MULTIMODAL TRACKING FOR ROBUST POSE ESTIMATION

A Thesis
Presented to
The Academic Faculty

by

Prateek Singhal

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Interactive Computing

Georgia Institute of Technology
May 2016

Copyright © 2016 by **Prateek Singhal**

MULTIMODAL TRACKING FOR ROBUST POSE ESTIMATION

Approved by:

Professor Henrik Christensen,
Committee Chair
School of Interactive Computing
Georgia Institute of Technology

Professor Byron Boots
School of Interactive Computing
Georgia Institute of Technology

Professor James Hays
School of Interactive Computing
Georgia Institute of Technology

Date Approved: 22 April 2016

To the force ..

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr Henrik Christensen, for believing in me and giving me an opportunity to work on a challenging problem. He helped me figure out the path and get me through my lows. I would also like to thank my labmates Ruffin, Siddharth, Varun and Samarth with whom I spent some cherished moments of my time. I am indebted to Siddharth and Rahul for being the ideal mentors, for being accommodating and helping me as I stumbled through the dark. I have shared my time here with awesome robotics students Yannick, Zhaoyang, Natesh and Carlos from whom I learnt about the diversity of research. Finally, I would like to thank Cheshta and my parents, for always being there for me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS OR ABBREVIATIONS

AR	Augmented Reality.
DOF	Degree of Freedom.
EKF	Extended Kalman Filter.
GPU	Graphical Processing Unit.
ICP	Iterative Closest Point.
RANSAC	Random Sample Consensus.
RGB	Red, Green and Blue colour.
RGBD	Red, Green and Blue colour with depth.
ROS	Robot Operating System.
SE(3)	Special Euclidean Group in 3 dimensions.
SFM	Structure from Motion.
SLAM	Simultaneous Localization and Mapping.
SO(3)	Special Orthogonal Group in 3 dimensions.
SURF	Speeded Up Robust Features.
SVO	Semi-Direct Visual Odomtery.
TSDF	Truncated Signed Distance Function.

SUMMARY

An on-line 3D visual object tracking framework for monocular cameras by incorporating spatial knowledge and uncertainty from semantic mapping along with high frequency measurements from visual odometry is presented. Using a combination of vision and odometry that are tightly integrated we can increase the overall performance of object based tracking for semantic mapping. We present a framework for integration of the two data-sources into a coherent framework through uncertainty based fusion/arbitration. We demonstrate the framework in the context of OmniMapper[35] and present results on challenging sequences of tabletop settings over multiple objects.

CHAPTER I

INTRODUCTION

Robotic perception is starting to become ubiquitous in the real world. Technologies from Autonomous Cars to Augmented Reality leverage perception. Perception is an old and diverse field, dealing with interpreting the world with sensors. In Computer Vision, perception of the world through cameras is the definition of the field. A sample use case is mapping the world and estimating the sensors position in the world. This problem is referred to as Simultaneous Localization and Mapping (SLAM) - task of building maps with mobile robots. The problem can be summarized by the following question: given data about the world collected from a moving platform, how can we solve for the structure (mapping) of the world together with the motion of the platform (localization)? SLAM has been a subject of extensive research with a large body of literature in the recent past. The current state of the art SLAM systems are able to localize and map the world in real time using sensors such as vision, laser or sonars with robustness. Though the field of SLAM has advanced significantly, open problems like building semantically rich maps which are memory and computationally efficient still exist. Sparse geometric features based methods were popularized in the early 2000's [10] [18][19] while direct local appearance based methods [16][14] have recently been more popular. Both of these methods don't exploit the semantics in the scene for map building or localization.

Another important use case for robotic perception is pick and place of objects, this task consists of detecting a diverse set of objects and then estimating the pose of the object to pick and place in a new position. This problem has been well studied in the form of object detection and pose estimation. An extension of this task is per frame pose estimation of objects, typically in household robot grasping scenario. That is by no means a new problem, early work dates back to the 60's [20]. Since then, there have been significant efforts to improve the robustness of the solution. Nonetheless, most of the available techniques are limited in terms of the scope of objects, robustness to background clutter and occlusions, and computational efficiency. Recently, pioneering work [8] utilizing geometric methods with known 3D models was proposed which alleviates several of these problems. Although this method, does well in typical object tracking scenarios, it fails in real world SLAM settings with heavy occlusion and long out of view scenarios. Even in static environment scenarios, there is scarce literature on combining object tracking and standard SLAM methods.

1.1 Motivation

Current SLAM systems operate at either the level of sparse features using edges or salient points [18][19] in the scene or do dense tracking [24] of the whole scene, these are further built into a map using either 3D point based meshes or have dense representation in the form of voxels. None of these methods contain any semantic information in the maps. Semantics are useful in maps do high level tasks such as navigation or exploration. We propose a method which integrates the semantic information of model based tracking and mapping with a sparse feature based tracking. There has been recent work building semantic maps incorporating semantics like planes and objects in the map. However, these methods try to use features which can be tracked over long periods of time. We go one level further to include objects in the map as semantic entities which can be effectively tracked over long periods. Our approach allows us to handle complex motions of the camera in a cluttered scene while preventing the map from growing too large or discrepancies developing in camera trajectory. SLAM methods typically operate without the knowledge of semantics and generally fall into the trap of being unscalable with time due to memory and computational complexity issues, while purely detection based methods from images rely solely on the image features without any knowledge of the 3D scene, leading to longer

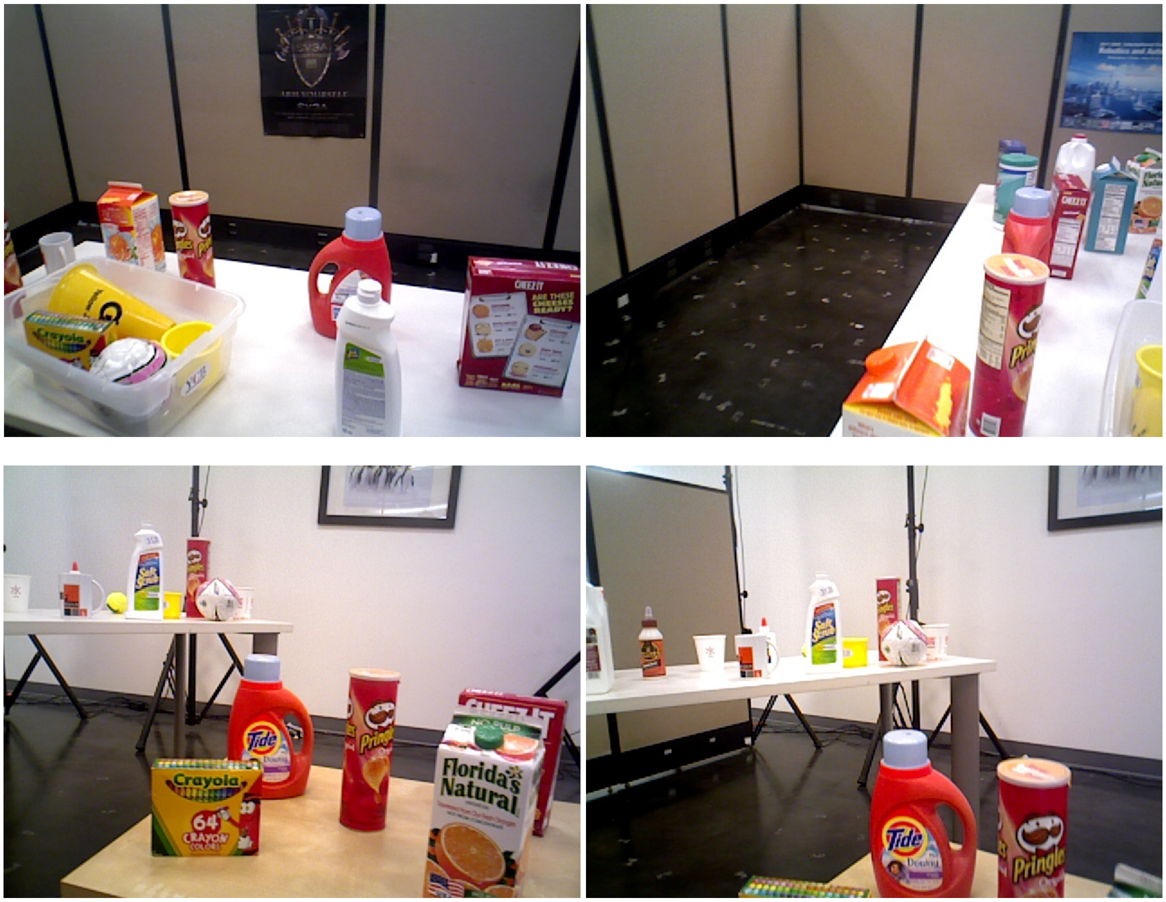


Figure 1: Challenges in Object Perception

Clockwise from Top left (a-d). **a,b:** Typical examples of foreground Occlusion due to clutter. The Tide Box is being tracked. **c:** Orange Juice box in the image being tracked, **d:** Out of view scenario, image without the object.(Orange Juice Box)

processing times. We exploit the relationship of objects in the scene to build a map which is scalable over time as well as allowing detection to benefit from the knowledge of the current scene to provide an accurate pose.

1.2 Challenges

Some of the challenges which are commonly encountered in real world SLAM settings for object tracking are clutter, long out of view sequences while SLAM suffers from drift and scale estimation. We detail these challenges in the subsequent sections:

1.2.1 Clutter

Varying degrees of clutter is one of the major difficulties in object detection and tracking. While most of the related efforts work well in backgrounds having a reasonable amount of clutter, it is still demanding to estimate the pose of an object with significant clutter as shown in Figure 1a and b. The clutter is not limited to just background and can lead to foreground occlusion. Clutter and occlusion cause data association errors with false estimates commonly making tracking get stuck in local minima.

For robust tracking, we build on the work of Changhyun and Christensen [8], a multiple pose hypotheses edge tracking based on a particle filtering framework. By maintaining multiple pose hypotheses, the tracking follows the global optimum despite significant clutter, an additional refinement process based on the RANSAC [4] algorithm is employed, in which wrong edge data associations are discarded to enhance the accuracy of pose estimates calculated from the measurement associations. We found that in heavily cluttered scenes, Canny Edge detector [5] finds a large number of edges which do not constitute an object boundary. This led us to utilize the recent works on learning edges which are semantically more meaningful in the form of closed contours. Our work is highly robust to clutter even with foreground occlusion.

1.2.2 Object Relocalization

An ideal tracking scenario is that the tracked object is visible during the entire tracking. However, in real world scenarios the object happens to be occluded by other objects, human, or robots. As shown in Figure 1 c and d, sometimes the object may be going out of the camera's field of view or blurred due to the motions of the object or the camera. This is referred to as object relocalization and is an essential part of a tracking setting. However, few efforts addressed such cases. Most of the works, emphasize on reintialization either based on features like SURF [3] or utilize AR markers, which are generally slow or require to augment the environment. In this work, we integrate object tracking in a factor graph formulation, which maintains the object pose in a global coordinate frame. We leverage this to relocalize the object in such scenarios by integrating this into a probabilistic feedback framework.

1.2.3 Scale

Monocular visual odometry systems suffer from being scaled in an arbitary space, rendering them useless for any real world metric problems. We utilize the known 3D models used for object tracking to accurately scale the odometry in the metric space. Features used for object detection are reused to provide the relative pose estimate between 2 keyframes.

1.3 Thesis Statement

A concise statement of this thesis is :

" *Model based tracking combined with visual odometry handles cluttered scenes and diverse objects with robustness.*"

1.4 Definition and Scope

Terminology in robotics and computer vision areas is convenient yet often confusing, so we explicitly introduce important terms used in this thesis and mention the addressed scope of the thesis in this section.

In this thesis, we explore the problem of visual localization and mapping with model based object tracking in unstructured environments. The meaning of model-based is that 3D object models are given a priori. We choose the most general format, 3D polygonal mesh models, since that representation has been widely adopted. A broad definition of localization and mapping would be estimating the current location of the robot or sensor in a map while building the map of the surrounding. In this thesis, we restrict ourselves to a camera as the sensor and parametrize its location with a 6-DOF pose and the map of the surroundings is represented only with objects which are parametrized by a 6-DOF pose as well. A 6-DOF pose is a point on the smooth manifold, SE(3) group, which represents a translation in 3D Euclidean space along with a 3D orientation in SO(3) group. Object pose tracking can be defined as local recursive estimation of the posterior pose hypothesis given the prior pose hypothesis of the previous time step. It usually describes a rigid body transformation between two coordinate systems, but here it mainly represents a rigid transformation from a sensor coordinate system to an object coordinate system. Visual Odometry is defined as the estimation of the pose of the sensor at the current time step, in this thesis we use relative odometry which is defined as pose hypothesis with respect to the previous time step. The unstructured environments represent indoor cluttered scenes occlusions and dynamic perturbations.

1.5 Contributions

- **Object tracking and mapping**: We proposed a pose graph formulation combining model based object tracking and visual odometry to do object based SLAM.
- **Object relocalization with map**: We show fast and accurate object relocalization using prebuilt map and visual odometry.

1.6 Outline

This thesis is organized as follows: In Chapter 2, an extensive literature survey on Visual Semantic SLAM with emphasis on object tracking and visual odometry is presented. In Chapter 3, we explain in brief the model based object tracking and visual odometry systems used. In Chapter 4, the pose graph formulation combining the two is presented with the feedback framework. We do extensive experimentation and show results in Chapter 5. Finally, we conclude the thesis and present promising future directions in Chapter 6.

CHAPTER II

RELATED WORK

Semantic SLAM has been an area of extensive research for the past decade. In this Chapter, we try to present a review of the relevant published works in the recent past which have been influential in the field. The chapter has been divided into 4 sections, Section 2.1 covers Semantic SLAM, reviewing the current SLAM and some of the recent semantic SLAM approaches, Section 2.2 details Object based SFM/SLAM, describing the past work instrumental in building up to the current work. In Section 2.3 and Section 2.4, we provide a brief review of the current model based 6-DOF Object Tracking and Monocular Visual Odometry methods respectively.

2.1 SLAM with Semantics

The Simultaneous Localization and Mapping (SLAM) problem was first proposed by Smith and Cheeseman, who used an Extended Kalman Filter (EKF) on landmark positions and the robot position, in [32]. SLAM has since become an important area of research for mobile robotics. A detailed overview of SLAM′s development is given by Durrant-Whyte and Bailey in [13]. Many modern SLAM techniques eschew the EKF formulation in favor of graph based representations. Instead of filtering and solving for only the current robot pose, these techniques typically solve the full SLAM problem and maintain a graph of the entire robot trajectory in addition to the landmark positions. This has the advantage of resulting in a more sparse representation which can be solved efficiently. Some examples of this type of approach involve Folkesson and Christensens GraphSLAM [15], and Dellaerts Square Root Smoothing and Mapping (SAM) [12]. SAM has been extended to allow incremental updates for improved online operation in [17]. We make use of the GTSAM library [11] based on these techniques as our optimization engine.

Feature based SLAM methods are widely used with a vision based sensor instead of a pose graph formulation. This method has been actively pursued as better feature representations were available like SURF since the turn of the millennium. Approaches with parallel tracking and mapping in multiple threads have also been proposed (PTAM [18]) which efficiently track points by matching them against the current map. Such approaches allow incremental map building and tracking. Semantics such as Planes and Objects have also been explored for both semantic mapping and tracking. With the introduction of cheap indoor depth sensors, plane based semantic localization and mapping have become quite common. Trevor et al. [34] recently presented an approach on using segmented planes from laser and RGBD data to perform localization by plane normal alignment combining tracking in branch and bound framework. While, Raposo et al. [28] combined depth sensors with rgb images to segment planes using depth data and estimate the plane normal, using image based photometric error minimization for localization. Planes are tracked across frames by tracking feature points across frames. Current state of the art plane based method was presented by Renato et al. [29] utilizing a dense tracking framework. They use a novel surface representation called Surfel for mapping and data association. These are aligned using Iterative Closest Point (ICP) method with a running average based refinement and integration into the map. ICP suffers from the problem of failing to converge without good initialization points leading to such methods being not robust to fast motion or occlusion while image based tracking methods degenerate in textureless areas.

Choudhary et al. [9] integrated objects into maps with object discovery in depth data,providing a richer representation of the environment in a service robot scenario. Pillai et al. [26] have explored improving on object detection using SLAM to predict object proposals. Such methods aim to improve the semantics of the map while also allowing higher level tasks such as object detection. Stuckler et al. [33] investigated semantic labelling of points in 3D point clouds, these semantically labelled point clouds are then used to perform ICP across frames to allow for robust data association. Semantic interpretation was given to the resulting maps with labels such as floor, wall, ceiling, or door. All these methods only augment the map with added constraints while not utilizing the semantics for tracking. We aim to integrate an object tracking framework to utilize semantics for both tracking and mapping.

2.2 Object based SLAM

In this work we focus in depth on Object based SLAM systems. Here we elaborate on two methods, SLAM++ Moreno et al. [30] , and SSFM from Bao and Savarese [2, 1], which have approached the same problem from different angles. Both of these methods, as we do, use objects as a semantic representation. They also exploit the structure of the factor graph to do efficient inference.

SLAM++: proposes an object oriented SLAM with a factor graph formulation using a RGBD sensor. They estimate the pose between the current camera pose and the object using a model based ICP, with the assumption that objects present in the scene are known. We make a similar assumption in our work. The odometry between the camera poses is found using ICP. One of the major emphasis of the paper is on model based ICP for estimating the pose of the object in the current frame. This method suffers from the problem of large pre-processing of the point cloud in the current camera frame to find the correct model estimate from the database. In spirit our work is similar to theirs but our method uses a monocular camera with sparse tracking.

SSFM: Semantic Structure from Motion approaches the traditional structure from motion problem by incorporating the semantic priors of objects in the formulation. A state of the art object detector is used to detect the bounding boxes of the object in the image and then standard CAD models are used to predict the scale and the pose probability maps. The points in the structure from motion are used to provide the camera pose. This method, though more generic than ours and SLAM++, requires more computationally extensive processing times due to class specific detection rather than instance specific detection. Furthermore, variations in the class can lead to failure of this system to estimate the pose of the object in the frame. Our take on object recognition for tracking needs to be instance specific rather than class specific favoring finer pose measurements over a large number of poses.

2.3 6-DOF Object Tracking

In this work, we build upon current state of the art model based object tracking. Here, we describe some of the recent approaches for real-time 6-DOF pose tracking using either RGB or RGBD data. In PWP3D, Prisacariu et al. [27] present a method using level sets to do real time segmentation of image into foreground and background. They evolve the contour of the projected model with respect to the pose and perform a dense matching between the projected model image and the foreground in the observed image to minimize the photometric error on a GPU. The model projected onto the image using the initialized pose is iteratively updated based on this error minimization. It has shown to perform badly when object is occluded. We utilize only sparse edge based features to avoid using a GPU and show robustness to occlusion due to a global pose hypothesis based approach.

DART(Dense Articulated Real-Time Tracking), presented by Schmidt et al. [31] described a method to do tracking using only depth data. They propose a modification to the Kalman filter, by replacing the correction step of the Kalman filter with a maximum likelihood estimate of the pose of the object. A TSDF parameterization of the surface is used which allows to incorporate free and occupied space based cost functions. They show results for articulated objects in real-time. Such maximum likelihood based methods are known to depend highly on the model used, as data association assumes that model is perfect.

Choi and Christensen [6] describe a combination of geometric and photometric based error metric using RGBD data for object tracking in a particle filter framework. They use a combination of color based matching and surface normal registration between the image and the model to estimate the current pose. They show an approach to implement a particle filter on the GPU allowing for real time computations even with large number of particles. The above mentioned methods rely on RGBD data to do robust object tracking while in this work we constrain ourselves to only monocular images.

In [7, 8] by Choi et al., a novel edge based tracking method using monocular images was proposed. They estimate the pose of the object by matching sparse correspondences between the contour of projected model and the edge image in a particle filter formulation. This is augmented with RANSAC outlier pose rejection leading it to be very robust to background clutter and motion blur. We use this tracking based approach as it works in real time on a CPU and is easily integrable into our ROS framework. They further extend this work to textureless objects using chamfer matching to initialize the pose.

2.4 Visual Odometry

Visual Odometry was a term coined by Nister in his seminal work [25]. He described a feature based tracking method over 2 views to estimate the pose and extended it over multiple frames using perspective resection. Several approaches since then were built on feature based methods to robustly reject outliers but all of these methods suffered in areas with textureless surfaces and motion blur. Recently, direct approaches using patch alignment have made a resurgence. These methods aim to align image patches by searching along the epipolar line and estimating the inverse depth of each patch to minimize the photometric error.

Semi-Direct Visual Odometry(SVO) was proposed by Forster et al. [16], which is combination of both direct image alignment and feature based method. Features in the images are used to find patches which are aligned with the next frame by minimizing the photometric error. The inverse depth of each point is computed and filtered using a gaussian filter. The odometry is refined by running a bundle adjustment step over windows if desired.

This method is robust to motion blur and has shown to run at 100 Hz, making it an ideal choice for our framework.

In Large Scale Direct (LSD) SLAM, Engel et al. [14] proposed a method which combined both photometric and geometric alignment to find the best registration between images. They match corresponding areas in images which have a spatial gradient like edges or points rather than computing features as used in SVO. They also maintain an inverse depth based estimate using a gaussian filter and minimize both geometric and photometric error. There method works well even in less textured areas but both the above mentioned methods suffer greatly due to outliers like moving objects or when the luminance of the object is not constant.

CHAPTER III

PRELIMINARIES: 3D OBJECT TRACKING & VISUAL ODOMETRY

In this chapter, we describe in detail the model based object tracking and the visual odometry methods used in this work, for the thesis to be self sufficient. Both these methods are essential for our approach and have been experimented and modified by us for our use. These methods only use monocular images as input and are one of the current state of the art methods for their task. We build on the work of Choi et al. [7] (Edge Based Tracking) for model based tracking with the assumptions that the objects are textured and the models for the objects are known a priori. For visual odometry, we extend Semi-Direct Visual Odometry [16] to provide a scaled metric odometry between frames. Both of these methods with their modifications are detailed as follows:

3.1 Edge Based Tracking

Choi and Christensen [8] demonstrated a method of 3D model-based visual tracking. This method of edge based tracking provides the requisite degree of flexibility and speed required for maintaining a high frequency estimate of the camera's pose relative to the object reference frame. By utilizing the pose obtained from object recognition to initialize the pose of the object model, the 3D model is projected onto the image using the camera's intrinsic and distortion parameters known a priori. The projected 3D model is rendered using only the salient edges within the original polygon mesh, salient edges in the model are found based on the curvature of the surfaces, identifying sharp edges in the object helps to match relevant edge features in the image. Once a hypothesis is initialized, the same edge-based measurement likelihood as illustrated in [8] is employed to guide a particle filter on the SE(3) group in tracking the object in the scene.

The particle filter begins in much the same way, where our density function $p(\mathbf{X}_t|\mathbf{Z}_{1:x})$is a set of weighted particles

$$\boldsymbol{S}_t = \{(\mathbf{X}_t^{(1)}, \pi_t^{(1)}), \ldots, (\mathbf{X}_t^{(N)}, \pi_t^{(N)})\} \tag{1}$$

With $\mathbf{X}_t^{(n)}$ representing the sample of the current state $\mathbf{X}_t$ in SE3, and $\pi_t^{(n)}$ the normalized weight proportional to the underlining likelihood function $p(\mathbf{Z}_{1:x}|\mathbf{X}_t)$, with N the total number of particles. Normally a large number of particles is necessary for robust tracking, but to reduce the computation, a fewer number of particles is used and these are locally optimized using iteratively reweighted least squares (IRLS) [7]. Detailed analysis of each component and results can be found in [8].

We discuss the relevant component which we found to be most important for our approach.

3.1.1 Edge Based Measurement Likelihood

In general edge-based tracking, a 3D wireframe model is projected into a 2D image according to a pose hypothesis $\mathbf{X}_t^{(n)}$. Then a set of points is sampled along edges in the wireframe model per a fixed distance. As some of sampled points are occluded by the object itself, a visibility test is necessary. The visible sampled points are then matched to edge points, which are obtained by using the Canny edge detector (Canny 1986), from the input image by performing 1D perpendicular search ([7]). In the matching, edge correspondences having significant differences in orientation are occluded. This can be achieved by defining an indicator function as

$$(I(p_i, q_i)) = \begin{cases} 1, & \text{if } |\theta_m(p_i) - \theta_c(q_i)| \leq \tau_\theta \\ 0, & \text{otherwise} \end{cases}$$

where θ_m and θ_c return the orientation of the model edge to which the sample point p_i belongs and of the image edge point q_i, respectively. After the orientation testing, the residual r_i which is the Euclidean distance between p_i and q_i is calculated.

For consistent refinement of the edge correspondences, a RANSAC on 3D sampled points $\mathbf{P}$ and their corresponding 2D closest edge points $\mathbf{p}$. The approach consistently discards outliers by estimating the best 3D pose containing large number of inliers. The RANSAC algorithm finds the refined edge correspondences $\hat{Z}$ = ($\hat{p}$, $\hat{P}$) given the current pose hypothesis X and the original edge correspondences Z = (p, P). Among the parameters, the maximum number of iterations is 1000, the minimum number of correspondences required for EPnP is 6, and ρ is the probability in which at least one set of randomly sampled m correspondences is from inliers. The ρ, typically is set to 0.99, is used to estimate the required number of iterations which is adaptively adjusted in the iteration.

Therefore, the measurement likelihood is evaluated as:

$$p(Z_t|X_t) \propto (exp^{-\lambda_v \frac{N_Z - N_{\hat{Z}}}{N_Z}})(exp^{-\lambda_r \hat{r}}) \tag{2}$$

where the first term signifies the effective number of visible points after RANSAC refinement, N_z is the total number of visible points and $N_{\hat{z}}$ is the number of points after RANSAC, the second term signifies the average error of the model where $\hat{r}$ is the average residual of the visible points. λ_r and λ_v control the sensitivity of the two terms.

In our current work, we found Canny edge detector to not perform well with clutter and occlusion. The edges found using Holistically Nested Edge [36] method better represented the object boundaries and neglected the background clutter such as text or texture. We used HED in our method, rather than Canny for this reason. We also use the tracker's measure of its health based on the ratio between the valid visible sampled points in the image and sampled points on the image to determine whether the tracker should be reset or not. A sampled point along the edge is deemed valid if the error along the perpendicular direction is less than the maximum error. Here we set 32 pixels as the maximum error possible.

3.1.2 Object Recognition and Pose Estimation

We use SURF features to model the objects in our database, as SURF is rotationally invariant compared to other faster features detectors allowing us to reinitialize robustly while tracking. During training each object in the database is stored with SURF feature descriptors, points and its corresponding 3D points of the model. During testing, we run a SURF descriptor on the image to recognize the objects in the scene. We build a kd-tree for each object in the database which is then matched using FLANN [22] with the SURF descriptors in the current image. Matched keypoints of each object are matched with its corresponding 3D points in the database to obtain the pose of the object in the current frame using Perspective-n-Point [21] algorithm. This process is carried out in a RANSAC framework to provide a robust estimate of the pose.

3.2 Visual Odometry

We use SVO for monocular visual odometry as it is open source and works in real-time. It follows the trend of having 2 parallel threads of mapping and tracking. The approach is a combination of feature and direct methods hence the name semi-direct. It finds features in the images and aligns patches between images to track features. This method has been shown to be very robust to motion blur but does suffer in low textured areas. SVO uses a Sparse Model-based Image Alignment between consecutive poses while refining the photometric error.

$$\delta I(T, u) = I_k(\pi(T.\pi^{-1}(u, d_u))) - I_{k-1}(u) \qquad \forall u \in R \tag{3}$$

where T is the pose of the camera, π is the mapping from the 3d point d_u to the image feature u. R represents the region for which the depth of the features is known. This equation can be minimized iteratively to find the current camera pose with respect to the previous camera ($T_{k,k-1}$).

This is then aligned globally with features in the closest keyframes.The pose obtained from image alignment with respect to the previous pose, has been shown to be not necessarily consistent with epipolar geometry. So, the current feature is aligned with the patch in a reference frame to refine the pose further. This can be written as:

$$u_i^{'} = argmin_{u_i^{'}} \frac{1}{2} ||I_k(u_i^{'}) - A_i.I_r(u_i)||^2 \quad \forall i \tag{4}$$

where A_i is an affine warping to the reference patch of (8x8) pixels, $u_i^{'}$ is the optimized position of the feature.

SVO uses a standard Bundle adjustment over the structure and pose to refine the estimate further. We bypass this step of making these refinements of 3d feature points because it represents a poor investment of computation time in moving toward our goal of generating high-level semantic maps.

One of the major contributions of SVO was effectively estimating the inverse depth of each feature in the image. The inverse depth is estimated by exploiting the epipolar geometry and searching along the epipolar line in the consecutive image. The pose obtained between consecutive frames is used to initialize the depth of the new features in the scene. This is formulated as a recursive Gaussian filter formulated as :

$$p(d_i^k | d_i, m_i) = m_i \mathcal{N}(d_i^{'k} | d_i, \tau_i^2) + (1 - m_i) \mathcal{U}(d_i^{'k} | d_i^{min}, d_i^{max}) \tag{5}$$

where $d_i^{'k}$ is the measured depth lying on the epipolar line between the reference and the current frame. τ is the variance of the distribution, d_i is the mean scene depth, d_i^{min} and d_i^{max} are empirically set. This does not work very well for forward looking cameras, which we modify to use only the mean of the depth points within 2m threshold.

3.2.1 Scale Estimation & Initialization

We initialize SVO using our object database based pose estimation at first keyframe, this process is repeated on the second keyframe to obtain a scaled estimate of the pose between the 2 keyframes. In the presence of multiple objects we chose the object with the highest number of inliers to intialize the keyframes. We take care while initializing that the first 2 keyframes have a minimum disparity between them. The initialization formulation can be written as :

$$T^{c}_{2,1} = inv(T^{c,o}_{1}) * T^{c,o}_{2} \tag{6}$$

where $T_{2,1}$ is the pose between the first 2 keyframes for intialization,$T^{c,o}_{i}$ are the pose between the object and keyframes obtained from the object detection. A RANSAC framework is used to find the pose hypothesis. The pose initialized from this method is a metric measurement and thus the inverse depth based map is also a metric map enabling to estimate a scaled odometry.

CHAPTER IV

OBJECT TRACKING AND MAPPING

In this chapter, we explain our approach and detail the main contributions of this thesis. We build on the work detailed in the previous chapter and provide a joint framework which combines both visual odometry and object tracking into a common metric map. The mathematical framework and the implemented pipeline are described as follows:

4.1 Joint Framework

We formulate our problem as a factor graph as shown in Fig 4.1. Here, the objects are used as landmarks while a pose represents the current camera location along the trajectory. The object tracker provides the factor between the landmark and the pose, while the factor between consecutive poses is provided by visual odometry. This framework is generic and can incorporate multiple modalities of measurements represented as factors; other geometric features such as points and planes as landmarks can also be expressed as factors [35]. In this work, we limit the modality of landmarks to that of pose measurements of objects found in the current scene.

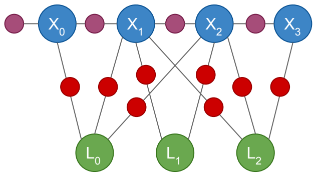


Figure 2: Joint Framework using FactorGraphs

The pose of the camera at i^{th} time step is x_i with $i \in 0 \ldots M$, a landmark is l_j with $j \in 1 \ldots N$ and a measurement is z_k, with $k \in 1 \ldots K$. Factor graph representation. Blue circles denote camera poses (X) and green circles denote landmarks (L). Small purple circles represent odometry constraints and red circles represent landmark-pose constraint(Z).

4.1.1 Pose Graph Optimization

Both visual tracking and visual odometry are combined together to form the complete factor graph. The factor graph constraints at time step k can be written down as :

$$T_k^{c,o} = T_k^{c,w} * T^{w,o} \tag{7}$$

$$T_{k,k-1}^{c} = T_{k-1}^{c,w} * (T_k^{w,c}) \tag{8}$$

where $T_k^{c,o}$ is the pose of the object in the current camera frame, $T_k^{c,w}$ is the pose of the camera in the current frame to the world frame, $T^{w,o}$ is the pose of the object in the world frame, $T_{k,k-1}^{c}$ is the pose of the camera between time steps k and k-1 in $(k-1)^{th}$ camera frame.

The energy to be minimized then can be written as:

$$E = min\|\sum_{k,o}((T_k^{c,o} - (T_k^{c,o})_m) + (T_{k,k-1} - (T_{k,k-1})_m))\|_2 \tag{9}$$

$(T_k^{c,o})_m$ is obtained from EBT, $(T_{k,k-1}^{c})_m$ is obtained from SVO, both of them are treated as measurements in the factor graph. The factor graph is optimized for $T_k^{c,w}$ and $T^{w,o}$ over all the objects and the poses.

We use GTSAM [11] to optimize the factor graph at each time step. Since this is an online process, we use ISAM2 [17] to estimate the current pose of the camera and the object.

4.1.2 Uncertainty based Feedback

Visual tracking often suffers from failure due to fast motion or occlusion leading to addition of erroneous measurements in the factor graph. This situation is analogous to adding wrong loop closure constraints in traditional SLAM systems. We use Sequential Compatibility Nearest Neighbour (SCNN) [23] algorithm to reject erroneous data association. This can be formulated as :

$$Res = ((T)^{w,o} - (T_k^{w,c} * (T_k^{c,o})_m)$$

$$(T_k^{c,o})_f = \begin{cases} (T_k^{c,o})_m, & \text{if Res} \leq \alpha * (\sum)_{(T)^{w,o}} \\ \text{NULL}, & \text{otherwise} \end{cases}$$

where Res is the residual between the current estimate of object in the world frame $(T)^{w,o}$ and the estimate from the current object tracking measurement $(T_k^{c,o})_m$ transformed to the world frame using the current estimate of the camera in the world frame $T_k^{w,c}$, $(T_k^{c,o})_f$ is the factor to be added and $\sum_{(T)^{w,o}}$ is the marginal covariance of the object in the world frame.

If the measurement is rejected we reset the object tracker with the current estimated pose of the object which can be written as:

$$(T_k^{c,o})_e = T_k^{c,w} * T^{w,o}; \tag{10}$$

$(T_k^{c,o})_e$ is the predicted pose of the object in the k_{th} camera frame. The method keeps resetting the pose of the object and increments a counter. When this counter overflows it triggers reinitialization of object tracking using SURF features from the object database. This is done as odometry accumulates drift and seldom leads to erroneous measurements of the predicted pose. The threshold *th* for the counter is set to 500 in our case.

We show the flow of our algorithm in Algorithm 1. EBT(SURF) corresponds to resetting object tracking using SURF, H($T_k^{c,o}$) is the function showing the status of the tracker. The α is the confidence of the object we would like our measurements to lie within, which is set to 2 in our case.

4.2 Pipeline

We show the outline of our system in Fig 3. Monocular images are given as inputs to both tracking and visual odometry in real time. As each algorithm operates at different frame rates, object tracking performance varying with the number of tracked objects in the

Algorithm 1 Uncertainity based Feedback

```
$I_k$ = Image at $k^{th}$ frame
counter = 0
init = true
while $I_k$ do
    $T_k^{c,o}$ = EBT($I_k$,init)
    $T_{k,k-1}$ = SVO($I_k$)
    if (H($T_k^{c,o}$) = good) then
        if Res $\leq \alpha * (\sum)_{(T)^{w,o}}$ then
            Add $(T_k^{c,o})_f$
            counter = 0
            init = false
        else
            Feedback()
        end if
    else
        Feedback()
    end if
end while
function FEEDBACK
    if counter $\leq$ th then
        Reset EBT($(T_k^{c,o})_e$)
        counter++
        init = false
    else
        init = true
        Reset EBT(SURF)
    end if
end function
```

current scene, the measurements are published asynchronously. Omnimapper allows us to create a factor graph online with such asynchronous measurements. The data association per pose is evaluated based on an uncertainty gating criterion, wrong data associations are found and rectifying feedback is given to the tracker. Based on the severity of the data association failure, the current solution may be used to quickly reinitialise the algorithm, or if the current estimate is highly uncertain or unstable, an event signal may be used to trigger the algorithm's own reset procedure. In this way, information is shared across the algorithms in order to arrive at a superior consensus.

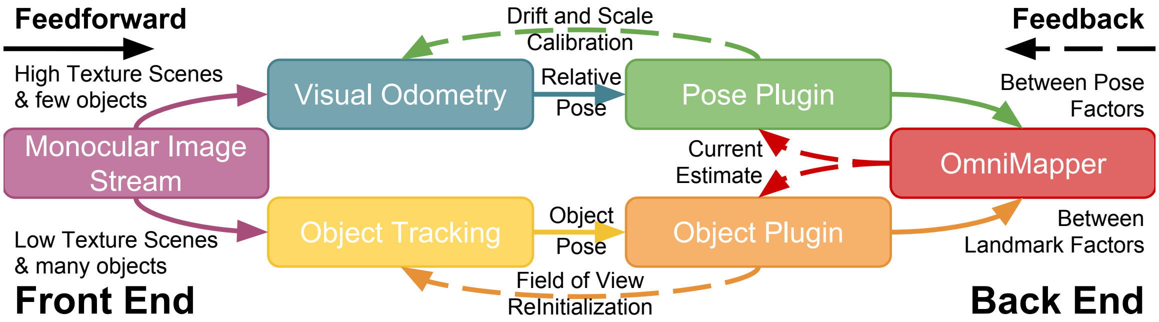


Figure 3: System Pipeline
Flow diagram for proposed tracking framework. Each plugin process measurements from the same image, but commit different levels of information into the graph. With this form of cascading feedback, both plugins influence each other as they supervise their own measurement source's performance

4.3 Implementation

We use ROS for implementation with the publicly available open source packages and using its communication framework for our system. This allows us to be modular and if required swap our current odometry and tracking systems for better methods. Available ROS package for SVO is used with modifications, we modify the plugin to scale the initialization as shown in Section 3.2.1. Our pose plugin subscribes to the relative pose measurement as a time stamped transformation from the odometry reference to the optical frame of the camera. The time dependent transform between the drifting odometry reference frame and the static world is one aspect that is continuously optimized and corrected for on-line. Additionally with our open source ROS package for 2D edge based tracking, we subscribe to the semantically labeled detections of objects. The health signal, along with the pose and covariance of the labeled detection is used by the object plugin in order to commit factors relating the optical frame and the objects reference frame using the open source OmniMapper ROS package. This time dependent sequence of detections is the second aspect continuously optimized. As shown in the flow diagram in Fig 3, for the tracking plugin, this outer feedback results in resetting the tracked pose of the object using the graph estimate for rapid recovery, while barring further measurements of its label from contributing to the graph until reinitialisation is acknowledged. OmniMapper publishes ROS messages which are subscribed by the object based tracker for pose based reinitialization or if the counter for the feedback overflows. If the reported detections persist being erroneous from the current landmark pose for a given amount of frames or time, the object plugin permits the use of the tracker's own reinitialization method. Using OmniMapper we can subscribe to multiple objects with the same framework each being added individually to the graph with the messages being subscribed and published individually.

CHAPTER V

RESULTS

In this chapter, we show experiments on a challenging tabletop dataset. Through these experiments and the corresponding results, we show empirical evidence supporting our claims. We also do an in depth analysis of the visual odometry and object tracking. Qualitative results are shown at the end, to highlight our claims.

5.1 Tabletop Dataset

The current dataset consists of 6 challenging sequences with common household objects from the Amazon picking challenge and the PCL library. These standard objects allow easy access to their CAD models. The sequences have a high level of difficulty compared to standard smooth tracking datasets with partial and full occlusion of the object. Currently the 3 objects in the dataset are: Tide Box, Ronzoni Box and Orange Juice Carton. Each of the objects poses a different level of difficulty for tracking. Tide is largely textureless and smooth, making it tough to detect from various poses; Ronzoni box is textured and small making it difficult to track in large tabletop settings and Orange Juice Carton has both sharp and smooth edges with lot of texture leading to a high number of false data associations for detections and tracking. A scene from the dataset is shown n Fig 4 and the objects used are shown in Fig 5. The dataset contains 2 kind of sequences for each object, in Object Sequences we move around the cluttered table with the object always in the frame but with instances of both foreground occlusion and heavy clutter. In Occluded Sequences, we move the camera in such a way that the object is out of view for significantly long periods of time. Both of these sequences are common scenarios in household setting and exemplify our hypothesis. We plan to release the dataset with the code and add more objects to the library in the future.

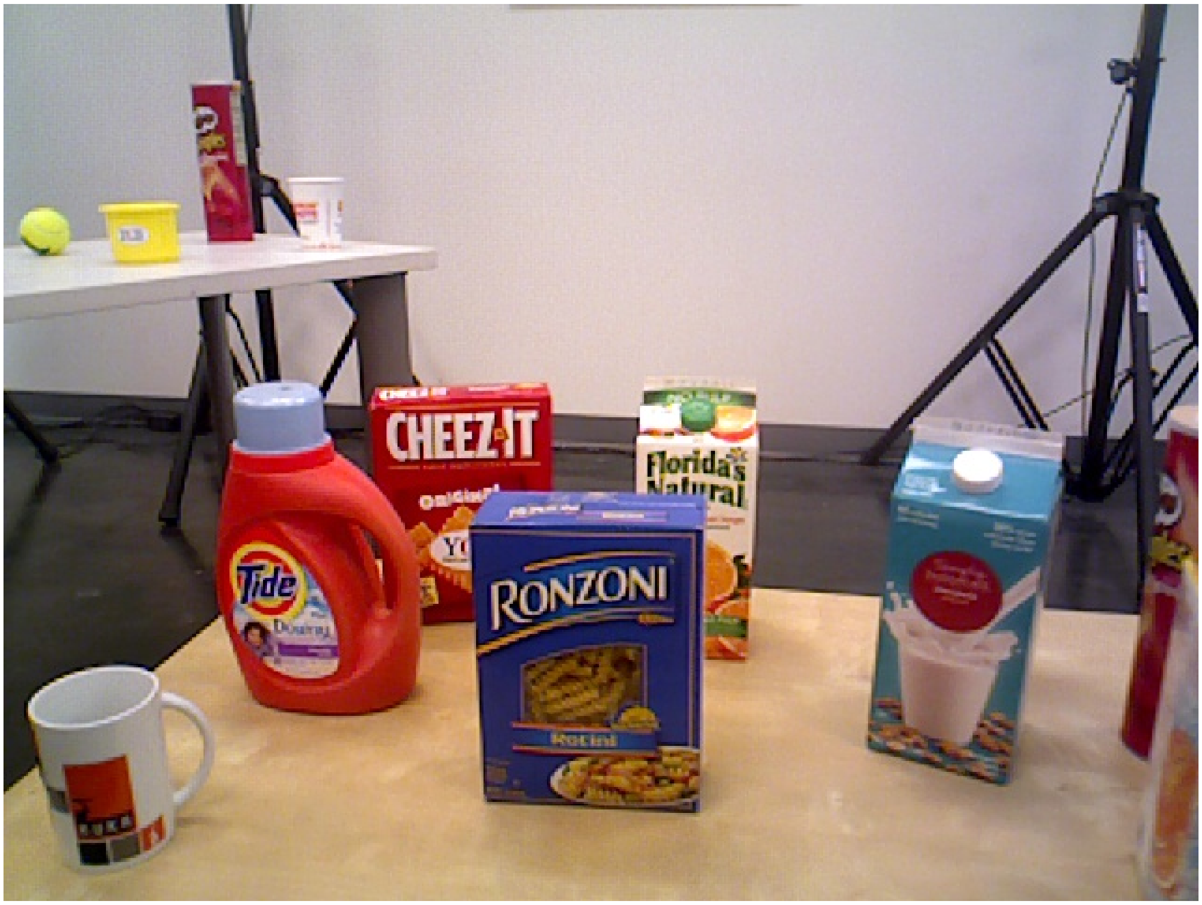
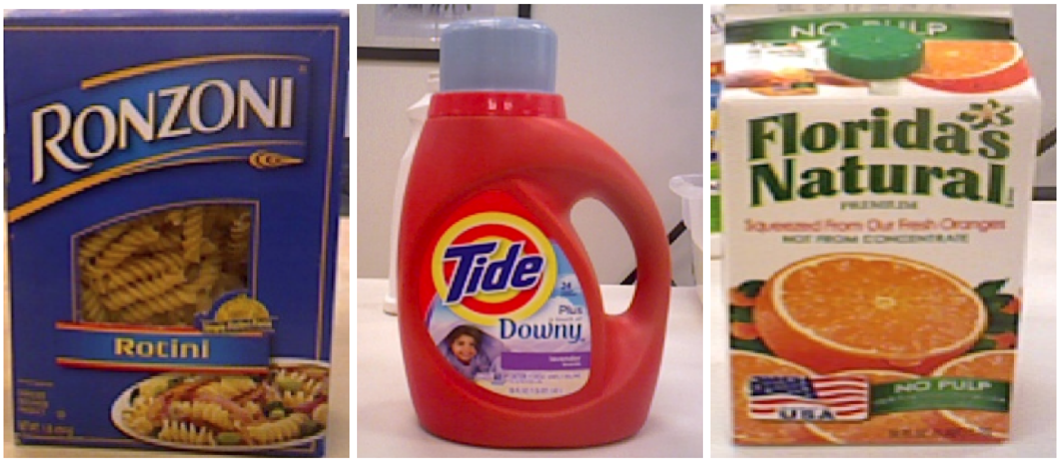


Figure 4: Example Scene from dataset



(a) Ronzoni Box **(b)** Tide Box **(c)** Orange Juice Box

Figure 5: Objects Used in Dataset
All of these are common household objects. We use CAD models from PCL library

5.2 Experimental Results

The parameters we set for SVO are: Minimum Disparity to start tracking = 10 pixels, as we start close to the object, Maximum number of Keyframes = 100, to allow less overhead in batch optimization, Maximum number features tracked = 400, to allow fast addition of features in small spaces, Minimum number of features to be tracked = 20, for cases of textureless settings and having a short baseline of 5cm for adding new Keyframes. Other parameters are set to default settings of SVO ROS. All the parameters are kept constant for all the sequences in the dataset to allow a fair comparison. Parameters set for Object Tracking[8] are : Sampling step = 1cm, interval in which points are sampled from the CAD model, Number of particles = 20 and valid visible point threshold = 0.5, to allow high degree of partial occlusion. The parameters are fixed for all the sequences in the dataset. The noise parameter for the landmark factor is set to 15cm and 30 degrees and noise parameter for pose factor is set to 5 cm and 5 degrees along each axis for translation and rotation respectively.

We do 3 experiments to test our hypotheses:

5.2.1 Object Tracking in Cluttered Scenes

In this experiment, we test how integrating object tracking and visual odometry affects pose estimation of objects. We show quantitative results for per frame object tracking in Table 1. Here we measure the median error per frame of a sequence with respect to ground truth pose of the object in the world frame given ground truth pose of the current frame in the world frame. We consider poses whose error is greater than 5 cm and 5 degrees to be lost and do not consider. We show comparison to EBT[8] and EBT combined with SVO[16] but without the online update step. As can be seen from Table 1 we do better than EBT in all the 6 sequences for the per frame success. We show an improvement of 43% in comparison to EBT in terms of robustness while the median mean error for our approach is 3cm. Our approach does better than SVO both in terms of per frame success % and mean error while

Table 1: Absolute Object Position Estimation Error

		Method		
Sequence		*SVO+EBT*	*EBT*	*Approach*
Ronzoni	median [m]	0.092	0.428	**0.030**
Box	ratio %	40.56	32.95	**71.93**
Orange	median [m]	0.077	0.073	**0.031**
Juice	ratio %	37.73	44.85	**66.42**
Tide	median [m]	0.058	0.413	**0.0525**
Bottle	ratio %	52.79	22.29	**54.70**
Ronzoni	median [m]	0.012	0.1574	**0.007**
Occlud	ratio %	82.99	43.40	**96.52**
Tide	median [m]	0.034	0.197	**0.030**
Bottle	ratio %	61.36	30.10	**89.74**
Tide	median [m]	0.11	0.3615	**0.047**
Occlud	ratio %	27.37	7.22	**62.86**
Mean	median [m]	0.0641	0.271	**0.033**
	ratio %	50.47	30.13	**73.69**

we found that odometry from SVO is very close to ground truth and does not suffer from drift. This leads us to believe that the online incremental feedback allows for more stable tracking.

We present some of our qualitative results in Fig 6. In Fig 6, we show results for the Tide sequence for the challenging case of partial occlusion and narrow viewing angle. As can be seen from the figure, the tracker is able to steadily track in both cases even in such difficult situations. The first and the second column show the current object pose projected in the current gray and edge image. It can be seen that even in large room cases our method is able to track efficiently.

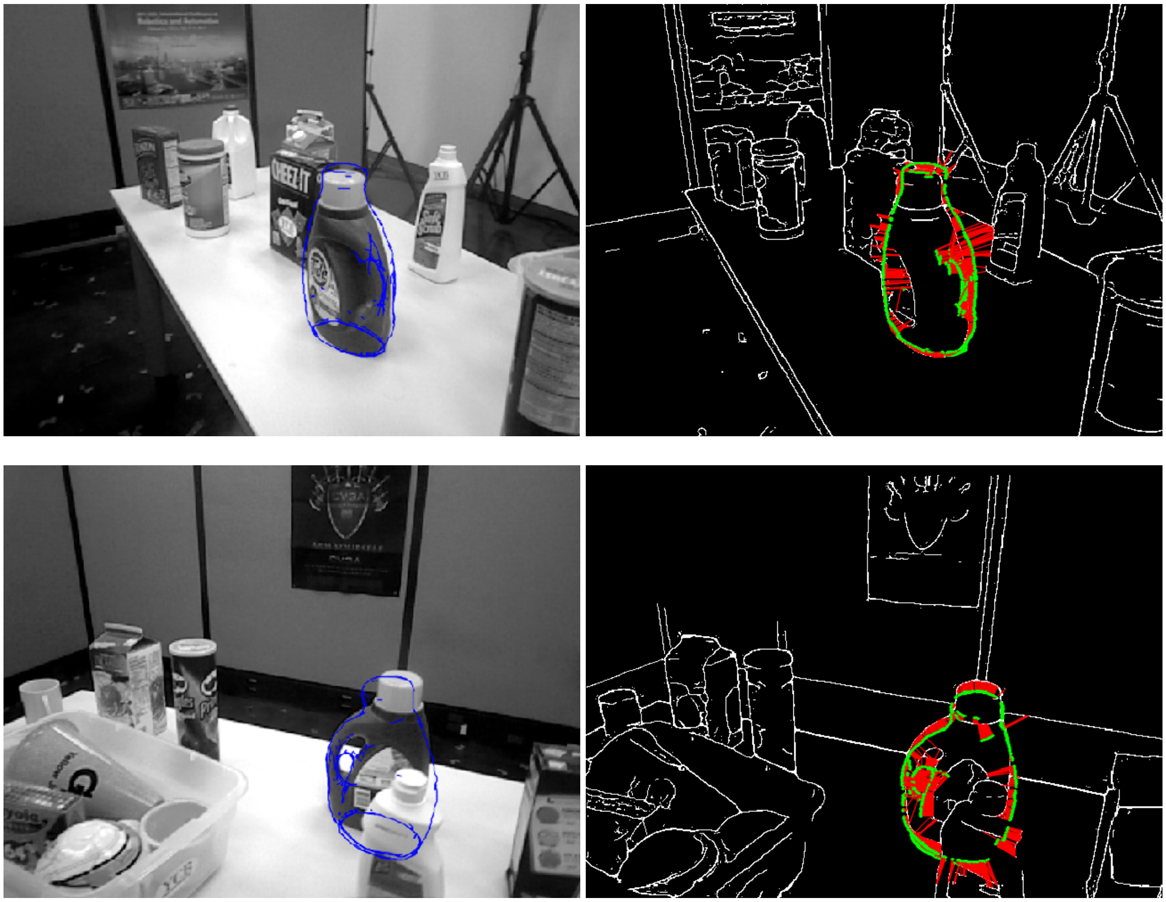


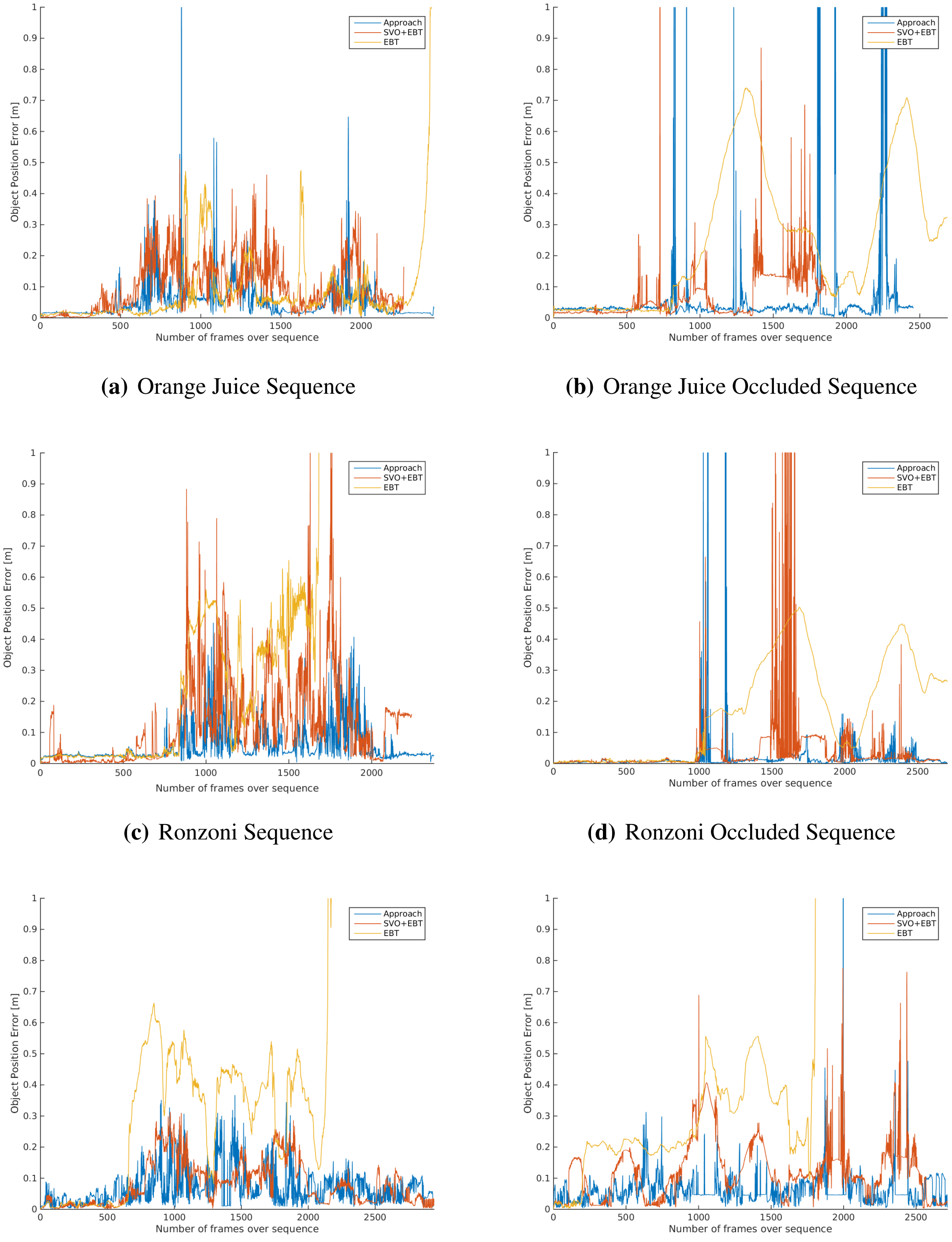
Figure 6: Qualitative Results

Above shown are frames from the Tide sequence (Left to Right): Tracked hypotheses superimposed on the image, and the edge based error using the select edges from the object model.Second row shows instances of foreground occlusion.

5.2.2 Relocalization using Prebuilt Map

With this experiment, we want to test how a prebuilt map effects relocalization of objects. We plot the error over time for each sequence in Figure 7 with respect to the other 2 methods. As it can be seen from the figure, EBT fails for most of the sequences midway while both SVO combined with EBT and our approach are able to track the object in the sequences. The spikes in the figures are the moments where EBT is about to fail but are reinitialized by feedback from Omnimapper. As can be seen from the figure, EBT takes longer time reinitialize than our approach. The map allows us to reintialize the object even when it is partially occluded. This shows that even in cases of full occlusion our method with feedback can relocalize and track the object successfully. This is shown in detail in the supplementary video.

In Fig 8 & Fig 9, we show a sequence of 2 images from Ronzoni Occluded Sequence and Orange Juice Occluded Sequence. Fig 8 shows how in consecutive frames even after more than 50% occlusion of these objects our tracker is able to track them effectively while only seeing a part of it as the object goes out of view of the camera. In Fig 9, we show an instance of the Orange Juice coming back into the scene and how it can be relocalized even with minimal view of the object. This shows our tracker is highly robust to occlusion and the semantic map allows us to relocalize objects without any adhoc detection. More qualitative results are presented in the supplementary video.



(a) Orange Juice Sequence

(b) Orange Juice Occluded Sequence

(c) Ronzoni Sequence

(d) Ronzoni Occluded Sequence

(e) Tide Sequence

(f) Tide Occluded Sequence

Figure 7: Error Plots over Time

Absolute Position error of objects over time in 6 sequences. Figures show positional error in meters over time in seconds. First coloumn shows tracking results as the camera is moved around a cluttered table, while the second coloumn shows out of scene occlusion comparison caused by camera rotations and translations in front of tracked object. Note the spikes in error for EBT particle tracking just before losing the object.

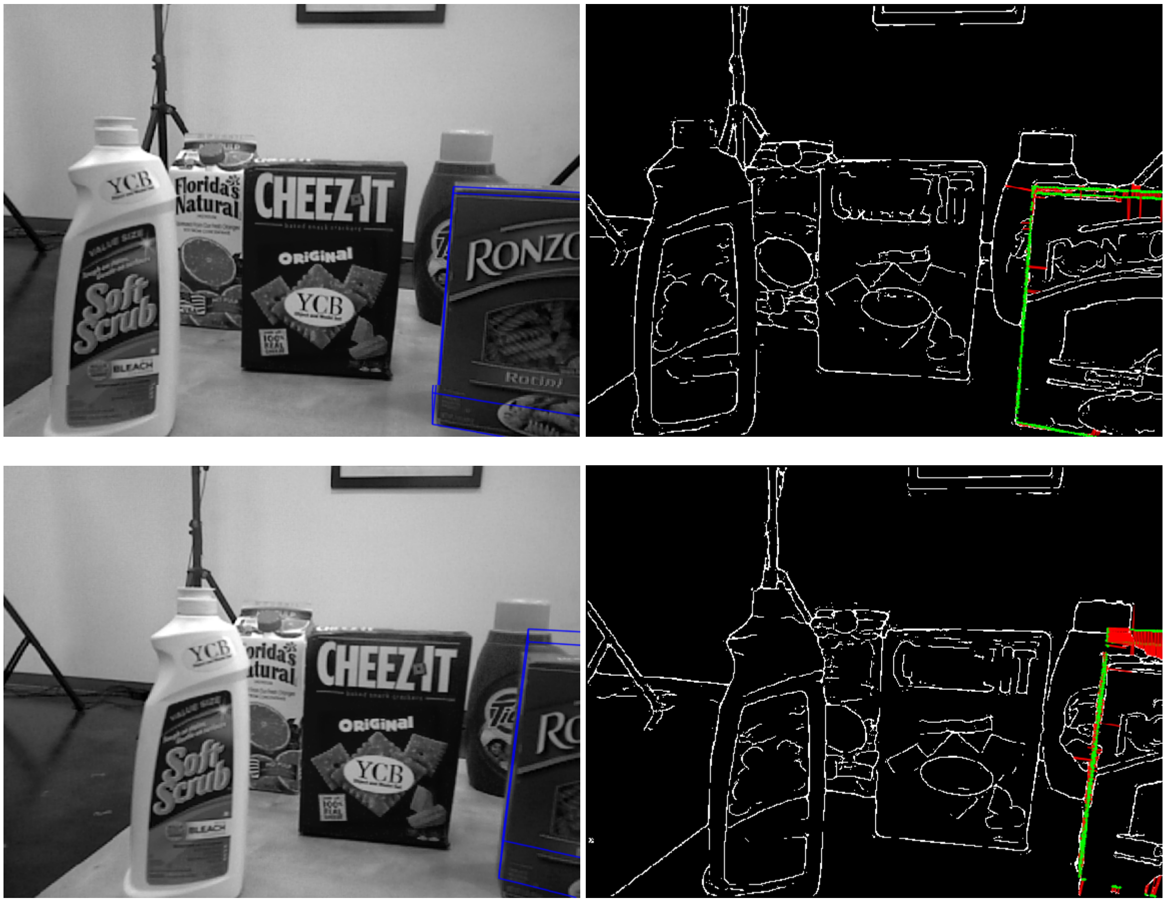


Figure 8: Qualitative Occlusion Result

Left: Object superimposed on the image , Right: Edge image superimposed with object. Shown above on the top is the view of the object in the Ronzoni Occlud sequence just before exiting the scene. On the right is the image of the object still being tracked as it is almost fully occluded while exiting the scene.

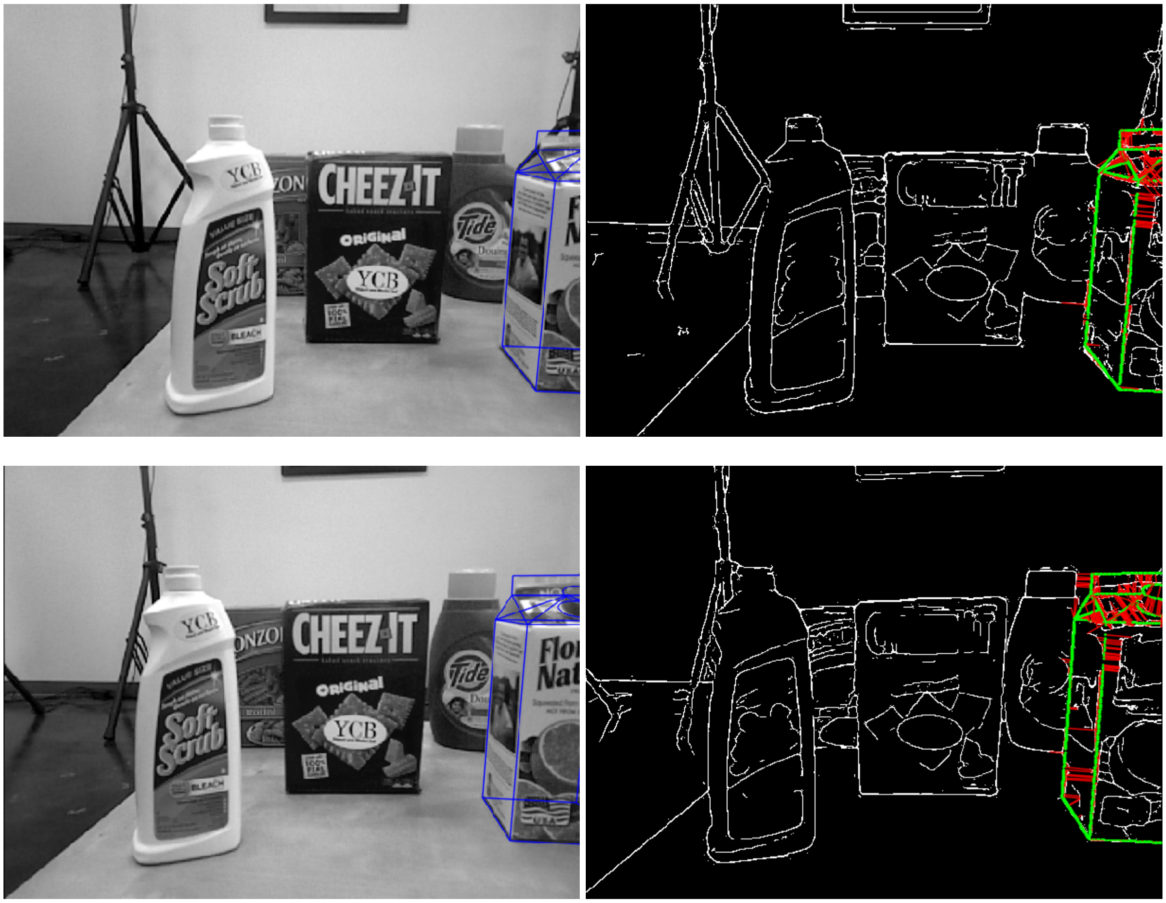


Figure 9: Relocalization Result

Left: Object superimposed on the image , Right: Edge image superimposed with object. Similar to Fig 8, the top row shows the Orange Occlud sequence just before the object exits the scene, while the bottom row shows the object being relocalized as it re enters into the cameras view frame.

Table 2: Absolute Trajectory Error

		Method	
Sequence		*SVO*	*Approach*
Orange Juice	mean [m]	0.090 ± 0.103	0.119 ± 0.094
	median [m]	0.063	0.094
Orange Occlud	mean [m]	0.095 ± 0.062	0.100 ± 0.060
	median [m]	0.073	0.083
Ronzoni Box	mean [m]	0.052 ± 0.028	0.117 ± 0.055
	median [m]	0.050	0.109
Ronzoni Occlud	mean [m]	0.015 ± 0.008	0.038 ± 0.021
	median [m]	0.013	0.030
Tide Bottle	mean [m]	0.135 ± 0.045	0.115 ± 0.053
	median [m]	0.126	0.100
Tide Occlud	mean [m]	0.108 ± 0.103	0.124 ± 0.099
	median [m]	0.069	0.083
All	mean [m]	0.083 ± 0.058	0.102 ± 0.064
	median [m]	0.066	0.083

5.2.3 Localization with Objects

In this experiment, we test our hypothesis whether objects as landmarks in a map help reduce localization error. We show quantitative results for error over the whole trajectory of the camera in each sequence. The comparison is done between SVO and our approach, with results being in Table 2. As it can be seen SVO does better than our approach at times, this is largely due to our use of ISAM2 which leads to large relinearization errors in case of wrong data associations. The mean error difference between the 2 approaches is less than 2 cm, showing that our method does not degrade to a large extent. Also another important finding is that in small spaces Object Based SLAM methods would not necessarily be beneficial over feature based methods in terms of localization error but they do allow to store the map efficiently.

CHAPTER VI

CONCLUSIONS

In this work we have demonstrated methods of improving object tracking reinitialization for monocular cameras by means of incorporating traditional tracking algorithms into a higher-level framework permitting modular measurement fusion, including position feedback from both visual odometry and semantic mapping uncertainties. In this manner, alternative tracking methods and motion estimation can be combined to generate sparse graph representations of world environments, reducing optimization problems for small memory and computationally limited applications, such as small mobile robotic platforms.

Future work within this domain could include evaluating the performance of various combinations of other available VO and object trackers, or perhaps collective of a multiple of each type, and charting the effects of different object types under degrees of occlusion other than for monocular imagery. Additionally, associating an object with its own pose-chain when recognized as non-static could provide a relatively simple extension of the current information sharing framework for dynamic environments, as VO with accompanying objects could be used to discern the relatively dynamic elements in a scene. However, perhaps the most promising extension of this framework would be for object learning, thus moving away from priori model dependent tracking.

REFERENCES

[1] Bao, S. Y., Bagra, M., Chao, Y., and Savarese, S., "Semantic structure from motion with points, regions, and objects," in *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pp. 2703–2710, 2012.

[2] Bao, S. Y. and Savarese, S., "Semantic structure from motion," in *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pp. 2025–2032, 2011.

[3] Bay, H., Tuytelaars, T., and Gool, L. J. V., "SURF: speeded up robust features," in *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part I*, pp. 404–417, 2006.

[4] Bolles, R. C. and Fischler, M. A., "A ransac-based approach to model fitting and its application to finding cylinders in range data," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81), Vancouver, BC, Canada, August 1981*, pp. 637–643, 1981.

[5] Canny, J. F., "A variational approach to edge detection," in *Proceedings of the National Conference on Artificial Intelligence. Washington, D.C., August 22-26, 1983.*, pp. 54–58, 1983.

[6] Choi, C. and Christensen, H. I., "RGB-D object tracking: A particle filter approach on GPU," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, November 3-7, 2013*, pp. 1084–1091, 2013.

[7] Choi, C. and Christensen, H., "Real-time 3d model-based tracking using edge and keypoint features for robotic manipulation," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 4048–4055, May 2010.

[8] Choi, C. and Christensen, H., "Robust 3d visual tracking using particle filtering on the special euclidean group: A combined approach of keypoint and edge features," *The International Journal of Robotics Research*, vol. 31, no. 4, pp. 498–519, 2012.

[9] Choudhary, S., Trevor, A. J. B., Christensen, H. I., and Dellaert, F., "SLAM with object discovery, modeling and mapping," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014*, pp. 1018–1025, 2014.

[10] Davison, A. J., Reid, I. D., Molton, N., and Stasse, O., "Monoslam: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, 2007.

[11] Dellaert, F., “Factor Graphs and GTSAM: A Hands-on Introduction,” Tech. Rep. GT-RIM-CP&R-2012-002, Georgia Tech, September 2012.

[12] Dellaert, F. and Kaess, M., “Square root SAM: simultaneous localization and mapping via square root information smoothing,” *I. J. Robotic Res.*, vol. 25, no. 12, pp. 1181–1203, 2006.

[13] Durrant-Whyte, H. F. and Bailey, T., “Simultaneous localization and mapping: part I,” *IEEE Robot. Automat. Mag.*, vol. 13, no. 2, pp. 99–110, 2006.

[14] Engel, J., Schöps, T., and Cremers, D., “LSD-SLAM: large-scale direct monocular SLAM,” in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II*, pp. 834–849, 2014.

[15] Folkesson, J. and Christensen, H. I., “Graphical SLAM - a self-correcting map,” in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA 2004, April 26 - May 1, 2004, New Orleans, LA, USA*, pp. 383–390, 2004.

[16] Forster, C., Pizzoli, M., and Scaramuzza, D., “Svo: Fast semi-direct monocular visual odometry,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 15–22, May 2014.

[17] Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., and Dellaert, F., “isam2: Incremental smoothing and mapping using the bayes tree,” *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[18] Klein, G. and Murray, D., “Parallel tracking and mapping for small ar workspaces,” in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pp. 225–234, Nov 2007.

[19] Klein, G. and Murray, D., “Parallel tracking and mapping for small ar workspaces,” in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pp. 225–234, Nov 2007.

[20] Lawrence, G. R., “Machine perception of three-dimensional solids,” *Ph. D. Thesis*, 1963.

[21] Lepetit, V., Moreno-Noguer, F., and Fua, P., “Ep*n*p: An accurate $O(n)$ solution to the p*n*p problem,” *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

[22] Muja, M. and Lowe, D. G., “Fast approximate nearest neighbors with automatic algorithm configuration,” in *VISAPP 2009 - Proceedings of the Fourth International Conference on Computer Vision Theory and Applications, Lisboa, Portugal, February 5-8, 2009 - Volume 1*, pp. 331–340, 2009.

[23] Neira, J. and Tardós, J. D., “Data association in stochastic mapping using the joint compatibility test,” *IEEE T. Robotics and Automation*, vol. 17, no. 6, pp. 890–897, 2001.

[24] Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S., and Fitzgibbon, A., "Kinectfusion: Real-time dense surface mapping and tracking," in *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pp. 127–136, IEEE, 2011.

[25] Nistér, D., Naroditsky, O., and Bergen, J. R., "Visual odometry," in *CVPR (1)*, pp. 652–659, 2004.

[26] Pillai, S. and Leonard, J. J., "Monocular SLAM supported object recognition," in *Robotics: Science and Systems XI, Sapienza University of Rome, Rome, Italy, July 13-17, 2015*, 2015.

[27] Prisacariu, V. A. and Reid, I. D., "PWP3D: real-time segmentation and tracking of 3d objects," in *British Machine Vision Conference, BMVC 2009, London, UK, September 7-10, 2009. Proceedings*, pp. 1–10, 2009.

[28] Raposo, C., Lourenço, M., Antunes, M., and Barreto, J. P., "Plane-based odometry using an RGB-D camera," in *British Machine Vision Conference, BMVC 2013, Bristol, UK, September 9-13, 2013*, 2013.

[29] Salas-Moreno, R. F., Glocker, B., Kelly, P. H. J., and Davison, A. J., "Dense planar SLAM," in *IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2014, Munich, Germany, September 10-12, 2014*, pp. 157–164, 2014.

[30] Salas-Moreno, R. F., Newcombe, R. A., Strasdat, H., Kelly, P. H. J., and Davison, A. J., "SLAM++: simultaneous localisation and mapping at the level of objects," in *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pp. 1352–1359, 2013.

[31] Schmidt, T., Newcombe, R. A., and Fox, D., "DART: dense articulated real-time tracking," in *Robotics: Science and Systems X, University of California, Berkeley, USA, July 12-16, 2014*, 2014.

[32] Smith, R., Self, M., and Cheeseman, P., "Estimating uncertain spatial relationships in robotics," in *UAI '86: Proceedings of the Second Annual Conference on Uncertainty in Artificial Intelligence, University of Pennsylvania, Philadelphia, PA, USA, August 8-10, 1986*, pp. 435–461, 1986.

[33] Stückler, J., Waldvogel, B., Schulz, H., and Behnke, S., "Dense real-time mapping of object-class semantics from RGB-D video," *J. Real-Time Image Processing*, vol. 10, no. 4, pp. 599–609, 2015.

[34] Trevor, A. J. B., III, J. G. R., and Christensen, H. I., "Planar surface SLAM with 3d and 2d sensors," in *IEEE International Conference on Robotics and Automation, ICRA 2012, 14-18 May, 2012, St. Paul, Minnesota, USA*, pp. 3041–3048, 2012.

[35] Trevor, A. J., Rogers III, J. G., and Christensen, H. I., "Omnimapper: A modular multimodal mapping framework," in *Proceedings on the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1983–1990, IEEE, 2014.

[36] Xie, S. and Tu, Z., “Holistically-nested edge detection,” in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pp. 1395–1403, 2015.