Micro Movement Tracking in Mixed Reality

Yifan Meng
University of Massachusetts Amherst
Department of Electrical and Computer Engineering
Amherst, United States of America
yifanmeng@umass.edu

Yanan Zhang
University of Massachusetts Amherst
Department of Electrical and Computer Engineering
Amherst, United States of America
yananzhang@umass.edu

Shaoyu Dai
University of Massachusetts Amherst
Department of Electrical and Computer Engineering
Amherst, United States of America
shaoyudai@umass.edu

Linghao Meng
University of Massachusetts Amherst
Department of Electrical and Computer Engineering
Amherst, United States of America
linghaomeng@umass.edu

***Abstract*—Mixed reality (MR) is a technology that combines real and virtual worlds to create immersive user experiences. Accurate tracking of hand movements is crucial for MR applications in the context of mixed reality, such as virtual reality gaming and augmented reality interfaces. Pose estimation is the process of determining the position and orientation of an object in space. However, tracking small, micro movements of the hand can be challenging due to the limitations of current tracking technologies. To address this issue, we propose a method for estimating the pose of micro hand movements using selective sensor fusion. Our approach combines information from multiple sensors, such as inertial measurement units and cameras, to accurately track the pose of the hand even when it is making small, subtle movements. We evaluate our approach using a dataset of real-world hand movements, and demonstrate its effectiveness in tracking micro movements with high accuracy. Our method for pose estimation of micro hand movements in mixed reality using selective sensor fusion shows promising results for tracking small movements with high accuracy. This technique has the potential to improve the user experience of MR applications by enabling more natural and intuitive interactions.**

Keywords—selective sensor fusion, mixed reality, micro movement, visual inertial odometry

1. Introduction

Mixed reality is a mixture of reality and virtual reality technology. It combines the real world with the virtual world to create new environments and visualizations. It requires real-time interaction between physical entities and digital objects, as well as tracking users and real world environments. Robots, smart phones, unmanned aerial vehicles (UAV) and many other fields are mixed reality applications [1]. In these applications, the tracking of users or environments is reflected. For example, an autonomous vehicle is equipped with GPS, imu, camera, laser radar and other sensors to track the user's whereabouts and road environment [2]. Multimode sensors are used for tracking users and spatial mapping of the environment. By integrating observations from different sensors, these applications can sense the environment and estimate system status, such as location and direction [1]. At present, the deep learning method of multi-mode odometer estimation and location has contributed to position tracking, but the problem of robust selective sensor fusion has not been solved. Robust selective sensor fusion is a necessary consideration in dealing with noisy or incomplete sensor observations in the real world. Moreover, the current depth and mileage measurement model lacks interpretability [2]. In addition, at present, most existing technologies support macro motion. However, surgery and other mixed reality applications also require micro tracking. We need to verify that these models also perform well in micro motion. Therefore, we chose to use the visual inert odometer (VIO) model and integrate the specific fusion strategy of SelectFusion to further explore and experiment with micro motion.

Visual inertial odometry (VIO) is widely used in the robot field because it can provide robust and accurate attitude information and the cost of camera and inertial sensor is relatively low [9],[10]. Traditional VIO methods usually follow a standard pipeline [1]. The pipeline process is to collect data monitored by different sensors, then detect and extract features, and then fine tune through sensor fusion strategies. Then input the results to the temporal model, and finally conduct pose estimation. However, these models rely on handmade features and fuse information based on filtering [3] or nonlinear optimization. Using all features before fusion will lead to unreliable state estimation. Because incorrect feature selection and extraction will establish a biased model, resulting in incorrect results.

Depth neural networks (DNNs) are also applied to visual inertial odometry [4], which extract advanced features representing self motion, thus improving the accuracy and robustness of the model as a whole. But it does not explicitly simulate the sources of data degradation in the real world. Therefore, if the input data is damaged or lost, it cannot guarantee the accuracy and security of VIO. Data loss, bad data association and data degradation can be caused by changes in lighting, texture free regions and motion blur. Data degradation includes camera occlusion or operation under low light conditions [5], excessive noise or drift in inertial sensor [6], time synchronization or spatial dislocation between two streams [7]. When possible sensor errors are not considered, all features will be directly input to the next module [8]. The accuracy of the system cannot be guaranteed, and the security of the system is directly affected by the accuracy of the input data. Therefore, the robustness of the system is not good.

Therefore, we use a select fusion framework to model feature selection for robust sensor fusion. The selection process is conditional on the reliability of measurement and

the dynamics of self movement and environment [1]. Select fusion has two optional feature weighting strategies: soft fusion, which is implemented in a deterministic way; Hard fusion, introducing random noise and intuitively learning to maintain the most relevant feature representation, while discarding useless or misleading information. Both architectures are trained end-to-end. By explicitly modeling the selection process, we can demonstrate the strong correlation between selected features and environment/measurement dynamics by visualizing the sensor fusion mask [1].

Select fusion is an end-to-end selective sensor fusion module, which can be applied to a variety of sensor modes, such as monocular image and inertial measurement, depth image and laser radar point cloud. There are different types of sensor fusions that are now widely used. Based on their different characteristics, they are used in specific areas based on that:

Visual-inertial odometry(VIO): VIO is a type of sensor fusion that combines data from an inertial measurement unit (IMU) and a camera to estimate the pose of an object or robot in real-time. The IMU provides measurements of linear and angular acceleration, while the camera provides visual information about the environment. By combining these two types of data, it is possible to estimate the pose of the object or robot even when the camera is not able to see any features in the environment.

Hierarchical sensor fusion: Hierarchical sensor fusion involves combining data from multiple sensors at different levels of a hierarchy. For example, a system might combine data from high-level sensors (such as cameras) and low-level sensors (such as inertial measurement units) in order to estimate the pose of an object or robot. By combining data from sensors at different levels of the hierarchy, it is possible to take advantage of the strengths of each type of sensor and improve the overall accuracy and robustness of the sensor fusion system.

Hybrid sensor fusion: Hybrid sensor fusion involves combining data from multiple sensors of different types (e.g., cameras, LIDAR, and inertial measurement units). By combining data from multiple sensors, it is possible to improve the accuracy and robustness of pose estimation in dynamic environments where a single sensor may not provide sufficient information on its own. For example, a system might use LIDAR data to estimate the pose of an object or robot when the camera is unable to see any features in the environment, and then switch to using camera data when features become visible again.

Adaptive sensor fusion: Adaptive sensor fusion involves adapting the combination of sensors based on the characteristics of the environment and the task at hand. For example, a system might switch between using LIDAR and camera data depending on the visibility of features in the environment. By adapting the combination of sensors based on the current conditions, it is possible to improve the accuracy and efficiency of the sensor fusion system.

Our contribution is to use the fusion strategy of hard fusion and the framework of VIO to track macro motion and micro motion in HoloSet data, calculate the loss, and compare the difference between macro motion and micro motion.

2. Background

2.1 Visual Inertial Odometry

The architecture of VIO is shown in Figure 1. The modularization of VIO includes visual sensor and inertial sensor, visual encoder and inertial encoder, feature fusion, time modeling and pose regression. The model used in this paper uses visual sensors and inertial sensors to obtain a series of original images, depth images and IMU measurements, uses visual encoders and inertial encoders for feature coding, uses a hard fusion method for sensor feature fusion, and generates the corresponding pose transformation. The IMU provides measurements of linear and angular acceleration, while the camera provides visual information about the environment. By combining these two types of data, it is possible to estimate the pose of the object or robot even when the camera is not able to see any features in the environment.

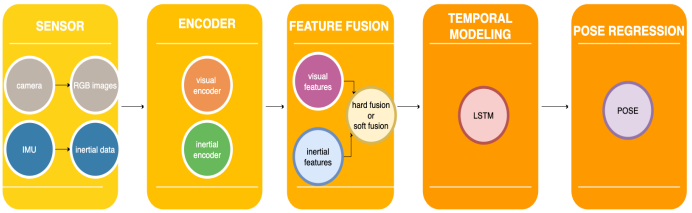

Figure 1 An overview of our neural visual-inertial odometry architecture with proposed selective sensor fusion, consisting of visual and inertial encoders, feature fusion, temporal modeling and pose regression

2.2 LIDAR odometry

LIDAR odometry is a technique that uses a LIDAR sensor to estimate the pose of an object or robot in real-time[19]. A LIDAR sensor works by emitting laser beams and measuring the time it takes for the beams to bounce back after hitting an object. By measuring the distance to nearby objects, it is possible to build up a 3D map of the environment and use this information to estimate the pose of the object or robot.

2.3 Camera relocalization

Camera relocalization is a technique that uses a camera to estimate the pose of an object or robot in an environment that it has previously been in. By matching features in the current camera image to a pre-built map of the environment, it is possible to estimate the pose of the object or robot relative to the map.

3. Selective Sensor Fusion

3.1 Direct Fusion

One method of sensor fusion is direct fusion. Direct fusion is the baseline model for comparison[2]. Direct fusion refers to combining the features of different channels (such as visual channel and inertial channel) directly through Multilayer Perceptrons (MLP) to achieve sensor fusion. Then transfer the selected features to the next module. Figure 2 shows the principle of hard fusion.

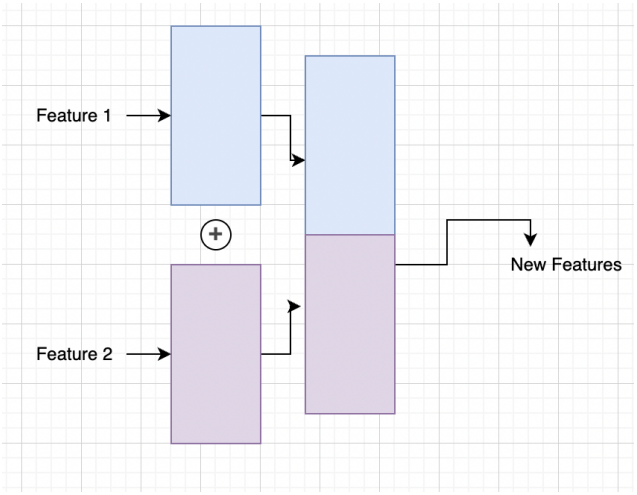

Figure 2 Direct Fusion

3.2 Soft Fusion

Soft fusion is deterministic fusion[3]. The principle of soft fusion is similar to the attention mechanism [17,18], which is achieved by readjusting the feature weight. The more a feature is considered, the higher its weight will be. Soft fusion adjusts the weight of each feature by adjusting different sensor channels, which also enables the feature selection process to be trained jointly with other modules. Taking the VIO framework as an example, the specific process of its soft fusion implementation is as follows: first, visual soft mask and inertial soft mask are introduced, then the inner product of visual features and inertial features and their corresponding soft mask is taken as a new weighting vector, and then these fused features are transferred to time modeling and attitude regression. Among them, the soft mask is parameterized by a neural network, which is a soft selection of extracted feature representation. Figure 3 shows the principle of hard fusion.

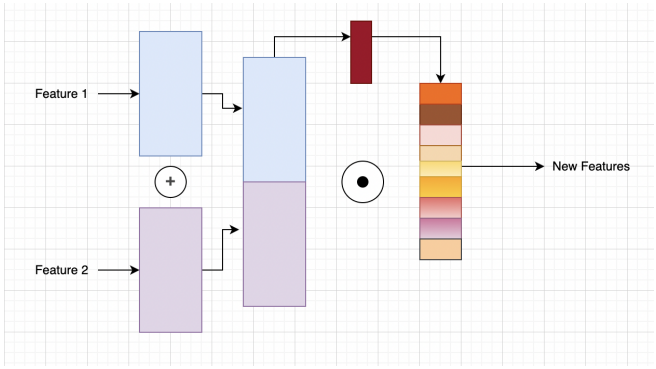

Figure 3 Soft Fusion

3.3 Hard Fusion

Hard sensor fusion refers to the process of combining data from different sensors in a way that is mathematically rigorous and well-defined. This type of fusion typically involves using statistical methods and probabilistic models to combine the data from different sensors and make inferences about the system being monitored. Hard sensor fusion is often used in applications where it is important to have a high level of accuracy and reliability, such as in autonomous vehicles, robotics, and aviation. In these applications, hard sensor fusion can be used to improve the performance of the system by providing a more accurate and reliable estimate of the state of the environment or system being monitored.

Hard sensor fusion can be used in robotics to improve the accuracy and reliability of movement tracking for tasks such as navigation, localization [20], and object tracking. In these applications, hard sensor fusion can be used to combine data from a variety of sensors, such as cameras, lidar, and IMUs, to create a more accurate estimate of the robot's position, orientation, and velocity. This can enable the robot to more accurately navigate its environment and perform tasks with a higher level of precision. In general, hard sensor fusion can be an effective tool for improving the accuracy and reliability of movement tracking in a variety of applications, by combining data from multiple sensors and using statistical methods and probabilistic models to make more accurate inferences about the movement of the system being monitored.

Instead of re-weighting each feature with a continuous value, hard fusion learns a stochastic function that generates a binary mask that either propagates the feature or blocks it [11]. This mechanism can be viewed as a switcher for each component of the feature map, which is a stochastic layer implemented by a parameterized Bernoulli distribution.

In our case, we employ a more lightweight method – Gumbel-Softmax resampling [12],[13] to infer the stochastic layer to tackle the problem that stochastic layer cannot be trained directly by back-propagation and makes hard fusion can be trained in an end-to-end fashion [14], [15]. The figure 4 shows our detailed workflow of our used Gumbel-Softmax resampling based hard fusion.

The distribution of the hard mask is unknown before training the model. Since each element in mask $s^{(i)}$ is a single random variable, we assume it follows a Bernoulli distribution with value 1 with probability p and value 0 with probability q = 1 - p. Since the entire mask **s** consists of n elements, it obeys a binomial distribution with parameter n, which is a discrete probability distribution of the number of successes in a sequence of n independent experiments. Therefore, instead of learning the mask deterministically from the features, the hard masks s1 and s2 representing the binary masks of the two modality features are resampled from a discrete binomial distribution. This discrete distribution is parameterized by α, learned by a deep neural network, and conditioned on features, but with random noise added:

$$s_1 \sim p(s_1|a_1, a_2) = \text{Binomial}(\alpha)$$

$$s_2 \sim p(s_2|a_1, a_2) = \text{Binomial}(\alpha),$$

Each mask s is a n-dimensional binary vector $s_{(i)}$, and each element of hard mask $s^{(i)}$ is a 2-dimensional categorical variable, deciding whether to select the ith feature or not. The total number of features is n. The element $s^{(i)}$ can be viewed as resampling from a Bernoulli distribution [11].

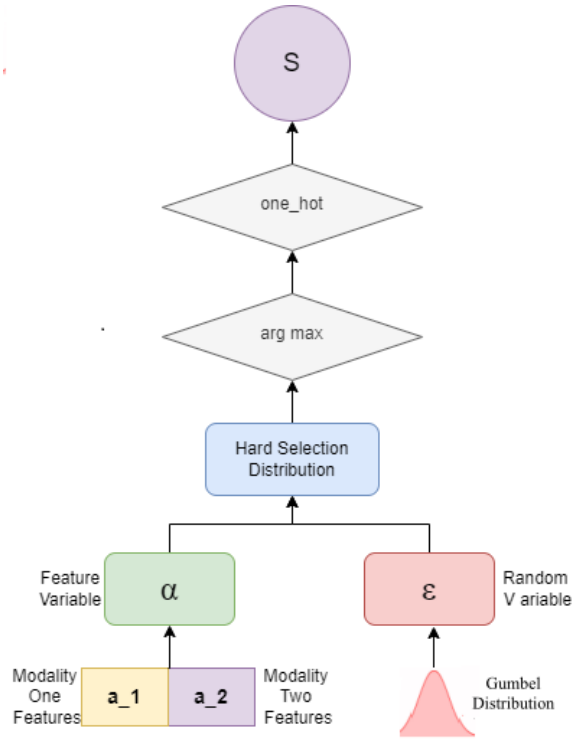

Figure 4 Hard Fusion (Stochastic)

To solve the problem of inferring this discrete distribution in order to generate hard mask **s**. We apply the so-called Gumbel-Softmax trick to convert the non-continuous function into a continuous approximation by using the fact that the distribution of a discrete random variable P (x = k) can be reparameterized by a random variable πk and a Gumbel random variable ε_k via:

$$x = \arg\max_k (\log \pi_k + {}_k).$$

The Gumbel-max trick [16] allows us to efficiently draw a hard mask $s^{(i)}$ from a categorical distribution given the class vector $\pi^{(i)}_k$ and a Gumbel random variable $\varepsilon^{(i)}_k$, and then an one-hot encoding performs 'binarization' of the category:

$$s^{(i)} = \text{one\_hot}(\arg\max_k [\varepsilon^{(i)}_k + \log \pi^{(i)}_k]),$$

4. System Design

Our system design ideas are as follows: First, conduct environment configuration. Then, the data is imported and processed. Next, build a pose regression prediction model, and then pass the processed data into the model for training. Finally, the data of macro motion and micro motion are compared and the results are evaluated. The figure 5 shows the overview of our system design.

4.1 Environment Configuration

Set up the Unity development environment, including installing any necessary packages or libraries for implementing hard selective sensor fusion. This may include the Unity engine itself, as well as any additional tools or frameworks that are needed for the project.

4.2 Import data

Import the HoloSet dataset into Unity. This dataset should include both normal pose data and depth images for both macro and micro movements. The pose data should consist of 3D coordinates for various points on the body, such as the hands, feet, and head. The depth images should be 2D representations of the same movements, captured using a depth camera or similar device.

Split the data into training and testing sets. This will allow you to evaluate the model's performance on data that it has not seen during training, which is important for assessing the model's generalization ability.

4.3 Build Prediction Model

Implement the hard selective sensor fusion model in Unity. This may involve designing and building the model from scratch, or using an existing model and adapting it for your needs. The model should be configured to accept the pose data as input, and should be designed to learn from the data in order to accurately track micro movements.

4.4 Train Data

Train the model on the macro movement data using the normal pose data as input. This will involve feeding the training data into the model and adjusting the model's internal parameters based on the data in order to learn the patterns and relationships in the data.

Evaluate the model's performance on the testing set for macro movements and record the results. This will involve using the model to make predictions on the testing data and comparing those predictions to the actual values in order to measure the model's accuracy.

Train the model on the micro movement data using the normal pose data as input. This process will be similar to the training step for macro movements, but will focus on learning patterns in the data that are specific to micro movements.

Evaluate the model's performance on the testing set for micro movements using the normal pose data as input and record the results. This will involve using the model to make predictions on the testing data and comparing those predictions to the actual values in order to measure the model's accuracy.

Train the model on the micro movement data again, this time using the depth images as input. This step will involve using the same training process as before, but with the depth images as the input data instead of the normal pose data.

Evaluate the model's performance on the testing set for micro movements using the depth images as input and record the results. This will involve using the model to make predictions on the testing data and comparing those

predictions to the actual values in order to measure the model's accuracy.

4.5 Data Comparison And Results Evaluation

Compare the results of the different training and testing scenarios to see how well the hard selective sensor fusion model performs on macro and micro movements using different types of input data. This may involve creating graphs or charts to visualize the results and comparing the performance of the model on different types of data.

Use the results of the comparison to identify any areas where the model could be improved and make any necessary adjustments. This may involve adjusting the model's architecture, training techniques, or other aspects of the model in order to improve its performance.

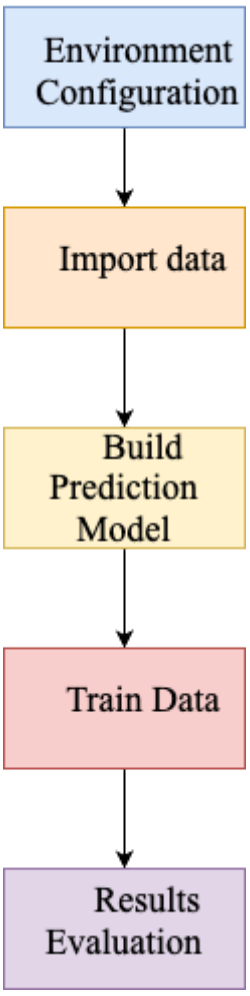

Figure 5 An overview of our system design

4.6 Challenges And Solutions

One of the main challenges of using hard selective sensor fusion models to train depth image data for micro movement tracking is the complexity of the data. Depth images often contain a large amount of information, and it can be difficult for the model to extract the relevant features and patterns from the data. Additionally, the data may be noisy or contain errors, which can further complicate the learning process.

One approach is to preprocess the data before training the model. This may involve filtering out noise, correcting errors, or scaling or normalizing the values to make them more suitable for the model to learn from. Moreover, as with any machine learning model, it is generally beneficial to have more data available for training. Increasing the amount of depth image data available for training may help to improve the model's accuracy and generalization ability.

5. Experiments

We evaluate our proposed approaches on Holoset, a new dataset which is collected by Hololens.

5.1 Experimental Setup

The architecture was implemented with PyTorch. Our experimental platform is the Unity server, a cloud server with NVIDIA RTX 2080 TI GPU.

As we said, our network chooses hard mode. The network was trained with a batch size of 4(According to our GPU memory capacity) using the Adam optimizer, with a learning rate . For a fair comparison, the hyperparameters inside the networks are the same.

We trained our model on Macro movement and Micro movement, then compared their results. For Macro movement, we only trained normal data. For Micro movement, we trained normal data and tried to train Depth images.

5.2 Dataset

HoloSet is a novel pose estimation dataset, collected using Microsoft Hololens 2, which is a state-of-the-art head mounted device for XR. It has an IMU sensor, one RGB camera, four grayscale cameras, and a depth camera[4].

HoloSet is provided by Professor Fatima M. Anwar's team. This is the first dataset collected using a head mounted device. It captures both macro and micro movement. For macro movement, this dataset has data samples of various environments (indoors, outdoors) and scene settings (trails, suburbs, downtown) under user action scenarios (walking, jogging). For micro movement, the dataset contains samples of depth camera images of articulated hands while users play games to exercise fine motor skills and hand-eye coordination.

Compared with the well-known dataset--KITTI Odometry dataset, Holoset provides human movement data and micro movement data. Because Holosens has depth, and gray-scale (4×) cameras, Holoset can provide some gray-scale and Depth images which have more detail.

5.3 Data Processing

For this experiment, what we need are PV images, ground truth data and IMU data. For micro movement, there are additional Depth images. For ground truth, Holosens report human pose trajectory, euler and quaternion. In our

experiment we used pose and euler data. For IMU data, Holosens's IMU consists of an accelerometer, gyroscope, and magnetometer; it reports related data. In our experiment, we used acceleration and gyroscope data.

Because selective sensor fusion is a prediction algorithm for macro scenarios. It is not suitable for grayscale images, we can't train Depth images directly. So we need to make some changes. We were going to modify the model and make it suitable for grayscale images. But it is hard for us, and we can't do it in the short term. So, we decided to convert the Depth image to RGB images. We know it is not the best way, so modifying the code is a future work.

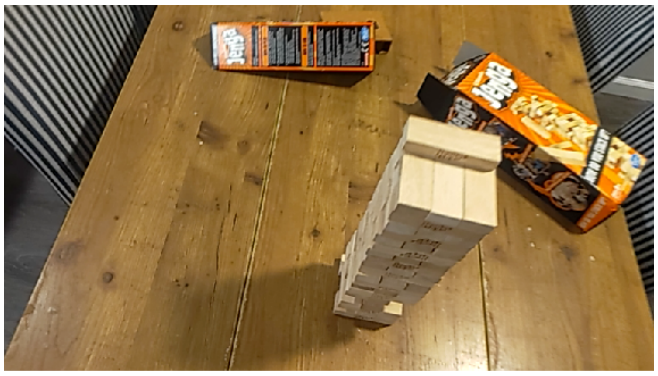

Figure 6: Example of two scenes, campus center(Top) and Jenga(botto*m)*.

5.4 Experimental Procedure

We chose two scenes–campus center and city center from the Macro movement and one scene–Jenga from the Micro movement for our experiment.

We divided the experiment into two parts. First, we trained 75% of Macro movement PV(Photo to Video) images(about 7300 images), and the remaining 25% for testing. Then using the model we trained, tested about 5000 micro movement PV images. Then we trained 75% Micro Movement Depth images(about 80000 images), and the remaining 25% for testing.

In these two phases, we obtained the result data respectively and compared them.

5.5 Results

We compare the predicted value of the experimental results with the actual value to obtain loss data. We got pose loss and euler loss, and use these loss, we calculated a total loss = pose_loss + euler_loss * 100.

Fig. 7 shows the loss data on PV images. The figure shows that for the model trained with macro movement data, when testing 25% of the macro movement data and 100% of the micro movement data, the test loss value of the micro movement is 0.0153, which is much larger than 0.0011 of the macro movement. This result is expected because the model is trained using macro movement data.

Fig. 8 shows the loss data on Depth images. The figure shows that the total loss value of the train is 0.2155, while the total loss value of the test is 0.1074. They are both larger than those of PV data. One possible explanation is that the macro movement data is more accurate and precise than the micro movement data. In this case, the macro movement data would provide more reliable information to the fusion model, resulting in a smaller data loss.Another possible explanation is that the selective sensor fusion model is better at combining macro movement data than micro movement data. In this case, the fusion model would be more effective at mitigating the effects of noise and other sources of error when using macro movement data, leading to a smaller data loss.It's also possible that the macro movement data and micro movement data are used in different ways by the fusion model, which could explain why the data loss is smaller for the macro movement data. For example, the macro movement data may be used to provide more accurate estimates of a particular quantity, while the micro movement data may be used to provide additional information about other aspects of the system. In this case, the macro movement data would contribute more to the overall accuracy of the fusion model, leading to a smaller data loss.

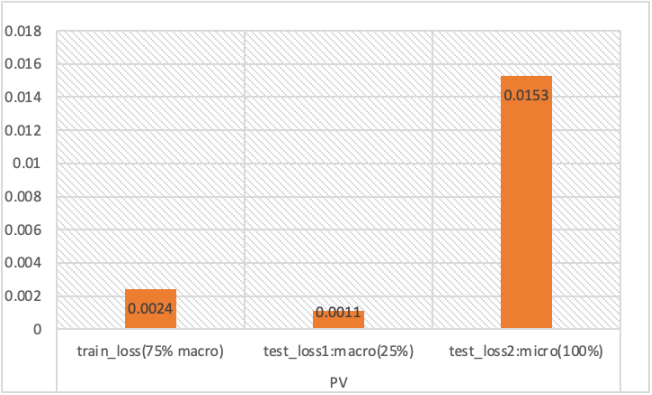

Figure 7: Evaluating total loss on PV images.

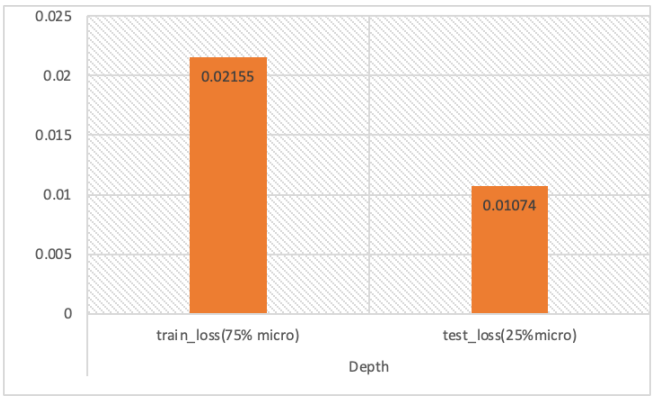

Figure 8: Evaluating total loss on Depth images.

Ultimately, the reason for the difference in data loss between macro movement data and micro movement data would depend on the specific details of the selective sensor fusion model and the data being used. Without more information, it's not possible to provide a more precise explanation.

6. Conclusion and Future Research

Using a model with hard selective sensor fusion for micro movement tracking can be a valuable contribution in certain situations. One of the main benefits of this approach is the ability to combine information from multiple sensors, which can provide a more accurate and comprehensive understanding of the movement being tracked. This can be particularly useful in situations where a single sensor may not be sufficient to accurately capture all of the relevant information. So our main contribution is to solve the problem of the hard fusion model which we used does not support depth image data process, we did data type conversion to convert depth image to normal data type just like the macro movement data we trained.

However, there are also limitations to this approach. One potential limitation is the need for the model to be specifically designed for the task at hand, which can be time-consuming and require specialized expertise. Additionally, the reliance on multiple sensors can also increase the complexity of the system and may require additional hardware and computational resources. Overall, the use of a model with hard selective sensor fusion for micro movement tracking can provide valuable insights and improve accuracy in certain situations, but it is important to carefully consider the limitations and requirements of this approach in order to determine if it is the most suitable solution.

In the future, there are some other directions that can be explored. For improving real-time performance, one of the main challenges in using hard selective sensor fusion for micro movement tracking is the need for fast and accurate tracking in real-time applications. Future research could focus on developing techniques and algorithms to improve the speed and efficiency of the model, allowing it to be used in more demanding and dynamic environments. Incorporating additional data sources: In addition to depth images, there are many other types of data that could potentially be used to improve the performance of hard selective sensor fusion models for micro movement tracking. Future research could focus on incorporating additional data sources, such as inertial measurement units (IMUs), thermal cameras, or other sensors, and analyzing their effects on the model's accuracy and robustness. Last but not least, applying the model to new domains: Hard selective sensor fusion has the potential to be applied to a wide range of applications beyond micro movement tracking, such as robotics, virtual reality, or medical rehabilitation. Future research could focus on exploring the use of the model in these and other domains, and adapting it to meet the specific needs and requirements of those applications.

References

[1] Bruno, F., Barbieri, L. & Muzzupappa, M. A Mixed Reality system for the ergonomic assessment of industrial workstations. *Int J Interact Des Manuf* 14, 805–812 (2020).

[2] Chen, Changhao, et al. "Selective sensor fusion for neural visual-inertial odometry." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

[3] [3] M. Li and A. I. Mourikis. High-precision, Consistent EKF-based Visual-Inertial Odometry. The International Journal of Robotics Research, 32(6):690–711, 2013.

[4] [4] H.W.A.M.N T. Ronald Clark, Sen Wang. Vinet: Visual- inertial odometry as a sequence-to-sequence learning problem. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. AAAI, 2017.

[5] [5] N. Yang, R. Wang, X. Gao, and D. Cremers. Challenges in Monocular Visual Odometry: Photometric Calibration, Motion Bias and Rolling Shutter Effect. IEEE ROBOTICS AND AUTOMATION LETTERS, pages 1–8, 2018.

[6] [6] N. Naser, El-Sheimy; Haiying, Hou; Xiaojii. Analysis and Modeling of Inertial Sensors Using Allan Variance. IEEE Transactions on Instrumentation and Measurement, 57(JANUARY):684–694, 2008.

[7] [7] Y.Ling,L.Bao,Z.Jie,F.Zhu,Z.Li,S.Tang,Y.Liu,W.Liu, and T. Zhang. Modeling Varying Camera-IMU Time Offset in Optimization-Based Visual-Inertial Odometry. In The Eu- ropean Conference on Computer Vision (ECCV), 2018.

[8] [8] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen. VidLoc: A Deep Spatio-Temporal Model for 6-DoF Video- Clip Relocalization. In CVPR, 2017.

[9] [9] E. S. Jones and S. Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. The International Journal of Robotics Research, 30(4):407–430, 2011.

[10] [10] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. On-manifold preintegration for real-time visual–inertial odome- try. IEEE Transactions on Robotics, 33(1):1–21, 2017.

[11] [11] Changhao Chen*, Stefano Rosa, Chris Xiaoxuan Lu, Bing Wang, Niki Trigoni, Andrew Markham: Learning Selective Sensor Fusion for States Estimation. In IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, May, 2022.

[12] [12] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. In Proceedings of the International Conference on Learning Representations, 2017.

[13] [13] C. J. Maddison, A. Mnih, and Y . W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. In Proceedings of the International Conference on Learning Representations, 2017.

[14] [14] A. Mnih and K. Gregor. Neural variational inference and learning in belief networks. In Proceedings of the International Conference on Machine Learning, volume 32, pages 1791–1799.

[15] [15] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning, 8(3-4):229–256, 1992.

[16] [16] C. J. Maddison, D. Tarlow, and T. Minka. A* Sampling. In Advances in Neural Information Processing Systems, pages 1–9, 2014.

[17] [17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. In NIPS, 2017.

[18] [18] C. Hori, T. Hori, T. Y. Lee, Z. Zhang, B. Harsham, J. R. Hershey, T. K. Marks, and K. Sumi. Attention-Based Mul- timodal Fusion for Video Description. Proceedings of the IEEE International Conference on Computer Vision, 2017-October:4203–4212, 2017.

[19] [19] Q. Li, S. Chen, C. Wang, X. Li, C. Wen, M. Cheng, and J. Li. Lo-net: Deep real-time lidar odometry. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8473–8482, 2019.

[20] [20] A. Geiger, P . Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. The International Journal of Robotics Research, 32(11):1231–1237, 2013.

[21] [21] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni. Vinet:

[22] Visual-inertial odometry as a sequence-to-sequence learning problem. In Proceedings of the AAAI Conference on Artificial Intelligence, pages 3995–4001, 2017.

[23] [22] E. J. Shamwell, K. Lindgren, S. Leung, and W. D. Nothwang. Unsu-pervised deep visual-inertial odometry with online error correction for rgb-d imagery. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(10):2478–2493, 2020.