https://doi.org/10.2352/ISSN.2470-1173.2020.6.IRIACV-014

Improving Multimodal Localization Through Self-Supervision

Robert Relyea, Darshan Bhanushali, Karan Manghi, Abhishek Vashist, Clark Hochgraf, Amlan Ganguly, Andres Kwasinski, Michael E. Kuhl, Raymond Ptucha; Rochester Institute of Technology; Rochester, New York/USA

Abstract

Modern warehouses utilize fleets of robots for inventory management. To ensure efficient and safe operation, real-time localization of each agent is essential. Most robots follow metal tracks buried in the floor and use a grid of precisely mounted RFID tags for localization. As robotic agents in warehouses and manufacturing plants become ubiquitous, it would be advantageous to eliminate the need for these metal wires and RFID tags. Not only do they suffer from significant installation costs, the removal of wires would allow agents to travel to any area inside the building. Sensors including cameras and LiDAR have provided meaningful localization information for many different positioning system implementations. Fusing localization features from multiple sensor sources is a challenging task especially when the target localization task's dataset is small. We propose a deep-learning based localization system which fuses features from an omnidirectional camera image and a 3D LiDAR point cloud to create a robust robot positioning model. Although the usage of vision and LiDAR eliminate the need for the precisely installed RFID tags, they do require the collection and annotation of ground truth training data. Deep neural networks thrive on lots of supervised data, and the collection of this data can be time consuming. Using a dataset collected in a warehouse environment, we evaluate the performance of two individual sensor models for localization accuracy. To minimize the need for extensive ground truth data collection, we introduce a self-supervised pre-training regimen to populate the image feature extraction network with meaningful weights before training on the target localization task with limited data. In this research, we demonstrate how our self-supervision improves accuracy and convergence of localization models without the need for additional sample annotation.

Introduction

Autonomy is rapidly growing in many industrial sectors including transportation, mining, and material handling. Accurate localization is integral for effective, efficient, and safe operations in these applications. Sensors such as omnidirectional cameras and 3D LiDARs provide many features but require specialized methods for extracting important information.

For applications where it is not economically feasible to collect and annotate large amounts of localization data, generating a model to automatically extract these features is challenging. Many find success training models on smaller datasets when applying transfer learning. This approach involves initializing a network with weights trained from a different task with a larger dataset. A commonly used dataset in transfer learning for image-based networks is ImageNet[1]. This dataset contains 1.2 million images with 1000 different classification labels.

An alternate approach for transfer learning is self-supervision. Self-supervised learning involves training a network on a task that can be automatically generated from unlabelled data. An advantage of this approach is the ability to pre-train a network using data from the same modality as the target task. In the case of this work, omnidirectional camera images with position labels serve as the dataset for the target localization task while unlabelled omnidirectional images are used for self-supervised pre-training. A self-supervised pre-training scheme can improve the performance of these supervised models, especially when the target task's dataset is limited in size. Unlabelled images are trivial to collect without the need for human annotation.

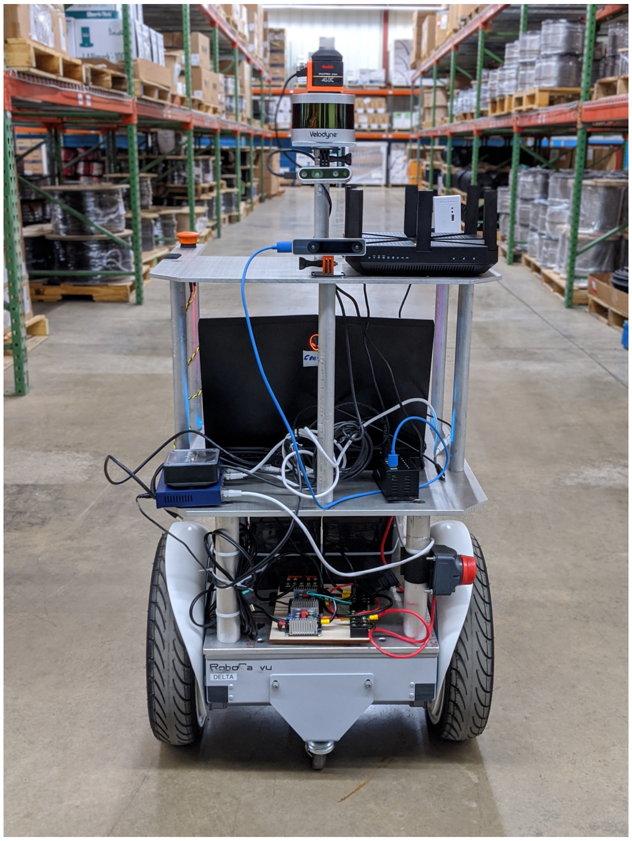


Figure 1. *The autonomous platform used to collect localization sensor samples for model training and evaluation. Several sensors including a Kodak PixPro sp360 4K omnidirectional camera (top) and a Velodyne VLP-16 3D LiDAR (below sp360 4K) provide an abundance of features useful for robot localization.*

A prototype of the robot platform the used for experimentation is shown in Figure 1. The differential-drive RoboSavvy platform houses a laptop used for data collection and navigation. Sensors equipped include a Kodak PixPro sp360 4K omnidirectional camera, a Velodyne VLP-16 3D LiDAR, and wheel encoders.

Related Work

Camera Localization

Many existing approaches utilize simultaneous localization and mapping (SLAM) and visual odometry (VO) techniques to process visio-spatial information and accumulate a representation of the operating environment. Within this representation the autonomous platform can localize itself through its onboard sensor readings. One such VO implementation presented by Forster et al. [2, 3] named semi-direct visual odometry (SVO). This implementation operates on pixel intensity values and boasts lower power consumption, making it ideal for smaller embedded platforms. The original work was extended by Zhang et al. [4] to evaluate the potential benefits and drawbacks that wider field-of-view (FOV) cameras present in this space. It was concluded that a wider FOV contributes to greater tracking performance in indoor environments while outdoor performance may degrade due to the decreased angular resolution of wide FOV images.

LiDAR Localization

SLAM approaches are also common in the LiDAR localization space. These methods can provide accurate position estimates and environment representations from laserscan and pointcloud readings. A LiDAR-based SLAM implementation with low overhead loop-closure was presented by Hess et al. [5] which serves as the backend for Google's popular Cartographer SLAM package. This packages can be used with both 2D laserscans and 3D pointclouds to produce accurate pose estimates and environment representations. The environment representations in SLAM implementations are iteratively updated over time as more information is obtained by the autonomous platform's sensors. SLAM approaches are ideal in situations where the structure of the environment is unknown to the platform and/or when many environment features are subject to change.

Multimodal Systems

Prior works have been presented concerning the fusion of both LiDAR and camera image information to provide a more robust position estimation. Wolcott et al. [6] synthesized several different camera views using dense LiDAR generated maps. An image retrieval scheme was created to compare images from live camera data to those generated from the LiDAR generated maps. A strong match between the live camera frames and a synthesized frame serves as the localization mechanic. A similar approach was presented by Caselitz et al. [7] which also utilizes LiDAR generated maps and a live camera feed. Instead of synthesizing images from the LiDAR map, this implementation produces sparse pointclouds from features detected in the live camera. These pointclouds are then geometrically matched to the original LiDAR map to provide location information.

Self-Supervised Learning

A crucial component for any learned model is weight initialization. When possible, it is desirable to initialize model weights from a network trained on a larger dataset such as ImageNet [1] for image-based architectures. This may not always be an ideal solution if pretrained model weights are not available, as training on a large dataset will consume a significant amount of time. Alternatives to ImageNet initialization for convolutional neural networks (CNNs) have been created through self-supervision. These self-supervised training schemes aim to learn important contextual features to serve as a better starting point for the intended target task.

Image patch localization is a common task used to learn these features as presented by Doersch et al. [8]. In this implementation, a small grid of patches are extracted and randomized from an image. A neural network is trained to determine the original locations of each patch, relative to one another. The resulting model is then used as a starting point for image classification, detection, and segmentation target tasks. This concept works quite well, as the fundamental image understanding tasks and filters to determine relative patch location are similar to the understanding tasks for image classification. Another context-based self-supervised scheme was presented by Pathak et al. [9] which removes regions from and image and trains a network to reconstruct the removed regions from the remaining image information. Both of these implementations achieved greater results on their final target tasks when initializing with self-supervised weights than when initializing with random weights.

Multimodal Sensors

Two types of sensors were utilized for the proposed localization system: an omnidirectional camera and a 3D LiDAR sensor. The wide field of view (FOV) and abundance of visual and spatial features from these two modalities make them ideal sensors for indoor localization. Both of these sensors accumulate readings as the agent navigates through the warehouse testing environment.

Omnidirectional Camera

Omnidirectional cameras take advantage of an extremely wide FOV to produce feature-rich images of the surrounding environment. The Kodak PixPro sp360 4K omnidirectional camera was utilized on our platform for capturing the warehouse testing environment. With a 235^o FOV and a full-frame capture resolution of 1440 by 1440 RGB pixels, the sp360 4K produces a detailed 360^o view of the robot's surroundings. With each frame including a view of the ceiling, floors, and walls, this sensor is ideal for extracting location-specific information as the robot navigates through the environment. An example image captured from the sp360 4K inside the warehouse testing environment is shown in Figure 2.

Velodyne 3D LiDAR

The Velodyne VLP-16 3D LiDAR contains 16 light distance and ranging channels that collect a 360 degree sweep of the surrounding environment. A complete sweep is performed at a varying frequency between 10 to 20 hz. Each sweep generates around 28,000 individual distance readings which are accumulated into a point cloud. Each distance reading has an associated X, Y, and Z coordinate relative to the sensor and a reflection intensity value. A visualization of a point cloud generated by the VLP-16 in the warehouse testing environment is shown in Figure 3. The VLP-16 rests one meter above the ground at the top of the platform shown in Figure 1 to provide a clear view of the operating environment. In addition to localization, this sensor is also used for obstacle detection.

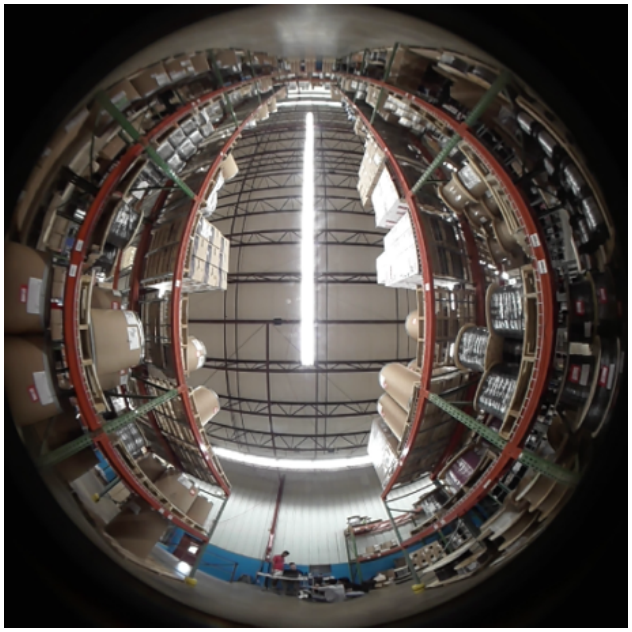


Figure 2. Omnidirectional camera image taken using the Kodak PixPro sp360 inside the warehouse testing environment.

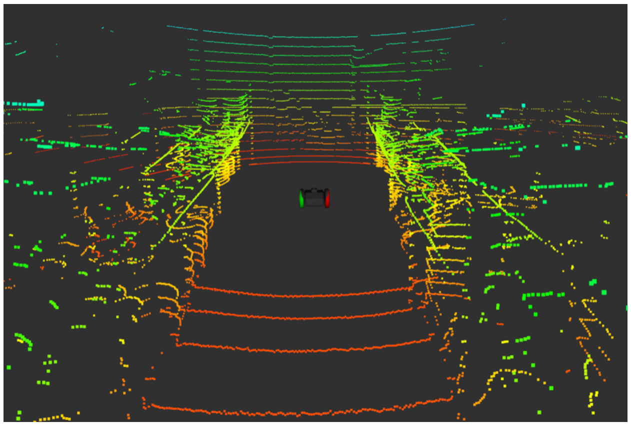


Figure 3. Point cloud visualization of Velodyne VLP-16 3D LiDAR readings inside the warehouse testing environment. The agent is in the same location shown in the omnidirectional image from Figure 2.

Self-Supervision for Omnidirectional Images

A patch-based self-supervised approach inspired by Doersch et al. [8] was designed specifically for omnidirectional cameras in this work. Several radial patches are extracted from an unlabelled omnidirectional camera image along with a center circular patch. Several randomized parameters were created for the radial patches including random starting and ending angle jitter, inner and outer radius jitter, and a random arc length. The center circular patch has a random radius as well as a random center coordinate. An illustration of these patches taken over an omnidirectional image is shown in Figure 4. The number of radial regions extracted can be varied as a training hyperparameter.

The images used for automatically generating patch samples are unlabelled. Since they don't require labels, a large number of samples can be collected by randomly driving the robot around the warehouse environment. Once the center and radial patches are extracted from the unlabelled image, the radial patches are randomly shuffled from their original positions. The resulting image contains the shuffled radial patches and the center patches with white padding around all the regions. A label is generated for each sample consisting of individual one-hot vectors for each patch position which comprise an overall multi-hot vector label for the sample. This vector provides the supervision needed to train the patch position classifier network. Illustrations of the self-supervision data generation pipeline and the transfer learning scheme are shown in Figure 5 with a patch count of four.

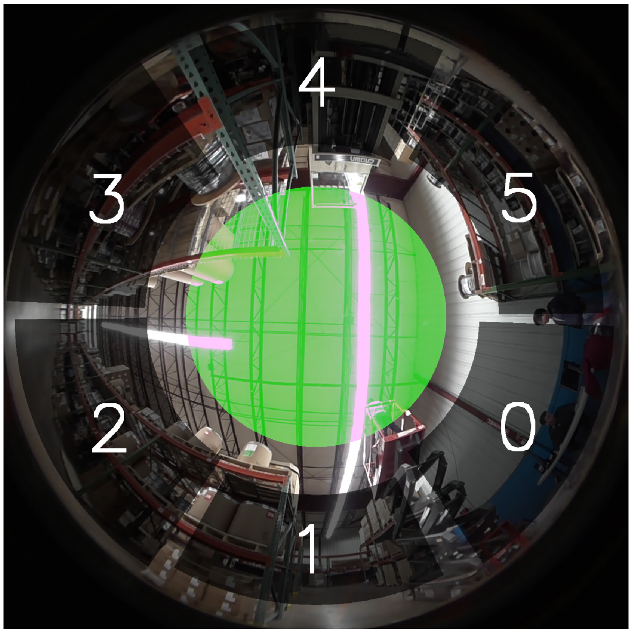


Figure 4. Omnidirectional camera image taken using the Kodak PixPro sp360 inside the warehouse testing environment.

Localization Network Architectures

Due to the differences in data structure between the omnidirectional camera samples and the 3D LiDAR point cloud samples, two independent architectures were instantiated for the individual sensors.

Omnidirectional Camera Localization Model

For omnidirectional camera localization, the ResNet CNN architecture originally presented by He et al. [10] was used. This architecture has been shown to outperform other networks such as VGG16 [11] and AlexNet [12] for image classification. This is largely due to the substantial depth achieved in the ResNet architecture made possible by residual connections. These residual, or skip connections require each layer to only learn the residual from one layer to the next. The skip connections additionally allowed a direct path for backpropagation data, alleviating performance reductions due to vanishing gradients even with very deep models. A similar architecture presented by Gao et al. [13] expands upon the original residual feature concept by concatenating features instead of summing them. In addition to this, more skip connections were created between layers in the DenseNet architecture. We uti-

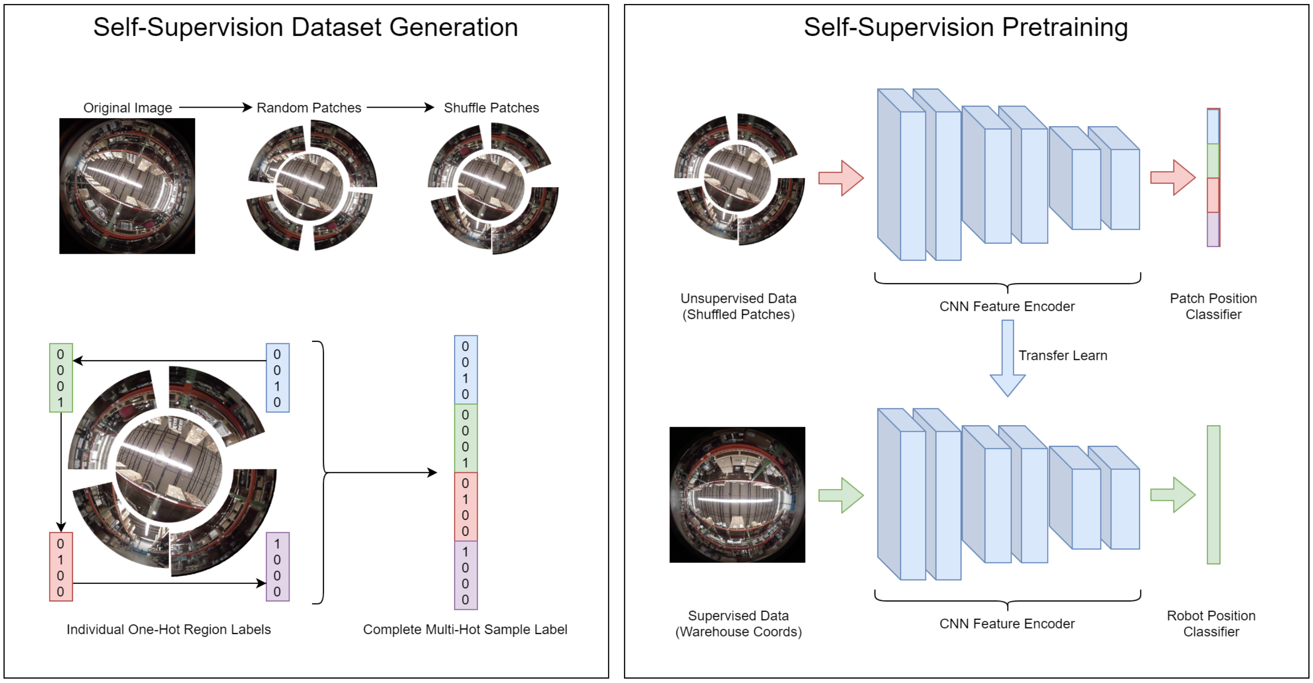

Self-Supervision Dataset Generation
Original Image
Random Patches
Shuffle Patches
0 0 0 1
0 0 1 0
0 1 0 0
1 0 0 0
0 0 1 0 0 0 0 1 0 1 0 0 1 0 0 0
Individual One-Hot Region Labels
Complete Multi-Hot Sample Label
Self-Supervision Pretraining
Unsupervised Data (Shuffled Patches)
CNN Feature Encoder
Patch Position Classifier
Transfer Learn
Supervised Data (Warehouse Coords)
CNN Feature Encoder
Robot Position Classifier

Figure 5. Self-supervision dataset generation illustration. An unlabeled omnidirectional image is split into a center patch and several random radial patches. These patches are shuffled from their original locations. The original locations serve as an automatically generated class label utilized during the self-supervised pre-training routine. The self-supervision pretraining pipeline serves as an initial pretraining step on unlabelled omnidirectional camera data with a patch position classifier output. Once this model has concluded training, the CNN weights are transferred to the target robot localization task network as a weight initialization starting point for training.

lize the lightweight ResNet18 architecture for the omnidirectional camera location classification network. In addition to classification for robot position, a regression model was also created for omnidirectional camera images. The classification network had one output node for each localization position, while the regression network had only two outputs corresponding to the two X and Y coordinates for each sample label.

3D LiDAR Localization Model

Processing of 3D point cloud data can be challenging due to the inherent irregular structure, sparse nature of points, and highly variable density and distribution over the 3D space. Qi et al. [14] proposed a deep neural network called PointNet which learns point-wise features directly from raw point clouds without the need for any kind of lossy data quantization. The network comprises of a set of shared multi-layer perceptrons (MLPs) which maps each of the n points in the point cloud to a higher dimensional space. These higher dimensional features are then aggregated together using a symmetrical max pooling function. The symmetric max pooling operation preserves the permutation invariance of the points in the point cloud. PointNet++ [15] is an improvement over the original PointNet. PointNet++ recursively applies PointNet on the input point cloud in hierarchical fashion. Each recursive call is followed by a clustering stage where the obtained features aggregate together using a max pooling operation.

PointNet++ additionally provides a point-wise learned vector representation at various contextual scales along with a global point features. Pointwise features allow more generalizability to the complex scene understanding. Both PointNet and PointNet++ are end-to-end trainable networks and can be used as fixed or variable encoders for point cloud data. The networks provide a learned spatial encoding of each point called as point-wise feature vector as well as an aggregated global point cloud feature vector. These features can be used for classification, regression or any other task. For 3D LiDAR-based robot localization, a PointNet++ network with classification output was initialized with enough classes to cover all of the warehouse testing environment positions.

Data Collection and Experimental Setup

The autonomous platform utilized for data collection and experimentation is shown in Figure 1. A differential-drive RoboSavvy Self-balancing platform is coupled with two casters for passive stability. A laptop equipped with an Intel i7 8750H, 32GB of DDR4 memory, and an NVIDIA GTX 1060 controls the platform and receives sensor information. Ubuntu 16.04 with the Robot Operating System (Kinetic) provides interfaces for the sensors and robot base. The Velodyne VLP-16 3D LiDAR and the Kodak PixPro sp360 4K camera are mounted at the top of the platform for the greatest visibility.

The environment for testing consists of a single warehouse aisle in a six foot by 73 foot region. Discrete data collection points are spaced out every foot which results in 438 independent locations in the testing region as shown in Figure 6. Ground truth localization samples from each sensor are gathered in a semi-autonomous procedure. An agent is manually placed at one of the six X coordinates at the start of the aisle. After being aligned with landmarks placed on the warehouse walls using lasers, the robot will slowly accelerate and travel 12 feet forward. While moving, the robot uses wheel encoders to determine the distance traveled from the starting point. Every foot traveled the robot will record all of the sensor readings and stamp them with the current location. After traveling 12 feet, the robot is realigned with the wall landmarks if necessary. The process is repeated until all of the 438 individual coordinates have been visited. Full sweeps of the testing region are performed over many days to ensure the warehouse shelf contents are varied from sample to sample.

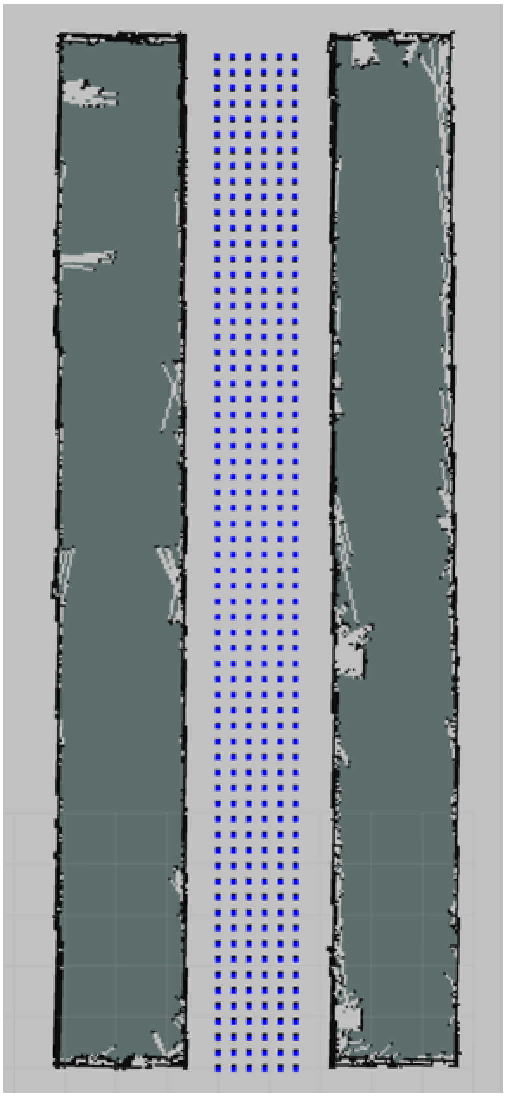


Figure 6. *Coordinate layout of localization evaluation environment. Six horizontally spanning intersection points are positioned 0.3048 meters apart. The vertically spanning intersection points are spaced 1.828 meters apart. An additional buffer region is placed between the recorded coordinates and the aisle shelves.*

Table I: Omnidirectional camera localization model performance. Several models with differing weight initialization methods are evaluated using a held-out validation dataset for warehouse localization.

Weight Initialization	Epochs	L1 (meters)	Classification Accuracy
ImageNet	100	0.066	0.9602
Self-Supervised (2 Patches)	100	0.126	0.1506
Self-Supervised (4 Patches)	100	**0.070**	**0.9176**
Self-Supervised (6 Patches)	100	0.076	0.8989
Random Gaussian	100	0.089	0.8656

Experimental Analysis

Omnidirectional Camera Model Results

All omnidirectional camera localization models were trained with 3,504 labelled training images with eight samples for each of the 438 different position classes. A held-out validation dataset containing 1,314 samples (three per class) served as the evaluation metric. The results for several omnidirectional models with different weight initialization methods are displayed in Table 1. The ImageNet pre-trained model performs the best out of all the evaluated models. The self-supervised pre-trained models with patch sizes of four and six outperformed the randomly initialized model. Overall, the self-supervised pre-trained model with four patches provided the best overall classification accuracy and regression accuracy when excluding the ImageNet pre-trained model. The ImageNet pre-trained model achieved a classification accuracy of 96.02% and an L1 loss of 0.066 meters for the regression model. The best performing self-supervised pre-trained model used four patches during the pretraining routine and achieved a classification accuracy of 91.76% and an L1 loss of 0.07 meters.

While the final result from the self-supervised pre-trained models was just over 4% worse for classification, the accuracy was still over 5% greater than the randomly initialized network. These results align with the findings from Doersch et al. [8] and Pathak et al. [9] for the self-supervision schemes presented in both works. The omnidirectional model presented by Relyea et al. [16] achieved a classification accuracy of 92.63% with ImageNet weight initialization and a deeper ResNet50 architecture on a similar dataset. The proposed ImageNet initialized model outperforms this previous work by over 3% and the best self-supervised pre-trained model is just 0.87% lower.

Validation loss convergence speed and stability were observed to be greater in the self-supervised pre-trained omnidirectional localization models when compared to the ImageNet and random initialized models. Validation loss curves obtained during training for the ImageNet, four patch self-supervised , and random initialized networks are shown in Figure 7. Figure 7 suggests that the features learned from the omnidirectional self-supervision task are more meaningful for the localization task in the earlier stages of training.

3D LiDAR Model Results

The PointNet++ based position classification model was trained with 2,190 training samples with five samples for each of the 438 different position classes. A held-out validation dataset containing 1,314 samples (three per class) served as the evalu-

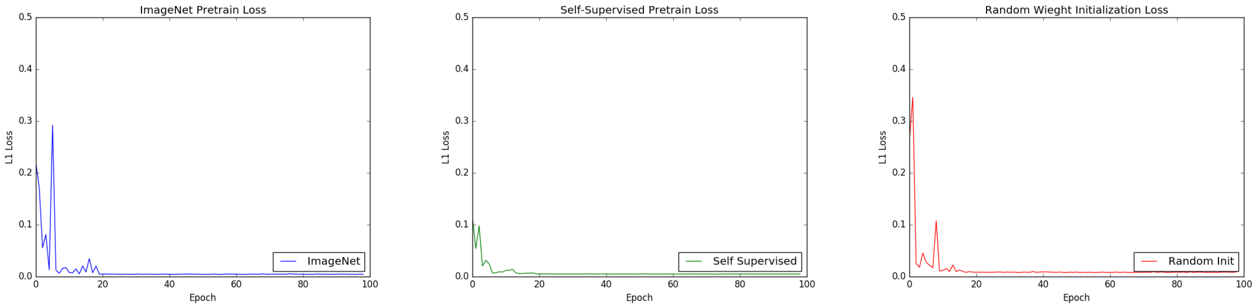


Figure 7. *Omnidirectional camera localization model validation dataset loss during training for the ImageNet (left), four patch self-supervised (center), and random initialized (right) networks. A learning rate decrease occurred every 20 epochs. Convergence speed and overall loss stability is greater for the self-supervised pre-trained network compared to the ImageNet and random initialized networks.*

ation metric. With a batch size of 16, the PointNet++ classification model was only able to achieve a classification accuracy of 46.36% on the held-out validation dataset after training for 100 epochs. The omnidirectional data lacks accurate object to lens distance measurements, while the LiDAR data lacks dense feature-rich details. The individual reflections or points in the point cloud are highly precise, but the sparsity of these points as compared to the feature-rich omnidirectional camera, inhibits LiDAR from being used as a localization sensor. While we expected lower localization accuracy of the point clouds as compared to the omnidirectional frames, we did not anticipate such large differences.

Conclusion

Warehouse robot localization models for an omnidirectional camera and a 3D LiDAR were trained and evaluated in this work. A novel self-supervision pre-training scheme for omnidirectional camera images was presented. This self-supervision technique improved training convergence and stability over random weight and transfer learning intialization methods. However, for optimal final validation performance, transfer learning performed best. We compared and contrasted using LiDAR vs. an omnidirectional sensor for indoor robot localization and found the omnidirectional sensor to perform best.

Acknowledgments

This research is sponsored in part by a grant from Toyota Material Handling of North America.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in *CVPR09*, 2009.

[2] C. Forster, M. Pizzoli, and D. Scaramuzza, “Svo: Fast semi-direct monocular visual odometry,” in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 15–22.

[3] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, “Svo: Semidirect visual odometry for monocular and multicamera systems,” *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2017.

[4] Z. Zhang, H. Rebecq, C. Forster, and D. Scaramuzza, “Benefit of large field-of-view cameras for visual odometry,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 801–808.

[5] W. Hess, D. Kohler, H. Rapp, and D. Andor, “Real-time loop closure in 2d lidar slam,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1271–1278.

[6] R. W. Wolcott and R. M. Eustice, “Visual localization within lidar maps for automated urban driving,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 176–183.

[7] T. Caselitz, B. Steder, M. Ruhnke, and W. Burgard, “Monocular camera localization in 3d lidar maps,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1926–1931.

[8] C. Doersch, A. Gupta, and A. A. Efros, “Unsupervised visual representation learning by context prediction,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1422–1430.

[9] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[10] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[11] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[13] G. Huang, Z. Liu, and K. Q. Weinberger, “Densely connected convolutional networks,” *CoRR*, vol. abs/1608.06993, 2016. [Online]. Available: http://arxiv.org/abs/1608.06993

[14] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[15] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric

space," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5099–5108. [Online]. Available: http://papers.nips.cc/paper/7095-pointnet-deep-hierarchical-feature-learning-on-point-sets-in-a-metric-space.pdf

[16] R. Relyea, D. Bhanushali, A. Vashist, A. Ganguly, A. Kwasinski, M. E. Kuhl, and R. Ptucha, "Multimodal localization for autonomous agents," *Electronic Imaging*, vol. 2019, no. 7, pp. 451–1, 2019.

Author Biography

Robert Relyea is a second-year computer engineering M.S. student at the Rochester Institute of Technology. He obtained his B.S. in computer engineering at the Rochester Institute of Technology in 2018. His research focus includes applications of computer vision and deep learning techniques to robot perception and localization.

Darshan Bhanushali is a second-year computer engineering M.S. student at the Rochester Institute of Technology. He obtained his Bachelor of Engineering in Electronics from the University of Mumbai in 2016. His research interests include computer vision, robot autonomy, and sensor modelling.

Karan Manghi is a third-year computer science M.S student at the Rochester Institute of Technology. He studied B.S in information technology at the University of Mumbai and received his degree in 2017. He is most interested in mobile robot path planning and navigation, computer vision and algorithm design for navigation of a mobile robot.

Abhishek Vashist is currently pursuing his Ph.D. degree in Department of Computer Engineering at Rochester Institute of Technology, Rochester, NY, USA. He received his M.S. degree in Electrical Engineering from Rochester Institute of Technology, Rochester, NY, USA in 2017 and B.Tech. degree from ABES Engineering College, India in 2014. His research interest includes machine learning, 60 GHz mmWave, and computer architecture.

Clark Hochgraf is an Associate Professor in the Department of Electrical, Computer, and Telecommunications Engineering Technology at Rochester Institute of Technology. He earned his Ph.D. in Electrical Engineering from the University of Wisconsin-Madison in 1997. He was a principal engineer at Visteon, held research positions at Westinghouse and General Motors, and has 12 US patents. He is a certified master trainer in Educational Mobile Computing. His areas of research interest include intelligent vehicles, power conversion, and education of engineers.

Amlan Ganguly is currently an Associate Professor in the Department of Computer Engineering at Rochester Institute of Technology, Rochester, NY, USA. He received his PhD and MS degrees from Washington State University, USA and BTech from Indian Institute of Technology, Kharagpur, India in 2010, 2007 and 2005 respectively. His research interests are in robust and scalable intra-chip and inter-chip interconnection architectures and novel datacenter networks with emerging technologies such as wireless interconnects. He is an Associate Editor for the Elsevier Journal of Sustainable Computing Systems (SUSCOM) and the MDPI Journal of Low Power Electronics and Applications (JLPEA). He is a member of the Technical Program Committee of several conferences such as International Green and Sustainable Computing (IGSC) and International Network-on-Chip Symposium (NOCS). He is a member of IEEE.

Andres Kwasinski received in 1992 his diploma in Electrical Engineering from the Buenos Aires Institute of Technology, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Maryland at College Park, in 2000 and 2004, respectively. He is currently a Professor at the Department of Computer Engineering, Rochester Institute of Technology. He is a Senior member of IEEE, Chief Editor of IEEE SigPort and Area Editor of the IEEE Signal Processing Magazine.

Michael E. Kuhl is a Professor in the Department of Industrial and Systems Engineering at Rochester Institute of Technology. He earned his Ph.D. in Industrial Engineering from North Carolina State University in 1997. His areas of research interest including simulation modeling and analysis, the design and development of autonomous material handling systems, and application of simulation to supply chain, healthcare, and cyber security systems.

Raymond Ptucha is an Assistant Professor in Computer Engineering and the Director of the Machine Intelligence Laboratory at the Rochester Institute of Technology. His research includes machine learning, computer vision, and robotics, with a specialization in deep learning. Ray was a research scientist with the Eastman Kodak Company where he worked on computational imaging algorithms and was awarded 31 U.S. patents. He earned a Ph.D. in computer science from RIT in 2013. Ray was awarded an NSF Graduate Research Fellowship in 2010 and his Ph.D. research earned the 2014 Best RIT Doctoral Dissertation Award. Ray is a passionate supporter of STEM education, an NVIDIA-certified Deep Learning Institute instructor, and the Chair of the Rochester area IEEE Signal Processing Society.

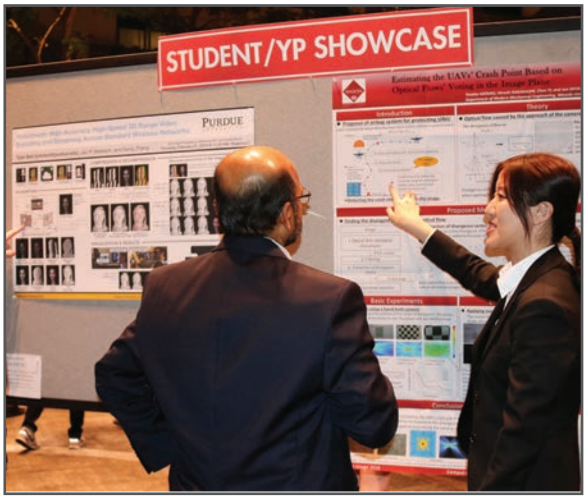
JOIN US AT THE NEXT EI!

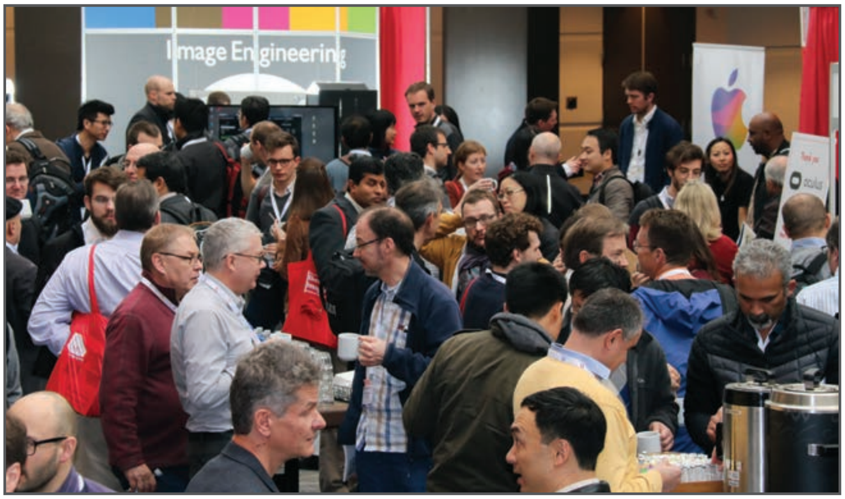
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





AR is a fundamental and transformative interface and platform.
20+ years from my initial working demo to today's AR displays
Optimistic that it won't take another 20 to reach ubiquitous consumer AR!
My first working AR prototype
Today's platforms
Ubiquitous consumer AR

• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org