Visual Relocalization With RGB-D SLAM and Convolutional Neural Networks

C. Birmingham

A dissertation submitted in partial fulfilment of the requirements of the University of Bristol and the University of the West of England, Bristol

Faculty of Engineering, University of West England, Bristol

September 2017

This study was completed for the PhD in Robotics and Autonomous Systems at the University of Bristol and the University of the West of England, Bristol. The work is my own. Where the work of others is used or drawn on it is attributed.

The dissertation may be made freely available immediately for academic purposes (this includes display in the Library for consultation purposes)

Word Count: ~12300

Approved: 

Chris Birmingham

Acknowledgements:

I would like to thank and acknowledge my supervisor, Dr. Andrew Calway who has guided and supervised me over the last year as well as Shuda Li who provided the SLAM algorithm and the guidance on how to use it.

1 Abstract

Modern RGBD SLAM systems produce impressive, robust results in navigation and map building but are limited by the need for a depth sensor. Current state of the art monocular depth estimation algorithms can produce realistic looking depth maps from single images. Paired together, these two systems could enable monocular SLAM near the level of RGBD SLAM without the depth sensor.

This dissertation investigates the utility of state of the art depth map estimators for SLAM applications. Specifically, this work tests the effectiveness of a depth map estimator combined with a RGBD SLAM system for relocalization on a 3D model.

In this work, a state of the art depth estimator is combined with a state of the art RGBD SLAM system and the pose error for relocalization on 3D models is measured, analyzed, and compared to other relocalization results. The average error for the combined system is 75 cm and 15 degrees, which is an order of magnitude worse than the current state of the art for monocular relocalization on a 3D model.

The main conclusion this work offers is that the current state of the art depth estimator is not effective for SLAM applications when treated as a replacement for the depth sensing element in a RGBD sensor. Future improvements in monocular depth estimation may change this conclusion and this work offers a method for a practical evaluation of estimated depth map utility going forward.

Contents

2 Introduction

2.1 Overview: Simultaneous Localization and Mapping

Before delving into the specifics of the problem for this work, it is important to address the overall area of work and define key terms. The general subject of autonomous localization and mapping is one of building an autonomous system that is capable of understanding the features and shape of its environment (mapping) as well as its place within its environment (localization). Because all sensors understand the environment from where they are located, the tasks of mapping and localization are best done simultaneously, which also allows the localization and mapping to improve one another. Because of this, the problem is now generally referred to in both the computer vision and robotics communities as *Simultaneous Localization and Mapping* (SLAM).

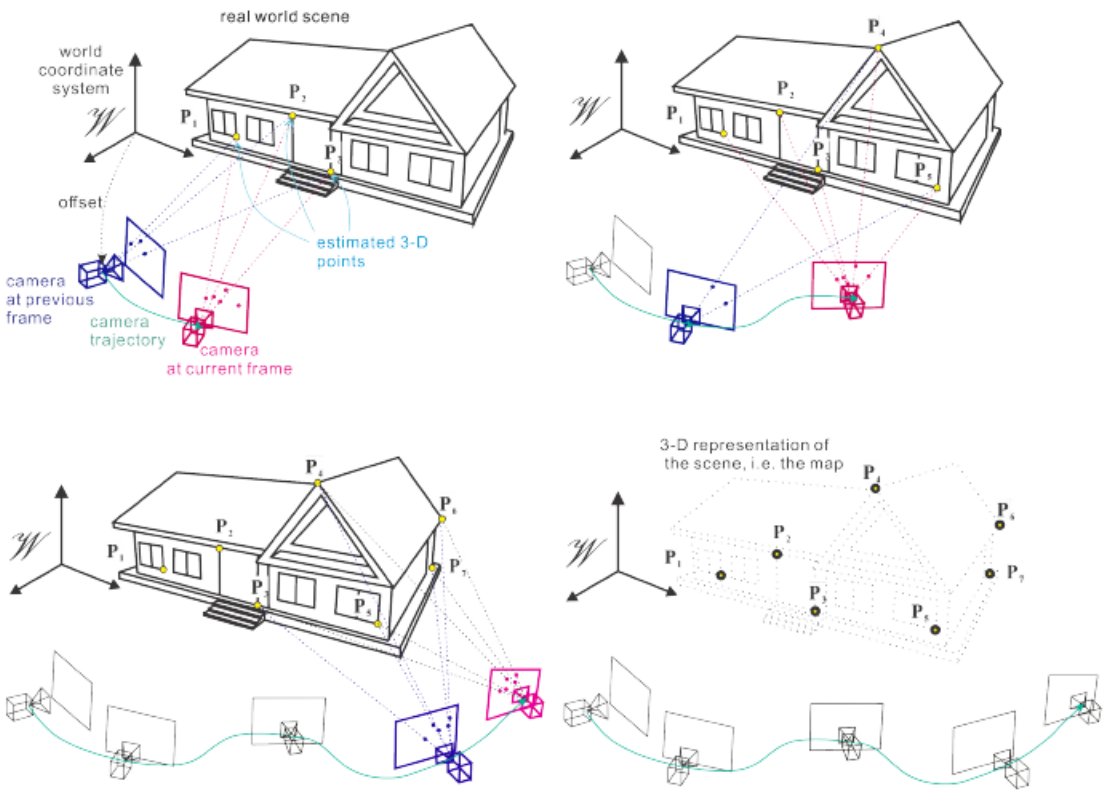

Figure 1: SLAM system in action. Figure used with permission of Shuda Li.

The most basic *SLAM algorithm* has three components. The first is an initialization module that will define the global coordinates of the map and construct the very first part of the map. The second is a tracking module that will locate subsequent sensor readings as they relate to the prebuilt map in order to understand the changing position of the

sensor. The third is a mapping module that is responsible for adding the structure of new regions to the map as they are discovered. Taken together, these three modules (initialization, tracking, and mapping) allow for rudimentary SLAM. High quality SLAM requires additional modules, which will be discussed later in the Literature Review chapter.

Figure 2: Types of Visual SLAM Sensors. Images sampled from the internet.

In addition to a SLAM algorithm, a *SLAM system* must also have a sensor capable of providing enough useful information about the world to build a map. Although it is possible to build SLAM systems using even the simplest of sensors, such as touch sensors or 1D range finders, the utility of such a system is limited to the simplest of situations, such as a two-dimensional maze. For more complicated three and four dimensional (time) situations the effective solutions are typically visual in nature. Visual SLAM systems use passive light sensors such as Red-Green-Blue (RGB) cameras (which provides two-dimensional color intensity maps - a picture), active depth sensors such as LiDaR (which provides two-dimensional depth maps), and a RGB-Depth (RGBD) camera such as a Kinect (which provides 2.5D maps that include both depth and color intensity). Examples of each of these can be seen in Figure 2. In essence, a Visual SLAM system takes a 2 or 2.5D projection of the world and use it to synthesize an accurate and detailed map, enabling use in a wide variety of complex environments.

SLAM is a hard problem and it is reasonable to ask why it is necessary and important to solve at all. After all, there are other methods and technologies that might enable an autonomous system to accomplish its goals without the use of SLAM. For example, GPS and dead reckoning are able to provide estimates of a system's global or relative position; or using rails and tracks with encoders may provide a very accurate position within a given workspace. However, neither of these technologies can solve the difficult

but valuable challenge of enabling an autonomous system to explore an unknown environment. Setting aside even their technical limitations, navigation by dead reckoning and GPS only provide accurate information about where the system is relative to the globe or to its prior location, they do not help with understanding the environment or the systems place relative to it. Thus, SLAM systems have proven useful in a variety of applications, ranging from self-driving cars to augmented reality to online 3D modelling.

Given the large body of research on SLAM and the incredible progress that has been made over the last few decades, some groups have begun to ask if SLAM is still an open problem or if it has been solved. This is a somewhat ill posed question however, because SLAM is an incredibly broad field. SLAM performance and requirements can vary wildly across different sensors, different situations (e.g. indoors or outdoors), or for different applications (e.g. navigation or augmented reality). In Cadena et al. (2016), it is argued that it is only possible to answer the question, "is SLAM solved?" for a given robot/environment/performance combination. They go on to argue that for some combinations (such as a delivery robot working in a hotel) the problem has been solved, while for others (a drone flying through a forest) it has not. Until there are solutions for most-if not all-of the robot/environment/performance combinations SLAM will continue to be a very active and worthwhile area of research.

Within the area of Visual SLAM, the work of this thesis is specific to the subcategory of *Monocular SLAM*, which is SLAM that only uses a single RGB camera. This is one of the oldest areas of Visual SLAM research and still one of the most difficult. This is because although pictures provide rich detail and information about the environment, they do not easily yield structural truth about the environment. Within Monocular SLAM this work specifically looks at *relocalization on pre-built 3D maps/models of the environment*. This is the task of estimating the location and orientation of the sensor with respect to a scene from scratch, using only visual cues.

2.2 Background: Pose Estimation on a 3D Model

Understanding where a subject is with respect to its environment is a problem that might easily be taken for granted, as most humans have the ability to almost instantly solve this problem with minimal disorientation from the moment they open their eyes. For autonomous systems however, the ability to localize their exact position and orientation with respect to a representation (i.e. map or model) of the world given just a single snapshot

from a camera is not trivial. As an example, imagine, taking a photograph of somewhere in a room, and attempting to find the exact position and direction of the camera in the room. For most rooms, matching the picture content with the items in the room is not difficult. Going further and aligning to exactly where the camera was, down to the centimeter and degree, would be difficult.

Fortunately, two techniques have been established to do just that for SLAM systems, called the Perspective *n* Points method (PnP) and the Absolute Orientation method (AO). PnP works by aligning the 2D points in the image plane and 3D points in the model to solve for the pose. It requires at least three matching points to be properly constrained but because of how the nonlinear equations work out it can still produce many solutions. AO is similar but it requires depth values for each of the matching pixels, enabling more constraints. Originally, because accurate depth estimates were hard to obtain, PnP was more accurate for relocalization than AO. As depth cameras have become more accurate however, AO has become the dominant method because it is easier to solve and more precise. The key difference between PnP and AO is that PnP is a 2D to 3D method whereas AO is a 3D to 3D method and requires a depth sensor. This work will attempt to utilize a system based on AO despite not having a depth sensor, through the use of monocular depth estimation, which is described in the following section.

2.3 Context: Progress in SLAM and Monocular Depth Estimation

Over the last decade or two, SLAM algorithms have progressed greatly. With RGBD SLAM it is possible to produce large, robust, detailed maps with fast and accurate localization. One of the only downsides to RGBD SLAM is the need for a depth sensor, primarily because most depth sensors only work in specific environments (e.g. inside vs. outside in sunlight) and fail on specific objects (e.g. reflective objects). These sensors also tend to be more expensive, heavier, and consume more power than just RGB cameras. On the other hand, Monocular SLAM has also come a very long way and is now capable of producing very realistic looking maps with highly robust tracking in a diverse set of environments. The issues still plaguing Monocular SLAM are a lack of true scale in the depth maps, low robustness to changing light conditions, low robustness to rotation only movements, and sub-optimal relocalization when tracking is lost.

Concurrently with the progress in SLAM algorithms, there has been a dramatic leap forward in the ability to estimate scene structure and depth from single monocular images.

New techniques in machine learning have combined with large open source datasets of RGBD images (created using RGBD cameras, like the Kinect) and have enabled the production of very realistic depth maps from RGB images. The essence of these techniques is nonlinear regression on the visual cues in the images to produce semi-accurate but realistic looking depth maps. A more detailed explanation of the history of depth estimation techniques will be provided in the Literature Review.

2.4 Motivation: Replacing Depth Sensors with Depth Estimation

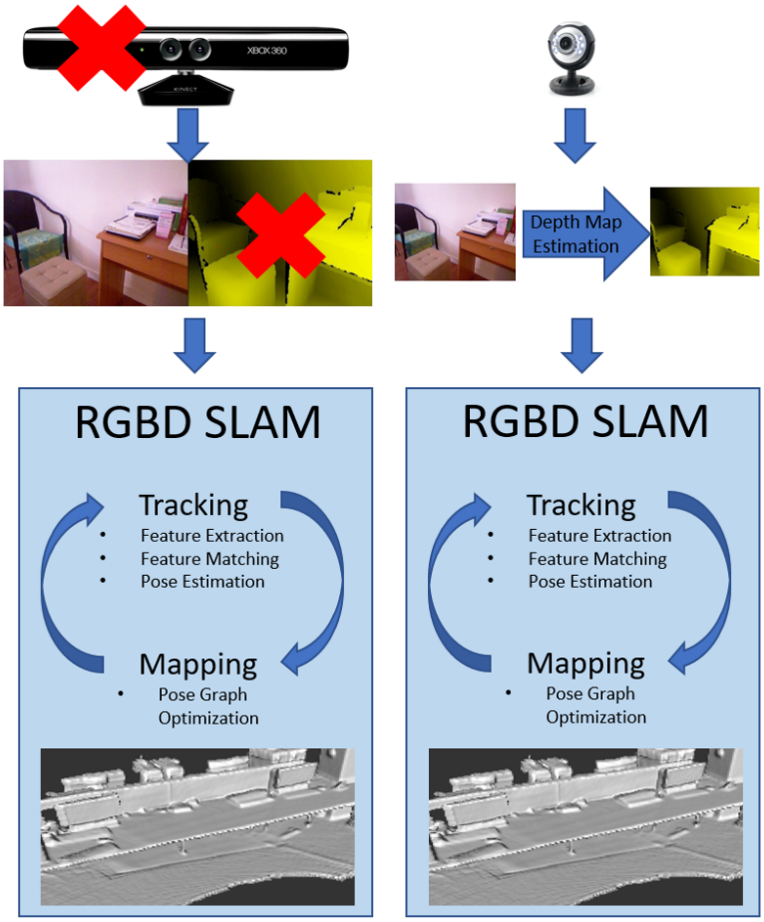

Figure 3: Replacing Depth Sensing with Depth Estimation.

As discussed above, one of the few limitations of RGBD SLAM algorithms is the requirement of a RGBD sensor, due to the cost, weight, power consumption, and technical limitations. Given the progress in producing relatively accurate depth maps from RGB images, it is desirable to find out if it is possible to replace the depth sensing element of an RGBD SLAM system with a depth estimator, as illustrated in Figure 3. This could poten-

tially remove the limitations on RGBD SLAM systems (sensor limitations) while keeping some of the advantages it currently holds over Monocular SLAM systems (true scale, robust relocalization, and rotational robustness).

In order to limit the scope of this project, instead of directly attempting to develop a complete Monocular SLAM system with monocular depth estimation and RGBD SLAM algorithms, this project explores if it is possible to use monocular depth estimation for accurate relocalization on models built with a RGBD SLAM system. This is a reasonable starting point because it directly tests two of the key advantages of current RGBD SLAM systems, true scale and accurate relocalization. The work is also motivated by the potential to enable systems that require monocular cameras to localize themselves on prebuilt maps, examples of which will be provided in the following section.

2.5 Significance: Lightweight Relocalization

If successful, this work would directly enable low power and lightweight monocular navigation through pre-mapped environments. This could be useful for an augmented reality application in a museum, for robots to navigate through a house, or for a swarm application where one robot might make a map for others to follow. Before creating and testing these applications, the problem of finding out just how suitable estimated depth maps are as replacements for the true depth maps provided by a depth sensor must be solved.

2.6 Problem: Estimated Depth Map Suitability

As part of the ongoing investigation into the creation of ever more accurate depth maps from single RGB images, many metrics have been used to assess the accuracy of the maps. Although metrics such as the Root Mean Square Error (RMSE) are useful for providing comparisons between methods, they tell very little about the suitability of the maps for use in SLAM applications. In fact, only one other work that this author is aware of has even tried to use the produced depth maps for any applications at all (Laina et al. 2016). This work attempts to address the suitability of current state of the art depth estimation algorithms for use as replacements for the depth sensor in RGBD SLAM systems. As mentioned above, in order to pose the problem in a limited, concrete, and quantitatively addressable form, this scope is limited to investigating how well these depth maps can be utilized for relocalization with RGBD SLAM algorithms on pre-built models.

2.7 Research Question

This work attempts to answer the question, can a state of the art monocular depth estimation algorithm be integrated with a traditional RGBD SLAM system to perform effective relocalization on an existing 3D model? It aims to evaluate the efficacy of the estimated depth maps and explore how they might be useful for SLAM applications. The objectives are as follows:

1. Integrate a RGBD SLAM system with a CNN depth map estimator
2. Evaluate the system effectiveness for relocalization on a model built separately with a RGBD SLAM system
3. Improve the system accuracy for relocalization

2.8 Thesis Outline

In the Introduction chapter, an overview of SLAM is provided as well as a background for the specifics of the monocular relocalization problem this thesis addresses. Subsequently in the Introduction, to provide some context the differences between the current state of the art in 3D and Monocular SLAM were explained and the concept of monocular depth estimation was introduced. It was then made clear that the motivation for combining monocular depth estimation with RGBD SLAM algorithms (which would then be a monocular SLAM system) was that it might overcome the deficiencies of both Monocular and RGBD SLAM. Next it noted that despite the obvious application of the depth map estimators, very little has been done toward actually testing or utilizing them for this or any other practical purpose. Finally, the limited scope of this project was outlined through the aims and objectives and the question was posed, can a state of the art monocular depth estimation algorithm be integrated with a traditional RGBD SLAM system to perform effective relocalization on existing 3D models?

In the Literature Review chapter, a deeper understanding of Visual SLAM generally is developed through a comparison and critical analysis of the published works related to the major algorithms in the field. A deeper dive is taken into the problem of the pose estimation and relocalization. Next, depth estimation using hardware, geometry, non-parametric techniques, and deep learning is analyzed and categorized. Finally, the gap in research into the utility of estimated depth maps is shown and the closest relevant

works are compared.

In the Research Methods chapter, the objectives are reviewed and a walkthrough of the entire project is undertaken. Specifically, the preliminary investigation and research is described, a detailed description of the system implementation and experimental setup is given, how the results are collected and evaluated, and finally what steps were taken to attempt to improve the results.

In the Results chapter, the results of the experiments run are presented and analyzed. The distribution of pose error for depth map estimator-RGBD SLAM system across six separate scenes is presented. The investigation into the source of the high error through an analysis of the depth maps is also presented. Finally, the attempts to improve performance by fine-tuning the depth estimation network are described and the small improvement is shown.

In the Discussion chapter, the results are further explored and the argument that the objectives have been met is made. The relocalization results are compared to those in the literature and the conclusion that the system is a failure for relocalization is presented. The fundamental limitations of the depth maps are explored. Finally, a reflection on the limitations of the scope of the work and suggestions for future work is presented.

In the Conclusion chapter, the contributions of this work are made clear. The work rules out the use of current state of the art depth map estimators with RGBD SLAM algorithms and presents a method for testing the real-world utility of future estimated depth maps.

3 Literature Review

3.1 Introduction

As established in the introduction, this research fundamentally seeks to answer the question: can a state of the art monocular depth estimation algorithm be integrated with a traditional RGBD SLAM system to perform effective relocalization on an existing 3D model? To answer this question, it is first necessary to understand the history and state of the art of both SLAM algorithms and depth estimation techniques. Towards that end this chapter is divided into two main sections, the first of which evaluates the state of published literature on visual SLAM, the second of which evaluates the literature on monocular depth estimation. The visual SLAM section is broken into two parts, a thorough examination of general SLAM techniques and a deeper dive into how the problem of pose estimation and relocalization has been addressed. The depth estimation section starts with a brief look at hardware based approaches to depth measurement to add context, and then examines the progression from geometric reasoning techniques to non-parametric depth reconstructions to deep learning depth estimation. The final section will conclude the chapter by reviewing the most relevant literature and contrasting it with this work, highlighting the gap this work will cover.

3.2 Visual SLAM

Over the last three decades the visual SLAM problem (or simply, “SLAM”) has enjoyed tremendous attention from both the computer vision and robotics communities and seen large improvements. Due to both algorithmic and hardware advances, SLAM systems have gone from poor performance on small workspaces to robust performance at building or even neighborhood scales. In their taxonomy of past, present and future SLAM, Cadena et al. (2016) divide the history of SLAM into a *classical age* from 1896 to 2004 and a *algorithmic-analysis age* from 2004 to 2015. Although some overlap in both directions makes the distinction not quite as clear cut as the distinction would make it seem, the classical age covers probabilistic formulations for SLAM such as Extended Kalman Filters, Rao-Blackwellised Particle Filters, and maximum likelihood estimation; whereas the algorithmic analysis age formulated SLAM as a maximum a posteriori estimation problem while using factor graphs and bundle adjustment to reason about variables. The earlier work in the classical age identified and focused on the challenges of efficient and robust

data association whereas later work in the algorithmic-analysis age would investigate fundamental properties of SLAM such as observability, consistency, and convergence. Although there is some overlap between approaches, this review will follow a similar line as Cadena et al. (2016) and draw the distinction between filter based approaches and graph based approaches to SLAM.

There are many possible ways to divide up the sub tasks that make up a SLAM algorithm, but one of the most helpful was introduced by Cadena et al. (2016), to divide a SLAM algorithm into a “front end” and a “back end”. The front end abstracts the sensor data into a usable form and will use that data for tracking the sensors location with respect to the model or the prior frame as described in the introduction. This part of the system is dependent on the sensor used. The back end will perform inference on the abstracted data to further build up the map and is therefore sensor independent. As noted in Taketomi et al. (2017), in addition to the initialization, tracking, and mapping modules, state of the art SLAM systems will also typically require a relocalization module for recovering when frame to frame tracking is lost and a global map optimization module for adjusting the inevitable drift. Global map optimization is typically part of the back end and relocalization is typically part of the front end, but because of the projects focus on the relocalization problem, it will have its own subsection following the subsections on the back and front ends.

3.2.1 Back End

The back end is primarily about the mapping and global map optimization modules. Techniques for these two modules can be divided into two categories: filter based and graph based. As noted in Strasdat et al. (2012), the key difference between the two is that filtering methods summarize the information learned over time from every frame as probability distributions. Whereas graph based methods will keep all the prior information explicitly available for global bundle adjustment but only from a few keyframes.

As discussed above, Cadena et al. (2016) notes that filter based methods were primarily used from 1896 to 2004, after which they were mostly discarded for graph based methods. Although still very limited, one of the earliest effective SLAM systems was called MonoSLAM and has been identified as representative of filter based SLAM techniques (Davison 2003, Cadena et al. 2016). Davison (2003) describe how their system use an extended Kalman filter to simultaneously estimate 3D structure and camera motion. Here,

the camera pose and feature locations are kept as sequentially updated probability distributions. Despite the fact that it could not relocalize properly and that the computational expense would increase in proportion to the size of the environment, MonoSLAM was one of the first true SLAM systems.

The current standard formulation of a SLAM back end is graph based. Although it only transitioned to popularity in 2004, the seminal paper on the approach was Lu & Milios (1997), in which a study of consistent registration of several poses instead of just one was done using maximum likelihood. This paper proved that it was possible to build a more consistent model using this approach, but was severely limited by a lack of efficiency. Subsequently, when this approach was taken by later systems such as Klein & Murray (2007), the computational load was split into different threads, enabling the system to run in real time. Thus, this new system was called Parallel Tracking and Mapping (PTAM). Because of the separated tracking and mapping threads as well as the use of keyframes, PTAM could use Bundle Adjustment (BA) with thousands of points in a map while still running in real time. However, because of the number of parameters, the system's BA could get stuck in local minima and fail at loop closure. While there have been many other graph-based systems since PTAM, the current state of the art system is ORB-SLAM. As of this writing, the work of Mur-Artal et al. (2015) and their follow up in Mur-Artal & Tardós (2017) offers the most complete feature based visual SLAM system.

3.2.2 Front End

The front end of a SLAM system refers to how the SLAM system performs the data extraction for a given frame and the tracking between frames. As described in Taketomi et al. (2017), the front end can be broken down into three categories of methods: feature based, direct, and RGBD. Although each of these methods have been used to develop very successful slam systems, they each have different strengths and weaknesses and work best under different scenarios.

Feature based SLAM algorithms employ careful feature detectors and descriptors on each image the system receives, and then compare the location of those features to matching features in either the prior frame or the global map. This can provide stable results in richly textured environments but provide poor results in low texture environments. In either case the resulting maps are typically very sparse as the most distinctive features tend to be corners, which don't appear densely in an image. MonoSLAM, PTAM, and

ORB-SLAM are all examples of feature based approaches (Mur-Artal et al. 2015, Klein & Murray 2007, Davison 2003).

Direct SLAM algorithms avoid abstracting hand-crafted features and are therefore also called featureless approaches. Most commonly, photometric consistency is used with pixel gradients on an input image and is compared with either the prior image or synthetic views on the map. This has the advantage of enabling dense tracking on a monocular camera as in DTAM: Dense Tracking and Mapping in Real-Time (Newcombe, Lovegrove & Davison 2011). Subsequently, Engel et al. (2014) found they could improve performance by switching to semi-dense visual odometry and only look at areas of high intensity gradients in LSD-SLAM: Large-scale direct monocular SLAM.

RGBD SLAM algorithms take advantage of the fact that the 3D structure of an environment can be directly obtained using RGBD cameras. This both simplifies the front end immensely and enables the map to have an accurate scale as well. A simple RGBD algorithm often only uses an Iterative Closest Point (ICP) algorithm to estimate the camera motion and then may directly combine the depth maps from the sensor into the global map. Since the release of cheap RGBD sensors in 2013 there have been many iterative improvements to RGBD SLAM, but most stay relatively true to the principles outlined above and established in Newcombe, Izadi, Hilliges, Molyneaux, Kim, Davison, Kohi, Shotton, Hodges & Fitzgibbon (2011).

3.2.3 Relocalization

Having reviewed and analyzed the elements of SLAM systems, it is now possible to take a deeper dive into the specifics of Relocalization, as this is the key challenged being used to test the effectiveness of monocular depth estimation combined with SLAM. The relocalization module of a SLAM system is utilized when tracking on the map has been lost or when a sensor comes back to a model built previously, and it is necessary to find one's position relative to a model of the environment from scratch. The relocalization module is typically made up of two elements: matching and pose estimation. Recalling the example from the introduction, matching refers to locating what part of the scene or model the current picture is looking at. Pose estimation is then lining up the picture precisely with the scene so as to find out the exact location and angle from which the picture was taken. Many methods for mapping have been developed however they can generally be categorized as either visual or metric methods. Once matching has been

accomplished there are two standard methods for estimating the pose, Perspective *n* Points and Absolute Orientation. Additionally, a few techniques utilizing deep learning have been created to estimate the pose without requiring matching. All of these methods and the related work will be explored below.

Within the matching module of most SLAM systems there is one of two categories of matching methods employed: visual methods and metric methods. Visual methods attempt to match keypoint descriptors from a given frame to either the model or to a keyframe. Those that attempted to match image keypoints with model keypoints had the advantage that they were able to recover poses that were significantly different from those in the map, but had limited scalability as the number of points grows rapidly with map size. Early examples of this approach such as Williams, Smith & Reid (2007), Williams, Klein & Reid (2007), and Chekhlov et al. (2008) focused mainly on improving the keypoint descriptors for faster identification and comparison. Li et al. (2010) added an interesting innovation when they flipped the algorithm on its head, matching parts of the map to the image instead of the image to the map, improving speed and accuracy. Later methods continued to add improvements to the speed and accuracy of the visual, image to map matching (Dong et al. 2012, Lim et al. 2012, Straub et al. 2013), but as noted in Li & Calway (2015) these methods continue to suffer from scalability and illumination problems.

The alternative visual approach, matching an image to a keyframe and deriving the pose from the keyframe, improved speed and does not have the same issues with scalability, but is limited in accuracy and the effective range of relocalization away from the original trajectory. Methods in this vein started by exploring different methods for matching images quickly and efficiently (Irschara et al. 2009) with representations such as lines (Reitmayr & Drummond 2006) or directly taking the sum-squared-difference of low resolution input frames and keyframes (Klein & Murray 2008). Further work in this area would investigate utilizing synthetic frames around a model, motivated by the dense maps provided by RGBD cameras (Jaramillo et al. 2013, Gee & Mayol-Cuevas 2012).

The alternative to visual matching methods are metric matching methods. Metric matching methods compare three dimensional features instead of visual, two dimensional features and thus require a 3D camera. This method has the benefit of ensuring 3D consistency in the matched points, for more effective and accurate matching. Because of the relatively late popularity of RGBD cameras this method has a smaller body of published literature, but improvements have been made from the original work (Martinez-

Carranza et al. 2013) to the use of pairwise geometry in (Li & Calway 2015). The current standard for relocalization is (Li & Calway 2016), in which a method for utilizing both visual and metric, or both 2D and 3D, is introduced, providing a very robust relocalization.

Compared with matching, the pose estimation part of a relocalization module has been relatively more standardized. The original method for pose estimation is called Perspective *n* Points (PnP). It solves for the perspective of a 2D camera frame on a 3D model given *n* matching points. Because there are six degrees of freedom and each matching point can constrain at most two degrees of freedom, at least three points are required to constrain all six degrees of freedom. Unfortunately, with only three points, four solutions are still possible when solving the equations. Consequently, four or more points are often used to find the pose.

The Absolute Orientation (AO) problem seeks to match 3D points on a RGBD frame to 3D points on a model. This can be done with three or more points as well, but is easier to solve with more points than PnP, making it more robust, though it does require a depth sensor. As noted in Alismail et al. (2010), the PnP method is more robust in situations in which there is some uncertainty about the depth values, such as when using a stereo camera; but generally, when reliable depth values are available, AO will outperform PnP.

The final and most recent method of pose estimation that has shown some promise is machine learning depth estimation. The first of these methods used a randomized tree classifier for keypoint recognition (Williams et al. 2011). It produced reasonable results but had a high memory footprint and failed to modify the uncertainty of the pose. Guzman-Rivera et al. (2014) improved upon this by training a random forest predictors to produce hypothesis poses and select the best one. This had better accuracy and a model for the uncertainty but it still required representative frames to choose from. The latest along these lines uses a CNN to directly regress the 6D pose of an image (Kendall et al. 2015). The advantages of this method are that the network size doesn't grow with the area that is modeled and that it could easily be retrained for other scenes, including across environment types (outdoors to indoors), but its disadvantage is its relatively low accuracy.

3.3 Depth Estimation

All SLAM systems inherently require some sort of depth estimation or measurement. This is done through a wide variety of methods. There are approaches based on active sen-

sors that utilize a projected light source to semi-directly measure the distance to objects in the environment. There are approaches based on passive sensors that utilize either the distance between two stereo images or the motion between images in a sequence of images. The sensor based approaches avoid the fundamental ambiguity of a single monocular image by adding more information from other sources. Recently there has been success at finding the geometry from just one RGB image, despite the inherent ambiguity. Techniques in this field have gone from using geometry to non-parametric sampling, to deep learning with increasing accuracy. In this section, we will take a brief look at sensor based techniques that are the current standard of SLAM systems and then an extended look at the literature related to monocular depth estimation.

3.3.1 Sensors

The sensors used for SLAM can be divided into active and passive categories. Active sensors project light or lasers onto the environment and then use the reflections to calculate the depth of each pixel in the image on a per image basis. Passive sensors such as RGB cameras or stereo use multiple images to find the depth by comparing features or gradients between the two images. This has been covered in the visual SLAM section, so here only the active sensors will be further explored. Additionally, because they produce depth maps for single images, active sensor based approaches to estimating depth make for a better comparison to the estimation techniques that follow.

Active sensors for creating depth maps can be broken down into three categories: Time of Flight (TOF), Structured Light (SL), and laser based. As described in Li (2014), TOF cameras will illuminate the environment with a modulated light source and based on the phase shift of the reflected light the distance is calculated. This modulation is either done through frequency modulation or by pulsing the light. SL sensors can be considered a form of active stereo vision. As described in Sarbolandi et al. (2015), a pattern is projected onto an object and the deformations in the pattern caused by the structure of the object are used to calculate its geometry. SL systems are simpler than TOF but are also slower. Both sensors typically work indoors, for smaller depth ranges. Laser based systems utilize a laser range finder for outdoor scenes with high depth range, accuracy, and reliability. They are also larger in size, require more power, and have moving parts for scanning the scene (Foix et al. 2011). As has been said before, these sensors can add major benefits to SLAM through accurate, dense depth maps but come with their

own technical limitations and environmental restrictions. Attempts to go beyond these restrictions with monocular depth estimation will be investigated in the following three sub-sections.

3.3.2 Monocular Geometry

Monocular scene reconstruction is an inherently ambiguous task. Each RGB image has an infinite number of potential ground truth structures that could cause that image. As a simple example, consider the fact that one could place a postcard in front of a camera and make a flat surface (the card) look like any shape desired. Approaching this ambiguous problem, early work began by making simplifying geometric assumptions. In Hoiem et al. (2005) it was assumed that the scene was made up of three components: ground, sky, and vertical. The scene was then cut and folded along the lines that divided the ground, sky, and vertical to create a 3D shape. They could automatically produce a somewhat realistic reconstruction of a scene, if it was outdoors and cleanly divided into ground, sky, and vertical components. Other works followed up with a similar approach, blocking city scenes using Manhattan (cubic shaped) and other physical assumptions (Gupta et al. 2010). Fouhey et al. (2014) is one of the most recent attempts at this sort of reconstruction, their work assumes that indoor scenes are made of planes that meet at convex and concave edges and corners and can be folded together into a coherent scene, much like Hoiem et al. (2005). The problem with these methods, besides their relatively poor accuracy, is that the geometric reconstructions were scale-less and provided little practical value for understanding the scenes.

3.3.3 Monocular Non-Parametric Techniques

Somewhat ironically, it would be the advent of a cheap and available depth sensor that was to cause the first major gains in monocular depth estimation. This was because with the advent of a cheap depth sensor, databases with RGB images and their corresponding true depths could be created. The non-parametric techniques described here take advantage of the database to match new monocular images with depths from the database and warp those depths to fit the new image. The seminal paper to introduce this idea was Karsch et al. (2012), though they also used optical flow from multiple frames of video to enhance the accuracy of the warping. This was improved upon by Liu et al. (2014), which formulated the depth estimation as a discrete-continuous optimization problem on

superpixels in the image. One of the most recent works on this method introduced using a depth dictionary, trained on a database, instead of referencing the database itself. This improved both the efficiency and accuracy. These data driven methods were far more accurate than the geometric methods that came before them, but were still bulky and too slow for real time applications, and not accurate enough for SLAM applications.

3.3.4 Monocular Deep Learning

Within the last several years there has been an explosion of deep learning algorithms for computer vision, driven by the large datasets now available and the fast, cheap computation available on the modern graphics card. Monocular depth estimation has not been excluded from this explosion, and the results are impressive. The seminal paper on machine learning depth estimation comes from Saxena et al. (2009), training a Markov Random Field to find the 3D location of patches on the Make3D dataset. One of the key disadvantages of the patched approach that was taken is that the predictions were made locally and lacked the context needed to generate realistic outputs. What may be considered the other seminal paper for this approach utilized deep learning with a multiscale Convolutional Neural Net (CNN) to predict the depth (Eigen et al. 2014). The authors then extended their work to show that the same architecture could be used to predict depth, normal, and semantic pixel labels for images. Although the resolution was low and the accuracy was limited, these works showed remarkable gains, benchmarking on the NYU test set with less than one meter error on average for each pixel in the image for the first time.

Following the seminal works there have been many papers offering improvements through novel architectures and training schemes. Liu et al. (2015) added a conditional random field to the CNN to improve the smoothness of the estimated depths. Fácil et al. (2016) fuses the estimated depths from single images with a Multiview depth algorithm to again achieve higher accuracy, though it is difficult to compare their results objectively with the single view algorithms. Laina et al. (2016) achieved higher accuracy through the use of a deeper network and a novel up projection method. The network was also shared freely online and was used for this work. Two novel methods for unsupervised training of a depth estimation network were developed in Godard et al. (2016) and Garg et al. (2016), utilizing stereo pairs of images and thus making training data easy to acquire.

Since this work has been underway, several other methods have been published that

continue to show small, incremental improvements on the state of the art. In Xu et al. (2017), outputs from multiple layers of the CNN are fused using a conditional random field in order to better extract the hierarchical information and improve the smoothness of the results. In Cao et al. (2017), instead of directly regressing the depth of each pixel, the depth estimation is treated as a classification task where each label is a bin of a range of depths. This also enables the estimation of the certainty of each label, which other networks fail to provide, at the cost of resolution lost due to the discretization of the depths. Common to all of the methods since Laina et al. (2016) are realistic looking, true scale depth maps with varying degrees of pixel wise accuracy. The maps are blurry and fail to capture small detail, but do provide a realistic structure with RMS pixel depth error around 0.5 meters.

3.4 Conclusion

At the outset of this work, the best publicly available depth estimation network was Laina et al. (2016). In the paper describing their results, they, like most other papers on depth estimation, describe their results in terms of the relative error, the Root-Mean-Square (RMS) error, and the log error. Unlike other papers on depth estimation, they also briefly highlight the potential for using their depth estimates in a SLAM application and show images of a scene reconstructed using their depth estimates. Unfortunately, they do not provide any metrics for the quality or effectiveness of their SLAM system. This work was the first to attempt to utilize the estimated depths for any practical purpose, and highlighted the need to evaluate estimated depth map utility. In a follow up work, published after the bulk of this work, the same authors introduce CNN-SLAM (Tateno et al. 2017). This SLAM system synthesizes predicted depth maps with the depths given by traditional monocular SLAM. Their paper also demonstrates very impressive results, showing true scale reconstruction, tracking during rotational movements that other monocular SLAM systems fail at, and even including semantic reconstruction. CNN-SLAM demonstrates impressive results, and validates the potential use for estimated depth maps, but it differs from this work in two key ways. The first is that in CNN-SLAM the estimated depth map is used to supplement a monocular SLAM system, not as a replacement for the true depth maps in a RGBD SLAM system. The second is that this work focuses on relocalization, which Tateno et al. (2017) do not cover. So though Tateno et al. (2017) validate the potential utility of depth maps, they do not answer the questions posed here.

Additionally, as explored in the pose estimation section, this work is not the first to attempt monocular relocalization on a 3D model. Though our method is unique and our motivation is different, Irschara et al. (2009) and Jaramillo et al. (2013) also perform monocular relocalization on 3D point clouds. The closest of these is Jaramillo et al. (2013), in which the relocalization is done on dense point clouds created with a depth sensor. Although they report very low translation and rotational error (4cm and 1degree), the impact of their work is limited by the fact that they utilize the prior frame's pose as a prior for relocalization, in essence actually doing more tracking than true relocalization. In conclusion, there has been little work to study the utility of estimated depth maps for SLAM and no work at all on their utility for replacing depth sensors in RGBD slam algorithms. The test case for this problem, monocular relocalization on pre-built maps has been studied before, but the closest instance of comparison has a faulty method which gives better results than might otherwise be expected, limiting its effect for comparison but still providing a decent benchmark.

4 Research Methods

4.1 Strategy

As described in the Introduction, this work attempts to answer the question: can a state of the art monocular depth estimation algorithms be integrated with a traditional RGBD SLAM system to perform effective relocalization on existing 3D models? It aims to evaluate the efficacy of the estimated depth maps and explore how they might be useful for SLAM applications. The objectives are as follows:

1. Integrate a RGBD SLAM system with a CNN depth map estimator
2. Evaluate the system effectiveness for relocalization on a model built separately with a RGBD SLAM system
3. Improve the system accuracy for relocalization

To accomplish these objectives and answer the research question, this work followed an experimental research strategy. The hypothesis formed at the start of this work was that monocular depth estimation combined with RGBD SLAM algorithms *is* effective at relocalization, and to test this hypothesis this work undertook four phases: investigation, implementation, evaluation, and improvement. Each of these will be explained in detail below. This work is both qualitative and quantitative in nature. It offers both a quantitative experimental result (the pose error) and a qualitative judgement of if that result constitutes "effective" relocalization, supported by that literature.

4.2 Investigation

This work requires a substantial amount of knowledge about SLAM and monocular depth estimation. In order to effectively implement a system that was capable of testing the hypothesis, a significant portion of the project was spent learning about SLAM algorithms, monocular depth estimation, and other deep learning techniques. The primary sources for this were free online courses and the published literature. The published literature in engineering journals was the primary source of this author's education because all the current state of the art techniques for SLAM and depth estimation are found there and it was easy to trace back the history through the references. This was supplemented with free online classes published through Youtube to answer questions about specific topics or methods as they arose.

4.3 Implementation

In order to accomplish the first objective, integrating a RGBD SLAM system with a CNN depth map estimator, it was decided to use open source methods rather than attempting to rebuild both systems from scratch. This was done because of time constraints on the project and because recreating other authors' work in this area was not essential to proving or disproving the hypothesis. It was also decided that though it may have been possible to integrate the two systems into a single real-time system, this was also unnecessary to the project. Consequently, an offline pipeline was developed for feeding data between the SLAM system and the depth estimator.

The SLAM system used for this work came from Li & Calway (2015), a windows application written in C++. As discussed in the literature Li & Calway (2015) is a fully functionally SLAM system that has been optimized specifically for fast RGBD relocalization and shows comparable results to state of the art. It was chosen because it is a representative of the metric relocalization methods that rely on depth maps for relocalization, thus giving the best test of whether the estimated depth maps were truly effective. Using a visual matching method might have better performance when the depth maps are not as accurate, but would have failed to test if the depth maps that were used were actually performing well for relocalization.

The depth map estimator used in this work came from the open source implementation of Laina et al. (2016), which has been published in both Matcaffe and Tensorflow. This was chosen because at the time of this work it was the current standard for depth estimation, with the lowest error of the published works. Laina et al. (2016) had also chosen to open source their code which allowed for quick implementation.

The SLAM system and depth estimator were both installed on a windows laptop with a GTX 680 GPU and a Structure Sensor from Openni. Because this work is not concerned with online performance or timing, the hardware specifics have no impact on the results. This hardware and software setup was chosen because of a mixture of easy availability and it met the performance standards needed to truly test the hypothesis.

The experiments were performed in the following steps. The first step consisted of building the model. Because of the limitations of the depth sensor and limitations in the SLAM system used, the model was restricted to indoor scenes approximately 5 meters cubic. Because of this, the corners of the lab where chosen as the model locations for maximum volume usage. Once the models were built and recorded, for the second

step a video sequence of 300 RGBD frames was recorded through the SLAM system, covering poses from all around the workspace. The pose of each frame was recorded along with the frame while tracking was enabled, ensuring the capture of the pose ground truth. Because these ground truth poses were measured through the SLAM system they are subject to some small amount of error; however, because the error is an order of magnitude smaller than the error in the estimated frames it can be neglected. After the sequence has been recorded, in the third step the depth estimating CNN is run on the sequence of RGB frames to produce a sequence of estimated depth maps. In the fourth step, each RGB frame and its corresponding estimated depth are independently run through the SLAM system to be relocalized on the model and the pose of that frame is recorded. It is worth noting that although the frames were captured as a sequence for the sake of convenience, they are truly relocalized – meaning that each frame is treated as a lost frame, without the context of the pose of the other frames around it. Fifth and finally the ground truth pose is compared to the relocalized pose of the estimated frame and the translational and rotational error is established . This simple experiment, run on several scenes, effectively accomplishes the second objective, and the scale of the error discovered will answer the research question.

4.4 Evaluation

There are several methods in the literature related to evaluating a system's ability to relocalize. Much of the earliest work was focused simply on whether the relocalization would enable a system to continue tracking, and thus success was measured as a percentage of relocalization instances in which the system would continue tracking after a tracking failure (Chekhlov et al. 2008). Other early works focused on feature matching speed and accuracy, defining success by how fast the largest number of points could be matched (Lim et al. 2012, Straub et al. 2013). Common to most works was an emphasis on speed and efficiency, in large part due to the smaller computational power available to systems at the time and the need for those systems to run in real-time (Chekhlov et al. 2008, Li et al. 2010, Sattler et al. 2011, Lim et al. 2012, Straub et al. 2013).

Although the need for speed and efficiency is still prevalent in recent publications, the focus has now shifted to relocalization pose accuracy. Unfortunately, a standardized metric for relocalization pose accuracy has not been adopted, but common to each of the measures is essentially an evaluation of the difference between the ground truth pose

and the pose produced as the output of the relocalization system. This has been done as a percentage difference, an absolute difference, and a RMS difference between the estimated translation (3D coordinates) and rotation (3D orientation). This work adopts the RMS error between the given ground truth (rotation and orientation, separately) and the relocalized pose. This method was chosen because it presents the average error on a given axis of measurement for both rotation and translation, providing two error metrics for each frame that is relocalized. These metrics are displayed and analyzed in a variety of ways (e.g. line, histogram, and box plots) across the sequence, to provide the greatest level of insight into how and why the system performs as it does.

To further understand the error in the relocalized poses, a thorough evaluation of the root causes of the error was undertaken. This involved an analysis of the individual depth maps produced by the CNN estimator and a comparison of those maps to the Kinect maps. Similar to analysis available in the literature, this work starts with a simple visual inspection of the estimated depth maps and the relative error in those maps compared to the truth provided by the Kinect. This is helpful but not sufficient to understand the sources of pose error. Going beyond the analysis available in the literature, this work also analyzes the error of the salient points in each frame, as these points are used by the relocalization module for pose estimation. Further explanation for each of these modes of analysis will be developed in the results and discussion, but at a high level, the relocalization effectiveness is measured through the RMS error in the 6D pose across all the frames for a given scene and the analysis for the error is found by examining individual frames error for all pixels and for key salient points.

4.5 Improvement

As the cause of the pose estimation error is established in the methods above, work to accomplish the third objective is undertaken by attempting to improve the relocalization. As the primary cause of the pose error in the relocalization is identified as the error in the depth maps, this work attempts to improve the estimated depth maps. Although there are a variety of techniques that can be used to improve the estimated depth maps in general (as evidenced by the recent continued progress in depth map estimation in Cao et al. (2017), Su et al. (2017), Mancini et al. (2017), Li et al. (2017), and Xu et al. (2017)) this work takes a different tactic. In part because recent improvements were not available during the execution of this work and in part because this improvement will continue to

apply to future developments of depth estimation; this work attempts to improve the depth estimation for a given scene by fine tuning the estimation network on the scene which localization will occur. The original network used in the work was trained on the NYU v2 dataset. Although this dataset is similar in some respects to the scenes used in the work, it is possible to improve the performance of the network on these scenes by additional training on real or synthetic views of the new scenes. This was the simplest method available for improving performance of the network while keeping the results relevant to the research aim of this work. It is also still plausible that for real world usage, training the network on the model that has been developed might be a viable strategy to improve performance for relocalization with a monocular camera on that model. The performance of the fine-tuned network system for relocalization was then evaluated in the same fashion as the original, described above.

4.6 Conclusion

This chapter had elaborated the simple yet effective experiment by which the research question is answered and the objectives are accomplished. It describes the implementation and evaluations strategy that directly answer the research question. In addition to accomplishing the aims and objectives, this strategy has the advantage of being simple and manageable in the allotted time. However, it also has several limitations. One of the key limitations of the strategy is applicability to other environments. Because this work uses an RGBD SLAM system with relatively small space constraints (5 meters cubic, indoors), it was impossible to test the performance on a larger, outdoor scene. This limits the generalizability of the findings and suggests that an outdoor test may be useful. This work is also limited in its conclusions about the utility of estimated depth maps to the state of the art at the conclusion of this work. As stated above, the state of the art is constantly improving, and future depth maps may improve to the point where they might be more useful for SLAM systems. Finally, it is worth noting that the scope of this work is limited to the utility of estimated depth maps with RGBD algorithms. This work does not provide any insight into how these estimated depth maps might perform with monocular algorithms or with algorithms explicitly designed to handle their inaccuracies. Although future work may be required to investigate beyond these limitations, within the scope of the aims and objectives this work presents a valid and reliable result. What that result is will be presented in the next chapter.

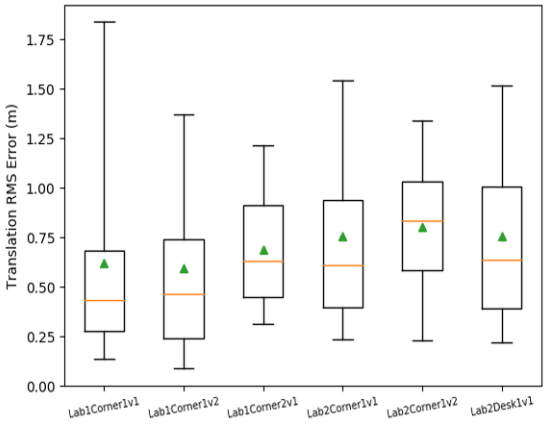

Figure 4: Translational RMS Error Across Six Scenes. For each scene the box represents 25-75%, the whiskers represent 5-95%, the yellow lines are the medians, and the green triangles are the means.

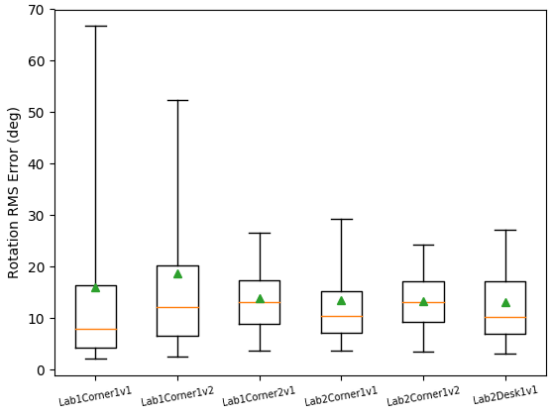

Figure 5: Rotational RMS Error Across Six Scenes. For each scene the box represents 25-75%, the whiskers represent 5-95%, the yellow lines are the medians, and the green triangles are the means.

5 Results

This chapter presents the results of the experiments and investigations conducted as a part of this work. The chapter starts with the primary finding of the relocalization experiments, then the results of an investigation into the source of the pose estimation error, and finally with the results of efforts to improve the relocalization accuracy. Here, only a quantitative description of the results is produced, a qualitative evaluation of the nature of these results is saved for the Discussion chapter.

5.1 Pose Accuracy

As described in the Methods chapter, this work conducted six experiments in which the pose of 300 lost frames were relocalized on a model of a scene using a depth estimating CNN and a RGBD SLAM algorithm. This produced a 6D pose for each frame, that could be compared to the ground truth pose, which was gathered using the Kinect depth and the RGBD SLAM system. The RMS error in the rotation and translation component was then calculated, producing a two-number error metric for each frame.

The distribution of those errors for each experiment can be seen in Figure 4 and Figure 5. These charts show that although they do vary between scenes, the majority of the error for a given axis of translation ranges between .25 and 1.0 meters and that the majority of the error for rotation on a given axis ranges between 5 and 20 degrees. Another way to describe these results is to say that on a given axis for any given frame,

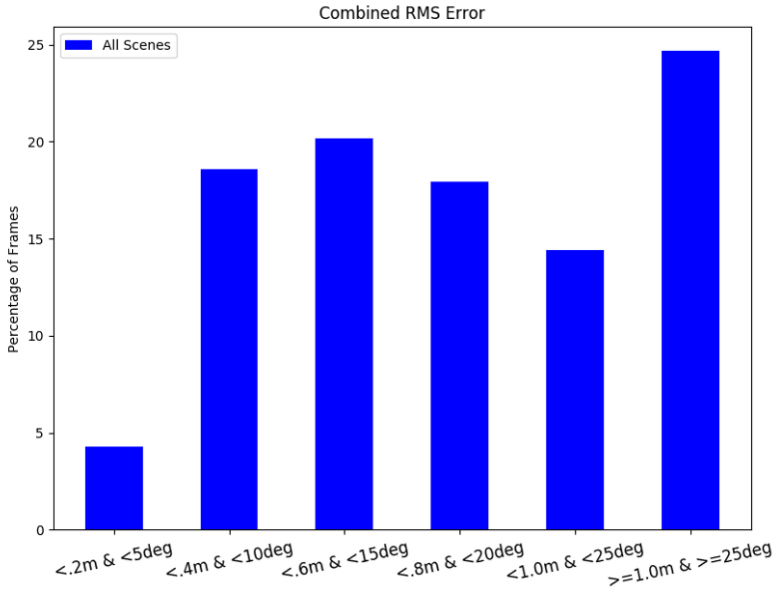

Figure 6: Error Bins for Rotation and Translation. Combined measure of error for the rotation and translation component of pose.

the calculated pose was likely *at least* .25 meters distant and looking 5 degrees away from the ground truth.

Using these two metrics independently paints a helpful but incomplete picture of the frame relocalization efficacy. It would be better to combine the two metrics towards a single metric of relocalization accuracy, because in real world applications location and orientation are both required to work together. Toward that end this work also presents an alternate representation in Figure 6, where the location and orientation error are binned together. Combining the two error metrics allows for more specific commentary about the systems accuracy, such as that at more than 1 meter and 25 degrees off target, about 25% of all frames can be considered abject relocalization failures. Between .2 meters and 1 meter and 5 degrees and 25 degrees there is a roughly uniform distribution of poor to very poor relocalization, making up about 70% frames. Finally, at less than .2 meters and 5 degrees RMS error, approximately 5% of all the frames across the experiments can be considered successfully relocalized by the system.

In addition to summaries across sequences or across experiments, it is also insightful to view the error for individual frames across a sequence. Here, Figure 7 and Figure 8 show the noisy error signal across the sequence of frames, in this case taken from experiment lab1corner1v1. Although the sequence is taken from a video, each of the frames is localized individually, and thus the error can jump dramatically from one frame

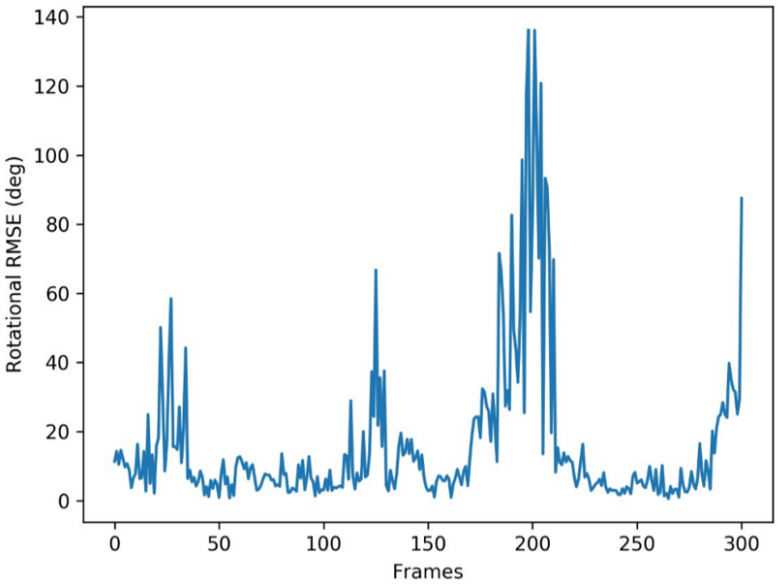

Figure 7: Rotational Pose Error Sequence. Rotational error for a sample sequence of frames. The frames are relocalized individually but the sequence is in order. Taken from sequence lab1corner1v1.

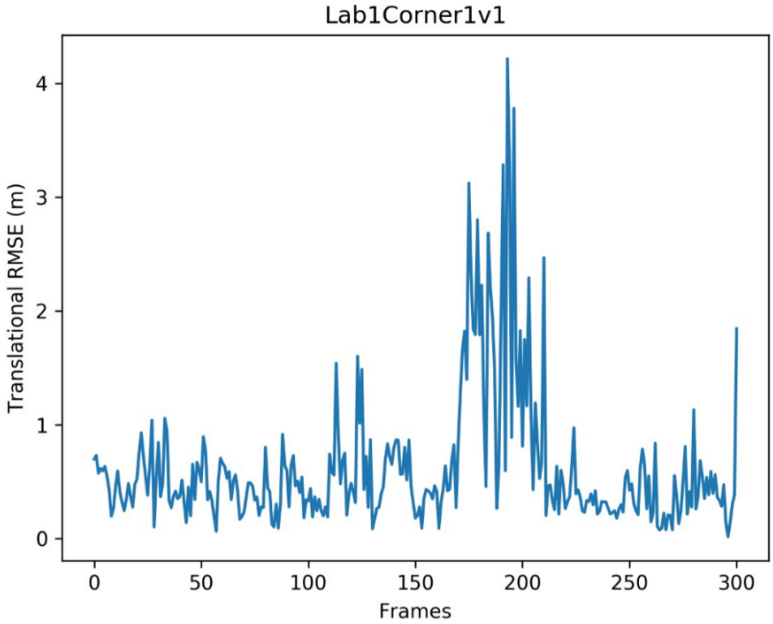

Figure 8: Translational Pose Error Sequence. Translational error for a sample sequence of frames. The frames are relocalized individually but the sequence is in order. Taken from sequence lab1corner1v1.

to the next. The clustering of frames in which the area is orders of magnitude higher than the rest suggest that these are sections of the video in which the camera was looking at an area or object that was more difficult to estimate depth.

5.2 Depth Estimation Accuracy

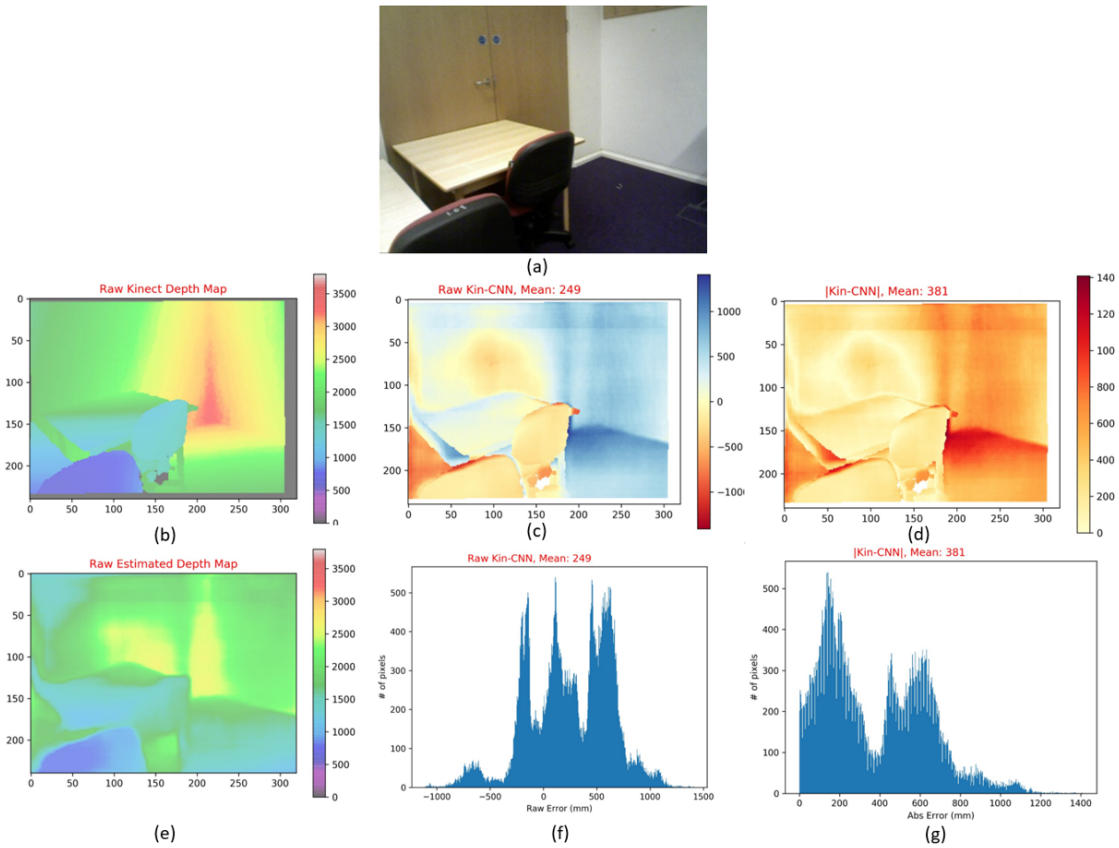

Figure 9: Kinect vs Estimated Depth Maps. Sample frame (33) randomly taken from Lab1Corner1v1 set. Top: RGB Frame. Middle Left: false color Kinect depth map. Middle Center: Estimated depth map subtracted from Kinect depth map, pixel-wise; darker is further from zero. Middle Right: Absolute difference between Kinect and estimated depth maps, darker is worse. Bottom Left: false color estimated depth map. Bottom center, histogram of pixel depth differences. Bottom Right: Absolute value of pixel depth differences.

To investigate the cause of the poor relocalization errors above, further study of differences between real and estimated depth maps was undertaken. This first involved a close visual inspection of the differences between the ground truth and estimated depths. As can be seen in the randomly chosen sample frames from Figure 9 and Figure 10, there are several visually obvious differences between the ground truth from the Kinect and the estimated frame from the CNN. The first point to note is that when comparing (b) and (e) for both figures, although the estimated depth map is structurally similar to the truth, it is blurred and many small details such as the gap between the two tables or the table leg in Figure 9 are lost. Similarly, as can be seen in both figures' (c) and (d), there is the highest error on the edges of objects, where the depth can transition sharply by several meters. Additionally, as can be seen in both figures' part (c), the depth error is not con-

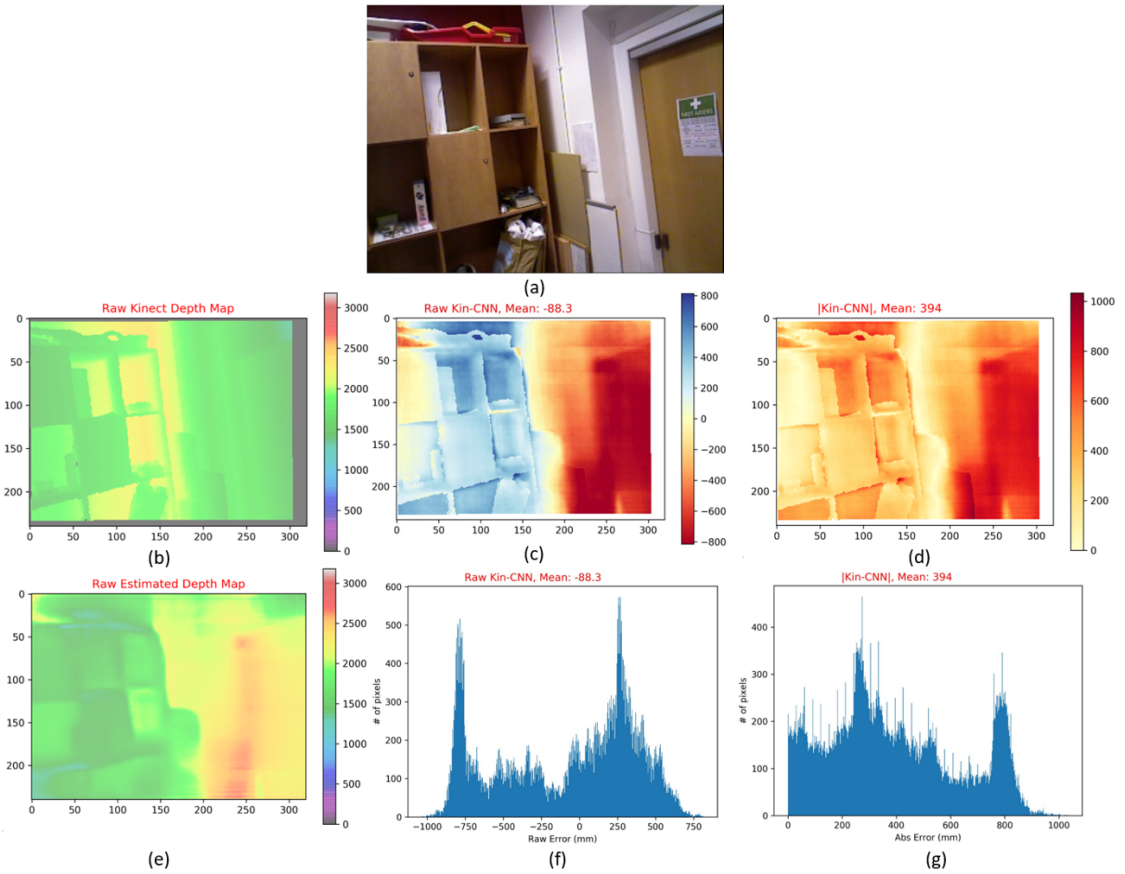

Figure 10: Kinect vs Estimated Depth Maps. Sample frame (58) randomly taken from Lab2Corner1v1 set. Top: RGB Frame. Middle Left: false color Kinect depth map. Middle Center: Estimated depth map subtracted from Kinect depth map, pixel-wise; darker is further from zero. Middle Right: Absolute difference between Kinect and estimated depth maps, darker is worse. Bottom Left: false color estimated depth map. Bottom center, histogram of pixel depth differences. Bottom Right: Absolute value of pixel depth differences.

sistent across the scene, with some objects (in red) closer than the CNN has estimated and some objects (in blue) further away than estimated. In (f) and (g) the different modes illustrate the error corresponding to the different objects in the scene. As can be seen in part (g) of both figures, the absolute pixelwise error is lumpy, but generally extends up to depth errors of .8 or 1 meter. This is quite significant (25-100%) when it is compared with the ground truth range of 0.5 to 3.5 meters.

To extend the analysis further, this work also investigates the error corresponding to the salient points used in the pose estimation for relocalization. This was done by finding the salient points in the image frames and examining the 3D distance between all combination of pairs of points projected into 3D space using the depth and a pinhole camera model. This was motivated by the fact that the relocalization algorithm uses the

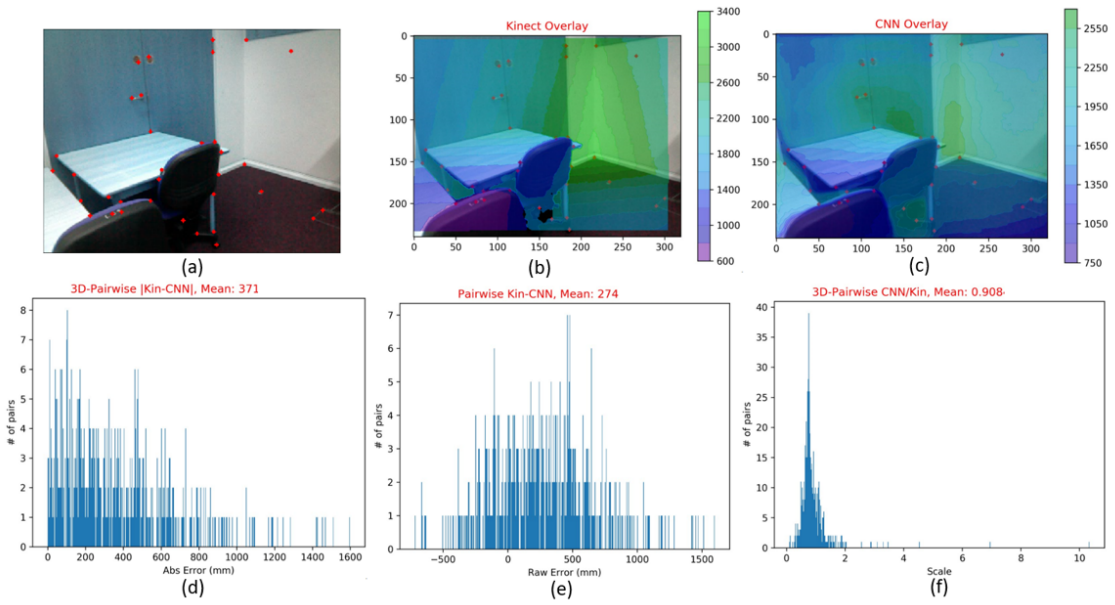

Figure 11: Kinect vs Estimated Pairwise Investigation. Sample frame (33) taken from Lab1Corner1v1 set. Top Left: RGB Frame with red dots corresponding to salient points. Top Middle: false color Kinect depth map overlaid on top of image with salient points. Top Right: Estimated depth map overlaid on top of image with salient points. Bottom Left: Histogram of absolute 3D difference between pairs of salient points. Bottom Middle, Histogram of 3D difference between pairs of salient points. Bottom Right: Histogram of scale differences between 3D pair distances.

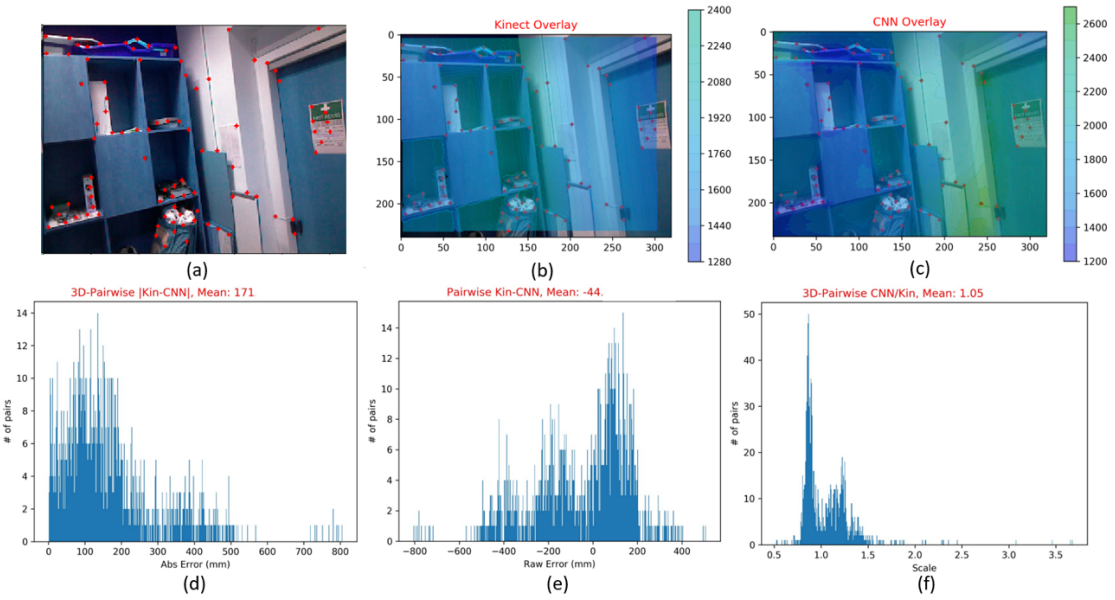

Figure 12: Kinect vs Estimated Pairwise Investigation. Sample frame (58) taken from Lab2Corner1v1 set. Top Left: RGB Frame with red dots corresponding to salient points. Top Middle: false color Kinect depth map overlaid on top of image with salient points. Top Right: Estimated depth map overlaid on top of image with salient points. Bottom Left: Histogram of absolute 3D difference between pairs of salient points. Bottom Middle, Histogram of 3D difference between pairs of salient points. Bottom Right: Histogram of scale differences between 3D pair distances.

relative geometry of the scene, not just the depth maps for pose estimation. To illustrate the results of this investigation the analysis is shown on the same frames as above in Figures 11 and 12. The overlays in both figures illustrate how well each depth map correlates to the image it is related to. As illustrated particularly well in Figure 11 (b) the Kinect depth map correlates very sharply with image structure, however it does not cover the entire image, leaving a gap on the bottom and right sides. The estimated map in (c) on the other hand, covers the entire image, but does not smoothly match the geometry as well.

For each pair of salient points in the image, the 3D distance between those pairs is calculated using both the real and estimated depth maps. The difference between those two is another measure for the structural error in the estimated depth map. Unlike in Figure 9 and Figure 10, the error in the these figures' histograms in (d) and (e) do not correspond to local image structure but instead show a more generalized error metric. Because the distance between the camera and the scene is larger than the distance between points in the scene, the average error is smaller than the pixelwise average error, as expected. The figure in (f) compares the relative sizes between the estimated and ground truth distances. The fact that the average scale factor is close to one but that the distributions is wide suggests that the error is not caused by an individual scale factor across the image, but small scale factors for local substructures.

5.3 Improvement Accuracy

As discussed in the Methods chapter, the final element of this work lay in attempting to rectify the error in the estimated depth maps through further training or fine tuning. The first attempt to improve the depth estimation utilized the sequence of frames from two scenes, the lab1corner1v1 and the lab2corner1v1 sequences. The network was then fine-tuned on these frames, until no more improvement could be made. It was then retested with all six sequences, and the results can be seen in Figure 13. Here, the pairs boxed in blue show the dramatic improvement of the relocalization on frames that the network had been trained on. In green, the small improvement on the same scene, but with a different sequence of frames is shown. In yellow, the slight improvement on scenes that were not trained on is shown. This experiment shows that it is possible to achieve real gains on relocalizing on scenes when the depth estimator is further trained on those scenes. It also shows that some of this benefit extend to scenes that are similar

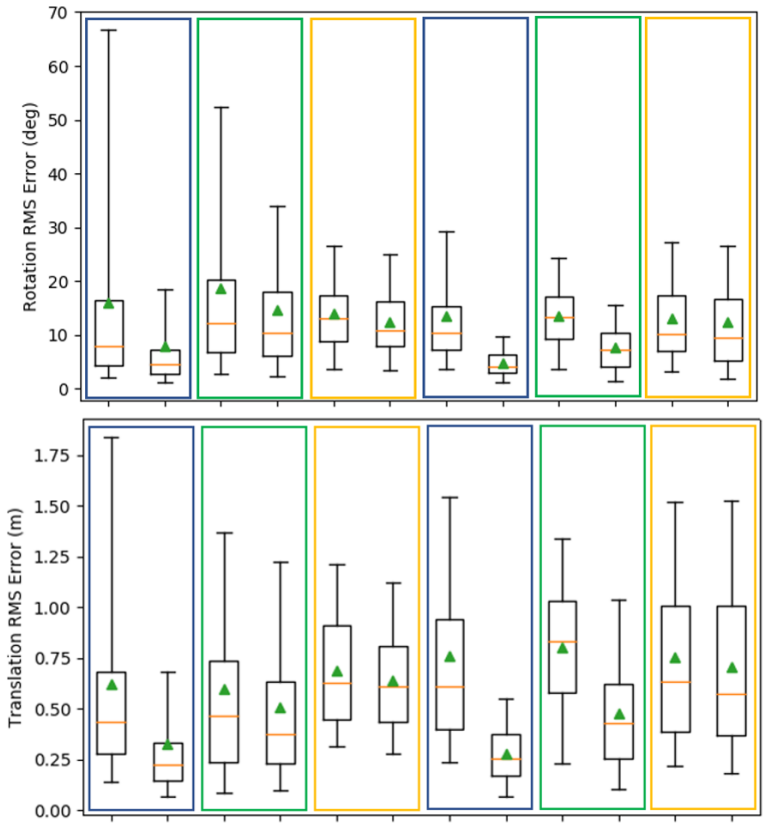

Figure 13: Rotation and Translational Improvement With Generally Fine-tuned CNN. Each pair shows the RMSE for the original (left) and fine-tuned CNN (right). Scenes highlighted in blue show the improvement when testing is conducted on the training sequence. Scenes highlighted in green show improvement for sequences from the same scene that was trained on but with different frames. Scenes highlighted in yellow show the improvement for unrelated scenes.

in nature, such as lab1corner2v1. The dramatic improvement in sequences which were used for fine tuning (highlighted in blue) is unsurprising and unhelpful, as the ground truth for these frames is already available (as they were what was trained on), rendering the estimator pointless.

The second attempt to improve the fine tuning of the depth estimator took 2000 image-depth pairs from around the lab1corner1 scene and trained the CNN using them. As can be seen in Figures 14 and 15, this method of training specifically for one scene had only modest improvements over the more generally fine-tuned network.

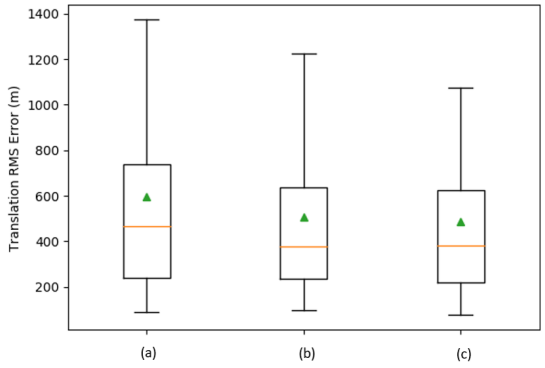

Figure 14: Translational RMS Error vs Tuning. From left to right: (a) Depth estimator trained just on NYUv2 Dataset, (b) Depth estimator fine-tuned on lab1corner1v1 and lab2corner1v1 sequences, (c) Depth estimator trained on 2000 images taken around lab1corner1 scene. Tested on lab1corner1v2.

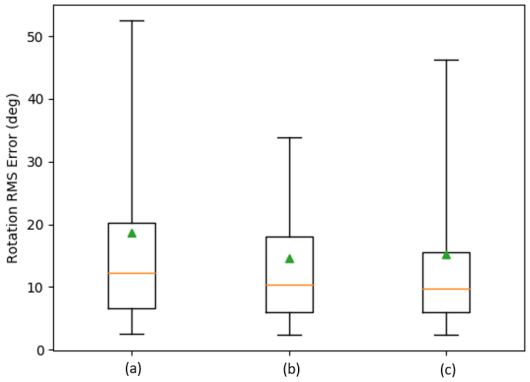

Figure 15: Rotational RMS Error vs Tuning. From left to right: (a) Depth estimator trained just on NYUv2 Dataset, (b) Depth estimator fine-tuned on lab1corner1v1 and lab2corner1v1 sequences, (c) Depth estimator trained on 2000 images taken around lab1corner1 scene. Tested on lab1corner1v2.

6 Discussion

As described in the Introduction and Research Methods, this work attempts to answer the question: can a state of the art monocular depth estimation algorithms be integrated with a traditional RGBD SLAM system to perform effective relocalization on existing 3D models? It aims to evaluate the efficacy of the estimated depth maps and explore how they might be useful for SLAM applications. The objectives are as follows:

1. Integrate a RGBD SLAM system with a CNN depth map estimator
2. Evaluate the system effectiveness for relocalization on a model built separately with a RGBD SLAM system
3. Improve the system accuracy for relocalization

In the Results chapter, this work presented and explained the outcome of the experiments defined in the Methods chapter. Here, this work will evaluate those results to answer the research question and reflect on the limitations and potential future work related to this body of work.

6.1 Evaluation

This work began with the motivation that recent work in the field of monocular depth estimation had produced reliable, relatively accurate, dense depth maps. Though the

progress in this area was evident by the improvement on standard benchmarks, none of the depth estimators had been tested for any practical purpose. This work sought to use a state of the art depth map estimator for a SLAM application both as a method for testing the utility of estimated depth maps and as a method for allowing some RGBD SLAM algorithms to be used with a monocular camera. The experiment that was set to test the Estimated Depth Map-RGBD SLAM combination was relocalization accuracy. In order to perform the experiment this work met the first objective by building a system that combined a depth map estimator and a RGBD SLAM system. Though it was only built for offline use, the system described in this work accomplishes this objective.

The quantitative answer to how well such an estimated depth map relocalization system performs is found in the Results chapter. To provide a qualitative answer to the research question, it is important to examine the results available in other relocalization literature for comparison. As discussed in the literature review chapter, relocalization accuracy is not well standardized and is reported by a variety of metrics for many different works. This work utilized the RGBD SLAM algorithm from Li & Calway (2015), in which they report (using a RGBD camera) their accuracy as a percentage below 5 cm of translational error and 5 degrees of rotational error. For the *7 scenes* dataset their performance varies between 85% and 100% success. On this work's dataset, approximately 4% of the frames lie below 20 cm of translational error and 5 degrees of rotational error. This is not a fair comparison as Li & Calway (2015) uses a RGBD camera, but it does demonstrate what should be possible with the RGBD SLAM system used by this work.

In the most closely related work to this one, Jaramillo et al. (2013) report an average error of 4 cm and 1 degrees using a monocular camera on a point cloud model of an indoor scene. The average error for this work is approximately 75 cm and 15 degrees, an order of magnitude worse. The closest work this author could find in terms of results is Kendall et al. (2015), in which the relocalized pose is directly regressed with a neural net, skipping the SLAM algorithm altogether. They report 50cm and 5-degree average error for indoor scenes, which is 25cm and 10 degrees better than this work, without a SLAM system or a 3D model. All of which is to say that the relocalization results for this system are terrible, and by most metrics would be considered a failure. The modest gains seen by fine tuning the network are not enough to change that evaluation.

The reason for the system failure lies with the error in the estimated depth maps and the geometry for pose estimation. RGBD SLAM systems utilize the fact that a true depth

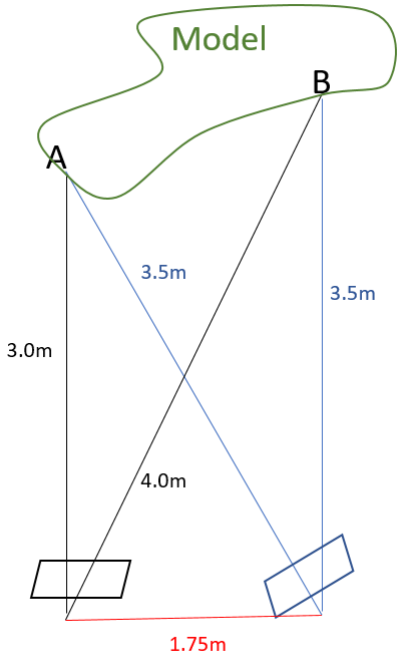

Figure 16: Pose Estimation Error Example. A and B are salient points on the model. Ground truth is in black and the estimated frame is in blue. The error is in red.

map is known and thus require that those depth maps be accurate, or at least contain salient points that are accurate. In Figure 16 a simplified 2D example shows how pose estimation can fail significantly when the error in the depths of salient points is large compared to the depths in the depth maps. Working with two key points (A and B), if the ground truth depths (in black) are 3 and 4 meters respectively, and if the error on each point is .5 meters, positive error for A and negative error for B, then the pose error for the relocalization will be 1.75 meters and 30 degrees. This is a simple example and takes the most extreme case, but it illustrates why, despite the realistic looking depth maps, the relocalization can go so wrong.

As they currently stand, the state of the art estimated depth maps are not effective replacements for the depth maps that are provided by depth sensors, despite their limitations. As noted in the Literature Review however, the state of the art is continuing to improve, and as the machine learning techniques that power these improvements are further developed, the estimated depth map may reach a point where its accuracy enables its use with RGBD SLAM techniques. For now, it seems that the current depth maps are incompatible with standard RGBD SLAM techniques. Because as the analysis in the Results chapter makes clear, the fact that the depth maps produce errors that vary

largely in both positive and negative directions for different objects in the scene forces the relocalized pose error to be large.

6.2 Reflection

This work has a very specific scope which limits the generalizability of the conclusions that can be drawn. As outlined in the methods section, this work deals with relatively small indoor scenes. The poor performance on such scenes suggests but does not confirm that the same methods would have similarly poor performance on larger scenes, either indoors or outdoors. Additionally, this work was limited in scope to testing the estimated depth maps as direct replacements for true depth maps in RGBD SLAM systems. It cannot draw conclusions about the suitability of depth maps for SLAM applications in general. Despite the errors and limitations of estimated depth maps, they still contain valuable information about scene structure that could be useful for SLAM applications. In the recently published work, Tateno et al. (2017) utilize the depth prediction with a monocular SLAM system to create a dense reconstruction at scale. This work offers some evidence that despite not being effective for RGBD SLAM algorithms, estimated depth maps contain information that is useful for monocular SLAM.

In the future, there are many potential avenues for research to continue. As has been evidenced by recent publications since this work has been completed, there is still room for improvement in monocular depth estimation, and further research along those lines may lead to depth map estimators that work with higher accuracy across a wide variety of scenes. As evidenced by the success of Tateno et al. (2017), integrating a depth map estimator with monocular SLAM algorithms leads to improved monocular SLAM, and further research along these lines could offer even more improvement.

7 Conclusion

The goal of this work was to explore and evaluate the use of monocular depth estimators as replacement for depth cameras in RGBD SLAM systems. This was accomplished by building a system that combined the depth estimator with the RGBD SLAM system and testing how effectively such a system could relocalize on pre-built 3D models. The result of this experiment was the conclusion that current state of the art depth estimators paired with RGBD SLAM systems are not effective at relocalization when compared with other monocular relocalization techniques. Thus, the final conclusion can be drawn that the monocular depth estimator is not suitable for replacing the depth sensing element in RGBD SLAM systems.

Despite the negative result, this work demonstrates a novel means of testing the practical value of estimated depth maps. Although testing the pixel-wise accuracy is helpful for comparing results against other depth map estimators, this work suggests that a more useful measure of improvement test the depth maps practical value as part of a SLAM system. This work also concludes that fine tuning a depth estimator on scenes similar to those to which it will be utilized for provides an easy boost to performance.

8 Bibliography

References

Alismail, H., Browning, B. & Dias, M. B. (2010), 'Evaluating pose estimation methods for stereo visual odometry on robots'.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I. & Leonard, J. J. (2016), 'Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age', *IEEE Transactions on Robotics* **32**(6), 1309–1332.

Cao, Y., Wu, Z. & Shen, C. (2017), 'Estimating depth from monocular images as classification using deep fully convolutional residual networks', *IEEE Transactions on Circuits and Systems for Video Technology* .

Chekhlov, D., Mayol-Cuevas, W. W. & Calway, A. (2008), Appearance based indexing for relocalisation in real-time visual slam., *in* 'BMVC', pp. 1–10.

Davison, A. J. (2003), Real-time simultaneous localisation and mapping with a single camera, *in* 'null', IEEE, p. 1403.

Dong, Q., Gu, Z. & Hu, Z. (2012), 'Automatic real-time slam relocalization based on a hierarchical bipartite graph model', *Science China Information Sciences* pp. 1–8.

Eigen, D., Puhrsch, C. & Fergus, R. (2014), Depth map prediction from a single image using a multi-scale deep network, *in* 'Advances in neural information processing systems', pp. 2366–2374.

Engel, J., Schöps, T. & Cremers, D. (2014), Lsd-slam: Large-scale direct monocular slam, *in* 'European Conference on Computer Vision', Springer, pp. 834–849.

Fácil, J. M., Concha, A., Montesano, L. & Civera, J. (2016), 'Deep single and direct multi-view depth fusion', *arXiv preprint arXiv:1611.07245* .

Foix, S., Alenya, G. & Torras, C. (2011), 'Lock-in time-of-flight (tof) cameras: A survey', *IEEE Sensors Journal* **11**(9), 1917–1926.

Fouhey, D. F., Gupta, A. & Hebert, M. (2014), Unfolding an indoor origami world, *in* 'European Conference on Computer Vision', Springer, pp. 687–702.

Garg, R., Carneiro, G. & Reid, I. (2016), Unsupervised cnn for single view depth estimation: Geometry to the rescue, *in* 'European Conference on Computer Vision', Springer, pp. 740–756.

Gee, A. P. & Mayol-Cuevas, W. W. (2012), 6d relocalisation for rgbd cameras using synthetic view regression., *in* 'BMVC', pp. 1–11.

Godard, C., Mac Aodha, O. & Brostow, G. J. (2016), 'Unsupervised monocular depth estimation with left-right consistency', *arXiv preprint arXiv:1609.03677* .

Gupta, A., Efros, A. & Hebert, M. (2010), 'Blocks world revisited: Image understanding using qualitative geometry and mechanics', *Computer Vision–ECCV 2010* pp. 482–496.

Guzman-Rivera, A., Kohli, P., Glocker, B., Shotton, J., Sharp, T., Fitzgibbon, A. & Izadi, S. (2014), Multi-output learning for camera relocalization, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 1114–1121.

Hoiem, D., Efros, A. A. & Hebert, M. (2005), 'Automatic photo pop-up', *ACM transactions on graphics (TOG)* **24**(3), 577–584.

Irschara, A., Zach, C., Frahm, J.-M. & Bischof, H. (2009), From structure-from-motion point clouds to fast location recognition, *in* 'Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on', IEEE, pp. 2599–2606.

Jaramillo, C., Dryanovski, I., Valenti, R. G. & Xiao, J. (2013), 6-dof pose localization in 3d point-cloud dense maps using a monocular camera, *in* 'Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on', IEEE, pp. 1747–1752.

Karsch, K., Liu, C. & Kang, S. B. (2012), Depth extraction from video using non-parametric sampling, *in* 'European Conference on Computer Vision', Springer, pp. 775–788.

Kendall, A., Grimes, M. & Cipolla, R. (2015), Posenet: A convolutional network for real-time 6-dof camera relocalization, *in* 'Proceedings of the IEEE international conference on computer vision', pp. 2938–2946.

Klein, G. & Murray, D. (2007), Parallel tracking and mapping for small ar workspaces, *in* 'Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on', IEEE, pp. 225–234.

Klein, G. & Murray, D. (2008), 'Improving the agility of keyframe-based slam', *Computer Vision–ECCV 2008* pp. 802–815.

Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F. & Navab, N. (2016), Deeper depth prediction with fully convolutional residual networks, *in* '3D Vision (3DV), 2016 Fourth International Conference on', IEEE, pp. 239–248.

Li, B., Dai, Y. & He, M. (2017), 'Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference', *arXiv preprint arXiv:1708.02287* .

Li, L. (2014), 'Time-of-flight camera–an introduction', *Technical white paper* (SLOA190B).

Li, S. & Calway, A. (2015), Rgbd relocalisation using pairwise geometry and concise key point sets, *in* 'Robotics and Automation (ICRA), 2015 IEEE International Conference on', IEEE, pp. 6374–6379.

Li, S. & Calway, A. (2016), Absolute pose estimation using multiple forms of correspondences from rgb-d frames, *in* 'Robotics and Automation (ICRA), 2016 IEEE International Conference on', IEEE, pp. 4756–4761.

Li, Y., Snavely, N. & Huttenlocher, D. P. (2010), Location recognition using prioritized feature matching, *in* 'European conference on computer vision', Springer, pp. 791–804.

Lim, H., Sinha, S. N., Cohen, M. F. & Uyttendaele, M. (2012), Real-time image-based 6-dof localization in large-scale environments, *in* 'Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on', IEEE, pp. 1043–1050.

Liu, F., Shen, C. & Lin, G. (2015), Deep convolutional neural fields for depth estimation from a single image, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 5162–5170.

Liu, M., Salzmann, M. & He, X. (2014), Discrete-continuous depth estimation from a single image, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 716–723.

Lu, F. & Milios, E. (1997), 'Globally consistent range scan alignment for environment mapping', *Autonomous robots* **4**(4), 333–349.

Mancini, M., Costante, G., Valigi, P., Ciarfuglia, T. A., Delmerico, J. & Scaramuzza, D. (2017), 'Towards domain independence for learning-based monocular depth estimation', *IEEE Robotics and Automation Letters* .

Martinez-Carranza, J., Calway, A. & Mayol-Cuevas, W. (2013), Enhancing 6d visual relocalisation with depth cameras, *in* 'Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on', IEEE, pp. 899–906.

Mur-Artal, R., Montiel, J. M. M. & Tardos, J. D. (2015), 'Orb-slam: a versatile and accurate monocular slam system', *IEEE Transactions on Robotics* **31**(5), 1147–1163.

Mur-Artal, R. & Tardós, J. D. (2017), 'Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras', *IEEE Transactions on Robotics* .

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

Newcombe, R. A., Lovegrove, S. J. & Davison, A. J. (2011), Dtam: Dense tracking and mapping in real-time, *in* 'Computer Vision (ICCV), 2011 IEEE International Conference on', IEEE, pp. 2320–2327.

Reitmayr, G. & Drummond, T. (2006), Going out: robust model-based tracking for outdoor augmented reality, *in* 'Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality', IEEE Computer Society, pp. 109–118.

Sarbolandi, H., Lefloch, D. & Kolb, A. (2015), 'Kinect range sensing: Structured-light versus time-of-flight kinect', *Computer Vision and Image Understanding* **139**, 1–20.

Sattler, T., Leibe, B. & Kobbelt, L. (2011), Fast image-based localization using direct 2d-to-3d matching, *in* 'Computer Vision (ICCV), 2011 IEEE International Conference on', IEEE, pp. 667–674.

Saxena, A., Sun, M. & Ng, A. Y. (2009), 'Make3d: Learning 3d scene structure from a single still image', *IEEE transactions on pattern analysis and machine intelligence* **31**(5), 824–840.

Strasdat, H., Montiel, J. M. & Davison, A. J. (2012), 'Visual slam: why filter?', *Image and Vision Computing* **30**(2), 65–77.

Straub, J., Hilsenbeck, S., Schroth, G., Huitl, R., Moller, A. & Steinbach, E. (2013), Fast relocalization for visual odometry using binary features, *in* 'Image Processing (ICIP), 2013 20th IEEE International Conference on', IEEE, pp. 2548–2552.

Su, C.-C., Cormack, L. K. & Bovik, A. C. (2017), 'Bayesian depth estimation from monocular natural imagessu, cormack, & bovik', *Journal of Vision* **17**(5), 22–22.

Taketomi, T., Uchiyama, H. & Ikeda, S. (2017), 'Visual slam algorithms: a survey from 2010 to 2016', *IPSJ Transactions on Computer Vision and Applications* **9**(1), 16.

Tateno, K., Tombari, F., Laina, I. & Navab, N. (2017), 'Cnn-slam: Real-time dense monocular slam with learned depth prediction', *arXiv preprint arXiv:1704.03489* .

Williams, B., Klein, G. & Reid, I. (2007), Real-time slam relocalisation, *in* 'Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on', IEEE, pp. 1–8.

Williams, B., Klein, G. & Reid, I. (2011), 'Automatic relocalization and loop closing for real-time monocular slam', *IEEE transactions on pattern analysis and machine intelligence* **33**(9), 1699–1712.

Williams, B., Smith, P. & Reid, I. (2007), Automatic relocalisation for a single-camera simultaneous localisation and mapping system, *in* 'Robotics and Automation, 2007 IEEE International Conference on', IEEE, pp. 2784–2790.

Xu, D., Ricci, E., Ouyang, W., Wang, X. & Sebe, N. (2017), 'Multi-scale continuous crfs as sequential deep networks for monocular depth estimation', *arXiv preprint arXiv:1704.02157* .