Landmine detection using 3 dimensions of vibration information

(Soil Characterization, Modeling and Sensor Understanding)

For NVESD Countermine

Contractor's Final Report

Submitted 3 August 2019

Administrative and Technical Point of Contact:

Kenneth D. Morton, Jr., Ph. D.

CoVar Applied Technologies, Inc.

1765 Greensboro Station Pl, Suite 900

McLean, VA 22102

DOD Sponsor Point of Contact

Brad Libbey, Ph.D.

Army C5ISR Center, Night Vision and Electronic Sensors Directorate

10221 Burbeck Rd

Ft. Belvoir, VA 22060

1 CONTENTS

2 INTRODUCTION

Working with NVESD CM and ERDC, CoVar was funded to conduct data analysis and algorithm development for the detection of explosive buried threats in surface vibrational data. For this effort, CoVar was tasked with using available simulated and field data in order to better understand the phenomenology of detecting buried explosive hazards using measurements of ground vibrations. Historically, we have investigated a laser Doppler vibrometer (LDV) which measures only the component of motion of the ground along the line of sight to the sensor. New sensors are currently under development which may be able to sense all three components of motion of the ground surface at standoff.

Using all three components of motion presents a much richer representation of the soil motion which can be used to detect anomalies and therefore buried threats. Much of how to utilize three component data is still unknown, as there is currently very limited data of ground motion over buried hazards.

There are two primary excitation modalities currently being investigated: acoustic and seismic. Both excitation modes have significantly different phenomenology, and how to approach the three-axis data may be different depending upon the excitation source. Simulating the acoustic excitation mode is challenging due to a requirement to model the air fluid in the pores of the soil to produce accurate seismic response results of the soil. Conversely, good simulation models exist in the seismic excitation modality to provide insight as to how to process three-axis data produced by seismic excitation.

ERDC's seismic simulation group has been working to provide detailed models of seismic waves exciting buried buckets filled with explosives. These seismic simulations have been done over a range of soil and target properties expected under arid conditions. We utilized this seismic simulation data under this effort to further the understanding of the detection problem.

The other primary source of data used under this effort is a collection over a buried target at Ft. Belvior using accelerometers placed on the surface. The accelerometer data was collected with both seismic and acoustic excitation over a three-day period. This limited data set enabled us to study both acoustic and seismic modalities and the impact of actual soil and target parameters, and sensor issues.

The work conducted by CoVar over this effort has expanded the understanding of methods that may be used to detect explosive hazards using surface vibrational measurements. Significant data analysis of all the above described data was conducted and several algorithms were developed to explore the potential of target detection using 3-axis data. In particular, this work culminated in a method, based on convolutional neural networks, that can learn the signature of buried disturbances from 3-axis vibrational data. This method can be trained from synthetic data, and when applied to the limited available field data, appears to offer performance that is better than other techniques that were previously developed and those developed during this effort. This CNN based approach appears to be very promising and further development and analysis of this method, using additional field data and more realistic simulations, should be a focus of future work.

3 SIMULATED SEISMIC DATA

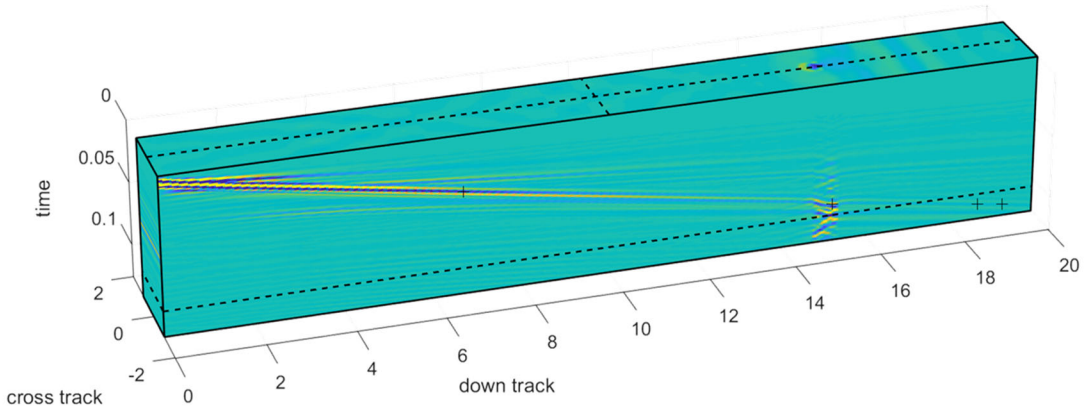

Figure 1: Seismic simulation cube. X & Y axes units are in meters, while the time axis is in seconds. Visualized here is the down-track component vector.

Figure 1 shows an example visualization of a seismic simulation cube. The seismic simulation results in a 3D cube of the 3D surface motion over time. The data can be thought of as three 3D cubes, or a 3D vector field, although the third dimension of the vector field is not strictly tied to the third temporal dimension in the way that the x & y dimensions are.

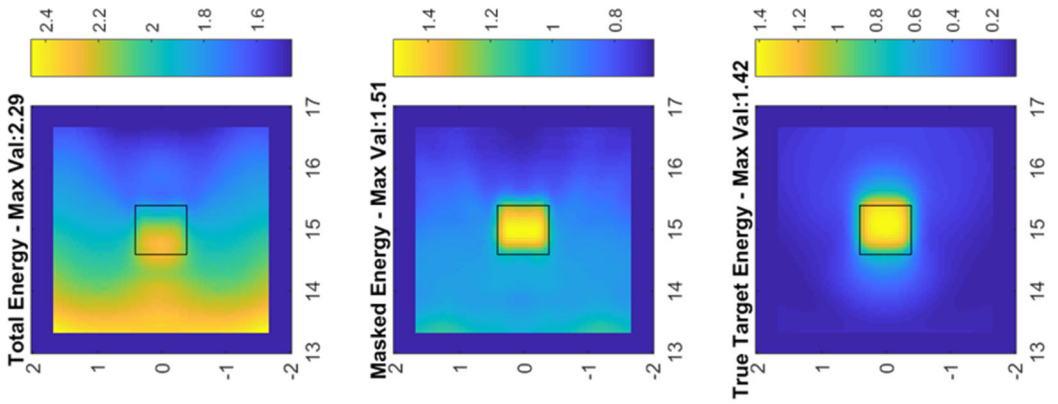

Figure 2: Example images from the background subtraction technique. The figure is turned sideways to fit better on the page. Here the x and y axes indicate the 2D offset to the seismic source in meters.

Figure 2 shows example images from a background subtraction technique developed for analyzing the simulated data. Based on lessons learned from the SAVI acoustic system, we believed that a good seismic detection algorithm could separate traveling waves from target responses in the 3D FFT domain, as the proven algorithm did for the moving system. Toward this end, we decided to do a background subtraction in the 3D FFT domain in order to separate the target response from the traveling wave to measure response energy. We hoped by doing this that we would come up with a reasonable metric with which to evaluate target detection rates as a function of the simulation parameters.

It turned out that much of the response energy was dictated by the magnitude of the passing wave – a function solely of the modeled soil layers and not the target parameters. We could then separate these two factors by normalizing by the wave energy. This produced a detection metric of the maximum background subtracted energy divided by wave energy.

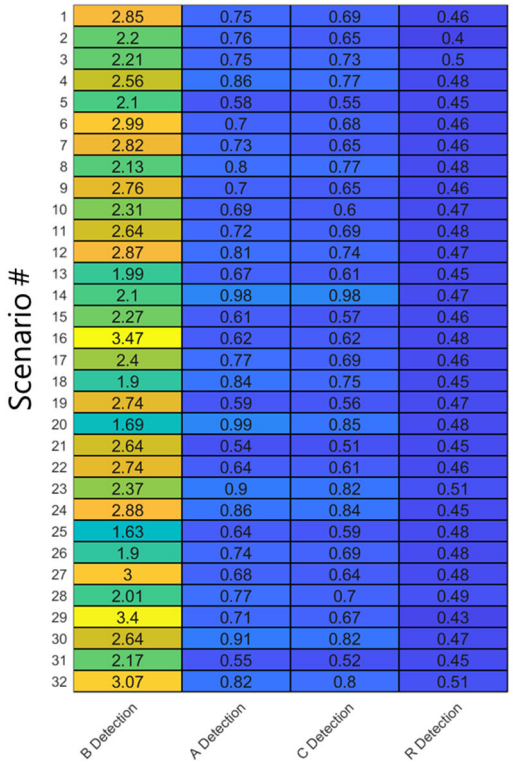

Figure 3: Table of detection metrics over different simulation parameter realizations. The scenario #s correspond to specific realizations of the soil characteristics as well as target properties (such as depth, thickness, air gap, etc.). B corresponds to a simulated environment with disturbed earth surrounding the target. A & C corresponds to a target emplaced with no disturbance, although C differs from A in the properties of the target lid. R corresponds to a scenario with no target.

Figure 3 shows the detection metrics for each provided simulation experiment. We found that the disturbed soil had a much larger effect on the target response than anything else. After sending these results to Sergey Vecherin at ERDC, we learned that the primary driving factor in the A & C columns was the target depth.

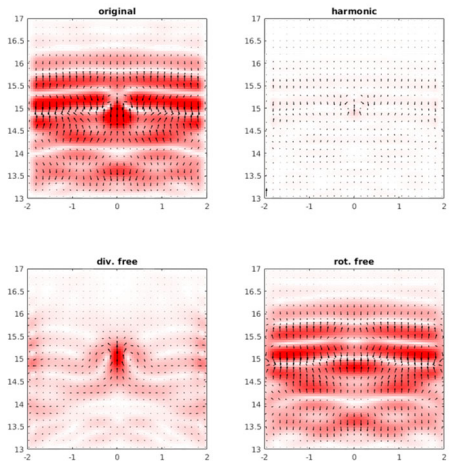

Figure 4: 2D Helmholtz Decomposition of a 2m section of the simulated surface. Black vectors show the horizontal and vertical components of the vector field, while the red image shows the magnitude of the vector field. Here the x and y axes indicate the 2D offset to the seismic source in meters.

Viewing the seismic excitation data as a vector field, we intended to find a way to combine the information contained in the three-axis data as well as determine the usefulness of the three-axis data as opposed to a single component sensor.

A traveling wave is typically a rotation-free vector field, which can be written as the gradient of some scalar field. If our vector field was a rotation-free vector field, one component of motion would be enough to sample the information contained in that field. Therefore, it was important to test that the vector field also contained divergence-free components of motion for the three-axis data to be useful.

The Helmholtz decomposition states that any vector field can be written as a sum of a divergence-free and rotation-free vector fields that are unique, provided they decay to zero sufficiently fast. If we had a sampling of a large enough area around the source, we could calculate exactly these unique vector fields. Unfortunately, we will never in practice see an area this large and had to come up with some computational shortcuts in order to approximate these unique vector fields.

Figure 4 clearly shows significant energy in the divergence-free component, particularly over the target response. This was an important finding as it suggested that the three-axis data would add a significant amount of information for detecting the target. A secondary hope was that this method could be used as a background separation technique that could isolate the traveling wave from the target response.

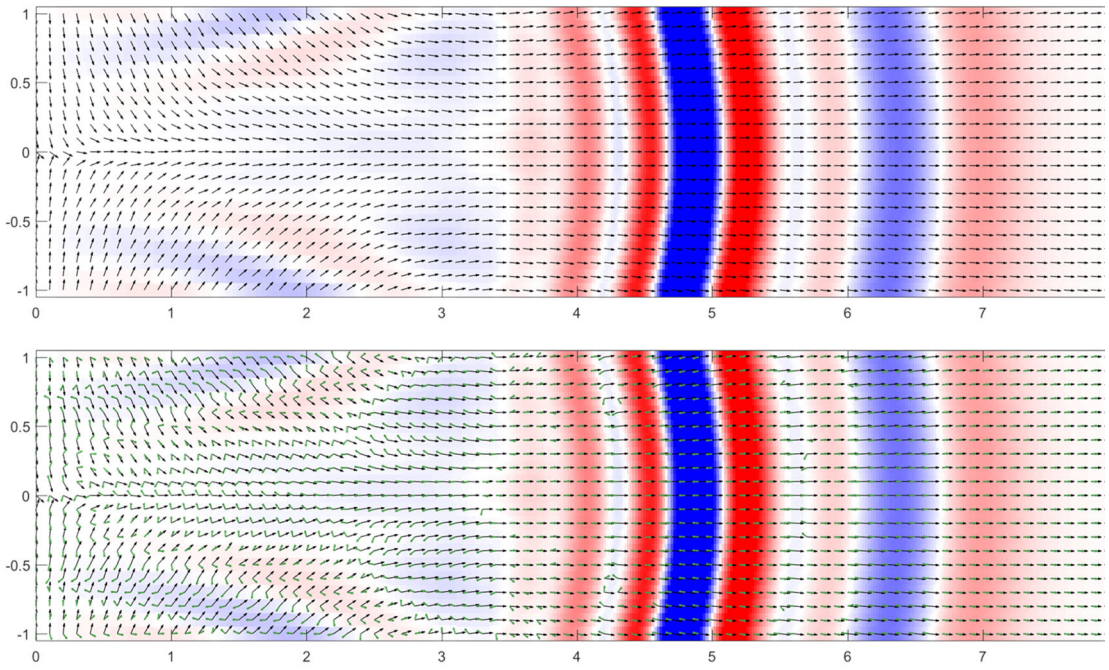

Figure 5: Top: black arrows indicate the direction of motion of the traveling waves in the simulation environment, while the color indicates the vertical surface velocity. The waves reflected off simulated road edges can be seen to travel inward. Bottom: The same visualization with the x & y components of motion are shown in green. The x and y axes indicate position relative to the source (in meters). Using this method, it can be seen which types of waves are present in the environment.

Next, we generated a simple algorithm that determines the orientation of propagating waves through the space. This is done by combining the spatial derivatives of the vector field with the temporal derivatives of the vector field. The derivatives are smoothed over space to reduce noise in the system. We also have a visualization, shown in the bottom of Figure 5, where we display the direction of the components of motion on the surface. Where these are aligned, we have Rayleigh or P-waves, and where they are not aligned, we see Love or Shear waves. It can be clearly seen that the waves propagating inward from the simulated road edge have components of motion which do not align with the direction of wave travel, meaning that they must contain some Shear or Love wave components.

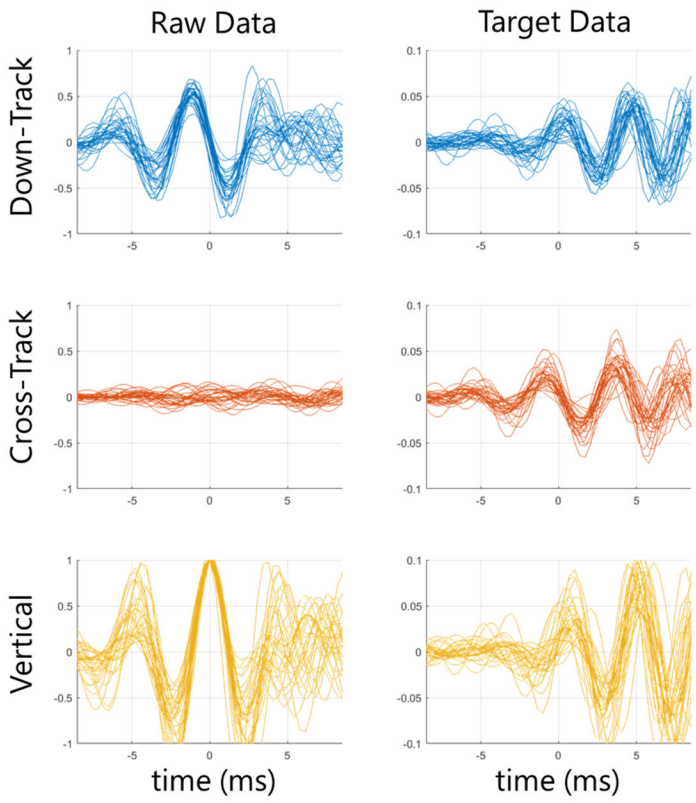

Figure 6: Time slices of three-axis (down-track in plane, cross-track in plane, and vertical out of plane) data over the target for the 32 simulation environments. Left: The data is aligned and normalized relative to the maximum Rayleigh wave response. Right: The pure background simulation data is subtracted, leaving only the target response.

Figure 6 shows an interesting way of examining the target simulation data. By aligning to the passage of the Rayleigh wave, we see that the typical response exhibited by the target is consistent in both magnitude and phase. The down-track component is aligned with the vertical component, while the cross-track component is out of phase with the vertical component.

4 Acoustic Accelerometer Data

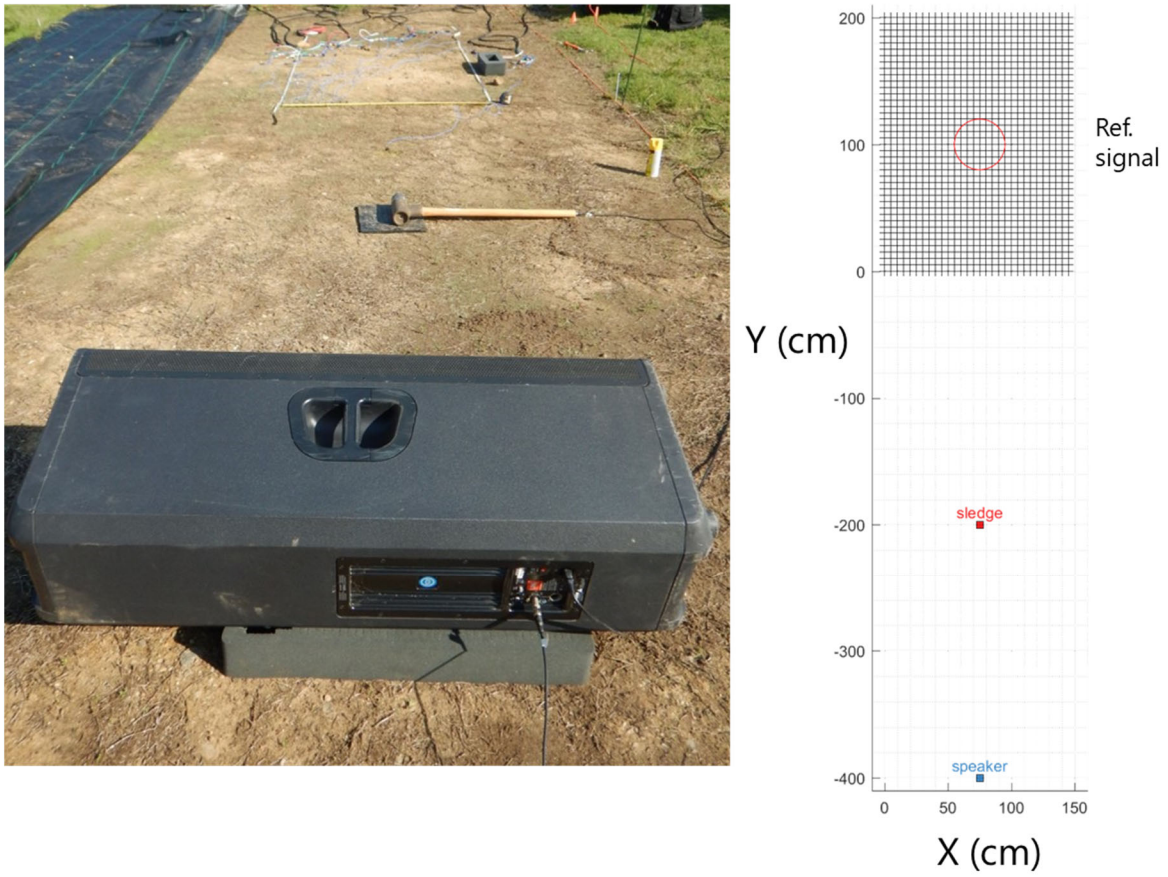

Figure 7: Accelerometer experimental setup. Left: photo of the test scene. Right: Simulated layout of the experiment.

Figure 7 shows a photo and computer representation of the experimental setup. The accelerometers are measured on a 31 x 40 grid with 5cm spacing at 5120 Hz. The target has been packed in to simulate a weathered in target. The speaker is used to broadcast a broadband signal from 50-500Hz.

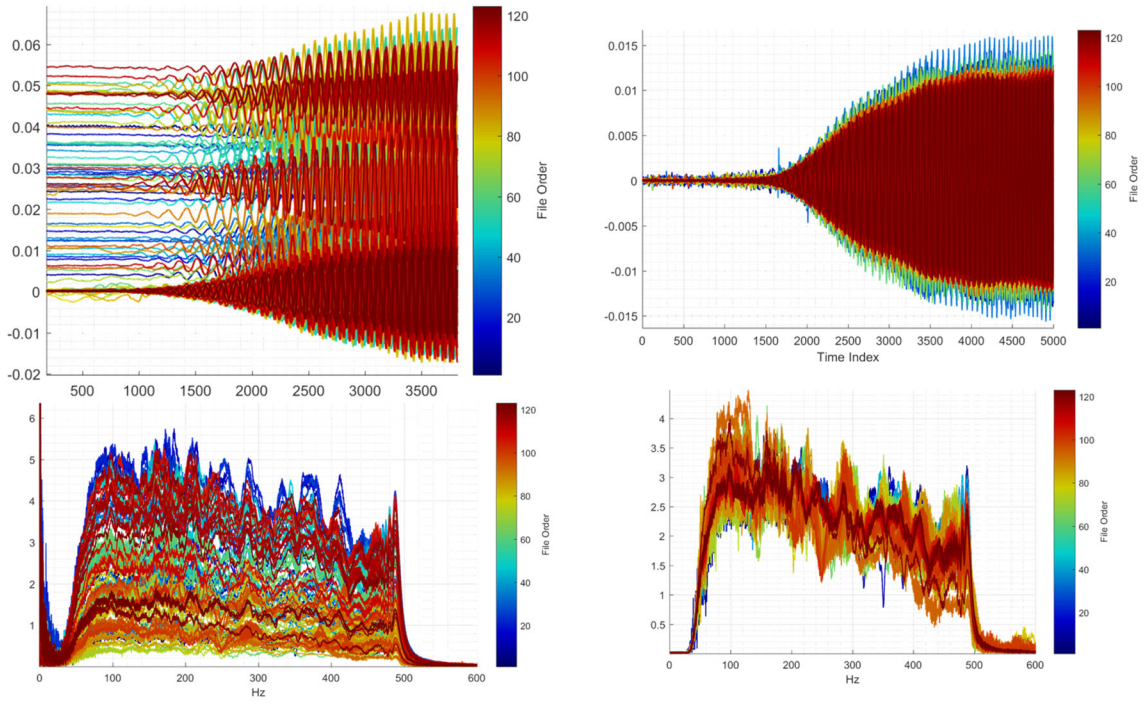

Figure 8: Left column: Raw accelerometer signals in time & Fourier domain. Right column: processed signals in time & Fourier domain.

Figure 8 shows the raw accelerometer data before and after some preprocessing. We can see a strong dependence upon the file order prior to the preprocessing. First, we apply a high pass filter to eliminate the low frequency components, as we are not even broadcasting on those frequencies. The MATLAB filter we used was butter(0.0156, 5, 'high'), and filtered using the filtfilt function. After that, each accelerometer is L2 normalized. The post-processed signals look much more uniform and do not show a strong dependence upon the file order.

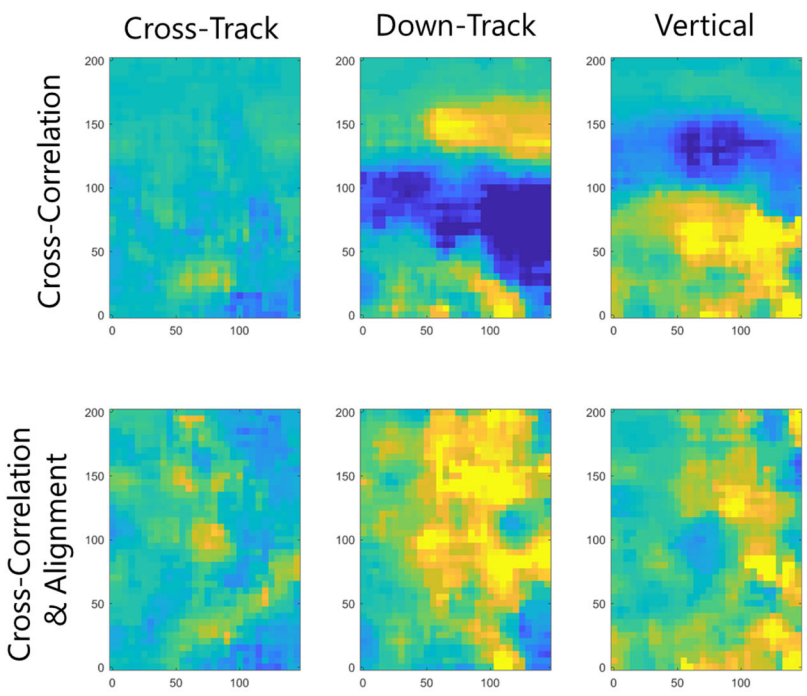

Figure 9: Example time slices before after time of flight alignment.

Figure 9 shows example time slices of the stitched acoustic excitation data over the collection surface. The data looks similar to data we received for the moving SAVI acoustic system.

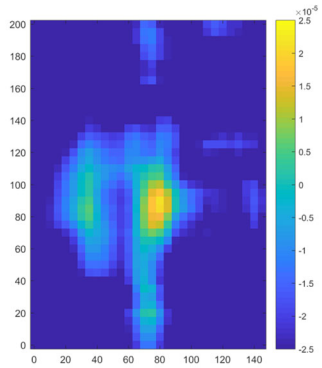

Figure 10: SAVI multi-scale 3D FFT classifier applied to the acoustic excitation data.

The multi-scale 3D FFT classifier can be applied directly to this pre-processed data by extracting 15cmx15cmx75ms cubes and running the classifier. Amazingly, the classifier developed on entirely different sensor, transmitted signal, and site generalized to pick out this target from the background, as evidenced by the detection map in Figure 10. Unfortunately, the preprocessing needed to get the data uniform across the grid removed the energy, and the units no longer match up with what SAVI detected, so these detection units are somewhat arbitrary.

5 SEISMIC ACCELEROMETER DATA

Next, we examined data that was collected over the same region and time period, but using a sledgehammer as a seismic source, also pictured in Figure 7. The change in excitation source has a drastic impact on the data, as the expected types of seismic waves change significantly.

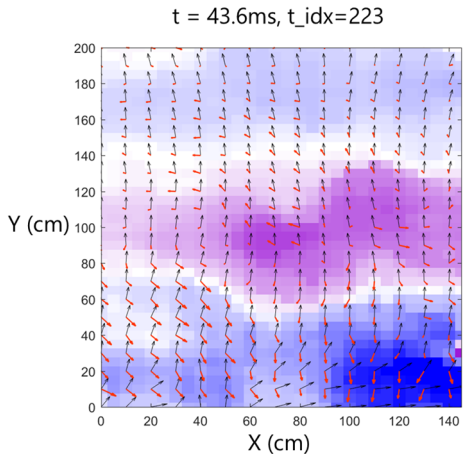

Figure 11: Time slice of the seismic excitation data. Black vectors indicate direction of wave propagation, while red vectors indicate the surface particle acceleration. The blue/purple mask indicates the vertical component of the surface acceleration.

First, in order to understand the seismic accelerometer data, we tried to use the wave direction estimation, and compare with the particle motion. It does appear that the wave estimation algorithm is working well, and indicates some interesting motion over the sensor domain. In Figure 11, the target in the center of the sensor region is clearly showing some stronger waves in the horizontal direction, as shown in the simulations in Figure 6.

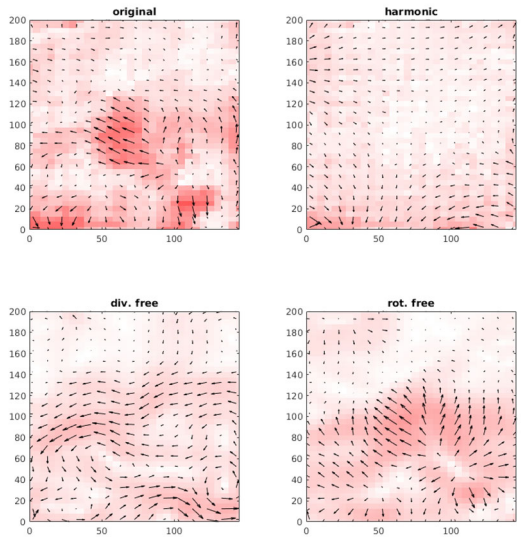

Figure 12: 2D Helmholtz decomposition on the seismic accelerometer data.

The Helmholtz decomposition as tested on the simulated data was not able to separate the components of motion which were target-like and wave-like, as was observed in the simulated data. After looking at the frames which appeared to contain target information, we decided to abandon this approach.

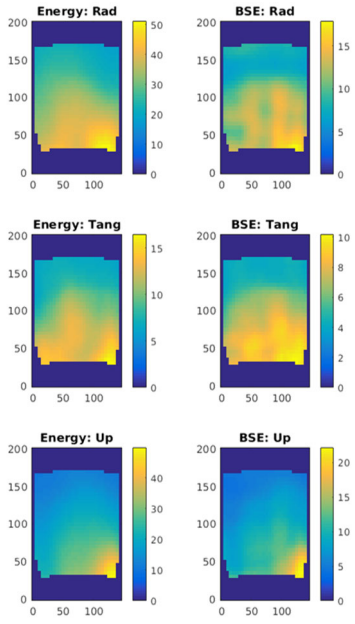

Figure 13: Background subtracted energy approach example on each component. The target is not visible.

Next, we tested the 3D FFT background subtraction technique used for the simulated data. Figure 13 shows the results, which show that the target is not detected using this technique. The vector fields in Figure 11 and Figure 12 shows that the waves are not very uniform as a function of space, even on a small 2m spatial scale.

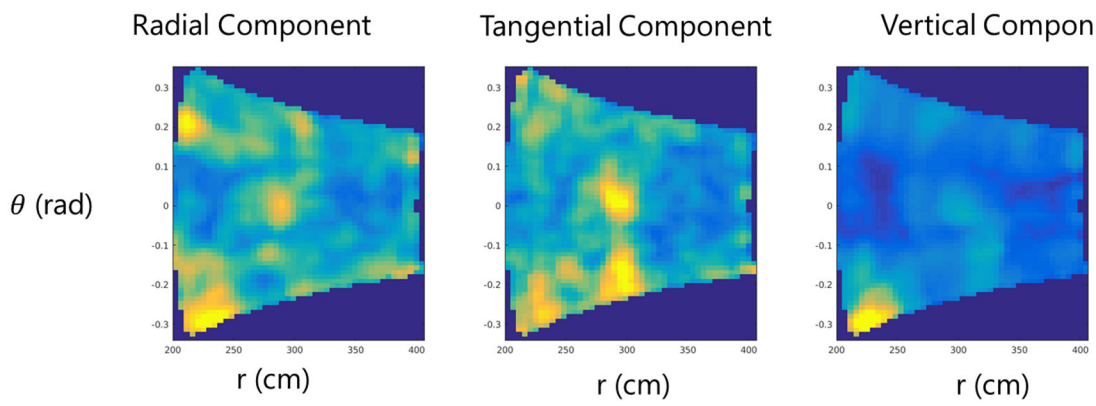

Figure 14: Simple background subtraction technique in radial coordinates.

Figure 14 shows the results of a simple background subtraction technique. The technique works as follows: first convert coordinates and vector field into radial coordinates relative to the seismic source. Next, subtract the mean along the angular dimension for each radius, time value. Last, we sum energy through time to produce a 2D anomaly map. The target clearly pops out in this approach, although there is some clutter, particularly in the lower left corner (corresponding to the lower right hand corner in the spatial domain).

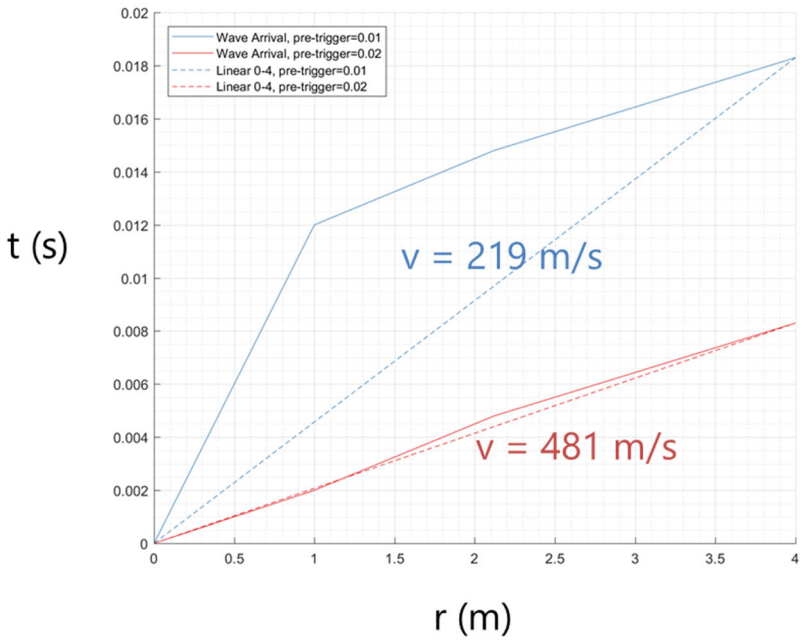

Figure 15: Wave speed plot for selected arrival times of the wave packet of the seismic data. The solid lines connect all selected points, while the dashed lines connect the first and last points to visually highlight a linear fit. The red & blue curves correspond to two different pre-trigger times.

In order to analyze the wave speeds or do dispersion analysis, we must have an accurate estimate of the time of impact of the seismic wave. In the experiment, a pre-trigger time of 0.01s was set, meaning that the accelerometers would save data beginning at 0.01s prior to the triggering event. Setting t0 = -0.01s results in a wave which propagates nonlinearly through the space. Setting t0=-0.02s results in a much better linear fit to the data, as shown in Figure 15, suggesting that there may be a timing offset of ~0.01s in the acquisition system.

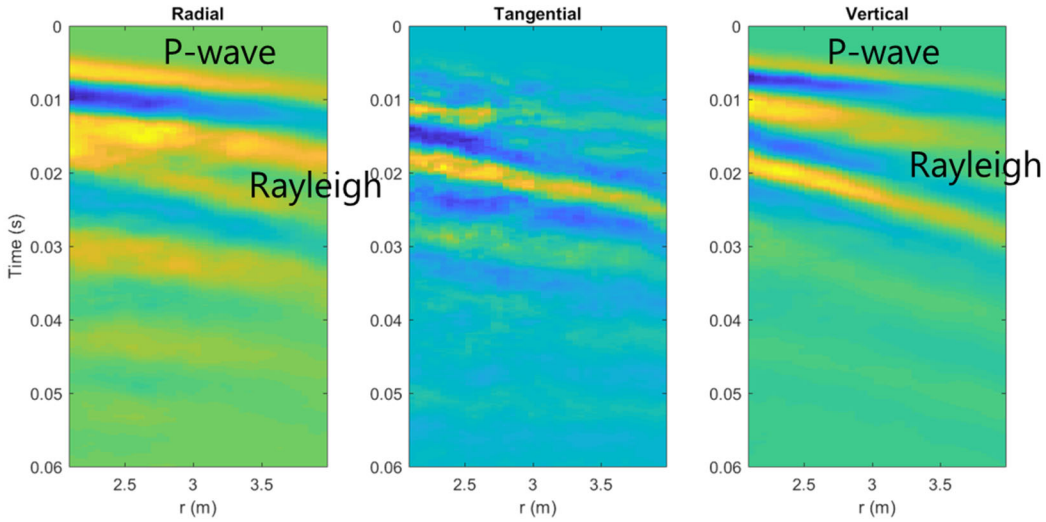

Figure 16: Image of the average wave pattern in radial coordinates. The P-wave and Rayleigh waves can clearly be seen in this image.

Figure 16 shows the average wave pattern in radial coordinates. In this case, the separation between the waves is large enough that they can be clearly separated. The Rayleigh wave is evident by the slower wave speed as well as the stronger vertical component and slower decay speed.

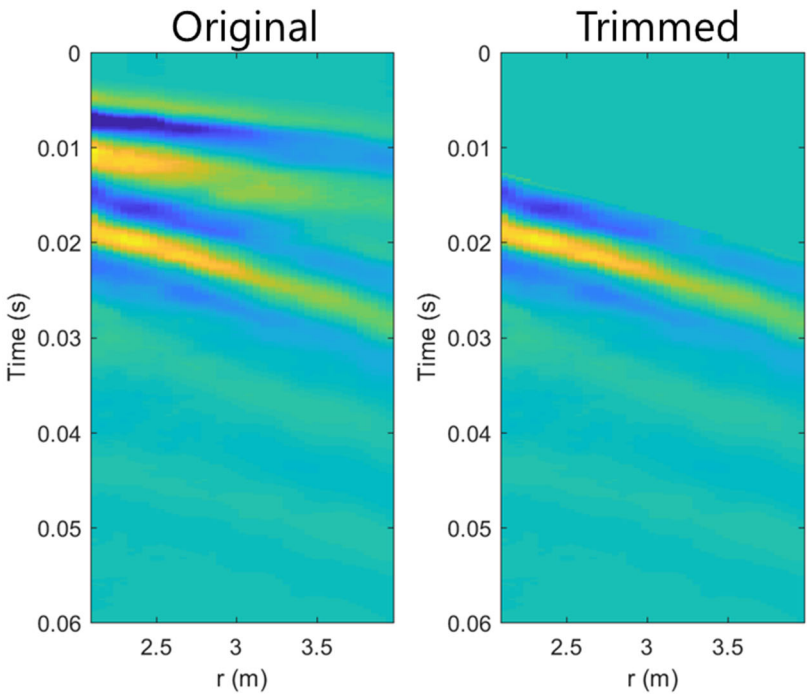

Figure 17: Left – original vertical data. Right – vertical component data with the P-wave removed.

In order to do dispersion analysis, we separate the P-wave from the Rayleigh wave, as it is the Rayleigh wave exhibiting dispersion. This is done simply by zeroing out a section of the data above the Rayleigh wave packet. The results are shown in Figure 17.

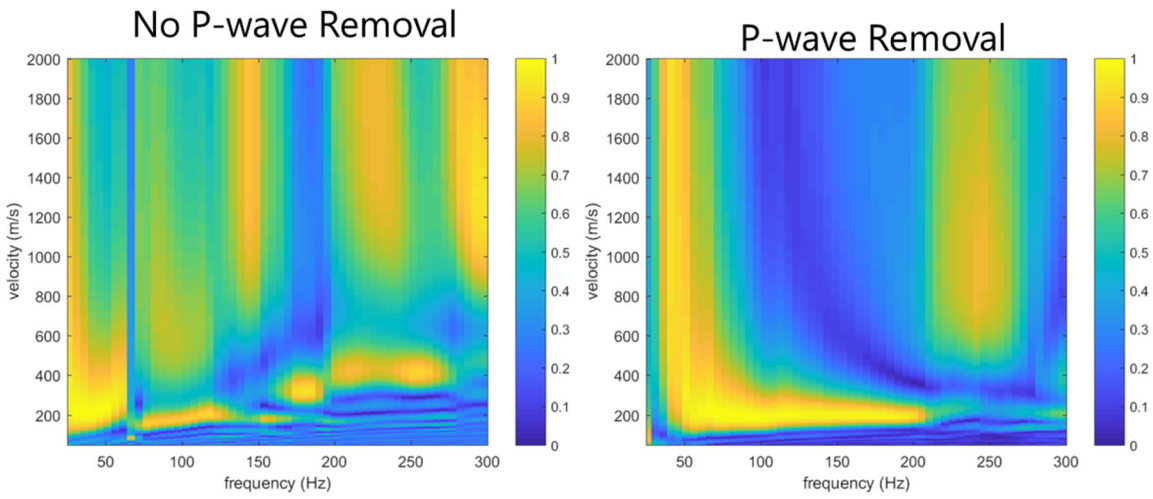

Figure 18: Dispersion images of the average vertical component. The left dispersion image is computed on the original image, while the right dispersion image is computed on the image with the P-wave removed.

Figure 18 shows the produced dispersion images with and without the P-wave removal. The images look completely different, and it appears the P-wave has a large impact on the resulting dispersion image. This is not typically the case in MASW approaches, as they use a much longer measurement section to determine the dispersion curve; this allows the P-wave to decay, and allows for finer estimates of the wave velocities.

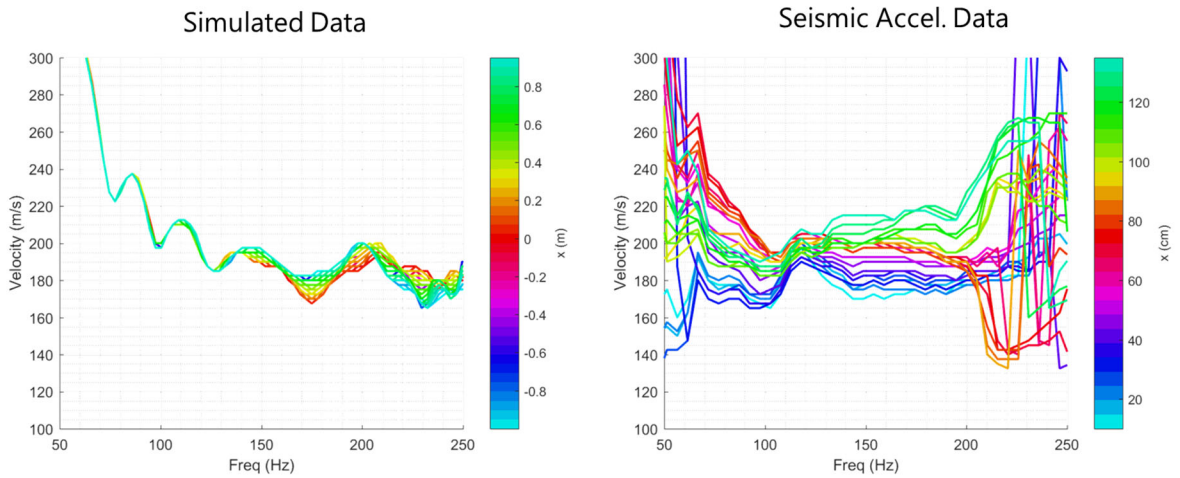

Figure 19: Dispersion curves generated as a function of the cross-track dimension for both simulated data and measured accelerometer data.

Figure 19 shows the dispersion curves for both simulated and experimental accelerometer data. The simulated data shows that the target (red curves) exhibits only subtle differences in the dispersion curve when the wave passes over the target. The real data varies wildly across the simulation domain, and the expected target response does not appear evident in this data. We believe this is because the target response is being mixed with many pixels which correspond to the background.

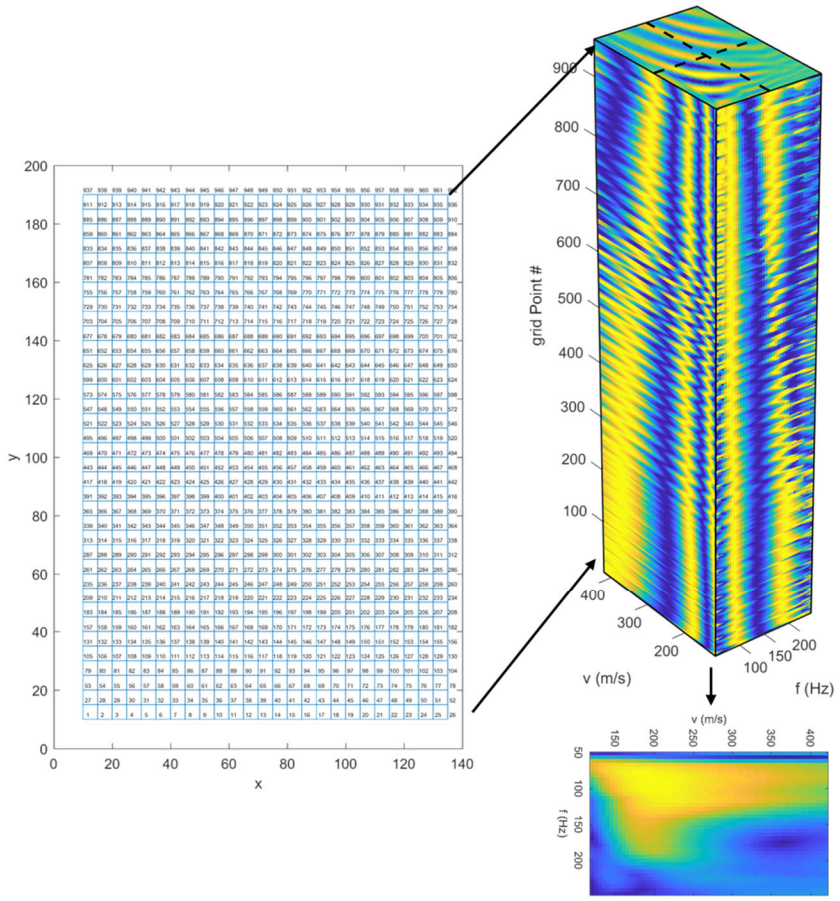

Figure 20: Illustration of the treatment of pixel-wise dispersion images which can be combined to form a single dispersion image.

Figure 20 shows a new way of thinking about the computation of the dispersion image. We can think of each pixel as contributing a complex valued image to the dispersion. Instead of selecting pixels from a given line to form different dispersion images, we can compare each pixel's dispersion image to the average dispersion experienced over the field.

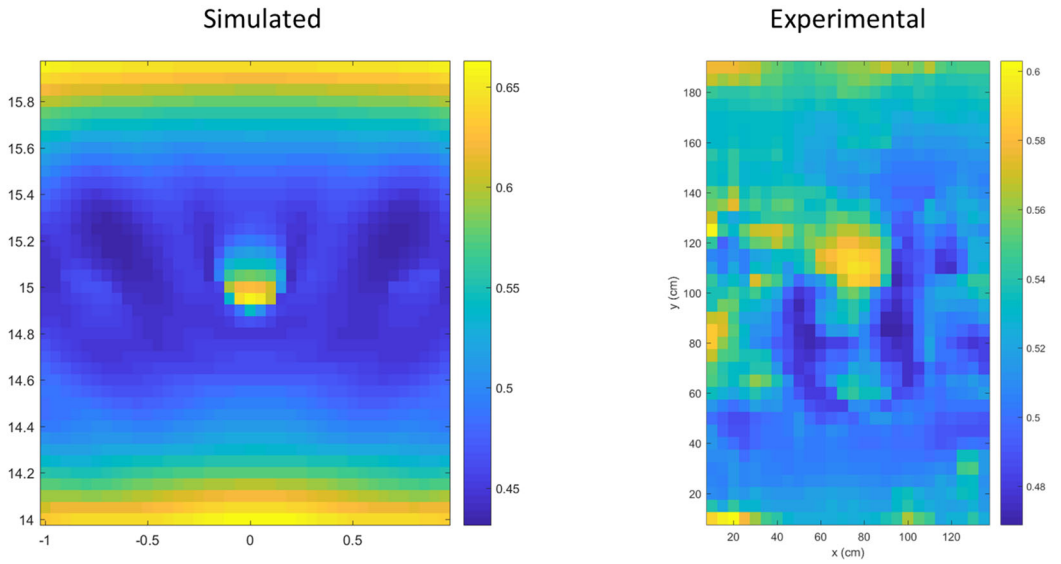

Figure 21: Dispersion irregularity maps for both simulated and experimental data.

Another way to think about the individual dispersion pixel maps is as a rasterized complex-valued vector. For each of these pixels, we compute the distance to the mean complex valued dispersion image. This results in the anomaly maps shown in Figure 21, which we call dispersion irregularity. This method is energy normalized, and discovers information generated from the phases only, one main benefit of which is that magnitudes of the dispersion irregularity are consistent across any domain – a very desirable property for any anomaly detection method. Notice the color bars in Figure 21 are the same for both images. It does appear that this method highlights the target, as there is a hotspot in the dispersion irregularity map in the center of the image (where the target is present). Given the limited amount of data at hand, it is difficult to tell if this method is truly detecting the anomaly caused by the target, or happened to get lucky in this instance. The large anomaly in the center of the image does appear to be about 10cm behind the target position, which does not appear to be the case in the simulated data. The dispersion irregularity method is promising since it relies upon the Rayleigh wave, which decays much less rapidly than the P-wave and can therefore be used for detection at greater ranges.

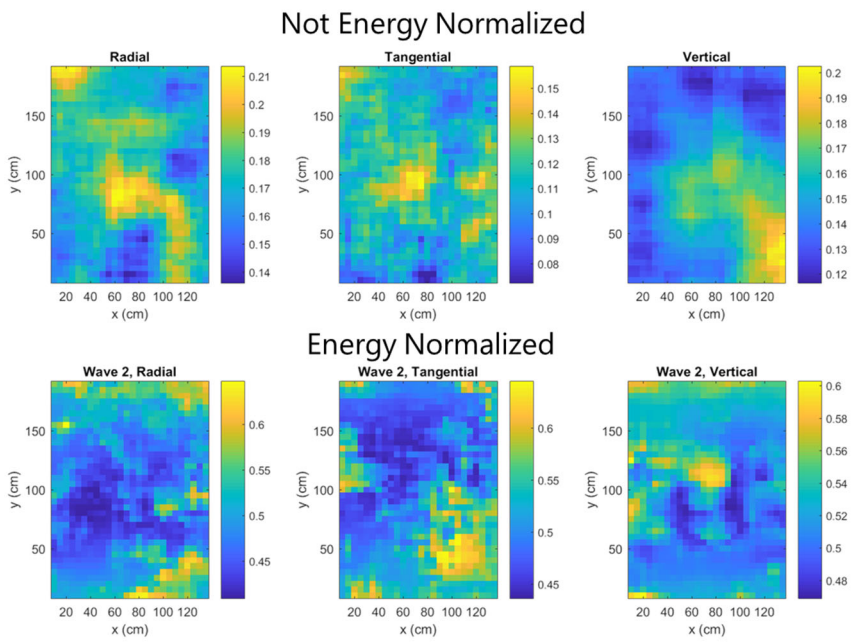

Figure 22: The dispersion irregularity method can be used with and without the energy normalization step. The energy normalization step is critical for the detection of the target dispersion in the vertical component. However, the target appears visible in the other components when the data is not energy normalized.

Figure 22 shows a surprising result; the dispersion irregularity algorithm appears to highlight the target only when the data is not energy normalized. We believe this is because the dispersion irregularity method without energy normalization is akin to a background subtraction technique and recovers similar results to those shown in Figure 14.

6 Seismic Detection Network

For the earlier part of this work, we were able to extract some physically meaningful features that appeared to do some form of anomaly detection. The main drawback to these approaches is that none of them were able to fuse information across the components of motion in a meaningful way. For this reason we decided to go in a completely different direction, and train a convolutional neural network (CNN) that fuses the components of motion directly at the input layer.

We have an interesting problem in that there is a scarcity of data at the target encounter level with only 32 simulations and a single field test; however, there is a wealth of data contained in each encounter, with 1000 time samples, 3 channels, and over 1000 measurement locations. Our approach to mitigate both problems at once is to train a network that operates on small patches at a single time slice. This converts each encounter to many samples while also significantly reducing the dimensionality of the input data.

The main problem with training a network in this case is that it is not convincing evidence that it works unless it can train on simulated data and test on the real accelerometer data, a typically challenging task for any detection problem.

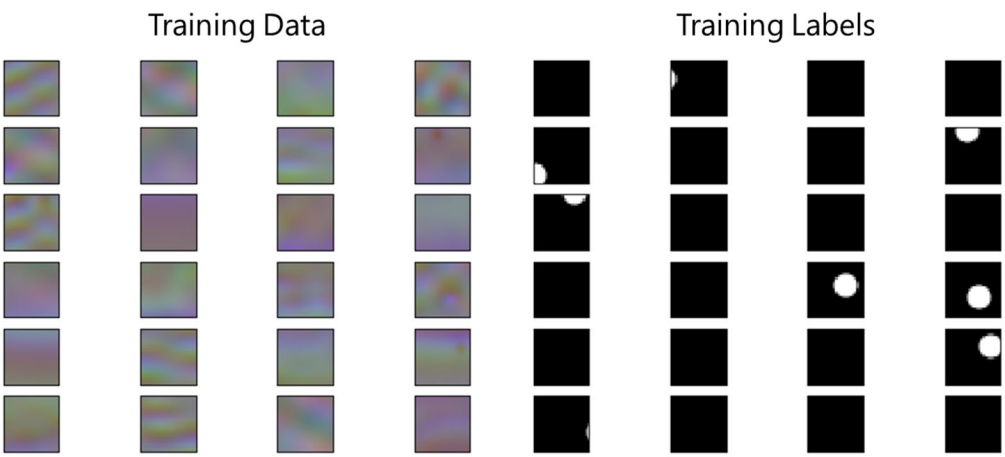

Figure 23: Left: 24x24 patches of the three-axis data viewed as RGB images. Right: labels on the example patches generated by putting a disk around the target location. The target appears in many locations due to randomly sampling patches from the simulated measurement domain.

Figure 23 shows a representation of the training data used for training. We select 24x24 patches at 5cm resolution. The training labels are images of the same size with 1's if the pixel is within 25cm of the target center, and 0's otherwise. Each patch is energy normalized. The visualization in Figure 23 shows the three-axis data as RGB, viewing acceleration directions as color channels; this makes for a simple way to visualize the patches, but the drawback is that it is difficult to parse the seismic meaning of the waves shown. As the acceleration can be positive or negative, these patches are shifted such that 0 is shifted to 0.5, meaning that a location with no acceleration will be gray, negative accelerations will be dark, and positive acceleration values will be bright. One last challenge is that the simulation data is so easy we were worried the network would learn to just look for energy. For this reason, we increase the difficulty of the problem by blending the target examples with their simulated background. We select how much background to add by choosing a uniform random number between 0 and 0.5. That way, the network sometimes sees very easy examples, and sometimes sees very difficult examples. This will also increase the variability of the training data and hopefully allow the network to learn generalizable features regardless of target difficulty.

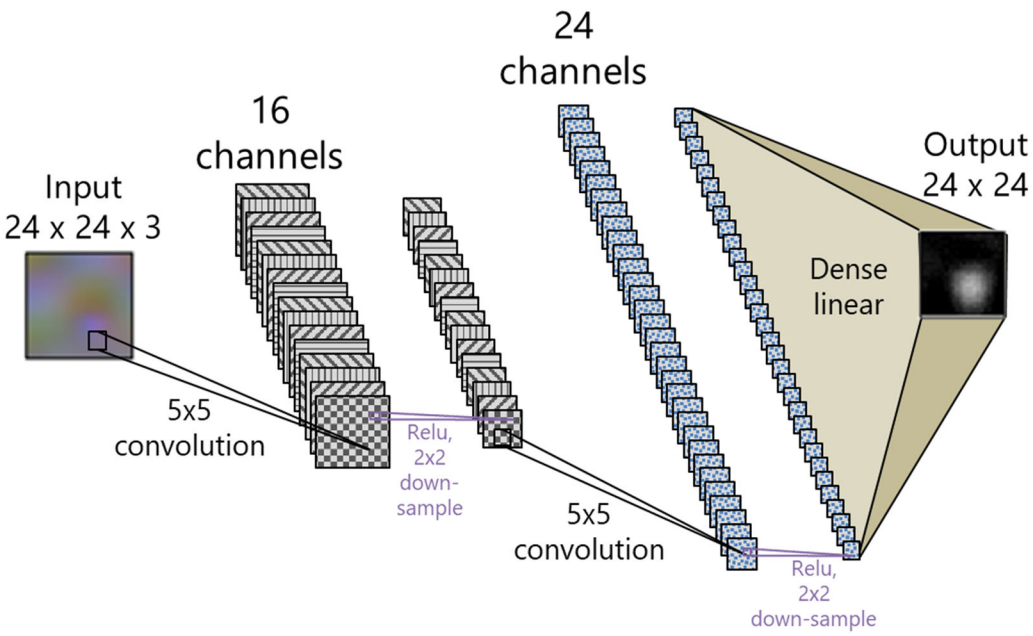

Figure 24: Graphic describing the architecture of the network. This is a simplified version of one found on https://pytorch.org/tutorials/beginner/blitz/neural_networks_tutorial.html used for handwritten digit identification.

Figure 24 shows a visual representation of the network architecture used for detection. This is a very small network with two convolutional layers and one dense layer to the output. The methodology behind this architecture was taking a basic example from the pytorch tutorial and modifying it to fit the problem. It is generally considered a bad plan to have any dense layers in the middle of a network, as it will very quickly lead to overtraining and poor generalization to unseen data. The dense layer was left at the end to produce the output mask simply because it was an easy way of generating the correct size output; if we had a chance to go back and change it, we would probably use convolutional transpose layers instead, which reduces the number of parameters in the system.

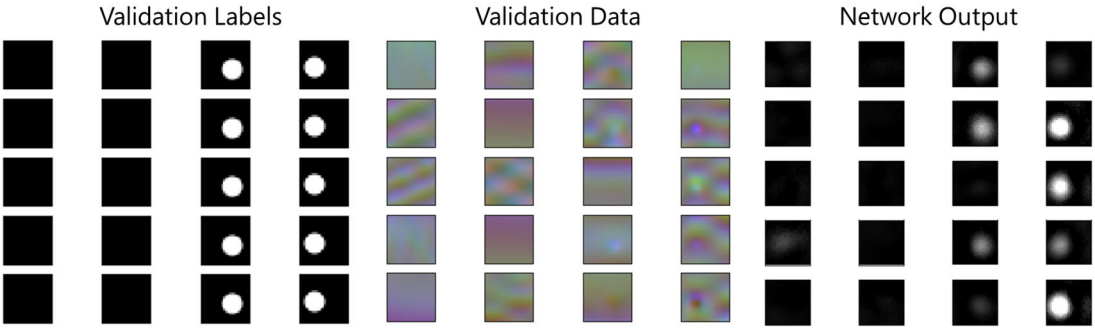

Figure 25: Example validation data shown with network outputs. Each column represents examples from each of 4 simulations held out from the training process.

Figure 25 shows some examples from 4 simulations held out from the training set. The network appears to do a good job of picking out the targets in the validation set when the target response is visible. We can see a few missed target

slices and a single weak false alarm. This data appears to suggest that the network is using reasonable features to detect the target and is unlikely to be too over-trained to the input data.

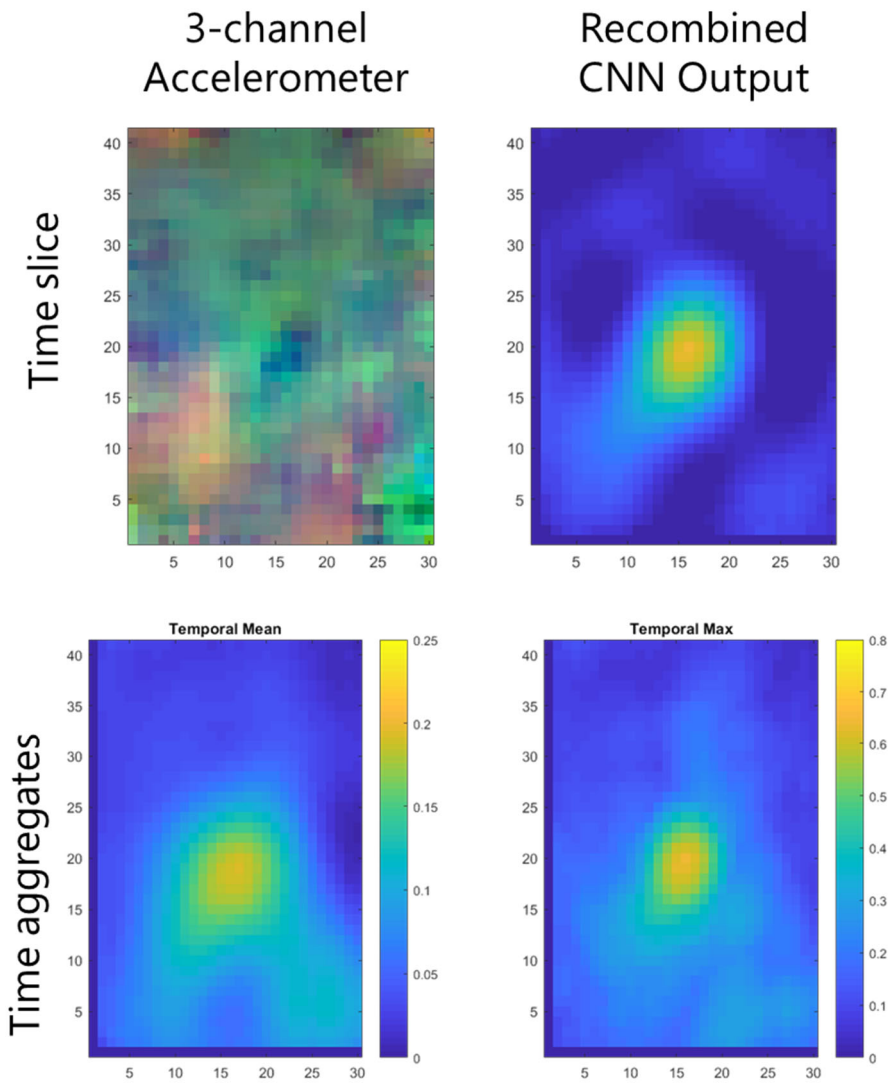

Figure 26: Network evaluated on experimental data. The top two plots show the network input and output from the time slice with maximum confidence, while the bottom two plots show time aggregates of the network confidence outputs.

In order to run the network on the experimental accelerometer data, we run the network on each 24x24 patch in the 31x40 input domain. As each patch generates a 24x24 output, we recombine the outputs at each slice by averaging the output result over all patches which saw that response. Figure 26 shows the output response of the network on the experimental accelerometer data. It appears the network can detect the target with reasonably high confidence and no significant false alarms present.

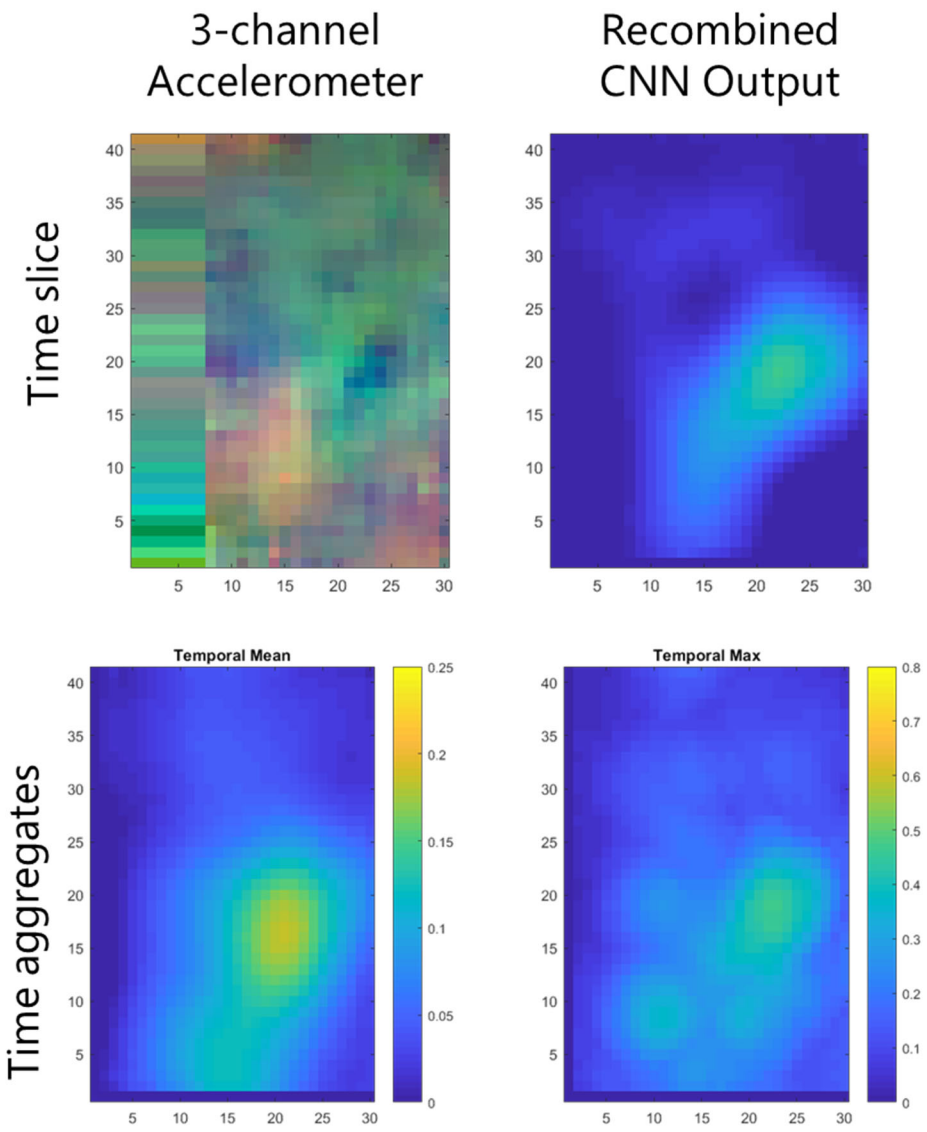

Figure 27: Same experiment as in Figure 26, but with the input data shifted to the right by 7 pixels.

A fair criticism of this algorithm would be that it is somehow learning where the 'center' of the field of view is and detecting the center instead of the target. In order to address these concerns, we shift the data to the right by as much as possible while still containing the target in the FOV. The data to the left is replicated from the nearest neighbor. Figure 27 shows by shifting the data that the target response can still be detected near the edge of the sensor domain. The confidence values have decreased slightly, although to be fair there is less background surrounding the target to help highlight the anomaly. This experiment convinces us that the network is detecting the features generated by the target rather than creating a location dependent response.

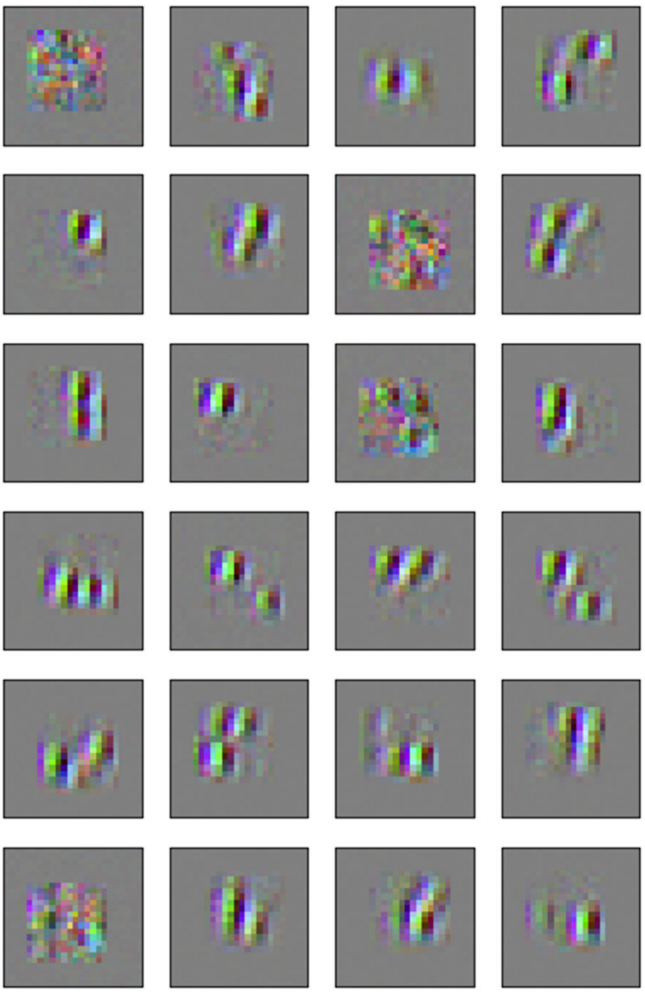

Figure 28: Input images which maximize the activations of neurons in the network.

One way to investigate the features that the network is cueing on is to generate images which maximize the activations of the neurons in the network. The neurons we activate are those prior to the dense linear layer at the output, of which there are 3x3x24. The 3x3 portions come from different convolutional regions and just correspond to a shift of these images around the domain. Figure 28 shows these maximum activation inputs which are generated through optimization of the inputs starting with random images.

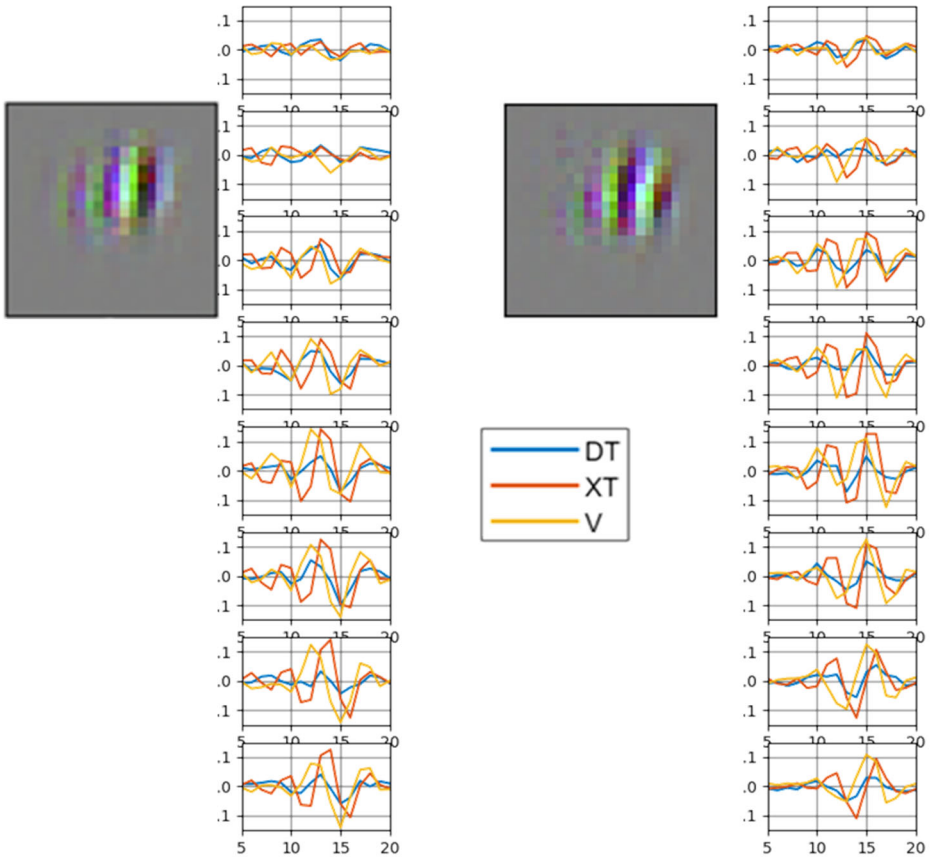

Figure 29: Horizontal line plots through some example maximization images.

Figure 29 takes some example chips output by a neuron maximization routine and plots the horizontal slices as line plots of the down-track (DT), cross-track (XT) and vertical (V) components. The reason for doing this is that the three channel images are confusing to interpret as seismic data. The line plots show that these input images represent XT & V components which are out of phase with one another, and a DT & V which are in phase. This exactly matches our expectations given the results on the simulated data shown in Figure 6. These neurons are excited by the patterns corresponding to the target response, and we expect that this will generalize well across varied scenes, sensors, standoff ranges, and possibly target types.

7 Conclusions

It appears that CNNs are an effective method for detecting target responses in seismic data and can even be trained using a small number of simulated encounters. This method will only become more powerful as more simulations become available and more data is collected.

The CNN appears to be identifying the cues generated by the targets in the simulated data as shown in Figure 6 and Figure 29. The targets all appear to have a strong vertical component which is out of phase with the cross-track component but in phase with the down-track component. This type of target signature appears to be the critical focus of this network which is able to identify the target signature in the experimental data.

It is surprising that the network can generalize across such differences as the training set and experimental data represent. More seismic data of difficult target encounters are needed to really prove out this approach, as even the convincing analysis of features identified by the network is not enough to show anything about false alarm rates on clutter in the field.

8 FUTURE WORK

The results shown in section 6 show very promising results for the future of seismic target detection. Many open questions remain, however. How well do these results generalize across many of the choices made during the network & training data generation? Will different network architectures reveal the same results? Will the same architecture trained with new random seeds or different held out simulations have similar results on the experimental data?

The frequency ranges exhibited in the simulated data (50-350Hz) are much greater than those produced by the sledgehammer (75-200Hz) in the experimental data. We have not taken this into account, and so it is surprising that even though the frequency ranges in simulation were significantly different, the network was still able to detect the target. What would happen if the simulated data were bandpass filtered to better match the excitation of the sledgehammer?

We would also like to see if this approach would work well on a single component of data. This could provide a new answer as to how many components are needed for the sensor to accurately identify targets, or if targets can be detected using old SAVI data (which contains only the down-track component).

We are also specifically interested in testing if the augmentation strategies used here, such as increasing the difficulty by blending with the background, is critical to success of the network. Training augmentation is an active field of research for other military problems as well and investigating these augmentation procedures may provide insights with wider generalizability across domains.

New simulation data is becoming available by ERDC which utilizes more soil variability in the spatial domain, which should produce simulations which are much more realistic than those we have been training on. It will be interesting to see if networks trained on more realistic scenarios with more background variation generates similar or better results.