The University of Melbourne
Faculty of Engineering & IT

SLAM in Mining Environment

A proposal for immediate control of sensor fusion

Tuan Khoi Nguyen
Department of Mechanical Engineering
The University of Melbourne
Parkville, VIC, Australia

Introduction

In this modern day of technology advancement, automatic operation is made to perform tasks in place of human. This has become very useful for hazardous environments, where various dangers can affect one's safety, and the mining industry is one among them: the mining environment have low light and dusty environment, where debris or timber drops can happen anytime. Simultaneous Localization and Mapping (SLAM) have been a commonly used method to automate Unmanned Vehicles (UV) to self-navigate in dark tunnels, yet they still have many limitations in resource and environments that needs an efficient approach of solving. Therefore, there is potential for methods that can extract useful information from sensors that produces minimal data, or the use of memory-efficient algorithms. This article aims to investigate the possibility of using sensor fusion in solving the SLAM problem, by proposing an Dual Feedback, Event-Triggered control model, with potential utilization of Wavelets for data combination and processing.

1 Context & Motivation

1.1 UV in mining industry

In mining environment, various factors are considered dangerous for human health. Such examples includes toxic fumes, falling debris or machinery noises [1]. Therefore, in mining industry, Unmanned Vehicles (UVs) are commonly used for exploration tasks in hazardous environment, such as navigation or investigating unexplored areas [2].

One of the main concerns when using UVs, is that it needs a method to be able to keep track of itself, as well as navigating in constrained environments. Simultaneous Localization and Mapping (SLAM) fits this requirement as its name suggests, by attempting to retrieve at the same time, the map and terrain of the environment (mapping), and where the device is with respect to that map (localisation). Not only being useful for an UV to locate itself, SLAM can also be carried out to gain useful information for its users, for instance, detecting slopes on the path, find optimal path load transport, or using object detection and tracking, locate out spillages or blockades.

While being a helpful potential to help UV operate without human control, SLAM may still have limitations that relate to very common problems of modern world processing. First, common SLAM algorithms rely on image (visual SLAM) or LIDAR (LIDAR SLAM) inputs, but the environments in the mining industry is not optimal for the data retrieval of these sensors. The majority of operating environment in mining is tunnel, where light can be limited and dusts can cause diffused lighting [2, 3]. Furthermore, in tunnels or underground, geo-magnetic information such as GPS may not be available [3], which makes evaluation of SLAM performance becoming more difficult. Another problem that existed is computational efficiency. Due to budget limitation and sizing constraints, the majority of industry computers have limited amount of processing power [4]. While SLAM produces good results, it needs to operate on every time step, hence requiring large processes to be done on high-resolution images within a short amount of time, which may not be feasible with industry computer's memory capacity. Overall, image quality degrade and lack of computational resources are the major challenges in mining that need to be overcome in order to operate SLAM efficiently.

1.2 Research Direction

While SLAM is known to rely on image or LIDAR sensory inputs, which is susceptible to lighting and perception problems, other types of sensors are not as affected. There are types of sensors that are not affected by darkness, such as ultrasonic, or resistant to diffused lighting, such as radio sensors. Therefore, there can be potential to use other sensory inputs to aid, or substitute image inputs when performing SLAM. These potentials may be able to reduce the problems that traditional SLAM might have in mining environments.

With the potential to use sensors for SLAM in mining, the main question focuses on how these ideas can be integrated into the already existing system, in order to avoid potential problems of the traditional image input framework, specifically environments with bad perception, and limited computation power. The initial research question can now be formally reworded as follows:

> "How can we utilise sensors to improve on our SLAM results, so that the UV can efficiently operate in visually degraded environments?"

With the initial research question now defined, the next parts in this article will have an in-depth exploration of possible ideas and methods, by reviewing literature that focused on relevant topics, and compare the proposed ideas on how relevant it is to the research problem, and how feasible it is to implement in real life. The research question will then be further refined, so that our problem can be better well-defined.

2 Literature Review

How to optimize the self-navigation of unmanned vehicles in visually degraded environments is a common research topic on its own, and the use of SLAM with aiding sensor components is also commonly considered in a wide variety of component choice and sensor fusion methods. Therefore, literature review will help in looking for potential ideas that can solve the research problem. In order to search for related articles with related ideas, multiple keywords were used in scholar search engines. From the main keywords "SLAM operation in challenging indoor environments", the search was expanded into more specific keywords such as "low light environments", "foggy view", "computationally efficient". Overall, 15 papers were found in total, and 11 among them directly tackle the problem of visually degraded environments. Through a more detailed reading, 5 papers we found to be relevant to the topic, where 2 of them were strongly related to our problem of interest - application for UV control using SLAM in tunnel-like environments. The rest 4 papers was not investigating to the same direction as the research problem, yet they provided useful insights that can be considered for the development process when finding a solution to the research problem.

2.1 Vehicle Types

There is a large variety of existing automated vehicles that can perform tasks in place of human work. In this report, we consider 2 most common types of unmanned vehicles being discussed in the literature - Unmanned Ground Vehicle (UGV) and Unmanned Aerial Vehicle (UAV). Both vehicles can be operated automatically without human control, and can keep the operating computation feasible for standard industry computers [5]. UAV has the capability to operate mid-air, making it easier to move in complex terrains, as well as gaining larger observation scope of the environment. However, UAV operation also requires more complex procedures to keep the vehicle balanced while airborne. It has up to 6 degrees of freedom that requires accurate supervision, and have low capability of carrying loads [6]. Furthermore, while the view scope is large, smaller details may have low accuracy, as they will be very small in the UAV perception. UGV also operates like UAV, but it only operates on ground, and have a smaller scope due to the limited view from ground. However, with the wheel support, UGV can carry much more load than UAV. UGV also requires less complexity for controlling, with only 3 degrees of freedom need to be supervised. UGV can also provide better detailed of closer objects, especially ground obstacles, as it can easily approach them at a closer distance than UGV [6].

Present research have looked on the possibilities of combining both UAVs and UGVs for operation, as their characteristics complement each other [5, 6]. However, with the limitation of computational resource in the mining industry, this may not yet be feasible, and requires more research to be done for this specifically. In mining environment, indoor areas like tunnels may be limited space-wise, making UAV not able to optimize its airborne advantages. Furthermore, the majority of common tasks in mining consist of carrying loads and close-up obstacle detection, which UGV has an advantage on. Therefore, UGV will be chosen to be our main operating vehicle of interest for the SLAM problem.

Processor Capacity While it is known that the computational power of industrial machines are limited, a review on this section is still essential to have a rough estimation of how feasible the UV processors are with respect to the amount of computation. In the past, most device operates on micro-controllers such as Arduino Uno or micro-processors like Raspberry Pi. However, these controllers are only feasible for small operations that read or process single signals at a time, which may not feasible with SLAM tasks that read in multiple signals at a time [7]. More modern controllers have considered the use of compact PC like Intel NUC, which have comparable capacity to a modern PC in terms of processing power, but may still not feasible enough for heavy memory consumption tasks such as image processing, deep learning, or task sets that requires concurrent working that can easily cause conflicts. For higher processing capacity, more combination architectures have been constructed, which either dedicate separate PCs or databases to specific tasks [8, 9], or combine multiple memory access to create more powerful system [7]. However, this may drastically increase the load that the UV have to carry, which may not be desired given that it may also need to carry the load from its own mining tasks. Therefore, for sufficient operation in the mining industry, it is feasible that the algorithm should only take up the processing power of a single mini PC, which is approximately 16 gigabytes of memory for Intel NUC 11 Pro, the current common modern mini PC up to date.

2.2 SLAM Methodologies

SLAM is a large and continuously expanding research topic in the field of Computer Vision. Within the topic of SLAM lies more specialized subsets of operations that suits more specified requirements. SLAM can be formally divided into 2 steps that can be represented as individual topics of research: front-end and back-end [10]. Front-end is the sensor-dependent part

of SLAM, where the sensory inputs are processed to provide an estimation or data structure of the information that user need, with notable examples including heading estimation and object detection. Back-end otherwise, is sensor-agnostic. Rather than working directly on sensor inputs, this stage uses the processed data given from the front-end, and use it to provide the essential information of a SLAM problem: the map of environment, and where the vehicle is located with respect to that map.

This section will perform literature review for both front-end and back-end part of SLAM. For the front-end part, different types of sensors will be reviewed to investigate each's feasibility in the mining environment, and for the back-end part, methodologies on how the chosen data inputs can be combined will be evaluated on to check for compatibility with the low-resource requirement of industrial computers.

2.2.1 Input Data

In existing literature, improvement ideas have been implemented on a wide variety of sensory components that provides additional information for consideration. These are ranged from attaching additional lighting for surroundings, to providing extra spatial information using waves. While many articles specified that they are tackling visually degraded environments, most only considered low-lighting as the main problem, while other problems were not considered, and one of our main problems - dusty environment, is among those problems. Therefore, other than the SLAM methodologies related to our research problem and their primary input source, review on plausible external sensors also extends to the use of sensors that can tackle the diffused view problem. A summary of this part is shown in Table 1.

Visual SLAM Visual SLAM works out the map, by taking picture inputs on an area from different angles, and use geometry method such as triangulation or epipolar geometry to reconstruct the 3D map of that area [11]. Camera is the essential component in visual SLAM, given that its image output is the primary input for the algorithm [12]. However, as mentioned in the previous sections, as the photo sensors operate on lighting, tunnel areas like mining environments can cause changing and challenging lighting conditions, which makes it hard for the camera to retrieve useful inputs. Several proposed the idea of attaching an Light Emitting Diode (LED) that lightens up the surrounding environment is a convenient way to help the camera getting better inputs in low lighting, with no extra computation cost [2, 4]. However, LEDs cannot shine through environments like dust or fog, hence not capable of fully solving our recurring problem.

LIDAR SLAM As the name have specified, this SLAM method is based on one or many LIDAR sensors, which captures information by shooting light beams and calculate the time they bounce back to the emitter [2]. There are 2 types of LIDAR, which differs in how the resulting measurement is returned, reflected by the name: 2D and 3D LIDAR. 2D LIDAR has a simple working mechanism, and work under low-lighting condition [4]. However, UV is not a stationary device, and movement can negatively affect the result quality. Furthermore, 2D LIDAR limits its results to planar projection, which makes the scope of view limited. In terms of measurement details and quality, 3D LIDAR provides more information than 2D LIDAR. It can retrieve information within 360^o view, as well as maximizing depth capture [13]. However, as with many sensors that provide information-rich data, 3D LIDAR also requires high computation to process, or just storing the data [13]. Furthermore, as light beams are discrete, further objects might create sparse data, where there are gaps with unknown depth in between the data points, which may be further distorted when the sensor moves [11].

RGB-D SLAM This SLAM method uses a RGB-D camera to retrieve its inputs. RGB-D camera works like a normal camera on visual SLAM, with the addition of extra depth measurement using a time-of-flight infrared (IR) sensor [3, 14]. Being able to provide high resolution input for the depth, it covers the sparse data problem that a LIDAR may have [13]. However, the depth information from this sensor has limited accuracy, due to data input being limited up to a specific depth [15], as well as susceptible to abnormal lighting such as occlusion [13], and needs further improvement to get more accurate depth information.

Internal Measurement Unit (IMU) IMU consists of 3 components: an accelerometer, a gyroscope, and a magnetometer. An IMU system will be able to help a vehicle to keep track of its acceleration and heading, in addition to the SLAM system [2, 14]. It is easy to implement, can provide measurement at high frequency and have negligible computation cost [14]. As it operates based on movement instead of ray projection, IMU cannot be affected with visual challenges. However, it is notable here that the mining environments, especially metal mines, are prone to magnetic field defection, hence can distort the magnetometer operation beyond calibration due to too varied magnetic fields across the mine. Furthermore, accelerometer and gyroscope can be strongly affected by drifting

overtime, which may cause squared error in further estimations like heading or velocity.

Radar Most inputs shown above works only in cases of dark environment, and is not guaranteed to work under diffusion such as fog or dust - common visual limitations in mining. Therefore, more sensor types can also be considered to solve this problem. Radar sensors working mechanism is similar to LIDAR: it sends radio waves, and gets the distance by measuring the travel time of that signal. As it does not rely on lighting, the radar signal is robust to different lighting conditions, including diffusion environment [16]. However, the radar sensor lack accuracy, especially when the sensor is moving, just like LIDAR. Distortion also happens when the object is close to the sensor [17], making it hard to operate if the UV is moving in small and narrow paths.

Evaluation & Choice Overall, it is notable that each type of sensor input has its own pros and cons. This proves that it will be beneficial to combine multiple sensors at once to ensure that SLAM estimation is robust to the visual challenges. Deciding between Visual SLAM, LIDAR SLAM and RGB-D SLAM is a complicated process that may require thoughtful analysis and rigid experimentation procedure, given that each have its own characteristics for operation in the mining environment. However, Visual SLAM can be a feasible choice to start, as it requires the least data space usage due to not containing depth information, as well as input being easier to process with many image processing algorithms readily available. Therefore, for the rest of this paper, especially the experimental section, will test on Visual SLAM, specifically trying procedures on image inputs.

2.2.2 SLAM Fusion Implementations

Not only on the type of inputs that SLAM can receive, how these inputs can be combined also has an abundance of ideas being proposed through different literature. Each method works differently, hence have different characteristics in computation, environment feasibility, or input-output requirements. Therefore, as we are specifically interested in the mining environment, each fusion idea will be evaluated, and see how well can it be integrated into the existing systems. The summary of this part is shown in Table 2.

Deep Learning With the arising trend of deep learning, many present papers considered the use of neural networks for sensor fusion [12, 13]. Neural networks learn the arbitrary representation of the data, and tries to interpret the information or surrounding properties, in a tailored way to be able to perform specific tasks. Neural networks have high reputation for having state-of-the-art accuracy when it is well-trained, and may even outperform human perception in some cases. There are 2 types of Deep Learning SLAM: supervised and unsupervised [11]. Supervised uses existing maps and data that are already available to train the model, so that it knows what to do using prior knowledge, and can be computationally efficient in making predictions given that the model is well-trained with information. Unsupervised models otherwise, does not require prior samples to be learned. These models can dynamically adjust themselves to adapt to changes within the environment, satisfying the online learning problem - where a model is required to quickly learn new problems coming without any certainty of what will come.

However, for both supervised and unsupervised learning, there may be problems being specific to the mining environments. For supervised learning firstly, it can only work well when sufficient amount of existing data is feed into the model, which in our case of SLAM problem, is the true map of the tunnel environments, and where the sensors are located within that map. This may then become a 'chicken-or-egg' problem: we use the model to retrieve information of the tunnels in place of human work, yet, we need the ground truth information to train that either requires human work, which counters the purpose of this model, or prior results of model, which we are trying to get at present. Without a way to get the ground truth, it may be hard to get a supervised model working. Unsupervised model can solve the problem of no prior knowledge, but it has another problem: efficiency. An unsupervised model have to do its training and prediction at the same time, and therefore will require much more computation power to store the newly learnt information prior to outputting its estimations. With the computation power of industrial resources being limited, this training process may not be efficient enough when the power consumption exceeds the machine capacity. This can either cause slower progress until a model can learn with high accuracy, or a system break down that might interrupt an ongoing progress. Therefore, while being state-of-the-art, deep learning might not be suitable for SLAM mining environment, due to the nature of the methods requiring either high computation power, or high amount of existing data.

Probabilistic Models (GraphSLAM) GraphSLAM represents the environment as a factor graph, where each node represents the object pose at a time instance, and each edge contains a transition measure-

Device	Advantage	Problem
Camera	Output is primary input for Visual SLAM [12]	Not working well under visual challenges
LED [2, 4]	No extra computation, efficient in low-light	Not efficient under diffusion (dust, fog)
2D LIDAR [4]	Input of LIDAR SLAM, efficient computation, work in low-light	Inefficient during movement, restricted to planar projection
3D LIDAR [13]	Input of LIDAR SLAM, 360^o cover, maximized depth capture [13]	Computationally expensive, sparse resolution [13]
RGB-D [3, 14]	Input of RGB-D SLAM, simple integration, high resolution data	Lack depth information [13]
IMU [2, 14]	Immune to all visual challenges	Inefficient if GPS not available, drifting
Radar [16, 17]	Robust to diffusion [16]	Low accuracy, hard to track movement, can be distorted in close-range [17]

Table 1: Summarizing table for evaluation of possible sensor inputs in SLAM

ment to move from one pose to another, such as acceleration or angular velocity [2, 4, 18]. The model considers SLAM as an optimization problem on probabilities, where the posterior of interest $p(y_t|c_t, z_t)$ - likelihood at time t of sequence $y_t = [x_1, x_2, ..., x_t]$ given controller estimation c_t and measurement z_t, is expanded as a recursive sequential product of timely transition priors using Bayes Theorem as follows:

$$p(y_t|c_t, z_t) = p(y_0) \prod_{i=1}^{t-1} p(x_i|x_{i-1}, c_{i-1}) p(z_i|y_{i-1}, c_{i-1})$$

GraphSLAM will then solve a system of equations to pick out the sequence that provides a critical value of log posterior, which represents the path that the vehicle have most likely taken. This system is robust to external disturbance and highly accurate, as it bases on how factors interact with each other, which can be accurate under sufficient samples. However, being made for offline SLAM problems [18], it requires many variables to be present in the computer, with quantity expansive proportionally to the number of factors, hence requiring significant power to perform simultaneous calculations, and the computation limit on industrial machines may not allow the process to happen. Furthermore, as GraphSLAM is based on Bayes Theorem, the assumption that factors are independent must hold well in order to keep the system error-free. As observed in the previous sections, the visual limitation of the mining environments may simultaneously affect the controllers and the sensors, which may in turn link these factors together, making the assumption invalid in the SLAM problem.

Feedback Models (EKF SLAM) Feedback model works in the form of a feedback controller system, where the output can reflect on the accuracy of model's estimation, and that information is then incorporated into the model to account for the error. EKF SLAM uses an advanced version of feedback controller called Extended Kalman Filter (EKF), where both model estimation and the measurement are considered to be prone to disturbance errors, and needed to compensate each other on a weighting basis [19]. The EKF model will attempt to dynamically adjust the weighting between the measurement and the estimation, so that downside of each such as drifting or white noise will be alleviated. The 'extended' term in EKF refers to the fact not only will the model predict the variable of interest, but it will also attempt to simplify the complex controller models to low-order polynomials, so that computation time can be significantly reduced. Figure 1 shows an example of EKF in pose estimation, where the estimation of the local position and its orientation is checked for correctness with the measurement given from the IMU.

The downside of using this method comparing to other methods relates to the fact that EKF SLAM is online, which means it will not keep the previously processed information [18]. While this helps to keep the computation use efficient, it requires the environment to be represented by a well-formed discrete state-space model so that information can be retained as much as possible in a compact-sized data structure. The more detailed the model is with larger quantity of state variables, the more accurate final prediction will be. This is possible to do for our interest of mining environment, given that enough time is spent on developing a robust model, which is in fact, more reasonably feasible than the need of human work to sample data for deep learning models, or constructing a large system of factor graphs. Therefore, EKF SLAM has potential to be the best choice of sensor fusion methods in an environment like mining, as it is not data hungry, nor in have high computation requirement. With this, EKF SLAM will be our primary focus for the later development stages of the research problem.

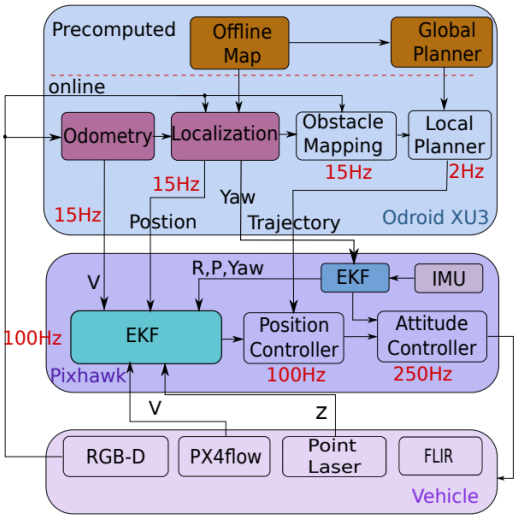


Figure 1: An EKF SLAM model used in pose estimation for UAV [3]

3 Problem Formulation & Evaluation

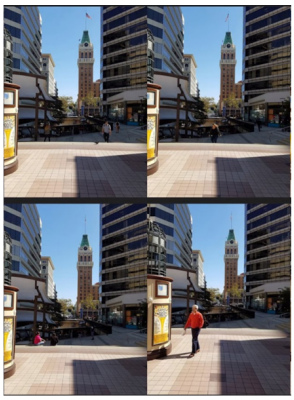
3.1 Research Problem

So far, literature review have narrowed down our research question to many specifications. For vehicle, UGV will be assumed as the main operating device, due to its advantageous ability in performing common mining tasks in limited space and high loading capacity. For upper bound on computation, the algorithms should be feasible to run at a reasonable time on the processing power of a mini-PC. For sensor choices, the combination of multiple sensors have been found to be potentially beneficial, given that the characteristics of each will be reciprocal altogether. And for how to combine the sensor inputs, specifically for the case of low computational resources and dark or dusty operating environment, Extended Kalman Filter, or formally known as the main mechanism of EKF SLAM, will be the methodology of interest. With all of the algorithms and choices made, the research problem now can be further refined, on how these choices can come together for our interest:

> "How can we utilise measurements of multiple sensors to improve on our SLAM results, so that the UGV can operate in visually degraded environments, with sufficient use of computation similar to a standard industrial PC?"

The next part of this section will explore deeper on possible directions of our research problem, in order to come up with suitable methods that can solve our problem efficiently.

3.2 Semi-discrete environment (SDE)



$l(t) + h(t) = L + \Delta l(t) + h(t)$

$L + \Delta l(t) \approx L$

Figure 2: Example of SDE on a set of images taken at the same location but different times

The articles that attempted to use EKF SLAM in the literature review have a common structure in the algorithm flowchart. This structure includes the processing of 2 distinct signals: one with high resolution that samples at a low frequency, and the other samples at a much quicker frequency, but provides less information. The latter of these 2 signals will provide supplementary information for the other one, which serves as a 'frame' that waits for more information to be put in. In literature, the high frequency - low resolution measurements are taken from sensors that can operate quickly and provide a measurement without heavy use of computation, with notable examples including IMU or low-resolution camera. On the other hand, the low frequency - high resolution measurements can be retrieved from information-rich sensor or sensor systems such as LIDAR, stereo cameras, or RGB-D camera. However, as mentioned in the literature review section, these types of sensors can be costly in terms of computation, as larger amount of information will need to be handled spontaneously.

As seen in the literature review, the systems in the EKF SLAM designs are providing additional information from the high-frequency signals to low-frequency ones, as a feedback term [3, 14, 16]. However, given that the high-frequency measurements are more susceptible to external disturbances, which can easily make the feedback unreliable, the high-frequency signals (HFS) will also need correction. It is also noticed here that the low-frequency sensors (LFS) have a larger amount of information, which brings robustness to the measurement noises. Therefore, the 'frame'

Fusion Method	Advantage	Problem
Deep Learning (CNN) [12, 13]	If well-trained: High accuracy, can learn useful representations without many parameters [12, 13]	Either requires offline pre-training, or online update with large computation cost
Probabilistic model (GraphSLAM) [2, 4]	Robust, accurate if factors are independent [2, 4, 18]	Unreliable in larger spaces, where dependencies is high, computation cost expansive [4]
Extended Kalman Filter (EKF SLAM) [3, 14, 16]	Online, computation-efficient [3, 14]	Requires an accurate state model to be constructed

Table 2: Summary table of possible sensor fusion methods in SLAM

provided by the LFS may also contains information that can help to correct the HFS outputs, and as those information are low-frequency, they should be measurements that does not change too rapidly over time.

To formalize the idea above, a general formulation of a semi-discrete environment (SDE) model $E(t)$ can be shown as follows:

$$E(t) = l(t) + h(t) \approx L + h(t)$$

Here, $l(t)$ represents the low-frequency elements, which changes so little over time t that we can approximate it to the constant L. $h(t)$ here indicates the high-frequency elements that change fast over time. A visual example is shown in Figure 2, where dynamic factors like human movements represents the high-frequency $h(t)$.

Now that the formulation is set, focus is now on the proposal of an immediate controller in the feedback model structure. This controller will need to be able to extract L from LFS, and then combine it with $h(t)$ information from HFS to provide a good estimation of the surrounding environment $E(t)$.

3.3 SDE problem in SLAM

There are multiple ways to interpret an SDE in the SLAM problem. However, we are more interested in how the inputs can be processed, so the focus will be on analysing the components in sensor signals. First, it can be observed that in a SLAM result, specifically for mining, low-frequency components that will not change rapidly with respect to time will be the true environment map (the ground truth), along with the static obstacles on the way such as timber or boulders. These can be represented presence of frequencies and their temporal location in visual SLAM, or a signal representing distance across an axis on LIDAR SLAM. High-frequency otherwise, changes as the vehicle moves along the map. The apparent variable is then the vehicle's pose with respect to the map, which consists of its location and heading. This in the SLAM inputs, can be represented with how much the signals have shifted with respect to the previous sample. Figure 3 demonstrates an example case, where if the camera moves slightly, the distance between the cars or the background will not change much, and the most observable change is the car near the camera being shifted over. In mining and for our approach, there may not be simultaneous input retrieval like this example, but as most objects are static in mining, it can apply to consecutive tracking from a single input as well. Therefore, it can be noticed that for our SLAM problem, the frequencies remain in form spatially, yet they can be translated through each time step. This hints the use of a method that can extract the frequencies in the signal, as well as where they lie with respect to a moving frame of reference, commonly attached to the UGV's chassis.

Figure 3: Image inputs [20] from a stereo camera pair, demonstrating the effect of shifting camera location

4 Methodology

Now that the direction of research is more refined, an overview on the proposed idea can be made. We need to be able to perform 2-way communication between LFS and HFS. While HFS provides quick information that LFS cannot capture in time, HFS can in turn pro-

vide a checking to ensure that LFS is valid, and make any necessary adjustments. A figurative demonstration for the logic is shown in Figure 4.

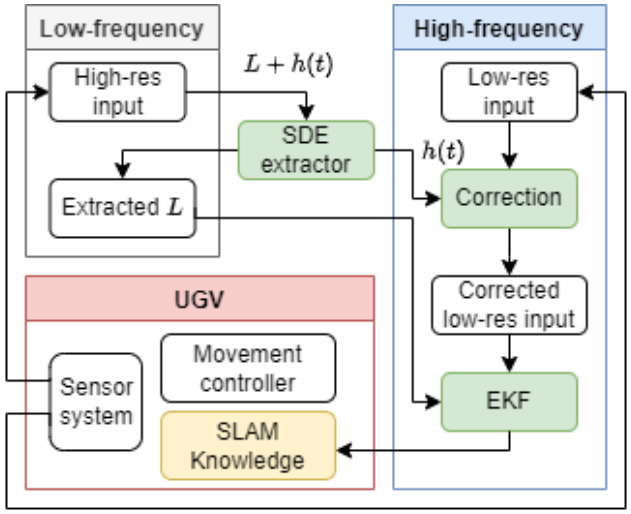


Figure 4: General overview of the proposed system

As we observed that LFS are frequency components, frequency decomposition methods will be best suit in this context. The next part in this section will perform review on 2 most common frequency analysis methods: Fourier Transform and Wavelet Transform, and evaluate their feasibility with respect to our research problem.

4.1 Fourier Transform

Fourier transform works by representing a signal as the summation of multiple smaller sinusoids [21]. It has been one of the most common methods for frequency analysis up to date, and have had many applications in image processing or wave analysis in general. However, Fourier Transform has quite a few limitations. First, it assumes that signals are periodic, which may not be true for abrupt changes in a signal, such as non-patterned images. Second, with the representation as a linear combination of sinusoid at different frequencies, the Fourier representation will only be able to say what frequencies exists in the signal and at what proportion, but does not specify where they may be located in the time-shift domain. Many approaches considered Short Time Fourier Transform (STFT) [22, 23], which have modified Fourier Transform to operate separately on short segments of a signal, giving it more spatial information in the representation. However, STFT will only work on a case-by-case basis, which means it will need to continuously change the division of segments when a signal changes [23], which will add the extra cost of checking for abrupt changes within the signal for every analysis. Therefore, a more dynamic method is preferred, in order to efficiently capture the spatial information of different signals.

4.2 Wavelet Transform

Wavelet transform works by performing convolution on a given signal with a pre-defined basis function. Like Fourier transform, this operation will also return multiple coefficients. But while the Fourier coefficients only represent the magnitude of each frequency, each coefficient in wavelet transform will represent the time and the scale, specifically, the magnitude of the mean change within an area around the specified time and scale [24]. Therefore, this method is suitable for our research problem. The problem now lies in how to choose the basis function for convolution.

Picking Wavelet

There are many wavelet basis functions existing in literature. The most common categorization method at present is to group the wavelets into families. Each family of wavelets contains a mother wavelet that determine the support length - the number of consecutive non-zero coefficients in a sequence of coefficients, and members within the family differ by the number of vanishing moments - the number of order where coefficients are 0, representing complexity of signal and sparseness of coefficient distribution [25]. We will investigate first on the characteristics of the mother wavelet in each family, and then look at number of vanish points in a wavelet group.

Mother Wavelet Mother wavelet is the wavelet form that defines the whole wavelet family's characteristics. It determines the spanning of non-zero coefficients, and determine what scale and frequency each coefficient can cover. We will inspect few of the most common wavelets next. Haar is the most simple wavelet, which is computationally efficient [24], but also capture the least amount of details. Lower support length wavelets, such as Haar, Daubechies and Symlet, are better for detecting dense features in a signal [23], which can be suitable for our SLAM problem, when obstacles on the way needs to be detected. These wavelets are also orthogonal - using the same procedures for decomposition and reconstruction, which preserves the average pixel values. If these processes use different transformations, they are biorthogonal wavelets, which does not preserve the mean pixel value, but have a linear phase property that can perform interpolation well for detail reconstruction and minimizes distortion [23]. Each wavelet prioritises a different strength, and in our case, it might be useful to start with a low-support, orthogonal wavelet such

as Daubechies or Symlet for simplification and all-rounded purpose satisfaction, and change to biorthogonal wavelets if detail reconstruction becomes more essential.

Vanishing points Vanishing point is a hyperparameter that can be selected by hand after having chosen the mother wavelet. It represents the 'resolution' of the coefficients that the transformation provides, and the larger the value, the more complex signals that wavelet transformation can represent [26]. However, this also means that each type of signal will suit with a different value for the optimal number of vanishing points. Therefore, the hyperparameter tuning process will always need to be done for every new environment, in accordance to No Free Lunch Theorem. Therefore, choosing the number of vanishing points can be considered on a case-by-case basis later in implementation stages, rather than in ideation, which is the current main focus of our problem.

5 Wavelet Experiments

In this section, experiments will be carried out to confirm if wavelet is really suitable for our research problem, by trying 2 basic operations that is commonly used with wavelet, namely image denoising and image compression.

5.1 Setup

Figure 5: Sample images from DARPA dataset [27]

Data The wavelet processes will be done on 2 datasets: the Driving Stereo Dataset (DSD) [20] images, a set of images that contain stereo image pairs taken on the street, and DARPA Subterranean Challenge Dataset (DCD) [27], which consists of images taken inside a tunnel. Images from DCD are chosen due to the fact the tunnel environments where these images are taken closely resembles a mining environment, making the experiment results more likely to resemble the outcomes if it were applied to the mines. DSD otherwise, while not reflecting any similarity to a mining environment, may still be able to use for assessing correctness of SDE extractions, as they display image pairs that are taken only a short distance away from each other, resembling an UGV taking consecutive pictures while moving. Examples of instances from the 2 datasets are shown in Figure 3 and 5.

Processor The operation will be done on the CPU of the virtual machine of Google Colab, which has 13 gigabytes of memory, similar to that of a standard Intel NUC 11 Pro. This should be able to reflect how operable the processes will be.

Evaluation Metrics As our main purpose is to assess computational feasibility, there will be 2 measurements of interest from the experiment that relates to online learning problem: the memory consumption for processing, and running time for operation speed. The lower these values are, the better performance of the corresponding method, proving it to be more suitable for our problem. To evaluate and compare different wavelet families, we will also use Mean Squared Error (MSE) for correctness check, and inspect by eye the results of image compression and image denoising, to see what effect can different wavelets bring to the image.

Steps For each image, there are 3 steps in common wavelet operation. First, the image will be decomposed to wavelet using discrete wavelet transform. Second, the main operation is done in the wavelet domain, where modifications or value replacements are done within the coefficients. Finally, after all the modification process is completed, reconstruction is done, where the resulting coefficients will be inversely transformed back to the resulting final image.

The experiment will attempt to record the runtime and memory consumption of all 3 steps, under different image sizes or wavelets. And for different wavelet families, the denoising and compression results will also be shown across different wavelets to demonstrate the feasibility in each operation. Tuning the number of vanishing points is not the main concern for this problem, but the wavelets with 1 vanishing point is just equivalent to Haar wavelet [23, 24], so the wavelets that have this variable as parameter will have the value set to second lowest value possible for orthogonal wavelets, and 1.3 for biorthogonal wavelets - the lowest complexity resolutions available, for computation simplicity.

The first and last step will be considered initially, as they are fixed steps and is expected to not too varied across different images with the multi-thread

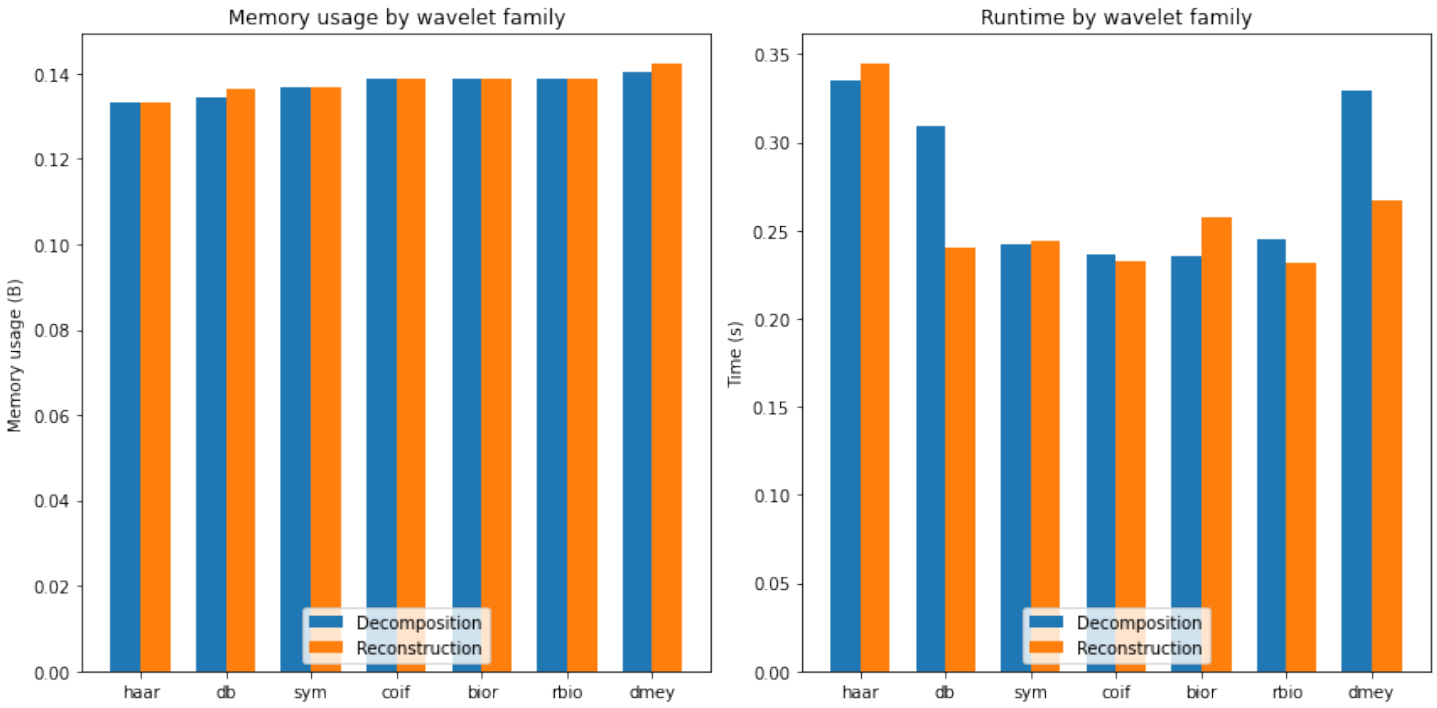


Figure 6: Progressing time and memory consumption of wavelet processes across different families

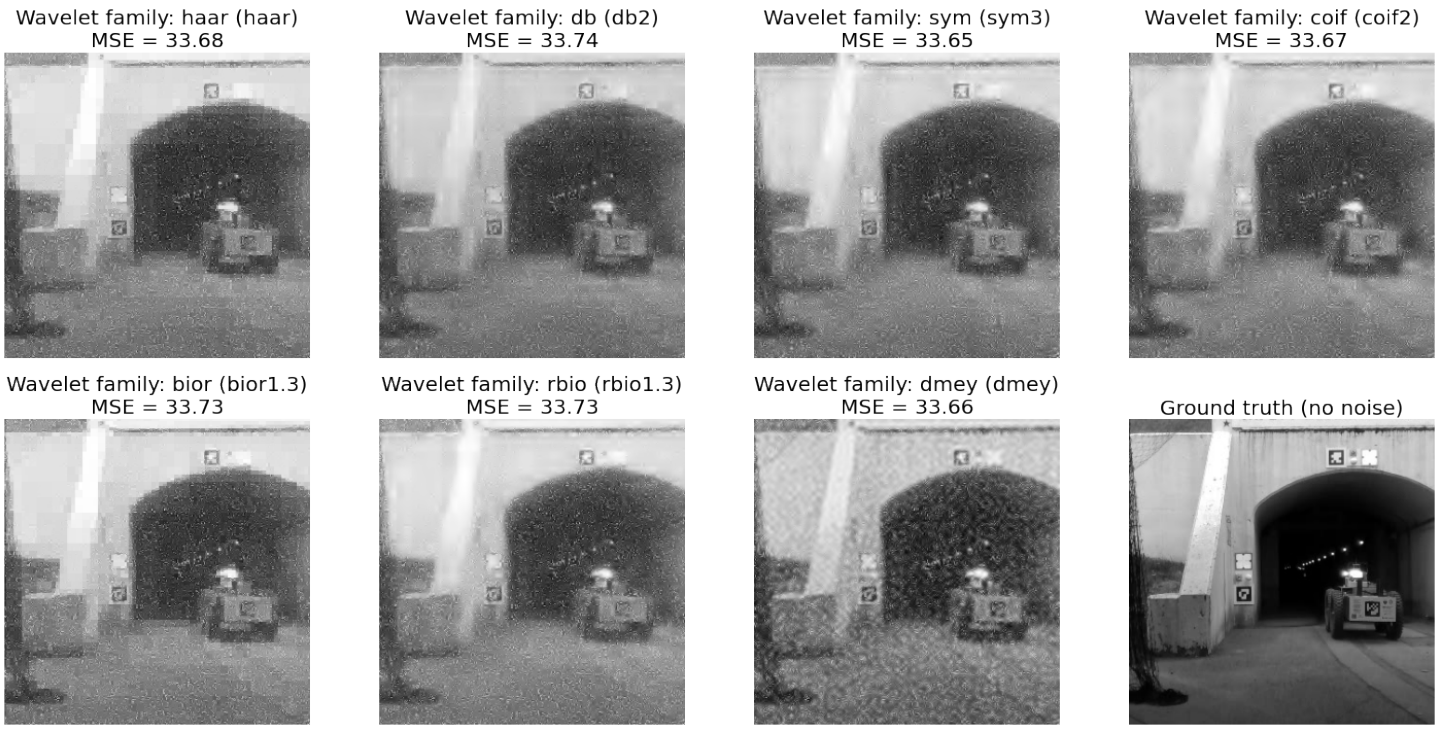


Figure 7: Image denoising using different families of wavelets

code optimization from existing Python libraries. The following inspection will then look into the second step, and see how influential it can be, by seeing how affected the operation complexity will be if the second step complexity rises.

Operations It is also important to see how the image processing works, in order to get an estimation of the expected complexity.

Image denoising works by thresholding, which sets all coefficients with absolute value smaller than a threshold to 0 [23]. This is expected to be performed at constant time for each image, as thresholding an array is $O(1)$ constant.

Image compressing also perform thresholding like denoising on small coefficients, but rather than hard-assigning a value, it cuts down a specified number of smallest coefficients and retain the rest, corresponding to the most significant features [24]. This requires an extra step of iterating through the coefficients to find the top values for setting boundary, which makes the complexity being linear $O(n)$.

5.2 Wavelet Family Comparison

Computation Usage From the results in Figure 6, there is not much difference of runtime or memory usage between wavelet families. In fact, Haar wavelet, despite being the most computationally efficient, was the slowest in terms of runtime, hinting that differences are more likely caused by computation overheading, which may have taken most of the processing time. This proves that when choosing a wavelet, the computation usage do not need to be considered.

Task Feasibility Figure 7 confirms that orthogonal wavelets on first row like db2 (Dabeuchies) and sym3 (Symlet) have great capability in connecting features and remove noises, most of which have lowest MSE for denoising. Haar wavelet has one of the lowest denoising MSE, but visualisation shows that grid-like shapes are still visible in the picture, proving that Haar wavelet can have quite discrete distribution. Figure 8 shows some interesting results on image compression. 2 biorthogonal wavelets (bior and rbio) have significantly small MSE, proving that the linear phase helps to maintain the details well. However, Haar wavelet, despite showing visible abrupt color changes, have the best MSE at nearly no error. This shows that even though orthogonal wavelets discretized the color, they performed it in a way that minimize the redundancy of the details [23], thus preserving the color correctness. Overall, sym3 - Symlet wavelet with 3 vanishing points, seems to be the all-rounder for both tasks, having significantly low MSE comparing to other wavelets performing the same tasks. Therefore, sym3 is feasible to be the initial starting wavelet for the immediate controller in SLAM, given that it is safely well-rounded to satisfy all tasks.

5.3 Computation Feasibility

Correct to the expectation, image denoising has constant runtime complexity, fluctuating around 0.2 seconds of operating time as shown in Figure 9, and image compression has linear runtime complexity, with runtime of less than 0.3 seconds for a decent-sized picture of 1000×320 pixels, as shown in Figure 10. Subtracting these runtimes from the computation overhead, which can be estimated from Figure 6 to be around 0.2 to 0.3 seconds, we will get running time close to 0, which corresponds to high frequency.

The memory usage of the processes, not including the image, is also considered to be negligible and constant across image sizes. This is because the operations do not need to store any extra variables for denoising and compression. However, this is also notable, as we do not know how affected will memory consumption be if too many local variables are specified in the progress. But given that only spatial details are needed to be retrieved, few to none extra variables will be needed.

These results have showed that the denoising and compression processes are feasible on our setup, and has very high potential for more complex functions. However, care still needs to be taken on the wavelet processing step. In order to make the operations computationally possible, the number of local variables should be minimized, and the main algorithm itself should not have too high complexity. Figure 10 have hinted that a linear complexity $O(n)$ operation will be well-fitted for the given computation power and online learning requirement.

6 Conclusion

Overall, a rigid thought process has been carried in detail to tackle the use of SLAM in the mining environment, to efficiently perform tasks where deemed hazardous for human work. We realized that each sensor can effectively complement each other when retrieving inputs for SLAM, hence can be combined to improve the prediction results of online SLAM. Visual SLAM can be a good start when choosing the main SLAM input, as it uses less memory and have more supporting programming libraries. For operating vehicle, we decided that UGV is best suited for

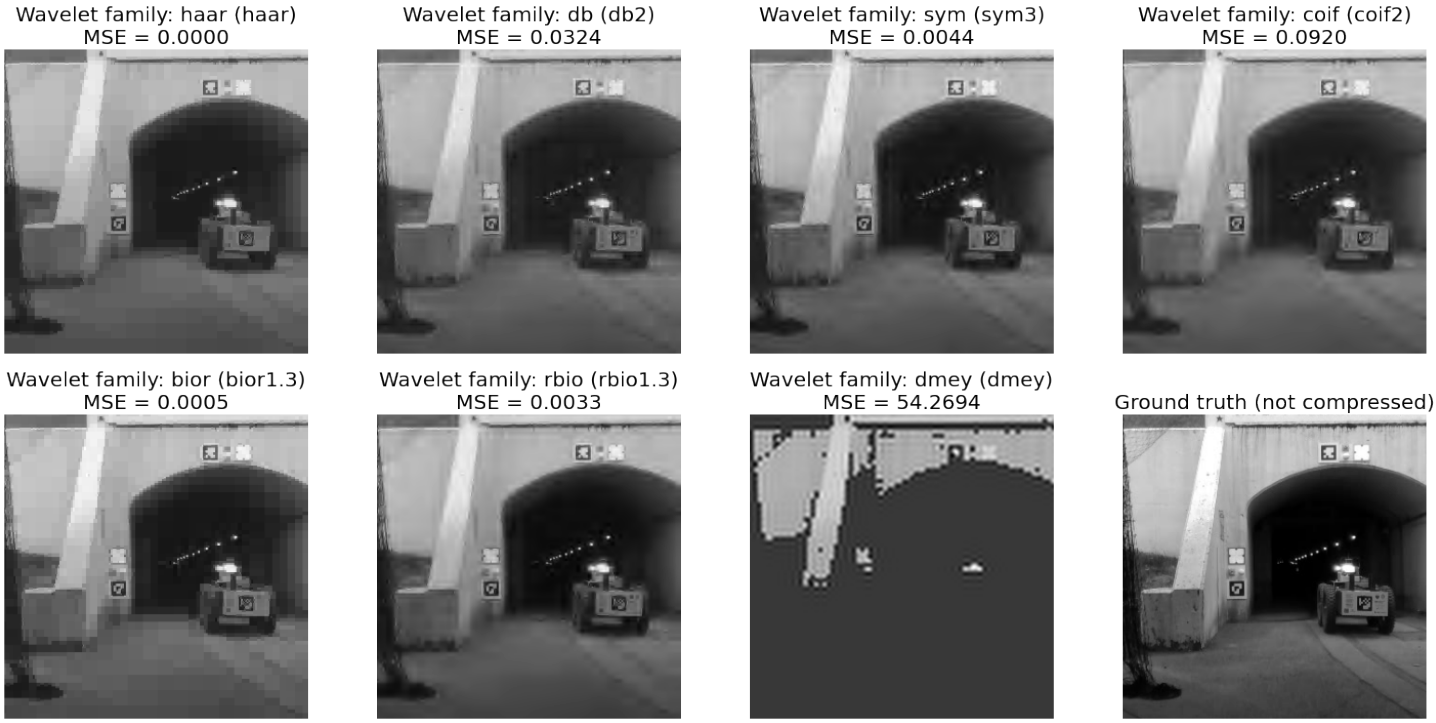


Figure 8: Image compressing using different families of wavelets

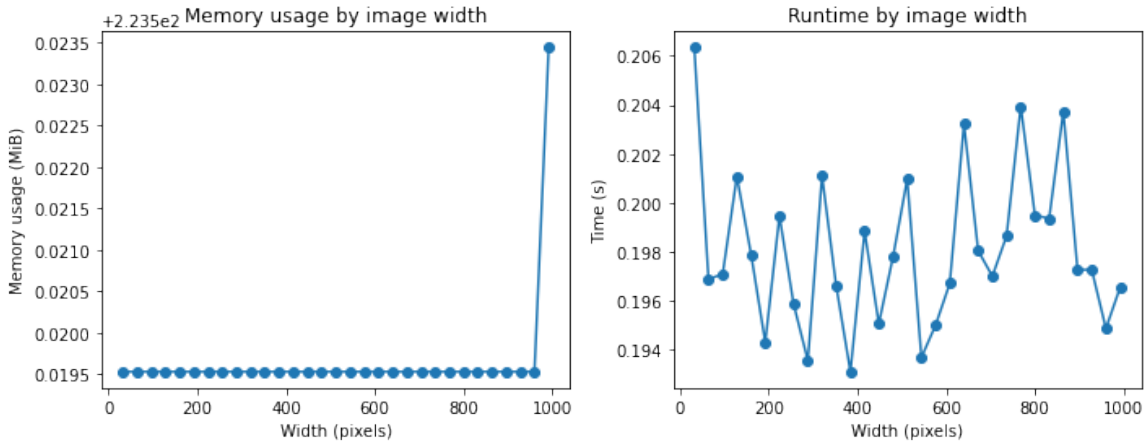


Figure 9: Denoising progress metrics across images with varied sizes

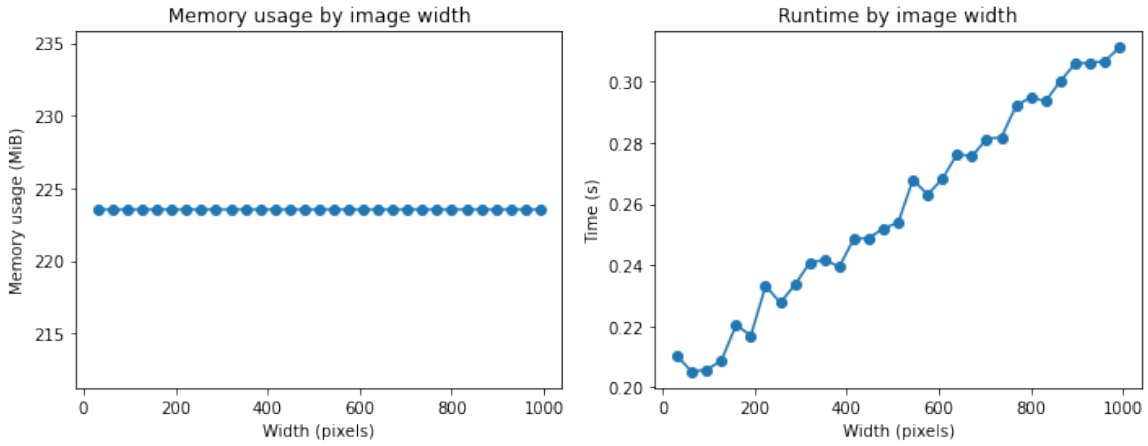


Figure 10: Image compression progress metrics across images with varied sizes

this problem due to its suitability in carrying larger loads and navigation in tight spaces. To maximize the load capacity of the UGV, we also worked out that the computation should not exceed that of an industrial PC's capacity, highlighting the importance of the need to keep computation efficient for online SLAM problem. With that reason, EKF was decided as the algorithm that perform the sensor fusion, given that it consider both sensor and model disturbances, and keeps the computation cost feasible, by only storing the most recent state variables, as well as approximating the controller model to lower order polynomials.

The EKF SLAM model performs by combining high-frequency information to low-frequency information to create an accurate real-time estimation. While common procedure is that the high-frequency signals will provide additional information to low-frequency signals to make a more complete prediction, the former may still subject to noise and uncertainties due to being low-resolution. To counter this, the low-frequency signal, having more information, can also provide correction adjustments to provide a feasibility check on the high-frequency signals. This raises the idea of Semi-Discrete Environment, where low and high frequency signal coexists in an environment, where the former will not change much with respect to time. This can be applied to the inputs of SLAM through various methods, and wavelets - a frequency analysis method is chosen as the method of interest for experiment, as it contains both frequency and temporal information, and have reasonable computation with modern multi-thread code optimizations.

Wavelet transform represents an input signal into different smaller wavelets, which display both frequency information and spatial information of that signal. Through experiments, it is shown that the Symlets family of wavelets are most suitable as initial start for our problem of interest, as it can retain dense features, which can be useful for obstacle detection, and minimizes distribution redundancy. Experiment on real world images from the Driving Stereo and DARPA datasets have shown that wavelet transform has feasible computation time of constant for decomposition or reconstruction, and the internal wavelet algorithm will determine the processing time, which is shown to be efficient at less than 1 second for basic operations like image denoising and image compression, therefore make wavelet transform feasible as the component extraction method in Semi-Discrete Environment.

Now that the algorithm structure is well-defined, further directions to continue improving the SLAM problem from here includes systematically choosing a sensor combination that works best for SLAM in mining. So far, we have determined that the resulting system has one sensor that provides high resolution data, and one to many sensors that provide low resolution data at fast sampling rate. Such feasible examples include a pair of high-resolution camera and a low-resolution camera, or a 3D LIDAR and an IMU, etc. and they can all get tested during in-person experiments to check for compatibility with the system. Another direction is to construct and improve the mechanism that incorporates information from low-frequency signals to correct high-frequency signals and estimate the environment with higher accuracy. This is a type of data stitching, commonly applied for merging images in Computer Vision, and evaluation or development in this field alone can be quite extensive, hence research on this problem can be a topic of its own, where it needs to develop the algorithm to achieve great accuracy, while still keeping the computation feasible within PC capacity.

Bibliography

[1] Neal R. Haddaway, Steven J. Cooke, Pamela Lesser, Biljana Macura, Annika E. Nilsson, Jessica J. Taylor, and Kaisa Raito. Evidence of the impacts of metal mining and the effectiveness of mining mitigation measures on social–ecological systems in arctic and boreal regions: a systematic map protocol. *Environmental Evidence*, 8(1), February 2019.

[2] Frank Mascarich, Shehryar Khattak, Christos Papachristos, and Kostas Alexis. A multi-modal mapping unit for autonomous exploration and mapping of underground tunnels. In *2018 IEEE Aerospace Conference*, pages 1–7, 2018.

[3] Zheng Fang, Shichao Yang, Sezal Jain, Geetesh Dubey, Silvio Maeta, Stephan Roth, Sebastian Scherer, Yu Zhang, and Stephen Nuske. *Robust Autonomous Flight in Constrained and Visually Degraded Environments*, pages 411–425. Springer International Publishing, Cham, 2016.

[4] Shehryar Khattak, Christos Papachristos, and Kostas Alexis. Vision-depth landmarks and inertial fusion for navigation in degraded visual environments. In *ISVC*, 2018.

[5] Tam Willy Nguyen, Laurent Catoire, and Emanuele Garone. Control of a quadrotor and a ground vehicle manipulating an object. *Automatica*, 105:384–390, July 2019.

[6] Jie Chen, Xing Zhang, Bin Xin, and Hao Fang. Coordination between unmanned aerial and ground vehicles: A taxonomy and optimization perspective. *IEEE Transactions on Cybernetics*, 46(4):959–972, 2016.

[7] Adam Stager, Herbert G. Tanner, and Erin Sparks. Design and construction of unmanned ground vehicles for sub-canopy plant phenotyping. In *Methods in Molecular Biology*, pages 191–211. Springer US, 2022.

[8] B.A. Draper, G. Kutlu, E.M. Riseman, and A.R. Hanson. ISR3: communication and data storage for an unmanned ground vehicle. In *Proceedings of 12th International Conference on Pattern Recognition*. IEEE Comput. Soc. Press.

[9] Tim Mueller-Sim, Merritt Jenkins, Justin Abel, and George Kantor. The robotanist: A ground-based agricultural robot for high-throughput crop phenotyping. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2017.

[10] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, Jose Neira, Ian Reid, and John Leonard. Simultaneous localization and mapping: Present, future, and the robust-perception age. *IEEE Transactions on Robotics*, 32, 06 2016.

[11] Yi An, Jin Shi, Dongbing Gu, and Qiang Liu. Visual-LiDAR SLAM based on unsupervised multi-channel deep neural networks. *Cognitive Computation*, 14(4):1496–1508, April 2022.

[12] Michael Milford, Eleonora Vig, Walter J. Scheirer, and David Cox. Vision-based slam in changing outdoor environments. 2014.

[13] Chen Fu, Christoph Mertz, and John M. Dolan. Lidar and monocular camera fusion: On-road depth completion for autonomous driving. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 273–278, 2019.

[14] Mikhail Sizintsev, Abhinav Rajvanshi, Han-Pang Chiu, Kevin Kaighn, Supun Samarasekera, and David P. Snyder. Multi-sensor fusion for motion estimation in visually-degraded environments. In *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 7–14, 2019.

[15] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. Visual SLAM algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1), June 2017.

[16] Ankith Manjunath, Ying Liu, Bernardo Henriques, and Armin Engstle. Radar based object detection and tracking for autonomous driving. In *2018 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, pages 1–4, 2018.

[17] Ziyang Hong, Yvan Petillot, and Sen Wang. Radarslam: Radar based large-scale slam in all weathers. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5164–5170, 2020.

[18] Sebastian Thrun and Michael Montemerlo. The graph SLAM algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, 25(5-6):403–429, May 2006.

[19] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[20] Guorun Yang, Xiao Song, Chaoqin Huang, Zhidong Deng, Jianping Shi, and Bolei Zhou. Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[21] David H. Bailey and Paul N. Swarztrauber. A fast method for the numerical evaluation of continuous fourier and laplace transforms. *SIAM Journal on Scientific Computing*, 15(5):1105–1110, September 1994.

[22] Bashar Rajoub. Characterization of biomedical signals: Feature engineering and extraction. In *Biomedical Signal Processing and Artificial Intelligence in Healthcare*, pages 29–50. Elsevier, 2020.

[23] Ingrid Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, January 1992.

[24] Cristina Stolojescu, Ion Railean, Sorin Moga, and Alexandru Isar. Comparison of wavelet families with application to WiMAX traffic forecasting. In *2010 12th International Conference on Optimization of Electrical and Electronic Equipment*. IEEE, May 2010.

[25] Jonas Gomes and Luiz Velho. *From Fourier analysis to wavelets*. IMPA Monographs. Springer International Publishing, Cham, Switzerland, 1 edition, September 2015.

[26] Ziran Peng and Guojun Wang. Study on optimal selection of wavelet vanishing moments for ECG denoising. *Scientific Reports*, 7(1), July 2017.

[27] Chen Wang, Wenshan Wang, Yuheng Qiu, Yafei Hu, and Sebastian Scherer. Visual memorability for robotic interestingness via unsupervised online learning. In *European Conference on Computer Vision (ECCV)*, 2020.