Monocular Camera Localization Using a Bag of Visual Words from Virtual World Data

THESIS

Joshua A Rinaldi, Captain, USAF

AFIT-ENG-MS-22-M-057

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-22-M-057

Monocular Camera Localization Using a Bag of Visual Words from Virtual World Data

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Computer Science

Joshua A Rinaldi, B.S.C.S

Captain, USAF

March 24, 2022

AFIT-ENG-MS-22-M-057

Monocular Camera Localization Using a Bag of Visual Words from Virtual World Data

THESIS

Joshua A Rinaldi, B.S.C.S
Captain, USAF

Committee Membership:

Clark N. Taylor, Ph.D
Chair

Scott L. Nykl, Ph.D
Member

Douglas D. Hodson, Ph.D
Member

Abstract

The Visual Localization (VL) problem is the question of how to take in a query image and determine the pose of the camera that took that photo. The fact that Red Green Blue (RGB) cameras have become incredibly common and inexpensive, coupled with the huge amount of data they are able to capture have made building VL pipelines that accurately generate results increasingly interesting and important. These pipelines are useful in everything from Simultaneous Localization and Mapping (SLAM) and Smoothing and Mapping (SAM) for robotics to applications in Augmented Reality (AR).

The work detailed in this paper seeks to determine if a Bag of Visual Words (BOVW) is adequately able to look past repetitious features in an indoor environment and localize the camera that captured an image. This pipeline is intended to be used as a truth system to verify results from other navigation techniques in development by the Autonomy and Navigation Technology (ANT) Center at the Air Force Institute of Technology (AFIT) in lieu of more expensive solutions such as motion capture systems or the Global Positioning System (GPS) which is unreliable in an indoor environment.

Acknowledgements

I would like to thank my advisor, without his guidance and advice I would have been totally lost throughout this entire process. I would also like to thank my wife for her unending patience and support for me throughout this entire process, both of which kept me moving forward and sane, I promise to never say "Bag of Words" within your hearing again. Thank you to all my instructors at AFIT, the knowledge you have imparted during my time here has been invaluable. Finally I would like to thank my fellow classmates at AFIT, their constant input and companionship made this process all the more enjoyable.

Joshua A Rinaldi

Table of Contents

Page

List of Figures

Figure Page

List of Tables

Table Page

Monocular Camera Localization Using a Bag of Visual Words from Virtual World Data

I. Introduction

1.1 Problem Background

When developing a new navigation technology, a critical aspect of that development process is having access to truth data. This allows the researchers developing that technology the ability to know how accurately their new system is performing and further refine the system. Typically a Global Navigation Satellite System (GNSS) such as the Global Positioning System (GPS) is used for this purpose, as it provides a high level of accuracy to almost any position on the globe. One notable exception to this though is in indoor environments where the signals from satellites can struggle to penetrate buildings and provide accurate information to the devices trying to connect to them.

This characteristic of GPS can make it difficult to use for generating truth data when testing a navigation technology that is intended to work in an indoor environment. As such, it is necessary to have another system to generate that truth data and validate the results from the technology being developed. There are several methods for accomplishing this in an indoor environment, including motion tracking systems, Radio Frequency Identification (RFID), Bluetooth Low Energy (BLE) and other technologies that have been developed. [1, 2]

One method of accomplishing localization, which has been thoroughly researched for use within the robotics community is Visual Localization (VL), which uses Red Green Blue (RGB) imagery generated by a monocular or stereo camera system in order to determine the position of the camera that created the image. The appeal of VL comes from how much information can be gathered by a camera, and how readily available such cameras are. To that end, this research aims to use images generated from a monocular camera, such as the camera that can be found on most mobile devices today, to generate the position of that camera within the second floor of Building 640 on the Air Force Institute of Technology (AFIT) campus to be used as the truth data for the development of indoor localization and navigation systems that come out of the Autonomy and Navigation Technology (ANT) Center.

The overall approach for this research is to use a Bag of Visual Words (BOVW) to determine which of 5 sections of the building the camera is in, and then a direct feature matching method in order to generate correspondences between the 2D coordinates of features present in the query image to the 3D real world coordinates of those features, and from that information determine the three dimensional position of the camera.

1.2 Research Objectives

This research aims to answer the following questions:

- Is the BOVW model sufficiently able to overcome the issue of repeated visual features within a building to narrow down the general location of an image to one of a few possibilities?
- Is a homogeneous sampling of a set of image features, such as that generated by Suppression via Square Covering (SSC) better suited for indoor localization on the AFIT campus than sparse sampling?

- Is a 3D virtual model of a building generated from a Structure from Motion (SfM) application accurate enough to generate the data used to train a BOVW, as well as the reference data used for 2D-3D correspondence detection?

1.3 Document Overview

Chapter II provides a look into the requisite background information in order for the reader to understand the technologies being used in this research. Chapter III details the process used for development of the localization system. Chapter IV provides a breakdown of tests performed to determine the efficacy of the developed data pipeline, as well as a discussion of the results of those tests. Finally Chapter V provides a summary of the information within this document, as well as details of how this research will benefit future efforts at the Air Force Institute of Technology.

II. Background and Literature Review

Starting with a high level look at the Visual Localization problem, this chapter provides the reader an overview of the technologies used throughout this research as well as previous efforts at accomplishing this task through other methods.

2.1 Visual Localization and Visual Place Recognition

Originally called the Location Determination Problem (LDP), the issue of determining the location from which a camera took a photograph, now referred to as Visual Localization, is one that has been occupying the minds of researchers since the middle of the twentieth century. Originally, this problem was solved by having a person assign 3D coordinates to features inside of a query image and then applying the Least Squares Method to select points with which to solve the Perspective-N-Point (PNP) problem and return the position of the camera [3]. This method was improved upon in the 1980s, when the random sample consensus (RANSAC) paradigm allowed this process to be fully automated [3]. To this day, RANSAC is still an incredibly useful tool that is widely used for outlier detection in various applications, such as Computer Vision.

Visual Place Recognition (VPR) is essentially a less precise version of Visual Localization (VL), in which the computer tries to determine the general area in which a photograph was taken, rather than the precise location of the camera. This is incredibly useful in robotics as knowing generally where a photograph was taken gives a rough starting area for localization. Knowing this general area reduces the number of possibilities that need to be considered, thus leading to faster and more accurate localization results. As indicated in [4] an accurate VPR result is essential for VL, as an estimate of where a photo was taken is necessary in order to more

precisely determine the location of the camera that took that photo.

VL, as well as VPR are both problems that have been extensively researched within the robotics community for use within Simultaneous Localization and Mapping (SLAM) and Smoothing and Mapping (SAM) algorithms. The overall appeal of using visual methods of determining the location or a robot, or other devices, such as a mobile phone, is the fact that Red Green Blue (RGB) cameras are relatively much less expensive than other equipment which might be used for this task, and they are also widely available; additionally, RGB cameras provide huge amounts of information that can be incredibly useful if used correctly [5]. For an overview of SLAM and SAM technologies and comparisons of some of the more recognized methods for each, refer to [6, 7].

Overall there are three approaches to solving the issue of Visual Localization [8]. These are:

- Structure Based methods, where a 3D model of the area in question is known ahead of time and used to draw correspondences to real world coordinates within the area [9, 10, 11, 12, 13].

- Image Based methods, where a large database of sample images, tagged with the location they were taken from is available to compare against the query image and generate a position [14, 15, 16].

- Learning Based methods, where a model of the area is learned through discovery and used to represent the scene in much the same way as Structure Based methods [17, 18, 19, 13].

As the work in this paper is intended to produce truth data to be used in other experiments taking place in a known environment, a Structure Based approach was chosen. The success of Structure from Motion (SfM) methods found in [20] was

a factor in this decision, as were the difficulties that typically face VL in indoor environments. Due to the relatively small size of the area in question, construction of a 3D model using SfM was feasible. Having such a model also opened the possibility of utilizing the virtual world, the advantages of which are discussed later in this chapter, and would greatly benefit future research potentially taking place at the Air Force Institute of Technology (AFIT).

The overall data pipeline for a structure-based VL approach typically follows the following steps:

1. Capture a query image.
2. Extract relevant features from the query image.
3. Find general location from which the query image was taken (VPR).
4. Draw correspondences between image coordinates and real world coordinates for features found within the image.
5. Execute the PNP problem using those feature coordinate correspondences.
6. Return the pose of the camera when the image was taken.

This pipeline is illustrated in Figure 1.

Typically feature extraction is accomplished using one of the standard Feature Detectors, such as SIFT, SURF, BRIEF and ORB. These are tried and true feature detection algorithms that are available in the OpenCV library. These algorithms tend to look at individual pixels and the intensity of a pixel versus that of its neighbors. [21] offers a comparison of the efficacy and efficiency of these algorithms, and led to the decision to use SIFT as the primary feature detector for this research. For more information on the specifics of SIFT, refer to [22].

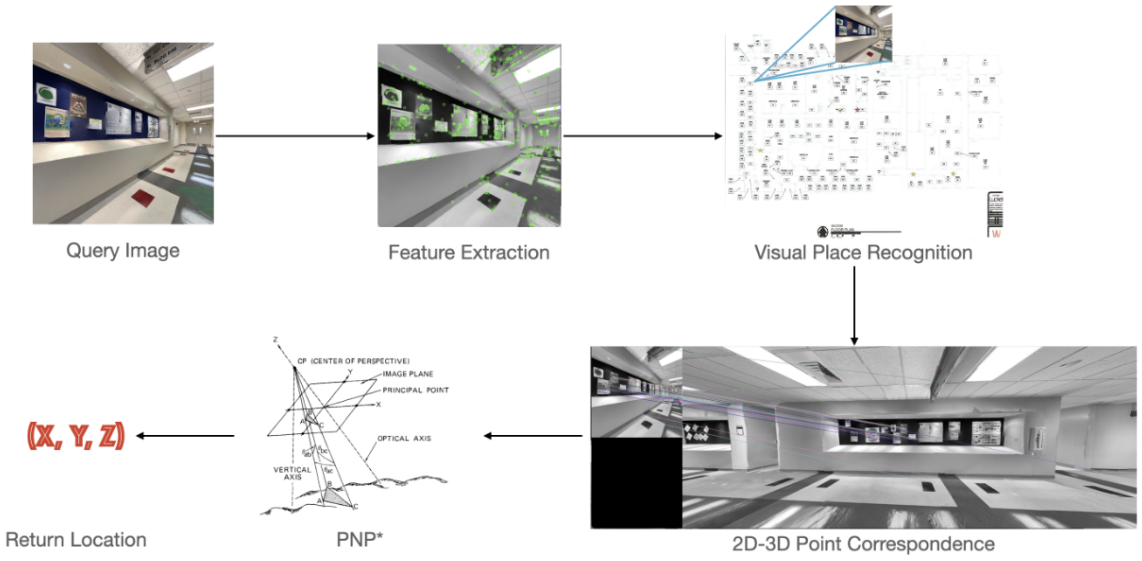


Figure 1: Illustration of generalized VL data pipeline, * courtesy of [3]

The PNP problem, in short, is the process of taking the known 3D coordinates of points in a photograph and determining the location of the camera that took that photo based upon the projection of those points. We know the distance in the real world between the 3D points, as well as the angles of the lines that connect those points from the image. Using this information, along with the specific parameters of the camera used to capture the query image, it is possible to determine where, relative to those features, the camera was when the image was taken. In order to solve this problem, it is necessary that at least three points in the image be known, as with one or two known points there are an infinity of possible solutions. Thus, it is necessary that there be no less than three known points when solving the PNP problem, but there is not necessarily an upper limit on the number of points used. For more information on the mathematics behind solving the PNP problem, refer to [3]. Fortunately, because of the importance of being able to solve this problem, a ready made implementation, along with several variations of the PNP problem are available for use in the OpenCV library.

2.2 The Difficulties of Urban and Indoor Visual Localization

One key area where Visual Localization struggles tends to be in indoor or large urban environments. The cause of this difficulty is repeated features. In indoor environments there tend to be large patches of blank wall that occur throughout the building, or other features such as water fountains, directory signs or poster boards that are all repeated and can confuse localization algorithms [12]. In outdoor urban environments, building features such as windows or doors, as well as items such as signs, all of which tend to look very similar to one another, can also confuse VL pipelines [23, 9]. Figure 2 shows some features from the building that repeat. Figure 3 shows how they can be falsely matched to one another. This issue is one that has been successfully overcome with NetVLAD, a convolutional neural network (CNN) developed for urban VPR in [23] and utilized as a part of an indoor VL pipeline in [12].

A key aspect of NetVLAD that made it successful in overcoming the obstacle of repetitious features is the usage of dense, rather than sparse, feature sampling. Sparse sampling tends to yield feature descriptors which cluster around objects in the scene. Dense sampling on the other hand, will break an image into a grid and provide feature descriptors for each point on that grid [24]. The generation of feature descriptors for an entire image, rather than just those portions of an image typically considered "interesting" allows for less interesting regions of imagery, such as an empty wall, to become more relevant in the feature matching process. Another approach is to split the difference between these two methods through the use of a technique such as Adaptive Non-Maximal Suppression (ANMS), like the Suppression via Square Covering (SSC) technique proposed in [25]. [26] showed that a more even distribution of detected points led to better feature matching. In SSC, keypoints in an image are detected using a FAST feature detector, and then filtered, providing a

Figure 2: Examples of repeated features throughout Building 640

much more evenly spaced set of features for consideration. Descriptors are generated from those keypoints using a feature descriptor of choice. A comparison of these three approaches to feature sampling is presented in Figure 4. Notice that in the SSC filtered image, feature descriptors are generated in the bottom right corner of the image that were not generated in sparse sampling. However, there are not so many

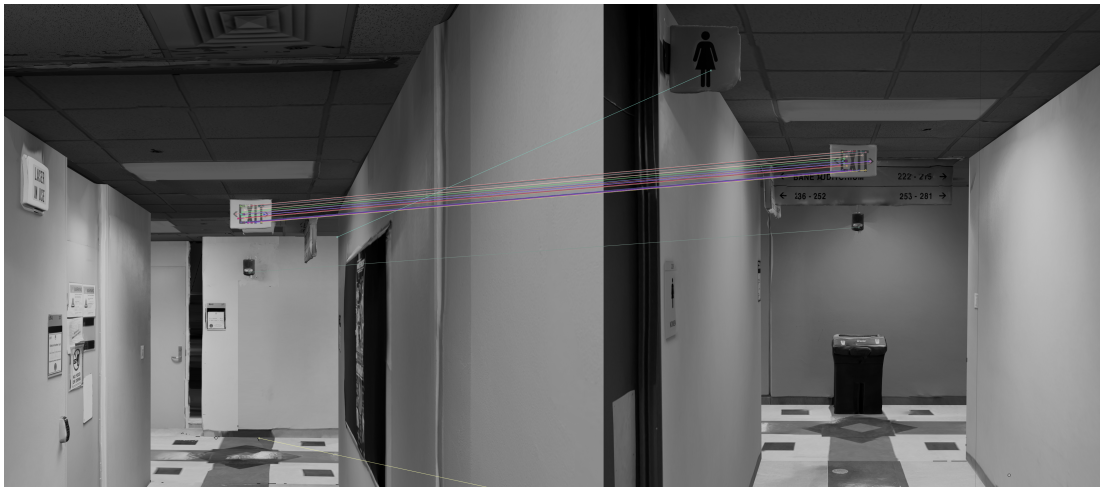


Figure 3: Feature matches occurring between different exit signs

descriptors generated that the actual structures within the image are lost, as is the case in the densely sampled image.

2.3 Machine Learning and Visual Localization

Machine Learning (ML) has been incorporated with Computer Vision with quite a bit of success over the last several years. Fields of research that have reaped the rewards of these efforts include VPR, VL, SLAM and SAM. There is a range in the degree to which the VL pipeline is taught to a specific ML model. The most extreme version of this would be Absolute Pose Regression (APR) algorithms, which attempt to have the model learn the entire localization pipeline and regress the pose of the camera [27, 28, 29, 30, 18]. Unfortunately, as indicated in [31] these approaches tend to not be as successful as the more traditional structure based approaches. Additionally, attempts to learn the VL pipeline end to end without the use of some sort of 3D model or RGBD data for initialization run into issues of overfitting based on global patterns present in the scene [17].

Other methods of incorporating ML into the pipeline involve having a model learn a part of the pipeline, and this has been shown to be incredibly successful, especially

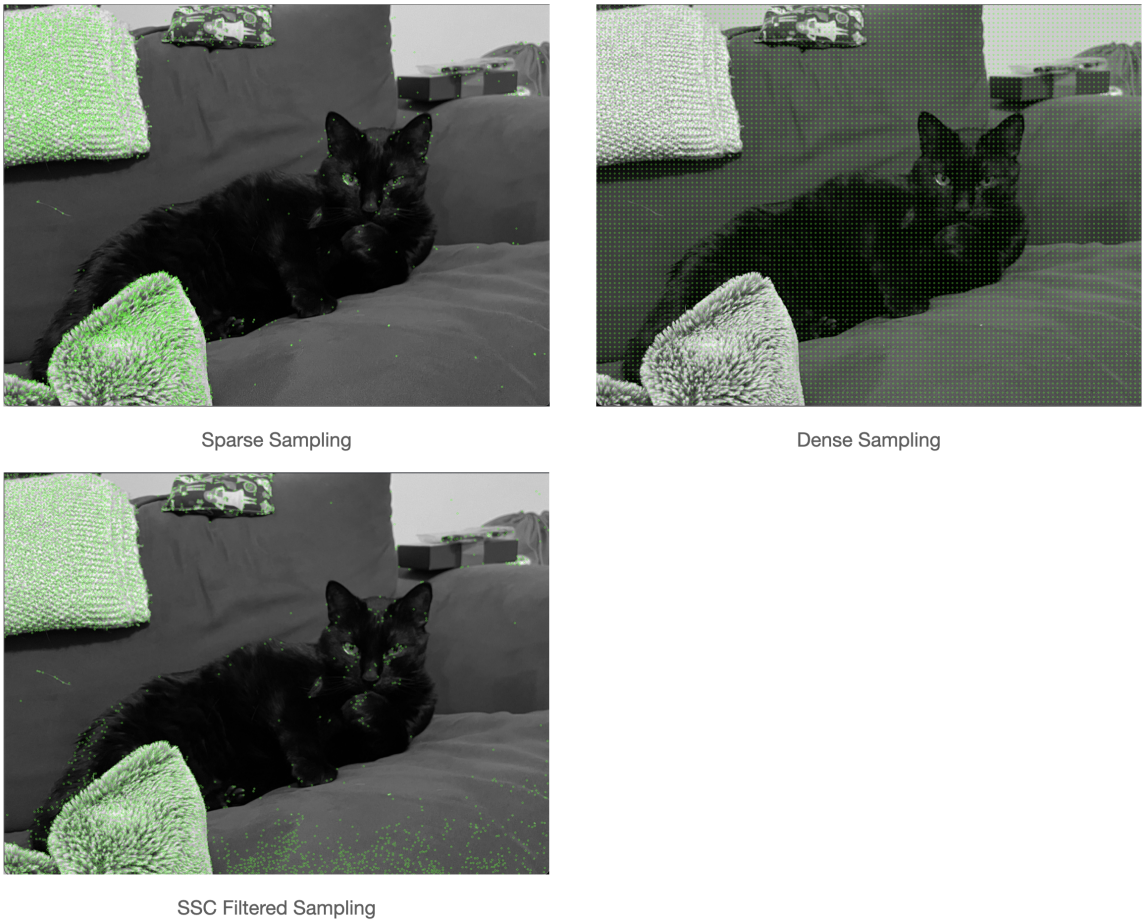


Figure 4: Comparison of Sparse, Dense and SSC Filtered feature sampling

when it comes to VPR, the first step in determining the precise location of the camera [12, 32, 23, 4, 2, 33]. [34] offers a comparison of the usefulness of three different CNNs for use in VPR. These and other lines of effort have shown CNNs to be effective at VPR. Another portion of the VL pipeline researchers have experimented with training a model on is the matching of 2D features in the query image to 3D features in the known model. Typically researchers have used regression forests for this task, which do come with the limitation of needing to be trained offline [19]. Attempts at using regression forests for this task have struggled with an imbalance in the left and right trees, and the attempt made in [35] to overcome this difficulty produced results much slower than the authors considered acceptable.

Another, more rudimentary version of ML that has been used in the VL pipeline is the Bag of Visual Words (BOVW). Explained in greater detail in the next section, this model was successfully used in [36] to determine what room a robot was in, thus solving the VPR problem and providing the first step in the VL pipeline. Additionally, [8] used a semantics based approach to solving VL under different lighting conditions. This approach looked at visual words labeled with semantic information to generate the 2D-3D feature correspondences needed for localization. Image semantics were also shown to be effective as part of a localization pipeline that uses multiple sensors, not just image data, in [37]. It is also possible for a robot to navigate a known path under various lighting changes based on experience, and the use of a BOVW in order to pick a supervised driving experience most visually similar to the scenario in which the robot finds itself [38]. One common thread in these efforts though is a lack of repeated features. [36] looked at features specific to various rooms in a home (kitchen appliances, furniture, etc.) that are very distinctive from one another, [8] looked at an outdoor environment, and [38] specifically looked for repeated features under lighting changes to choose an experience to replicate, rather than actually trying to localize the camera producing the query image.

2.4 The Bag of Visual Words (BOVW) Model

To understand the BOVW model, it is necessary to first understand the Bag of Words (BOW), a natural language processing model that was adapted into the BOVW for use in Computer Vision. The BOW was developed as a means of categorizing text documents simply by looking at the number of occurrences of each word within the document. [39] determined that a State Vector Machine (SVM) could be adequately trained to determine the category of a text document based upon the frequency with which words occur in that document, independent of the order in which they appear.

This model was later adapted to computer vision by the use of Visual Words.

When extracting visual features from an image, it is common for those features to form clusters around objects within the image. These clusters of features are what constitute the visual words in the BOVW. In the same manner that a text document is broken apart into its key features (the frequency of word occurrences), an image can also be broken apart and described by the number of occurrences of the individual visual words contained within, and a SVM can also be used to accurately determine the category of that image [40]. See Figure 5 for a visual representation of this.

In this manner [36] was able to determine the room from which an image was captured by training the model to recognize which features were common in various

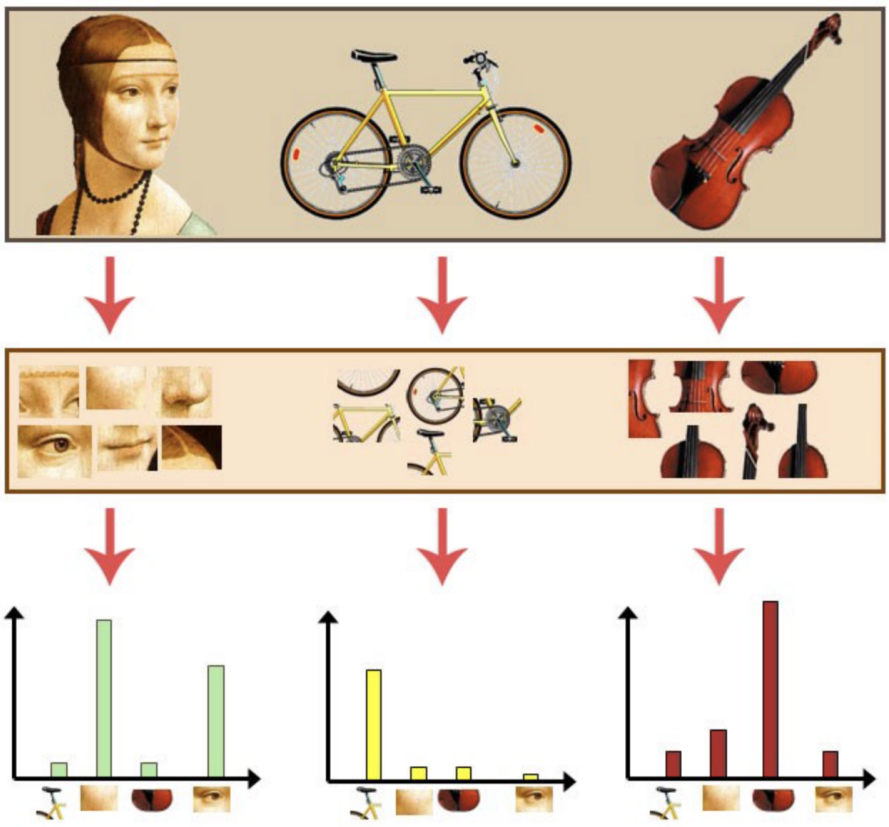


Figure 5: BOVW, courtesy of https://towardsdatascience.com/bag-of-visual-words-in-a-nutshell-9ceea97ce0fb

rooms in a home. In [41] a BOVW was used for VPR in a localization pipeline used outdoors. [42] also discusses several methods that utilize a BOVW to determine the general area within which an image originated. A key component to the BOVW is an availability of labeled data upon which the SVM can be trained prior to deployment. This is a characteristic of the BOVW that makes it difficult to use without retraining in a new environment.

2.5 Virtual Worlds, Digital Twins, and Their Benefits

The use of virtual worlds is a concept that has been gaining more traction in recent years. The many benefits of using a virtual world in research all lead to the same end result: time and money saved for the researcher. As an example, the Automated Aerial Refueling (AAR) project out of the Autonomy and Navigation Technology (ANT) center at AFIT uses a virtual world to run multitudes of tests of algorithms without the need for actually flying aircraft, which is logistically infeasible, as well as hugely expensive and potentially dangerous in the case that automated systems collide during flight [43].

In [44] the authors test collision avoidance in navigation algorithms for an unmanned aerial vehicle (UAV) through the use of a virtual world generated by the Unity game engine. In reality, the UAV is flying through empty space, but the game engine feeds imagery data from the virtual room that is being navigated to the computer on board the UAV while a motion capture system tracks the location of the vehicle. In this way, the authors were able to test algorithms without running the risk of having their equipment collide with real world obstacles, thus damaging equipment. Another advantage to using virtual worlds can be seen in [45] where the authors created simulation software, dubbed "UnrealNavigation," for testing SLAM algorithms intended to be used in environments such as outer-space, other planets or

underwater, which are inaccessible for testing purposes except at great expense. Not only were the authors able to accurately test these algorithms, the use of a totally virtual world also offered access to accurate truth data which could be used to verify the results coming out of the algorithms being tested.

Relating this to the problem of VL, the data pipeline can be implemented and results from that pipeline can be verified by the engine running the virtual environment, because the engine will know the exact location of the camera that generated the image. Additionally, thanks to SfM technologies, it is possible to quickly and easily generate to-scale, photorealistic models of an environment such as the inside of a building or even large outdoor environments [9, 13, 46].

Refer to Figure 6 and Figure 7 for side by side images taken from the real world and the 3D model used in this research. The accuracy of these SfM reconstructions is to such a degree that even a cockroach trapped in a light fixture was picked up and can be seen in the 3D model used in this research, reference Figure 8.

Figure 6: Real vs. Reconstructed Hallway

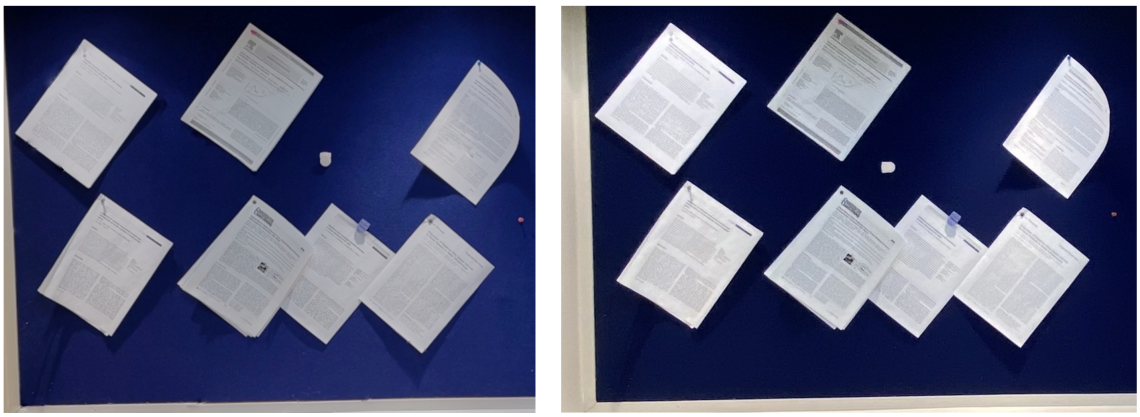


Figure 7: Real vs. Reconstrcuted Poster Board

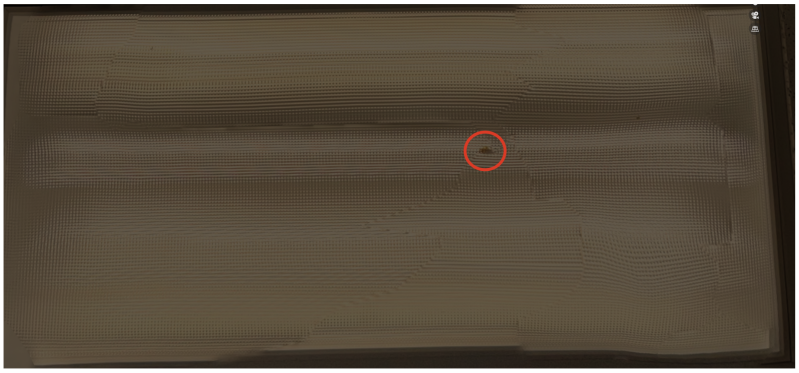


Figure 8: Reconstructed Cockroach

Another area where the use of virtual worlds is keenly advantageous is in the generation of training data for ML models. Not only is it a simple matter to script out image capture locations, thus negating the need to manually take photos of the environment, it is also possible within the virtual world to generate thousands of photos for training purposes within a matter of minutes, as well as randomly alter the environment to account for changes such as objects moving and changes in lighting due to time of day or year. This leads to a more robust ability for a model to recognize images and features within them that are useful for VPR and VL [47, 48]. In the case of [49] the researchers generated so many randomly altered scenarios of the training data that the model in question treated any sort of abnormalities in the real world

not accounted for in the virtual world as yet another variation that could have been generated virtually.

III. Methodology

3.1 Preamble

The primary goals of this work are as follows:

- Build a data pipeline able to take in a monocular photo taken within the second floor of building 640 and return to the user the three degrees of freedom position of the camera that took the photo within a local coordinate frame.
- Construct a scale 3D virtual model of the second floor of Building 640 to use for generating the data which will be used in the data pipeline as well as testing of the constructed data pipeline.
- Engineer a module for the Aftrburner Engine which incorporates the aforementioned virtual model, which allows for easy integration and testing of future indoor localization algorithms developed by students and faculty within the ANT Center.

Starting with an overview of the data pipeline, this chapter will cover how and why the pipeline was constructed as it is, and how the data used was collected.

3.2 Data Pipeline and Methodology Overview

From image capture to return of camera position, the data pipeline implemented for this work follows this series of steps:

1. Image capture
2. SIFT feature extraction
3. K-Means clustering of features

4. Bag of Visual Words (BOVW) determination of two most likely building regions

5. SIFT descriptors generated for SSC filtered FAST keypoints from query image

6. Comparison of query image data against reference photos for BOVW determined building region

7. Feature matching between query and selected reference images

8. Run Perspective-N-Point (PNP) on feature matches

9. Return XYZ coordinate of the camera that captured the query image

This data pipeline is illustrated in Figure 9.

While there has been some success recently in the use of a convolutional neural network (CNN) for Visual Place Recognition (VPR) in indoor environments [12], one key issue which this work seeks to address with using a CNN is the need for huge amounts of training data, as well as extensive amounts of time and computing

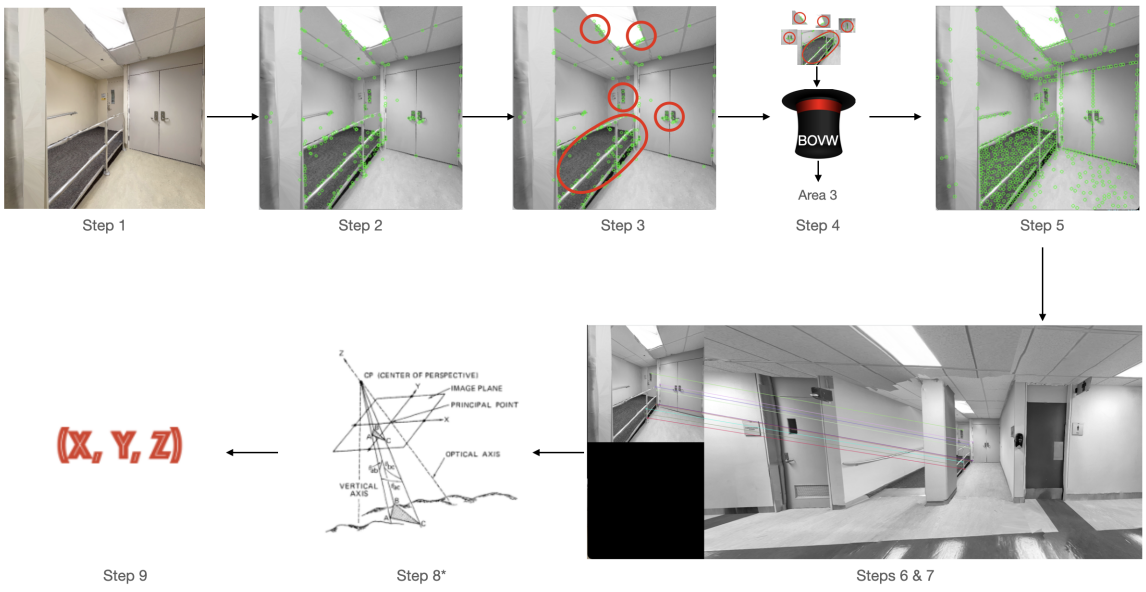


Figure 9: The data pipeline used in this work, * courtesy of [3]

resources needed to train a CNN. For example, the NetVLAD CNN used in [23] used more than 83k training images. While it would be possible to use a CNN like NetVLAD which has already been trained for VPR, a BOVW can be trained on a personal computer in a matter of minutes using a smaller data set. [50] showed Visual Words could be used for accurate localization and [41] constructed a Visual Localization (VL) pipeline that used a BOVW for VPR. These papers show promise for the usefulness of the BOVW in VL, but neither focused specifically on indoor environments. If the issue of repeated features indoors could be overcome, a BOVW would make an easily constructed and implemented portion of a VL pipeline.

SIFT feature descriptors and keypoints generated using the built-in OpenCV SIFT feature detector were used for training as well as querying the BOVW. This decision was made considering that a homogeneous spread of keypoints, such as that generated by Suppression via Square Covering (SSC) filtering, may not produce the image clusters necessary to create a robust visual vocabulary. SSC filtering is used later in the data pipeline for feature matching between query and reference images considering the success found in [25] and [26] for matching features based on homogeneous keypoint distributions. SIFT was used as the primary feature detection algorithm based upon the results of [21] which showed SIFT to be more accurate than other popular feature detection algorithms. Based upon experimental results within this work, ORB was chosen as a backup in the event that SIFT failed to generate the necessary keypoint matches. Additionally, based on experimental results, it was decided the data pipeline would consider the top two choices of area generated by the BOVW, as opposed to just the first choice. This is discussed more in the next chapter, but the use of top two most likely areas of the building greatly increased the likelihood with which the correct area of the building was chosen.

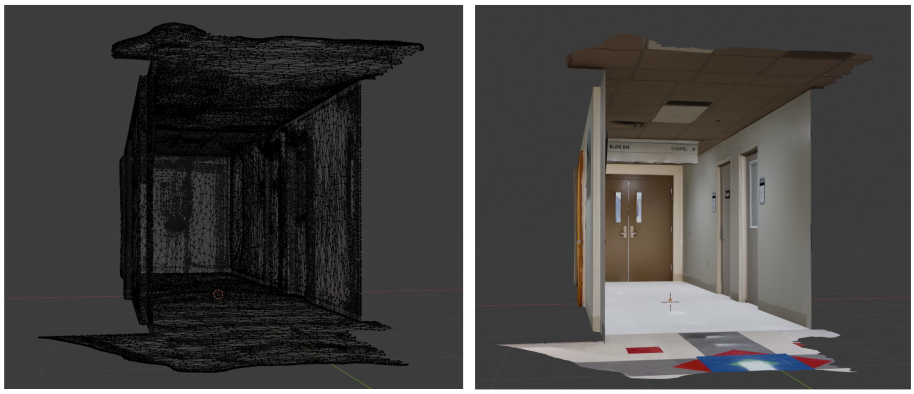
All data used for training the BOVW as well as initial testing was gathered from

a 3D model of the second floor of Building 640 on the Air Force Institute of Technology (AFIT) campus. Data capture and pipeline testing were performed using the Aftrburner game engine, developed by Dr. Scott Nykl and used in [43] and [2] for their research. The decision to use Aftrburner as opposed to another game engine was predicated on the integration of OpenCV within the engine, and the ease with which image textures can be captured, processed and returned within the engine. Additionally, the Aftrburner engine provided an easy means of verifying results from the data pipeline as it is able to write out the exact location from which an image was captured.

3.3 3D Model Construction

The 3D model used in this research was constructed through the use of Scaniverse, an application designed for iPhone which makes use of the on-board camera and LIDAR sensor to generate a 3D point cloud (see Figure 10). The phone used for data collection was an iPhone 12 Pro. Due to performance limitations of the application, the building had to be scanned in sections, which were then exported and combined using Blender, a free 3D modeling software. While piecing together the different sections of the model it was also necessary to correct for some errors that were generated during the scanning process. Many of these errors, however, were not able to be corrected and do appear as cosmetic blemishes in the completed model. The impact of these blemishes is discussed in chapter IV. See Figure 11 and Figure 12 for examples of Scaniverse generated errors that could be corrected in Blender. Figure 13 and Figure 14 give a few side by side comparisons demonstrating blemishes that persist after correction. After all pieces of the model were put together in Blender, the scale measurements of the model were verified through the use of the program's measurement tool and independent measurements of the real building

captured using a laser distance measuring tool. Once completed, the model of the building was exported from Blender as a .fbx file and then loaded into the Aftrburner engine for data collection and pipeline testing.



Scaniverse generated 3D point cloud

Scan w/ image texture overlay

Figure 10: 3D point cloud generated by Scaniverse, with and without image texture overlay

Figure 11: Example of blemishes that could be corrected using Blender. Note the hole in the wall.

Figure 12: More blemishes that could be corrected. Note the holes and the warping in the door.

SfM Reconstruction iPhone image

Figure 13: Blemishes that could not be corrected. Note the edges of the sign, as well as the room number.

Figure 14: More incorrectable blemishes. Note the seam in the floor, legs of the chairs and warping around door frames.

Figure 15: An overhead view of the full model loaded into Blender

Figure 16: An overhead view of the full model loaded into Aftrburner

3.3.1 Variations on the 3D Model and Data Pipeline

It should be noted that there are several objects that were modeled within the building that are considered "movable" such as trash cans and chairs. Throughout the course of the this research the majority of these objects remained in roughly the same position in the building, however, this does not mean that the day to day position of such objects do not vary slightly. For example, a trashcan, after being emptied, might be put back a little to the left of where it was initially, it is also possible that such objects might be removed from the halls of the building entirely.

Due to the strong likelihood that such shifts in the environment could make it difficult for the data pipeline proposed in this research to produce accurate results, two additional models of Building 640 were produced in order to generate additional training and testing data to cover these cases. This means that in total, three models were created. The first model contains these movable objects in their original "starting" positions. In the second model, these objects were shifted slightly, in ways that might seen in the real world, such as a slight move to one side, rotation or removal from the scene altogether. The third model was created with all such movable objects removed. Using data created from these three models, there are three variations on the data pipeline that were built, they are:

Figure 17: Images showing the change between the three versions of the model. Note the trashcan at the end of the hall

1. A pipeline in which the BOVW training images and panoramic reference images (discussed later) are generated using only the model with movable objects in their starting positions.

2. A pipeline in which the BOVW training images come from model with movable objects in their starting positions, and the panoramic reference images are pulled from the model without movable objects.

3. A pipeline in which the BOVW training images and panoramic reference images both come from the model that does not have any movable objects

Without testing, it would seem that the first data pipeline, consisting of images all taken from the model with movable objects would be the least robust. In the scenario that a movable object is present, but shifted, in both query and reference image, any feature matches used in PNP would likely negatively impact the final result. The third data pipeline seems likely to overcome this challenge, as by removing the movable objects entirely and the data pipeline is forced to localize entirely based off of immovable features of the building, such as walls and doors. Since these objects tend to be in roughly the same position, it seemed reasonable to keep them present in the training data use for the BOVW. Keeping these objects would create more visual words with which the BOVW could determine the rough area of the building, but removing them from reference data would maintain the advantage in forcing the pipeline to only use feature matches generated on immovable features of the building for PNP.

3.4 Data Collection

There were three datasets that were collected for this pipeline. The first dataset consists of training data for the BOVW, the second dataset contains reference images

used for feature matching, 2D-3D point correspondence and PNP, and the third dataset contains test images collected from all three versions of the model, as well as the real world.

3.4.1 BOVW Training Data

To begin with, the building was split into 5 different regions, each corresponding to one of the five major hallways in the building as shown in Figure 18. Training data was then collected by placing a starting point at one end of the hall, and an ending point at the other end of the hallway. A camera in the virtual world was then moved along the line connecting these two points in increments of two meters. At each position along the line, the camera would be rotated in 30 degree increments around the vertical axis, after each rotation, three images would be captured, one with the camera positioned perfectly level, one with the camera panned up thirty degrees from level, and one with the camera panned down thirty degrees from level. The camera used for training image collection was a virtual camera which generated images that were 500 pixels wide by 500 pixels tall, with a horizontal field of view of 97 degrees.

This process was used to collect training images from the building model where movable objects are kept in the starting positions, as well as the empty building model. Additionally, a third set of training images was collected which only captured distinctive features of each area of the building. Features that qualified as distinctive were features that only appear in that area of the building, such as artwork on the walls or the "Laser in Use" signs which only appear in area three of the building, see Figure 19. This reduced training set was captured to cover the possibility that numerous photos of repeated features such as doors, fire alarms or light fixtures would make it difficult for the BOVW to distinguish regions of the building. This was done

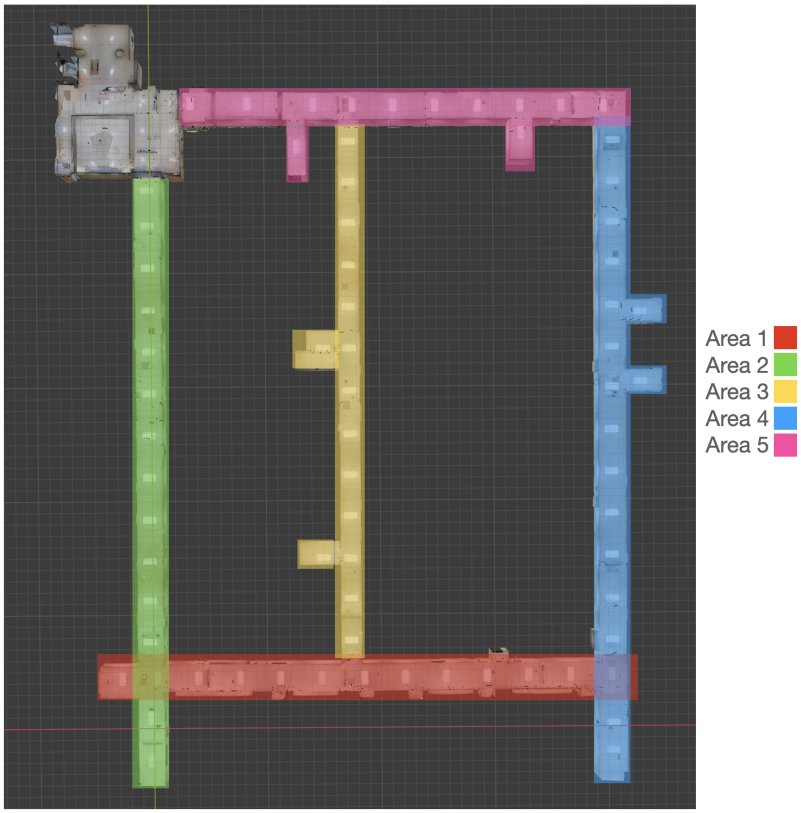


Figure 18: The five areas into which the building was divided

to determine what effect excluding such repeated features from training would have on the accuracy of the BOVW. See Figure 20 for examples of these features. In total, the empty building model and model with movable objects were used to generate around 4,400 images each to train the BOVW, while only 109 images were collected for the reduced training set that only focused on distinctive features.

3.4.2 Reference Data

Following the completion of image collection for training the BOVW, Aftrburner was then used to collect reference images for use in the next part of the data pipeline. Images collected were 2,500 pixels wide by 1,000 pixels tall, with a horizontal field of view of 150 degrees. See Figure 21 for a comparison of reference and training images. At the same time that each reference image was captured, Aftrburner's built-in

Figure 19: A "Laser in Use" sign, there are several of these, but in only one area of the building

Bane Auditorium Entrance

Weather School Emblem

Distinctive Board

Figure 20: Examples of distinctive features captured for the reduced dataset

Figure 21: Comparison of images generated for reference and images generated for BOVW training

Virtual Flash Lidar (VFL) module was used to generate a 3D point cloud corresponding to the image. The VFL module inverses the OpenGL Rendering Pipeline to determine the 3D world coordinate of each individual pixel captured within the viewing frustum of the camera attached to the module. These coordinates are then returned as a point cloud which is written to file. The end result of this process is a reference image containing 2,500,000 pixels and a .txt file associated with each image containing the 3D world coordinate of every single pixel in that image. Once feature matches are established between the query image and reference image, the feature pixel coordinates from the reference image are taken from the corresponding pixel coordinate file and then used in conjunction with the 2D pixel coordinates for those same features in the query image as inputs to PNP.

3.4.3 Test Data

Finally Test Data was collected in order to test the accuracy of the data pipeline. A total of thirty eight test images were collected across four different datasets. Those datasets are:

1. “objects” dataset - images generated from the model containing movable objects

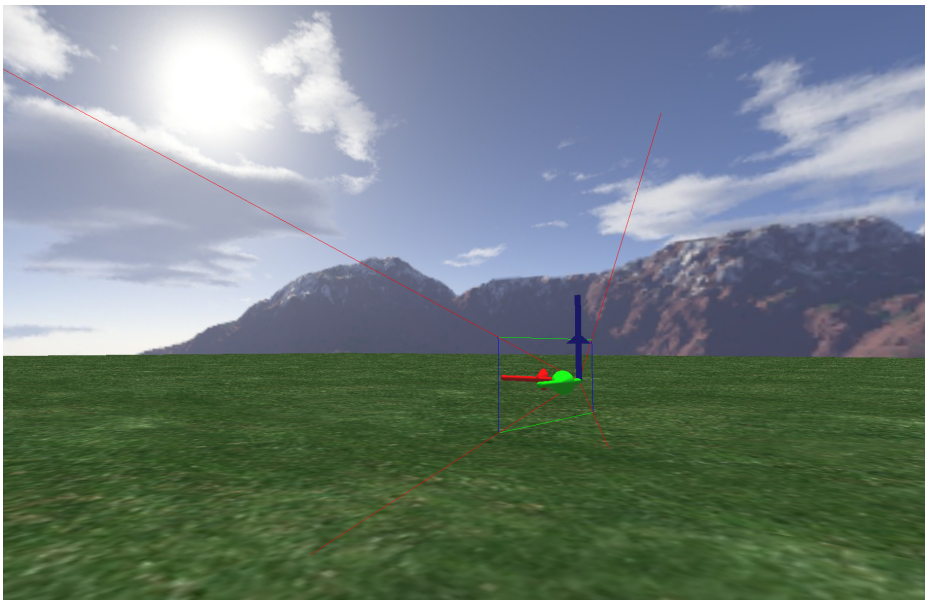


Figure 22: The view frustum of the camera used to collect reference and 3D point data

in their start positions. Consists of fifteen images.

2. "empty" dataset - images generated from the model containing no movable objects. Consists of eleven images.

3. "moved" dataset - images captured in the model with movable objects shifted from their start positions. Contains seven images.

4. "realworld" dataset - images captured from the real world. Contains five images.

Test images captured in the virtual world were done so using the same virtual camera module that was used to capture training images for the BOVW, a 500x500 pixel camera with 97 degree horizontal field of view. Accompanying the three datasets generated in Aftrburner is a positions file containing the three coordinate position of the camera when each image was captured. The real world images were captured using an iPhone 12 Pro. These images do not have the associated truth position, but were used to verify that the data pipeline is able to match each query image with an

"appropriate" reference image (as deemed by a human overseeing the matching).

IV. Results and Analysis

4.1 Preamble

Upon completion of constructing the data pipeline, as well as generation of the datasets discussed in Chapter III the various steps contained within the data pipeline were tested for efficacy. First, the Bag of Visual Words (BOVW) was tested to determine which dataset provided the most accurate area determination, along with a test of whether considering the two most likely results from the BOVW resulted in higher accuracy. The pipeline's ability to determine an appropriate reference image using homogeneous keypoint filtering via Suppression via Square Covering (SSC) was also tested.

In addition to the tests above, a naive data pipeline was also implemented against which the BOVW pipeline was compared. This naive pipeline cut out the BOVW and instead compared a given query image against every reference image for the building. In order to determine if SSC filtering generated better feature matches, a multi-filter control was also utilized in which generic SIFT and ORB detectors, as well as an SSC filtered ORB detector were all allowed to compete with one another for reference image determination.

The results and methodologies of these tests are discussed in this chapter.

4.2 BOVW Testing

This section discusses the methodology used to test the efficacy of the BOVW as well as the results that came out of those tests.

4.2.1 Testing Methodology

As mentioned in Chapter III training data for the BOVW was generated in three different sets. The first set used images gathered from the model of the building containing movable objects in their respective "starting" positions, this is the "Objects" training set. The second set, called the "Empty" training set, contains images generated from the building model where those movable objects are not present. Finally is the "Reduced" trainig set which is comprised of images taken only of those features in each area that would be considered significant and unique to that specific area of the building.

4.2.2 Testing Results

An overview of results from each of the tests presented in table 1. More detailed results are available in Appendix 1.1 in table 10, table 11, table 12, and table 13. In table 1 the number of correct area guesses out of the thirty eight test images is presented when the BOVW is trained of each of the three training datasets. So, for example, when trained on the Objects dataset, the BOVW was able to accurately guess the area of the building for 13 out of 38 test images on the first guess and 15 out of 38 test images on the second guess.

Table 1: BOVW Test Results Summary

BOVW Training Set	**# Correct 1st Choice**	**# Correct 1st or 2nd Choice**
Objects	13	15
Empty	7	10
Reduced	7	20

4.2.3 Discussion of Results

If looking at only the first choice provided by the BOVW the training data generated within the model containing movable objects showed the best results, with a 34% success rate. However, when looking at the top two choices provided by the BOVW, this number pales in comparison to the 53% correct guess rate provided when the BOVW is trained on the reduced dataset. However, upon further inspection, it becomes apparent that the success rate of the reduced dataset is perhaps not to be trusted. Every single result generated from this training data was area three for the first choice and area four for the second choice. This is what led to the dramatic jump in correct guesses when looking at the top two choices, out of the test images, seven were from area three, and thirteen were from area four. This skew towards these two areas likely comes from the fact that both of these areas provided the highest number of images for training. In fact, of 109 images in the reduced training set, 57 of them come from these two areas of the building. Taking this into consideration, it stands to reason that the tendency towards picking these two areas comes from the fact that over half the training data in this set comes from these two areas.

Based upon the results presented in the confusion matrices in Appendix 1.1 it was apparent that a BOVW trained using data from the "objects" training set produced the correct area of the building more frequently than when the BOVW is trained using the other two datasets. This is highlighted in table 2 below. For this reason, it was decided that the "objects" training set would be used for training the BOVW.

Table 2: BOVW Success Percentages

Training Set	% Correct 1st Choice	% Correct 2nd Choice	% Correct Overall
Objects	31%	13%	44%
Empty	22%	11%	33%
Reduced	20%	20%	40%

4.3 Reference Image Selection Testing

This section discusses the methodology used to test the pipeline's ability to decide on an appropriate reference image, as well as the results that came out of those tests.

4.3.1 Testing Methodology

Once the two potential areas of the building are identified, a reference image is then chosen. In order to determine the efficacy of SSC filtering for this task there were two tests performed. In both tests the returned reference image was given a pass or fail depending on whether it was deemed to adequately capture the portion of the building contained within the query image. See Figure 23 for an example of a well selected reference photo, and Figure 24 for an example of an unacceptable reference photo.

Figure 23: An example of good reference photo selection. Notice that the board is the same board in both photos.

Figure 24: An example of bad reference photo selection. Notice the only shared features are the design on the floor, a pattern repeated throughout the building.

The first test looked at reference photos only chosen by SSC filtered SIFT keypoints. The second test looked at reference images selected by SSC filtered SIFT keypoints, along with SSC filtered ORB keypoints, as well as reference images selected using the regular SIFT and ORB feature detectors. Of the four provided reference images, the pipeline then chose what it considered to the best match. These tests were run on both the naive data pipeline, as well as the full pipeline which utilized a BOVW trained on the "Objects" dataset. Every test image was used during this test for two reasons. The first reason being the need to test whether the BOVW would lead to better reference image selection than the naive approach. The second reason is that due to overlap in the different areas of the building, such as at intersections in the hallways, there is a chance that even if the BOVW incorrectly determined the area of the building for a query image, the reference image selected might still be appropriate and give a correct localization.

4.3.2 Testing Results

A summary of test results is provided below in table 3. In this table the total number of acceptable reference images selected out of all thirty eight total test images, is listed, along with the filter that was chosen most often when the respective pipelines were allowed to choose from multiple filters. A more detailed breakdown of results is

available in Appendix 1.2 in table 20 and table 21.

Table 3: Reference Image Selection Summary

Pipeline	SSC SIFT Only	All Feature Detectors	Detector Chosen
Full Pipeline	10	13	Regular SIFT
Naive Pipeline	15	18	Regular SIFT

4.3.3 Discussion of Results

Upon completion of reference image selection, it is apparent that the BOVW pipeline does not out perform the naive pipeline. Additionally, it appears that when given the choice between using reference image matches generated by homogeneous filtering based on SSC or the typically generated image features, the pipeline tended to favor SIFT features generated normally over the homogeneously filtered features. It was also noticed during testing that every reference image generated via ORB or SSC filtered ORB descriptors did not prove to be acceptable. As such, another test was performed comparing only the results of SIFT and SSC filtered SIFT descriptors. This test was also performed using only the Naive Pipeline, as that showed a greater chance of selecting the correct reference image than the full pipeline using the BOVW. The summary of these test results can be seen in table 4, with the more detailed breakdown available in table 22 in Appendix 1.2.

4.4 Homogeneous Feature Detection Filtering

It was noticed in the previous test that ORB was not chosen to provide a reference image for any of the test images, and in all cases where homogeneously distributed ORB descriptors were used for reference image selection, the chosen reference image was not a good match. Additionally it was noticed that the normal SIFT algorithm

routinely was chosen more frequently than homogeneously distributed SIFT descriptors. Thus, another test was performed, similar to the last, which excluded ORB altogether, and only looked at reference images generated by traditional SIFT and homogeneously filtered SIFT. The intent behind this test was two fold. First, to determine if the number of acceptable reference images selected would improve when the pipeline only used SIFT descriptors. Second, to determine if the trend towards traditional SIFT over homogeneously distributed SIFT held true under this shift in parameters.

4.4.1 Testing Methodology

Given the naive pipeline provided a higher rate of acceptable reference image selection than the full pipeline using BOVW, this test only utilized the naive pipeline. For this test, all thirty eight test images were run through the naive pipeline, using both unfiltered SIFT descriptors and SSC filtered descriptors.

4.4.2 Testing Results

The results of this test are presented in table 4. A more detailed breakdown of the results can be seen in table 22 in Appendix 1.3.

Table 4: SIFT Reference Image Selection Summary

	SSC SIFT	**Regular SIFT**
# Images Acceptable	8	26

4.4.3 Discussion of Results

This test made it apparent that SIFT not filtered homogeneously outperformed SSC filtered SIFT descriptors. While the results in [25] and [26] are that a more even, rather than clustered spread of descriptors leads to better feature matching, this has

not held true inside of Building 640 on the Air Force Institute of Technology (AFIT) campus. Likely, this is due to the fact that there are so many repeated features within the environment that tighter clusters of features led to the improved results when matching features, either through the full pipeline or the naive pipeline.

Another result determined by this test that is worth mentioning is that SSC failed to produce any acceptable reference images for query images generated from the real world, whereas unfiltered SIFT features were able to determine the correct reference image for four out of the five test images. This fact alone, that SSC is less capable of bridging the gap between real and virtual data, makes it a less appealing choice for indoor localization on the AFIT campus than regular SIFT features detection.

4.5 Perspective-N-Point (PNP) Testing

This section presents the results of comparisons in efficacy between the full and naive pipelines in determining camera position.

4.5.1 Testing Methodology

While the naive pipeline proved better at generating an acceptable reference image off which to work, there is still the chance that the end result of the pipelines, the camera's actual position, is more reliable from the full pipeline. When a reference image is chosen, both the naive pipeline and full pipelines begin to look for feature matches between the query and reference images which are used for 2D-3D coordinate correspondences. When the reference image chosen does not contain the appropriate features from the query image, there are three possibilities for the result returned from PNP. They are:

- Too few 2D-3D correspondences are generated and PNP fails, returning a "None" result.

- Enough 2D-3D correspondences are generated, but the result returned is outside the realm of possible (e.g. a negative Z coordinate would require the camera to be under the floor, something which is impossible).

- Enough 2D-3D correspondences are found and a feasible result is returned.

In the event that an incorrect reference image is chosen, the first two scenarios listed above are ideal. They are easily caught with basic error checking, and would thus be considered favorable failures. The third scenario however requires more intricate methods to catch, such as virtual view synthesis, which was used in [12], but not implemented in this paper, and would thus be considered unfavorable failures. Because the full and naive pipeline choose different reference images, they will also have differing rates of favorable and unfavorable failures. This means that a BOVW approach might outperform the naive pipeline when it comes to how favorible their respective failures are.

In order to rule out this possibility, both pipelines were used to generate camera positions for all test images. Additionally, because of the results from homogeneous feature testing and reference image determination testing, pipelines used only either SIFT or SSC filtered SIFT keypoints for both reference image determination and 2D-3D correspondence determination. To address the issue of movable objects and the effect they might have on the accuracy of PNP, both the "Objects" and "Empty" reference datasets were used in testing.

Due to limitations in accuracy of PNP when the number of keypoints is too low, any query image/reference image pairing that generated less than four feature correspondences were discarded. If a result was successfully generated, it was also run through a check to validate that it made sense. The checking function looked to see if the generated result contained any values listed as "Not a Number" or whether or not the numbers generated could feasibly exist within the bounds of the building. For

example if a determined position returned a negative Z coordinate, the camera would have had to be beneath the floor of the building, so the returned position would be discarded.

4.5.2 Testing Results

The summary of results of this test are presented in table 5 with a more detailed breakdown of the results available in Appendix 1.4. Results of this test are presented as the error in the result, meaning the magnitude of the vector connecting the truth position to the calculated position, and the number of test images for which a result was returned. Distances are measured in meters. Table 5 only presents findings for test images generated in the virtual environment. As the images generated in the real world were not tagged with an exact location, it was only possible to generate an approximate truth location against which to compare the PNP results. Due to the fact that on virtually generated test data, results were consistently more accurate using the Object reference dataset, this was the only reference dataset used when testing real world test images. These findings are presented in table 6.

Table 5: PNP Results Summary

Pipeline	**"Objects" # Found**	**Avg Error**	**"Empty" # Found**	**Avg Error**
Full	13	7.034	11	8.91
Naive	22	2.765	20	6.27

Table 6: Real World PNP Results

Pipeline	**# Found**	**Avg Estimated Error**
Full	1	44.837
Naive	3	1.531

4.5.3 Discussion of Results

From these results it is clear that the naive pipeline significantly outperforms the full pipeline. Not only was the naive pipeline able to generate results for more images, it also averaged a significantly higher level of accuracy across the three virtually generated datasets. At the time of collection, truth data was not generated for the real world test images. Instead, this truth data was determined via an estimation of the camera's position using Aftrburner. As precise truth data was not available for the real world test set, and only one camera position was able to be generated by the full pipeline, it is not possible to definitively say that the naive pipeline outperformed the full pipeline. However, in terms of area selection from the BOVW, as well as reference image selection, the performance of both pipelines on real world test images was comparable to their performance on virtual world test images. It stands to reason that with accurate truth data for real world test images, the performance of PNP on those images will also stand up to performance of PNP on virtual world test images. Further test data collection and testing will be required to definitively determine this.

When looking at the individual image results generated by both pipelines for virtual test data, there are several results that stand out as being particularly erroneous. A summary of test results removing images with an error greater than one meter can be seen below in table 7. The percentages of results that are considered to be erroneous are detailed in table 8.

Table 7: Pruned PNP Results Summary

Pipeline	**"Objects" # Found**	**Avg Error**	**"Empty" # Found**	**Avg Error**
Full	10	0.08	8	0.064
Naive	15	0.058	14	0.046

Table 8: PNP Erroneous Returns Percentages

	"Objects" Ref Images	**"Empty" Ref Images**
Full	23%	27%
Naive	32%	30%

Taking this into consideration, after pruning, the naive pipeline still performed better overall, however, the full pipeline, using a BOVW had a lower percentage of results that were egregiously incorrect, twenty three percent as opposed to thirty two percent. Additionally with a difference in average error of slightly more than two centimeters, the advantage gained in ability to trust the results coming out of the pipeline might make the full pipeline a preferred candidate.

4.6 Pipeline Execution Time Comparisons

The final set of tests detailed in this chapter pertain to the execution time required by each implementation of the pipeline.

4.6.1 Testing Methodology

This test was performed by finding the average execution time of each pipeline on every image generated in the virtual environment, thirty three in total. The BOVW was trained using the "Objects" training set, and reference images were generated from the "Objects" reference set. Additionally image features were found using SIFT descriptors not sorted via SSC. Pipeline implementations were written using Python script and run on a 2020 iMac equipped with a 3.6 GHz 10-Core Intel i9 processor.

4.6.2 Testing Results

Results of execution time testing can be seen in table 9.

Table 9: Execution Time Test Results

Pipeline	**# Executions**	**Avg Time/Image**
Full	100	5.10966 sec
Naive	100	2.5829 sec

4.6.3 Discussion of Results

The results of this test show that the naive pipeline also runs far more quickly than the full pipeline, taking about half the time to process an image and return a result that the BOVW requires to do the same. There are several improvements that might be able to be made on the full pipeline that could improve this execution time, such as multithreading reference image determination, but these improvements would also be able to be implemented on the naive pipeline as well, meaning that the naive pipeline would likely still be faster to execute than the full pipeline.

V. Conclusions

The primary aims of this work were to determine the following:

- Is the Bag of Visual Words (BOVW) model sufficiently able to overcome the issue of repeated visual features within a building to narrow down the general location of an image to one of a few possibilities?
- Is a homogeneous sampling of a set of image features, such as that generated by Suppression via Square Covering (SSC) better suited for indoor localization on the Air Force Institute of Technology (AFIT) campus than sparse sampling?
- Is a 3D virtual model of a building generated from a Structure from Motion (SfM) application accurate enough to generate the data used to train a BOVW, as well as the reference data used for 2D-3D correspondence detection?

The research conducted in this paper determined that the BOVW model is likely not sufficiently able to overcome the challenge of repeated features in a building. It was also determined that homogeneous distributions of image features, such as those created by the SSC filter do not provide an advantage over sparse feature distributions when it comes to matching features between images in an indoor environment. In all tests, a naive approach to Visual Localization (VL) employing sparse SIFT feature sampling was able to generate more results, with a higher level of accuracy than a VL approach that employed a BOVW to narrow the search space and homogeneous feature distributions for matching query and reference images.

There was some success in the use of the BOVW when it came to the frequency with which an erroneous result was returned, however without a means by which to automatically determine if a result is erroneous, this advantage falls somewhat

flat. Additionally, the results of this work showed that image data generated in the virtual world is sufficiently able to stand in for data generated in the real world for VL pipelines, at least as far as Building 640 on the AFIT campus is concerned. This determination, combined with the 3D model of Building 640 built during the course of this research should augment future research out of the Autonomy and Navigation Technology (ANT) center in this area.

5.1 Future Work

The next steps that might build upon this work include the following:

- Common features within the building that led to false reference image selection, as well as incorrect feature matches were the patterns on the floor or ceiling. The implementation of a feature mask that removes those features from query and reference images might improve the performance of the data pipeline.

- Other features in the building that are totally unique are room signs. While the general shape and coloring of these signs is standardized, the actual text on the signs is unique. The ability to single out these signs and use them for both Visual Place Recognition (VPR) and even generation of 2D-3D correspondences would likely lead to very accurate results. In order to make this happen though, the 3D model constructed for this research would have to be updated to ensure all room signs are legible and picture perfect. This is something that could be achieved through the use of an application such as Blender.

- The construction of a means of filtering out erroneous returned results from the data pipeline without a prior knowledge of where in the building the camera has been.

- The implementation of a Neural Network, such as NetVLAD or the Neural Net constructed in [2] in an architecture specific to the AFIT campus could lead to better VPR and determination of general area of the building.

Appendix A. Detailed Test Results

1.1 Section 4.2 Detailed Results

Below are results from testing BOVW models. The first choice is the top number and second choice the bottom number in the cell.

Table 10: BOVW Objects Dataset Results

Image	Truth	Objects BOVW	Empty BOVW	Reduced BOVW
image0	1	3 1	4 5	3 4
image1	1	3 4	4 3	3 4
image2	1	3 1	1 2	3 4
image3	4	3 1	4 5	3 4
image4	4	4 1	2 1	3 4
image5	4	2 4	2 1	3 4
image6	4	4 1	2 1	3 4
image7	4	3 2	2 3	3 4
image8	5	4 2	1 2	3 4
image9	5	4 1	2 1	3 4
image10	5	3 2	4 2	3 4
image11	3	3 1	4 1	3 4
image12	3	3 2	4 2	3 4
image13	3	4 1	2 1	3 4
image14	3	3 2	5 3	3 4

Table 11: BOVW Empty Dataset Results

Image	**Truth**	**Objects BOVW**	**Empty BOVW**	**Reduced BOVW**
image0	5	3	5	3
		2	3	4
image1	5	3	5	3
		2	3	4
image2	4	4	2	3
		1	1	4
image3	4	3	2	3
		2	3	4
image4	4	4	2	3
		1	1	4
image5	4	3	5	3
		1	1	4
image6	1	3	4	3
		1	1	4
image7	3	3	5	3
		2	3	4
image8	2	3	4	3
		1	1	4
image9	2	4	2	3
		1	1	4
image10	2	2	2	3
		3	1	4

Table 12: BOVW Moved Dataset Results

Image	**Truth**	**Objects BOVW**	**Empty BOVW**	**Reduced BOVW**
image0	1	3	4	3
		1	3	4
image1	4	3	1	3
		2	2	4
image2	4	3	2	3
		2	1	4
image3	4	3	5	3
		1	1	4
image4	5	3	1	3
		1	2	4
image5	3	3	1	3
		2	2	4
image6	2	3	4	3
		1	3	4

Table 13: BOVW Real World Dataset Results

Image	**Truth**	**Objects BOVW**	**Empty BOVW**	**Reduced BOVW**
image0	1	1	3	3
		5	5	4
image1	1	3	2	3
		2	3	4
image2	3	3	5	3
		2	3	4
image3	3	4	2	3
		1	1	4
image4	4	4	4	3
		1	1	4

1.1.1 Section 4.2 Confusion Matrices

Below are the confusion matrices for each of the three sets of training data. Rows are the truth area, and columns are the area of the building chosen by the BOVW. In the matrices with raw numbers (not percentages), the bottom right most cell of each matrix is the total number of correct guesses, e.g. the sum total of the diagonal line running from top left to bottom right. In the percentage matrices the bottom right cell contains the average success rate across all 5 areas.

1.1.1.1 "Objects" Training Data Set

Table 14: Objects Confusion Matrix

First Choice							Second Choice						
	BOVW Area							BOVW Area					
Truth	**1**	**2**	**3**	**4**	**5**	**Totals**	**Truth**	**1**	**2**	**3**	**4**	**5**	**Totals**
1	1	0	6	0	0	7	**1**	4	1	0	1	1	7
2	0	1	2	1	0	4	**2**	3	0	1	0	0	4
3	0	0	6	2	0	8	**3**	3	5	0	0	0	8
4	0	1	7	5	0	13	**4**	8	4	0	1	0	13
5	0	0	4	2	0	6	**5**	2	4	0	0	0	6
Totals	1	2	25	10	0	13	**Totals**	20	14	1	2	1	5

Table 15: Objects Confusion Matrix Percentages

First Choice							Second Choice						
	BOVW Area							BOVW Area					
Truth	**1**	**2**	**3**	**4**	**5**		**Truth**	**1**	**2**	**3**	**4**	**5**	
1	14	0	86	0	0		**1**	58	14	0	14	14	
2	0	25	50	25	0		**2**	75	0	25	0	0	
3	0	0	75	25	0		**3**	38	62	0	0	0	
4	0	8	54	38	0		**4**	61	31	0	8	0	
5	0	0	67	33	0		**5**	33	67	0	0	0	
						31							13

1.1.1.2 "Empty" Training Data Set

Table 16: Empty Confusion Matrix

First Choice							Second Choice						
	BOVW Area							BOVW Area					
Truth	**1**	**2**	**3**	**4**	**5**	**Totals**	**Truth**	**1**	**2**	**3**	**4**	**5**	**Totals**
1	1	0	0	4	0	5	**1**	1	1	2	0	1	5
2	0	2	0	2	0	4	**2**	3	0	1	0	0	4
3	1	1	0	2	2	6	**3**	2	2	2	0	0	6
4	1	8	0	1	2	12	**4**	8	2	1	0	1	12
5	2	1	0	1	2	6	**5**	1	3	2	0	0	6
Totals	5	12	0	10	6	6	**Totals**	15	8	8	0	2	3

Table 17: Empty Confusion Matrix Percentages

First Choice							Second Choice						
	BOVW Area							BOVW Area					
Truth	**1**	**2**	**3**	**4**	**5**		**Truth**	**1**	**2**	**3**	**4**	**5**	
1	20	0	0	80	0		**1**	20	20	40	0	20	
2	0	50	0	50	0		**2**	75	0	25	0	0	
3	17	17	0	33	33		**3**	33	33	33	0	0	
4	8	67	0	8	17		**4**	67	17	9	0	83	
5	33	17	0	17	33		**5**	17	50	33	0	0	
						22							11

1.1.1.3 "Reduced" Training Data Set

Table 18: Reduced Confusion Matrix

First Choice							Second Choice						
	BOVW Area							BOVW Area					
Truth	**1**	**2**	**3**	**4**	**5**	**Totals**	**Truth**	**1**	**2**	**3**	**4**	**5**	**Totals**
1	0	0	5	0	0	5	**1**	0	0	0	5	0	5
2	0	0	4	0	0	4	**2**	0	0	0	4	0	4
3	0	0	6	0	0	6	**3**	0	0	0	6	0	6
4	0	0	12	0	0	12	**4**	0	0	0	12	0	12
5	0	0	6	0	0	6	**5**	0	0	0	6	0	6
Totals	0	0	33	0	0	6	**Totals**	0	0	0	33	0	12

Table 19: Reduced Confusion Matrix Percentages

First Choice							Second Choice						
	BOVW Area							BOVW Area					
Truth	**1**	**2**	**3**	**4**	**5**		**Truth**	**1**	**2**	**3**	**4**	**5**	
1	0	0	100	0	0	100	**1**	0	0	0	100	0	100
2	0	0	100	0	0	100	**2**	0	0	0	100	0	100
3	0	0	100	0	0	100	**3**	0	0	0	100	0	100
4	0	0	100	0	0	100	**4**	0	0	0	100	0	100
5	0	0	100	0	0	100	**5**	0	0	0	100	0	100
						20							20

1.2 Section 4.3 Detailed Results

Detailed results from Reference Image Selection testing. Listed is acceptability of chosen reference image and chosen feature detector in the multi-filter test.

Table 20: BOVW Pipeline Reference Image Selection

Image	**SSC SIFT Only**	**All Feature Detectors**	**Detector Chosen**
Objects Set			
image0	good	good	SIFT
image1	bad	bad	ORBSSC
image2	bad	bad	ORBSSC
image3	bad	bad	ORBSSC
image4	good	good	SIFT
image5	bad	bad	SIFTSSC
image6	good	good	SIFTSSC
image7	bad	bad	SIFTSSC
image8	bad	bad	ORBSSC
image9	bad	bad	ORBSSC
image10	bad	bad	SIFTSSC
image11	bad	bad	ORBSSC
image12	good	good	SIFT
image13	bad	bad	SIFTSSC
image14	good	good	SIFT
Empty Set			
image0	bad	bad	SIFT
image1	bad	bad	SIFT
image2	good	good	SIFT
image3	bad	bad	SIFTSSC
image4	good	good	SIFT
image5	good	good	SIFT
image6	bad	bad	ORBSSC
image7	good	good	SIFT
image8	bad	bad	SIFT
image9	bad	bad	SIFT
image10	bad	bad	ORBSSC

Image	SSC SIFT Only	All Feature Detectors	Detector Chosen
Moved Set			
image0	good	good	SIFT
image1	bad	bad	ORBSSC
image2	bad	bad	SIFTSSC
image3	bad	bad	SIFT
image4	bad	bad	ORBSSC
image5	bad	bad	ORBSSC
image6	bad	bad	ORBSSC
Real World Set			
image0	bad	good	SIFT
image1	bad	bad	SIFTSSC
image2	bad	good	SIFT
image3	bad	bad	SIFT
image4	bad	good	SIFT

Table 21: Naive Pipeline Reference Image Selection

Image	SSC SIFT Only	All Feature Detectors	Detector Chosen
Objects Set			
image0	good	good	SIFT
image1	bad	bad	ORBSSC
image2	bad	bad	SIFT
image3	good	good	SIFT
image4	good	good	SIFT
image5	good	good	SIFT
image6	good	good	SIFTSSC
image7	bad	bad	SIFTSSC
image8	good	good	SIFT
image9	good	good	SIFTSSC
image10	good	good	SIFT
image11	bad	bad	ORBSSC
image12	bad	bad	ORBSSC
image13	good	good	SIFT
image14	good	good	SIFT

Image	SSC SIFT Only	All Feature Detectors	Detector Chosen
Empty Set			
image0	good	good	SIFT
image1	bad	bad	ORBSSC
image2	good	good	SIFT
image3	bad	bad	SIFTSSC
image4	good	good	SIFT
image5	bad	bad	SIFTSSC
image6	bad	bad	SIFTSSC
image7	bad	bad	SIFTSSC
image8	bad	bad	SIFT
image9	bad	bad	SIFTSSC
image10	bad	bad	ORBSSC
Moved Set			
image0	bad	bad	SIFT
image1	bad	bad	SIFTSSC
image2	bad	bad	SIFT
image3	good	good	SIFT
image4	bad	bad	ORBSSC
image5	bad	bad	ORBSSC
image6	good	good	SIFT
Real World Set			
image0	bad	good	SIFT
image1	bad	bad	SIFTSSC
image2	bad	good	SIFT
image3	bad	bad	SIFT
image4	bad	good	SIFT

1.3 Section 4.4 Detailed Results

Below are detailed results of the comparison of SSC filtered SIFT and non-filtered SIFT descriptor generation. For each image the result is annotated as good or bad.

Table 22: SIFT Reference Image Selection

Image	**SSC SIFT**	**SIFT**	**Image**	**SSC SIFT**	**SIFT**
Objects Set			**Moved Set**		
image0	good	good	image0	bad	bad
image1	bad	bad	image1	bad	bad
image2	bad	good	image2	bad	good
image3	bad	good	image3	bad	good
image4	good	good	image4	bad	bad
image5	bad	good	image5	bad	good
image6	good	good	image6	good	good
image7	bad	good	**Real World Set**		
image8	good	good	image0	bad	good
image9	good	good	image1	bad	good
image10	bad	good	image2	bad	good
image11	bad	bad	image3	bad	bad
image12	bad	good	image4	bad	good
image13	bad	good			
image14	bad	good			
Empty Set					
image0	good	good			
image1	bad	bad			
image2	bad	good			
image3	bad	bad			
image4	good	good			
image5	bad	bad			
image6	bad	good			
image7	bad	bad			
image8	bad	bad			
image9	bad	bad			
image10	bad	good			

1.4 Section 4.5 Detailed Results

Below are the detailed results of error generated for each test image when solving Perspective-N-Point (PNP) for that image. Rows left blank are images for which a result was not generated. Error is measured in meters off from truth.

Table 23: PNP Results: Full Pipeline, Objects Reference Data

Image	**Error**	**Filter**	**Image**	**Error**	**Filter**
Objects Set			**Moved Set**		
image0	0.079	SIFT	image0	0.165	SIFT
image1			image1		
image2	0.009	SIFT	image2	48.661	SIFT
image3			image3		
image4	0.026	SIFT	image4		
image5	0.07	SIFT	image5		
image6	0.005	SIFTSSC	image6		
image7			**Real World Set**		
image8			image0		
image9			image1		
image10			image2	44.837	SIFT
image11	0.282	SIFT	image3		
image12			image4		
image13					
image14	0.151	SIFT			
Empty Set					
image0					
image1					
image2	0.008	SIFT			
image3					
image4	0.005	SIFT			
image5	37.22	SIFTSSC			
image6					
image7					
image8					
image9					
image10	4.763	SIFT			

Table 24: PNP Results: Full Pipeline, Empty Reference Data

Image	**Error**	**Filter**	**Image**	**Error**	**Filter**
Objects Set			**Moved Set**		
image0	0.033	SIFT	image0	0.048	SIFT
image1			image1		
image2			image2	46.17	SIFTSSC
image3			image3		
image4			image4		
image5			image5		
image6	0.009	SIFTSSC	image6		
image7					
image8					
image9					
image10					
image11					
image12	0.189	SIFT			
image13					
image14	0.004	SIFT			
Empty Set					
image0					
image1					
image2	0.016	SIFT			
image3	46.731	SIFTSSC			
image4	0.025	SIFT			
image5	0.188	SIFT			
image6					
image7					
image8					
image9					
image10	4.597	SIFT			

Table 25: PNP Results: Naive Pipeline, Objects Reference Data

Image	**Error**	**Filter**	**Image**	**Error**	**Filter**
Objects Set			**Moved Set**		
image0	0.079	SIFT	image0	0.165	SIFT
image1	7.054	SIFTSSC	image1		
image2	0.009	SIFT	image2	3.933	SIFT
image3	0.009	SIFT	image3	2.188	SIFT
image4	0.026	SIFT	image4		
image5	0.07	SIFT	image5		
image6	0.005	SIFTSSC	image6	0.006	SIFT
image7			**Real World Set**		
image8	0.022	SIFT	image0	1.803	SIFT
image9	0.729	SIFTSSC	image1	1.299	SIFT
image10	0.016	SIFT	image2	1.492	SIFT
image11	0.282	SIFT	image3		
image12			image4		
image13	4.063	SIFT			
image14	0.151	SIFT			
Empty Set					
image0	0.017	SIFT			
image1					
image2	0.008	SIFT			
image3					
image4	0.005	SIFT			
image5	37.22	SIFTSSC			
image6					
image7					
image8					
image9					
image10	4.763	SIFT			

Table 26: PNP Results: Naive Pipeline, Empty Reference Data

Image	**Error**	**Filter**	**Image**	**Error**	**Filter**
Objects Set			**Moved Set**		
image0	0.033	SIFT	image0	0.048	SIFT
image1	6.964	SIFTSSC	image1		
image2			image2	0.065	SIFT
image3	0.015	SIFT	image3		
image4			image4		
image5			image5		
image6	0.009	SIFTSSC	image6	0.021	SIFT
image7	1.867	SIFT			
image8					
image9	0.146	SIFTSSC			
image10	0.018	SIFT			
image11					
image12	0.189	SIFT			
image13	0.021	SIFT			
image14	0.004	SIFT			
Empty Set					
image0	0.028	SIFT			
image1	22.533	SIFT			
image2	0.016	SIFT			
image3	46.731	SIFTSSC			
image4	0.025	SIFT			
image5					
image6	42.066	SIFTSSC			
image7					
image8					
image9					
image10	4.597	SIFT			

Bibliography

1. Faheem Zafari, Athanasios Gkelias, and Kin K. Leung. A Survey of Indoor Localization Systems and Technologies. *IEEE Communications Surveys and Tutorials*, 21(3):2568–2599, 2019.

2. Ricky Anderson. Indoor navigation Using Convolutional Neural Networks and Floor Plans. 2021.

3. Martin A Fischler and Robert C Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Graphics and Image Processing*, 24(6):381–395, 1981.

4. Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-NetVLAD : Multi-Scale Fusion of Locally-Global Descriptors. In *IEEE Conference on Computer vision and Pattern Recognition (CVPR)*, 2021.

5. Yuncheng Lu, Zhucun Xue, Gui Song Xia, and Liangpei Zhang. A survey on vision-based UAV navigation. *Geo-Spatial Information Science*, 21(1):21–32, 2018.

6. Maksim Filipenko and Ilya Afanasyev. Comparison of Various SLAM Systems for Mobile Robot in an Indoor Environment. *9th International Conference on Intelligent Systems 2018: Theory, Research and Innovation in Applications, IS 2018 - Proceedings*, pages 400–407, 2018.

7. T. J. Chong, X. J. Tang, C. H. Leng, M. Yogeswaran, O. E. Ng, and Y. Z. Chong. Sensor Technologies and Simultaneous Localization and Mapping (SLAM). *Procedia Computer Science*, 76(Iris):174–179, 2015.

8. Johannes L. Schonberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic Visual Localization. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 6896–6906, 2018.

9. Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3D point clouds. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7572 LNCS(PART 1):15–29, 2012.

10. Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient Effective Prioritized Matching for Large-Scale Image-Based Localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1744–1756, 2017.

11. Linus Svarm, Olof Enqvist, Fredrik Kahl, and Magnus Oskarsson. City-Scale Localization for Cameras with Known Vertical Direction. 39(7), 2017.

12. Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

13. Arnold Irschara, Christopher Zach, Jan Michael Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. *2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009*, pages 2599–2606, 2009.

14. Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 Place Recognition by View Synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2):257–271, 2018.

15. Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. Are large-scale 3D models really necessary for accurate visual localization? *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:6175–6184, 2017.

16. Zetao Chen, Adam Jacobson, Niko Sunderhauf, Ben Upcroft, Lingqiao Liu, Chunhua Shen, Ian Reid, and Miachael Milford. Deep Learning Features at Scale for Visual Place Recognition. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3223–3230, Singapore, 2017.

17. Eric Brachmann and Carsten Rother. Learning Less is More - 6D Camera Localization via 3D Surface Regression. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 4654–4662, 2018.

18. Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC – Differentiable RANSAC for Camera Localization – Supplementary Materials –. *Cvpr 2017*, (2):1–2, 2017.

19. Tommaso Cavallari, Stuart Golodetz, Nicholas A. Lord, Julien Valentin, Luigi Di Stefano, and Philip H.S. Torr. On-the-fly adaptation of regression forests for online camera relocalisation. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:218–227, 2017.

20. Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2D-to-3D matching. *Proceedings of the IEEE International Conference on Computer Vision*, pages 667–674, 2011.

21. Ebrahim Karami, Siva Prasad, and Mohamed Shehata. Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images. 2017.

22. David (University of British Columbia) Lowe. Distinctinve Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 2004.

23. Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1437–1451, 2016.

24. Baiying Lei, Yuan Yao, Siping Chen, Shengli Li, Wanjun Li, Dong Ni, and Tianfu Wang. Discriminative Learning for Automatic Staging of Placental Maturity via Multi-layer Fisher Vector. *Scientific Reports*, 5(January):1–11, 2015.

25. Oleksandr Bailo, Francois Rameau, Kyungdon Joo, Jinsun Park, Oleksandr Bogdan, and In So Kweon. Efficient adaptive non-maximal suppression algorithms for homogeneous spatial keypoint distribution. *Pattern Recognition Letters*, 106(February):53–60, 2018.

26. Konstantin Schauwecker, Reinhard Klette, and Andreas Zell. A new feature detector and stereo matching method for accurate high-performance sparse stereo matching. *IEEE International Conference on Intelligent Robots and Systems*, pages 5171–5176, 2012.

27. Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A convolutional network for real-time 6-dof camera relocalization. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 International Conference on Computer Vision, ICCV 2015:2938–2946, 2015.

28. Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:6555–6564, 2017.

29. Bandaraike. Published version. IV(3):223–233, 2015.

30. Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. [MapNet:PoseNet + ResNet34] Geometry-Aware Learning of Maps for Camera Localization. *Cvpr*, pages 2616–2625, 2018.

31. Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-taix. Understanding the Limitations of CNN-based Absolute Camera Pose Regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3302–3312, 2019.

32. Yaguang Kong, Wei Liu, and Zhangping Chen. Robust ConvNet Landmark-Based Visual Place Recognition by Optimizing Landmark Matching. *IEEE Access*, 7:30754–30767, 2019.

33. Paul Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:12708–12717, 2019.

34. Niko Sünderhauf, Sareh Shirazi, Feras Dayoub, Ben Upcroft, and Michael Milford. On the performance of ConvNet features for place recognition. *IEEE International Conference on Intelligent Robots and Systems*, 2015-Decem:4297–4304, 2015.

35. Lili Meng, Jianhui Chen, Frederick Tung, James J. Little, Julien Valentin, and Clarence W. De Silva. Backtracking regression forests for accurate camera re-

localization. *IEEE International Conference on Intelligent Robots and Systems*, 2017-September:6886–6893, 2017.

36. David Filliat. A visual bag of words method for interactive qualitative localization and mapping. *Proceedings - IEEE International Conference on Robotics and Automation*, (May):3921–3926, 2007.

37. Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: An Open-Source Library for Real-Time Metric-Semantic Localization and Mapping. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1689–1696, 2020.

38. Kirk Mactavish, Michael Paton, and Timothy D. Barfoot. Visual triage: A bag-of-words experience selector for long-term visual route following. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2065–2072, 2017.

39. Thorsten Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Europena Conference on Machine Learning*, page 70, 1998.

40. Jun Yang, Yu Gang Jiang, Alexander G. Hauptmann, and Chong Wah Ngo. Evaluating bag-of-visual-words representations in scene classification. *Proceedings of the ACM International Multimedia Conference and Exhibition*, pages 197–206, 2007.

41. Loukas Bampis and Antonios Gasteratos. Revisiting the Bag-of-Visual-Words model: A hierarchical localization architecture for mobile systems. *Robotics and Autonomous Systems*, 113:104–119, 2019.

42. Stephanie Lowry, Niko Sunderhauf, Paul Newman, John J. Leonard, David Cox, Peter Corke, and Michael J. Milford. Visual Place Recognition: A Survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016.

43. Vincent Bownes. Using Motion Capture and Augmented Reality to Test AAR with Boom Occlusion. 2021.

44. Thomas Sayre-Mccord, Winter Guerra, Amado Antonini, Jasper Arneberg, Austin Brown, Guilherme Cavalheiro, Yajun Fang, Alex Gorodetsky, Dave McCoy, Sebastian Quilter, Fabian Riether, Ezra Tal, Yunus Terzioglu, Luca Carlone, and Sertac Karaman. Visual-Inertial Navigation Algorithm Development Using Photorealistic Camera Simulation in the Loop. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2566–2573, 2018.

45. Anne Bettens, Benjamin Morrell, Mauricio Coen, Neil McHenry, Xiaofeng Wu, Peter Gibbens, and Gregory Chamitoff. Unrealnavigation: Simulation software for testing slam in virtual reality. *AIAA Scitech 2020 Forum*, 1 PartF(January), 2020.

46. Siddharth Choudhary and P J Narayanan. Visiblity Probability Structure from SfM Datasets and Applications. 7576(October), 2012.

47. Erik Bochinski, Volker Eiselein, and Tomas Sikora. Training a convolutional neural network for multi-class object detection using solely virtual world data. *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2016*, (August):278–285, 2016.

48. Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. VirtualWorlds as Proxy for Multi-object Tracking Analysis. *Proceedings of the IEEE Com-*

puter Society Conference on Computer Vision and Pattern Recognition, 2016-Decem:4340–4349, 2016.

49. Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. *IEEE International Conference on Intelligent Robots and Systems*, 2017-Septe:23–30, 2017.

50. Torsten Sattler, Michal Havlena, Filip Radenovic, Konrad Schindler, and Marc Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter:2102–2106, 2015.

Acronyms

AAR Automated Aerial Refueling. 14

AFIT Air Force Institute of Technology. iv, 2, 6, 14, 21, 40, 46, 47, 48

ANMS Adaptive Non-Maximal Suppression. 8

ANT Autonomy and Navigation Technology. iv, 2, 14, 47

APR Absolute Pose Regression. 10

AR Augmented Reality. iv

BLE Bluetooth Low Energy. 1

BOVW Bag of Visual Words. iv, vi, viii, x, 2, 3, 12, 13, 14, 19, 20, 26, 27, 28, 30, 31, 33, 34, 35, 36, 37, 38, 39, 41, 43, 44, 45, 46, 49, 50, 51, 52, 53, 54, 55

BOW Bag of Words. 12

CNN convolutional neural network. 8, 11, 19, 20

GNSS Global Navigation Satellite System. 1

GPS Global Positioning System. iv, 1

LDP Location Determination Problem. 4

ML Machine Learning. 10, 12, 16

PNP Perspective-N-Point. vii, 4, 6, 7, 19, 26, 27, 30, 40, 41, 42, 43, 59

RANSAC random sample consensus. 4

RFID Radio Frequency Identification. 1

RGB Red Green Blue. iv, 2, 5

SAM Smoothing and Mapping. iv, 5, 10

SfM Structure from Motion. viii, 3, 5, 6, 15, 46

SLAM Simultaneous Localization and Mapping. iv, 5, 10, 14

SSC Suppression via Square Covering. viii, 2, 8, 9, 11, 20, 33, 36, 37, 38, 39, 40, 41, 44, 46, 55, 56, 57, 58

SVM State Vector Machine. 12, 13, 14

UAV unmanned aerial vehicle. 14

VFL Virtual Flash Lidar. 30

VL Visual Localization. iv, viii, 2, 4, 5, 6, 8, 10, 11, 12, 15, 16, 20, 46, 47

VPR Visual Place Recognition. 4, 5, 6, 8, 10, 11, 12, 14, 16, 19, 20, 47, 48

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
19–03–2020	Master's Thesis	Sept 2018 — Mar 2020

4. TITLE AND SUBTITLE

Monocular Camera Localization Using a Bag of Visual Words from Virtual World Data

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Joshua A. Rinaldi

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-MS-22-M-057

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

USA DEVCOM/C5ISR
Daniel A. Dekowski
Building 6007
Aberdeen Proving Ground, MD 21005
Email: daniel.a.dekowski.civ@army.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

DEVCOM/C5ISR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Visual Localization problem is the question of how to use visual information to determine the location of the camera that captured that data. The wide availability and low price of RGB cameras has made this useful in many fields such as SLAM, SAM and AR. This research seeks to determine if a BOVW can adequately overcome repetitious features in indoor environments and be effectively incorporated into a VL pipeline.

15. SUBJECT TERMS

Visual Localization, Virtual Worlds, Bag of Visual Words

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Clark N. Taylor, AFIT/ENG
U	U	U	UU	84	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636, ext 4614; clark.taylor@afit.edu