to find out which of the previous objects corresponds to one of the new objects or whether a object appeared for the first time. Robust data association is essential in applications such as tracking airplanes with radar, cars on a highway or even how a cell population evolves.

In this project we will focus on tracking corner features extracted from a single camera. This is necessary in order to be able to estimate the position of such a corner feature in 3D space, since a single frame can only provide the direction of the feature but not the distance from the camera. Being able to estimate the position of features in the world is a prerequisite for all visual Simultaneous Localization and Map building (SLAM) systems. Tracking those corner features is clearly a problem of data association since we get a set of features in every frame received from the camera, that we want to associate with established landmarks i.e. features that we have observed in previous images.

Related Work

In the literature there exist several approaches to solve the feature to landmark association in visual SLAM. The most commonly used algorithm is Nearest Neighbor (NN) tracking [1]. This is a straight forward algorithm that simply finds for each feature the, in the euclidean sense, nearest landmark and associates them.

The Maximum Likelihood (ML) or individual compatibility data association [7, 5] takes the uncertainty of the position of a landmark into consideration. Features are associated with the landmark that they have the smallest Mahalanobis distance to.

In using the information about correlations between landmarks, the Joint Compatibility Branch and Bound (JCBB) algorithm by Neira [7, 5] achieves even more robust data association. The algorithm searches the space of all possible assignments for the most probable. Although this search space is exponentially large in the number of observed features the branch and bound part in the algorithm makes it feasible. Branch and bound discards sub-trees that would not improve the current data assignment. Although this is a powerful tool, it is in practice not possible to have much more than 20 features if real-time performance is desired [5].

Approach

The goal of this project was to implement ML data association for a monocular visual SLAM system based on iSAM [6]. On the basis of the GTSAM library which gives access to an iSAM implementation, a basic visual SLAM system has already been set up. This system relies on NN data association of SURF [2] features between two consecutive frames, which is often incorrect. Those wrong feature-landmark matches cause the whole system to get unstable, since they add incorrect constraints on the solution.

The fundamental difference of this approach to ML data association, is that instead of frame to frame matching of features, we now use the knowledge about the approximate position of a landmark in 3D space to find matches between the extracted corners in one frame and the landmark corners that have already been added to the SLAM system. GTSAM gives access to the mean and the covariance of the distribution of each landmark in 3D space. In order to be able to find a distance between 2D corner features in the frame and the 3D landmarks, the landmark distributions are projected back into the 2D coordinate system of the image.

Projecting a 3D point into the 2D coordinate system of the camera can be done given the knowledge of the camera calibration matrix K and the 3D pose of the camera, consisting of rotation R and translation t. K was retrieved using a camera calibration script [3] within matlab. According to [4] the augmented image coordinates $x = (u, v, 1)^T$ of a given augmented 3D point $X = (X_1, X_2, X_3, 1)^T$ can then be computed using

$$x = K \cdot [R|t] \cdot X \tag{1}$$

where $[R|t] \cdot X$ transforms X into the coordinate system attached to the camera and the multiplication with K transforms the result into image coordinates x. The pose estimate of the camera can be obtained from GTSAM.

The projection of the covariance C of a 3D landmark into the 2D image domain has to be approximated since the projection function of a pinhole camera is non-linear:

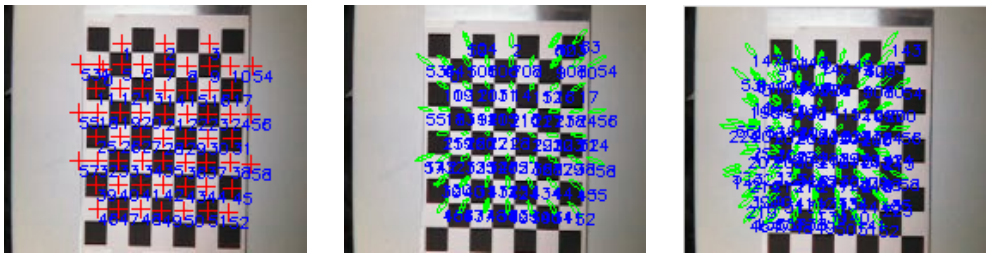

Figure 1: Left: Positions where SURF features have been extracted. Middle: Original first frame of the camera with projected landmark sigma curves. Right: Original second frame of the camera with projected landmark sigma curves.

$$\begin{bmatrix} u \\ v \end{bmatrix} = g(X) = \frac{f}{X_3} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \tag{2}$$

where f is the focal length of the camera.

To deal with that, we linearize this projection function around the mean μ_X of the landmarks distribution:

$$g(X + \Delta X) = g(X)|_{\mu_x} + \left.\frac{\partial g(X)}{\partial X}\right|_{\mu_X} \cdot \Delta X \tag{3}$$

where

$$\frac{\partial g(X)}{\partial X} = J_X = \begin{bmatrix} \frac{f}{X_3} & 0 & -\frac{f \cdot X_1}{X_3^2} \\ 0 & \frac{f}{X_3} & -\frac{f \cdot X_2}{X_3^2} \end{bmatrix} \tag{4}$$

Also, in order to be able to apply this linearized projection transformation, we have to have the covariance in the coordinate system attached to the camera. As described beforehand, the transformation into camera coordinates is a affine transformation $[R|t]\,X$. Hence the covariance in camera coordinates is given by RC_XR^T.

So all in all we can compute the mean and the covariance of a landmark in the 2D image domain using Equations 1 and3 to obtain the 2D Gaussian distribution:

$$N\left(K[R|t]\mu_X; J_X R C_X R^T J_X^T\right) = N\left(\mu_x; C_x\right) \tag{5}$$

Our naive implementation then computes the Mahalanobis distance between all feature-landmark pairs according to

$$d_{ij} = (x_i - \mu_{xj})^T \cdot C_{xj}^{-1} \cdot (x_i - \mu_{xj}) \tag{6}$$

where the subscript i denotes the i^{th} feature and j the j^{th} landmark.

A feature i is only associated with a landmark j iff

$$d_{ij} = \min_{j \in J}(d_{ij}) < 3 \tag{7}$$

holds, with J denoting the set of all landmarks that are currently estimated and 3 standing for a distance of three standard deviations from the mean.

In short the whole algorithm iterates over those four steps:

1. Project landmark distributions to 2D image plane using current estimates obtained from iSAM.
2. Compute Mahalanobis distances.
3. Associate features to landmarks according to Equation 7.
4. Insert those new measurements in the factor-graph in iSAM and compute an update.

Evaluation

Until the very end, there were issues in the code that resulted in the estimates of the camera being off. Therefore the goal of comparing NN against ML data association could not be reached.

Figure 1 shows the positions where SURF extracts features in the original frame one. Also the re-projection of the landmark distributions is shown for frame one and two in the form of their sigma curves.

Figure 2 shows the individual distributions after the first frame to enable a qualitative comparison between the re-projection shown in Figure 1.

Discussion

Unfortunately and despite of all efforts put into the project, no quantitative results comparing the NN and ML data association could be obtained in time. Issues in the code that could so far not be eliminated lead to an incorrect re-projection of the landmark

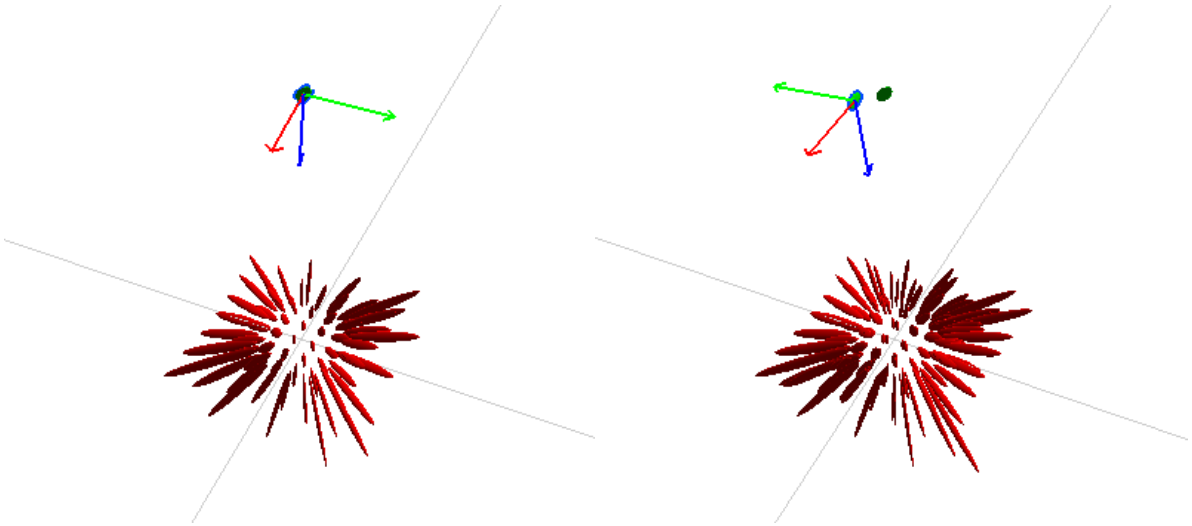

Figure 2: Left: Distributions after the first frame: 3D plot of the camera position estimate (green ellipse) and the landmarks (red ellipses). The ellipses show the 3D one-sigma surfaces of the respective Gaussian distributions. The landmarks were extracted using SURF features. Right: Distributions after the second frame. Note, that due to ambiguities resulting from the way the orientation of the ellipses is extracted, the y-axes (green) in the right image is wrong and should point in the opposite direction.

distributions in the image as can be seen in Figure 1 when comparing the extracted feature positions against the positions of the landmark ellipses in frames one and two. This in turn results in wrong feature-landmark associations and leads to fast divergence of the solution as can be inferred from the growing error of the re-projection from frame one to frame two.

However, the re-projection of the 3D landmark distributions into the image coordinates could be demonstrated. Also, qualitatively it can be seen from Figure 2 and the middle and left image in Figure1, that the re-projection produces reasonable 2D distributions for the landmarks. The problem that leads to the drift of the reprojected landmarks away from the extracted features is assumed to be the inaccurate estimation of the most recent camera pose. Small deviations in the orientation of the camera estimate lead to big shifts of the reprojected landmarks and thus to bad feature-landmark associations.

All in all it is clear that the workload was underestimated in this project. However a lot of progress has been made and the main problem has been identified. The hope is that the issue can be resolved until the deadline for the final paper and that more results can be presented in it.

After that it would be interesting to see how joint compatibility data association performs in comparison. Also, a possible and probably necessary improvement for both, the NN and ML data association, would be to ensure that one feature is only connected to one landmark and vice versa. This is a clear constraint that could help reduce the number of miss-associations.

References

[1] Y. Bar-Shalom. *Tracking and data association.* Academic Press Professional, Inc., San Diego, CA, USA, 1987.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. *Computer Vision–ECCV 2006*, pages 404–417, 2006.

[3] Jean-Yves Bouguet. Camera calibration toolbox for matlab http://www.vision.caltech.edu/bouguetj/calib_doc/, 2010.

[4] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, second edition, 2004.

[5] Michael Kaess and Frank Dellaert. Covariance recovery from a square root information matrix for data association. *Journal of Robotics and Autonomous Systems*, 2009.

[6] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 2008.

[7] José Neira and Juan D. Tardós. Data Association in Stochastic Mapping Using the Joint Compatibility Test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, December 2001.