Verifiably Robust Sonar Perception

Jack Zhang (CAS ‘24)*, Ryan Tong (SEAS ‘23)*, Sriram Tolety (SEAS & Wharton ‘23)*
Ivan Ruchkin, Radoslav Ivanov, Oleg Sokolsky, Insup Lee
*: Equal-contribution

Penn Research in Embedded Computing and Integrated Systems Engineering (PRECISE)
Penn Undergraduate Research Mentoring Program (PURM)

Motivation and Objective

Background

We are interested in using an Unmanned Underwater Vehicle (UUV) to follow an underwater pipeline by keeping parallel with it a certain distance away. The UUV has an Inertial Measurement Unit (IMU), Doppler Velocity Log (DVL), and multiple sonar sensors that allow it to know its state and scan its environment. We are concerned with sonar images gathered from the side-facing sonar.

Motivation

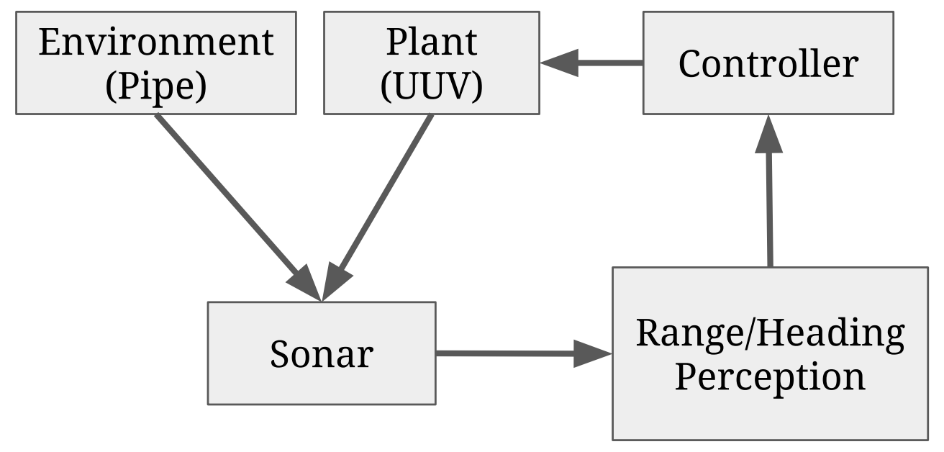
- Autonomous systems require robust perception of the environments
- Verifying that a complex system behaves correctly is difficult; however, verifying its neural net-based perception is a stepping stone to the whole-system guarantees

Goal

- Construct a perception system for the UUV’s lateral distance to the pipeline (“range”)
- Computationally verify the robustness of that perception system

Challenges

- Realistic sonar simulation and perturbations need to be designed
- Sonar scans are high-dimensional and noisy, making verification of model robustness non-trivial for large networks


Environment (Pipe)
Plant (UUV)
Controller
Sonar
Range/Heading Perception

Problem Statement

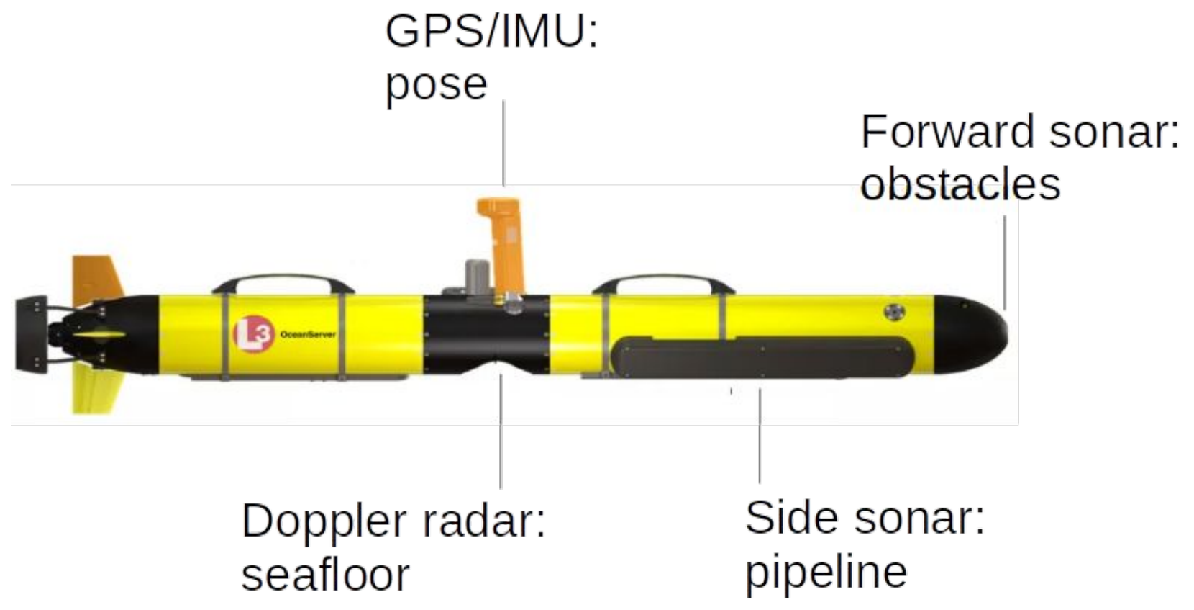
Our goal is to estimate the seafloor distance between the UUV and pipeline using the state of the UUV and sonar images. We then analyze the robustness of our range estimation by perturbing a given number of pixels within the sonar images.

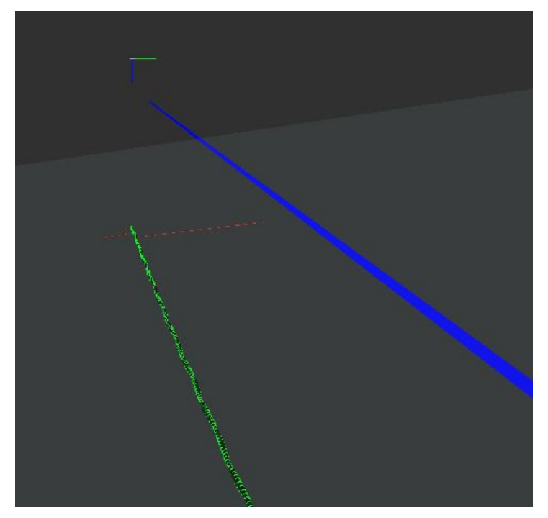
At each time step t, we are given a UUV pose $P := (P_{pitch}, P_{roll}, P_{height})$, a sonar intensity image M_i, a sonar distance image M_d, and wish to construct a range function $r(M_i, M_d, P)$ that estimates the range correctly. We are then interested in the robustness of this range function. In particular, we wish to calculate the range robustness interval $[min(r(M_i')), max(r(M_i'))]$ for all M_i' such that $\|M_i' - M_i\|_\infty \leq \epsilon$, where we want to minimize the size of the range interval.

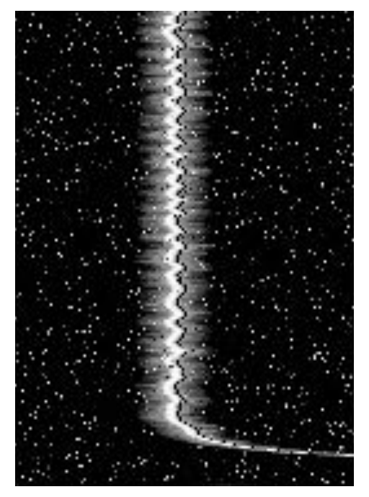
Perception System Overview

We model and test our control and perception system using a Gazebo and ROS based simulator. We are able to manipulate the starting position, heading, and controller of the UUV allowing us to gather a diverse dataset to be used to evaluate and train our range estimation pipeline. The UUV in the simulator will periodically send out sonar scans on the side sonar. Each ray in the sonar scan corresponds to a pixel in the sonar image reflecting intensity and the distance of the ray. We collect this data to generate a sonar image along with synchronized pose and true range information and feed this to our range estimation and evaluation pipeline.

The simulator environment lacked noise and variability in sonar images. We improved upon this by adding Gaussian noise and localized pipeline noise.



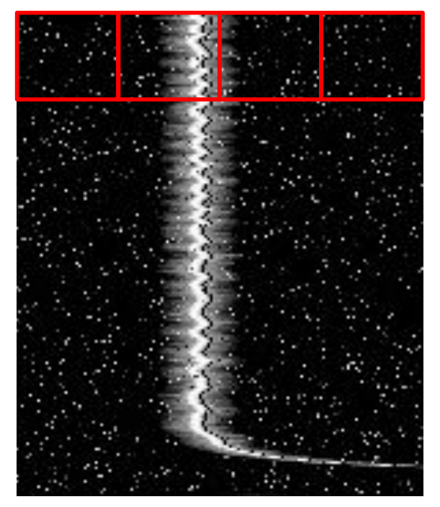

UUV Simulation (Green: UUV Path, Blue: Pipeline

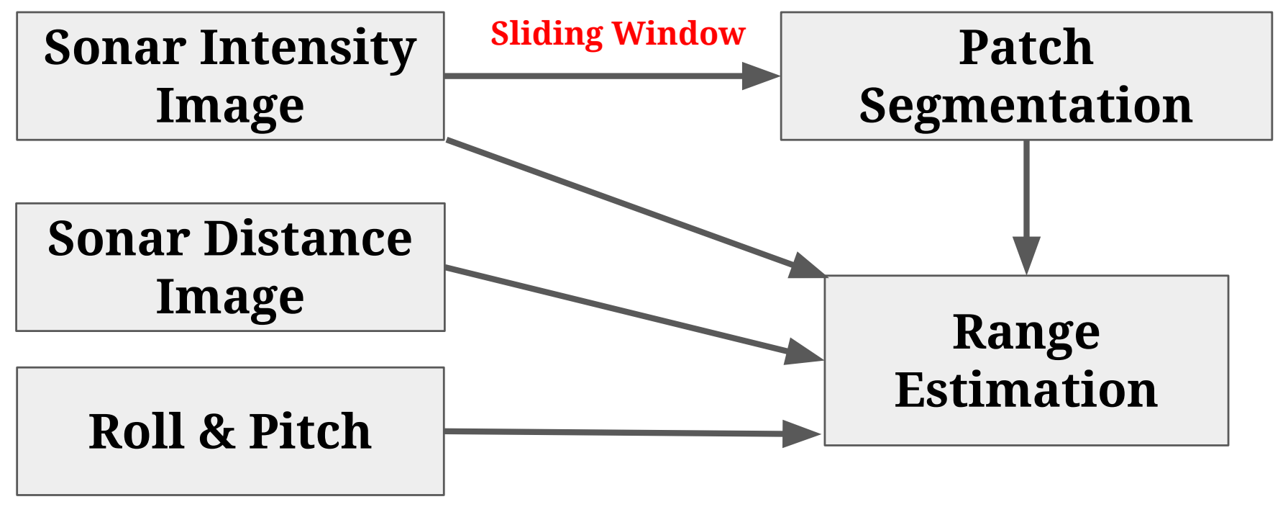

Sonar image with localized pipe noise

Perception Decomposition

It is difficult to verify robustness claims by considering the entire perception system. To make the problem tractable, we split it into patch segmentation using a CNN for pipeline detection and range perception using the patches. Verifying segmentation robustness implies end-to-end perception robustness:

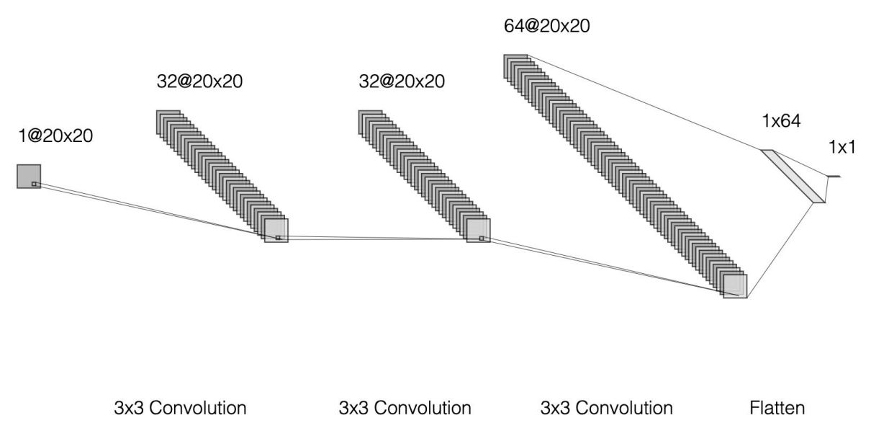
For some patch segmentation $S \in \{0,1\}^k$, given functions $r : X \to Y$ for $X \in (M_i, M_d, P)$ and $Y \in R^+, f : X \to Z$ with $Z \in (M_i, M_d, P, S)$, and $g : Z \to Y$, where $r = g \circ f$—i.e., $r(x) = g(f(x))$ then we note that if f is $(\Delta_\mathcal{X}, I_\mathcal{Z})$-robust, then r is $(\Delta_\mathcal{X}, I_\mathcal{Y})$-robust, i.e. the range estimate is robust. For instance, in the L_∞ norm, this implies that if our segmentation is invariant under perturbations δ to the sonar images such that $\|\delta\|_\infty \leq \epsilon$, then our range estimation is also invariant to these perturbations.




Sonar Intensity Image
Sliding Window
Patch Segmentation
Sonar Distance Image
Roll & Pitch
Range Estimation

Pipeline Patch Segmentation

A CNN was trained until convergence on an NVIDIA P100 GPU to detect the presence of pipeline in a given patch (segmentation) using 10 noisy simulated UUV runs. Two other runs were kept hidden as a holdout/validation set, and six runs were kept as a test set for robustness estimation.



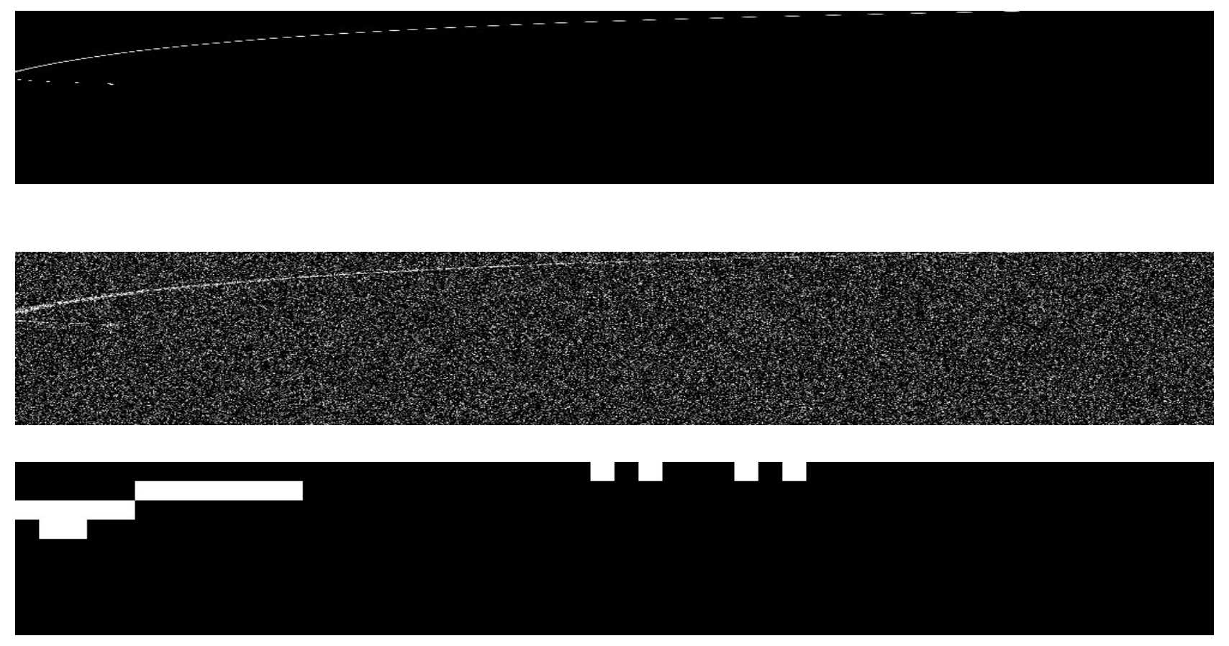
Visualization of the CNN architecture.

Standard data augmentations increased accuracy against noisy images. However, multiple other attempts to increase robustness (L1/L2 regularization, dropout layers, ensembling models) did not significantly improve accuracy. Adding more layers to the CNN did not significantly increase accuracy either; this may be due to the fact that our patch classification task for segmentation may not benefit from the additional non-linearity of a deeper neural network.

Pipeline Range Estimation

To estimate the Euclidean distance from the UUV to the pipeline in a given image, the classified patches **C** and the sonar image **S** are used to construct a weighted average of relevant values from the distance matrix **D**.

From the Euclidean distance estimate, the UUV height-from-seafloor estimate, roll, and pitch are used to compute the seafloor distance (range) using trigonometry.



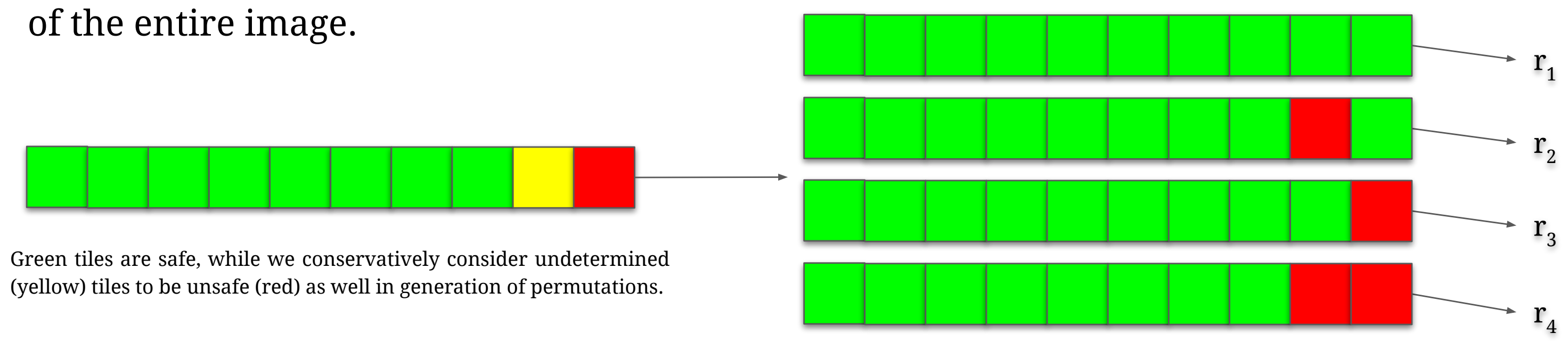
Visualization of an intensity scan (clean and noisy) and the patch segmentation.

Robustness Verification

In order to verify robustness of range estimation, our system considers the misclassification of each tile in an image, and how that affects range estimation. Neurify verified safety of the networks when faced with attack vector X’ generated from network input X as follows:

$$||X' - X||_\infty \leq \epsilon$$

In particular, for each tile, Neurify identifies whether the classification can be successfully attacked with an ε-bounded L-∞ perturbation. In order to provide a conservative robustness guarantee, we then calculate a range using all permutations of unsafe and undetermined tiles, which is equivalent to all possible segmentations within ε of the entire image.



Green tiles are safe, while we conservatively consider undetermined (yellow) tiles to be unsafe (red) as well in generation of permutations.

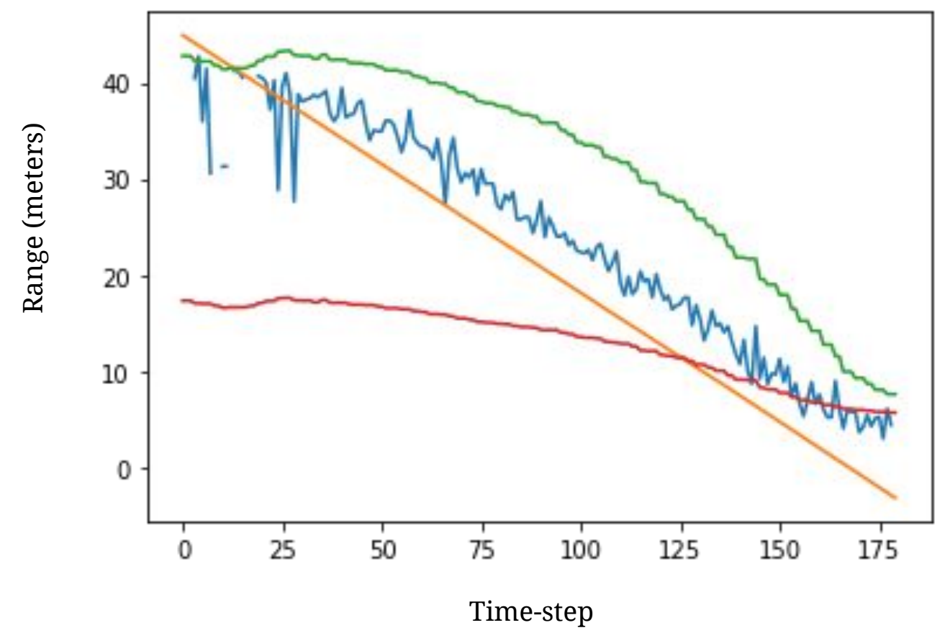
Using these attack vectors, our system generated all possible permutations of the binary mask, and determined the set of possible range estimates. We then store absolute differences in range estimates and our pipeline’s estimate, thereby enabling concrete statements regarding the robustness of our system in the presence of noise and other attacks.

Results & Conclusions

Range Estimation Accuracy: We compare our method against two existing baselines. However, these baselines were calculated using the clean images due to engineering constraints, meaning that their performance is actually *overstated* compared to our method, which uses noisy images. Still, we find our method performs significantly better than the two existing legacy methods.

	Our Method	Baseline 1	Baseline 2
MAE	**3.79**	11.22	10.51
RMSE	**4.31**	12.29	13.46

Summary of accuracy results.



Visualization of range estimates over the course of one simulated run. Orange, ground truth; blue, our method; green, baseline 1; red, baseline 2.

Range Estimation Robustness: By checking exhaustively the set of segmentation permutations and ranges for representative sample of the test set, we can provide strong guarantees of robustness against L-∞ norm-bounded perturbations. Even against significant perturbations, our network was very robust and no adversarial tiles were identified. Performance of our network decreased as the size of the attack increased—at ε = 100, performance was still strong, with an average range estimation error of 17.9%.

Epsilon	Average Mean Dif.	Average Max Dif.	Tiles Flipped
10	0	0	0%
25	0	0	0%
50	0	0	0%
75	0	0	0%
100	4.70	9.76	24%

Summary of robustness results; all epsilons are in L_∞-norm. The average ground-truth range was 26.2

We have studied an approach on improving the robustness of neural networks in real-world perception tasks by decomposing a complex function into tractably-verifiable parts. In particular, by utilizing patch segmentation, we were able to analyze and provide meaningful guarantees on the robustness of the end-to-end perception system. In doing so, we developed a highly accurate and robust range estimation method that significantly outperforms existing baselines. Future work mayentail further enhancements to the network, as well as perhaps a deeper look into how different attacks such as L-1 or L-0 affect system robustness.