Synthetic Aperture LADAR Automatic Target Recognizer Design and Performance Prediction via Geometric Properties of Targets

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

by

JACOB W. ROSS
B.S.C.S., Wright State University, 2013,
M.S., Wright State University, 2015

2022
Wright State University

Wright State University
GRADUATE SCHOOL

May 11, 2022

I HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER MY SUPERVISION BY Jacob W. Ross ENTITLED Synthetic Aperture LADAR Automatic Target Recognizer Design and Performance Prediction via Geometric Properties of Targets BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Doctor of Philosophy.

Michael Raymer, Ph.D.
Dissertation Director

Yong Pei, Ph.D.
Director, Computer Science and Engineering Ph.D. Program

Barry Milligan, Ph.D.
Vice Provost for Academic Affairs
Dean of the Graduate School

Committee on
Final Examination

Michael Raymer, Ph.D.

Krishnaprasad Thirunarayan, Ph.D.

Mateen Rizki, Ph.D.

Brian Rigling, Ph.D.

Frederick Garber, Ph.D.

Vincent Velten, Ph.D.

ABSTRACT

Ross, Jacob W. Ph.D., Department of Computer Science and Engineering, Wright State University, 2022. *Synthetic Aperture LADAR Automatic Target Recognizer Design and Performance Prediction via Geometric Properties of Targets.*

Synthetic Aperture LADAR (SAL) has several phenomenology differences from Synthetic Aperture RADAR (SAR) making it a promising candidate for automatic target recognition (ATR) purposes. The diffuse nature of SAL results in more pixels on target. Optical wavelengths offers centimeter class resolution with an aperture baseline that is 10,000 times smaller than an SAR baseline. While diffuse scattering and optical wavelengths have several advantages, there are also a number of challenges. The diffuse nature of SAL leads to a more pronounced speckle effect than in the SAR case. Optical wavelengths are more susceptible to atmospheric noise, leading to distortions in formed imagery. While these advantages and disadvantages are studied and understood in theory, they have yet to be put into practice. This dissertation aims to quantify the impact switching from specular SAR to diffuse SAL has on algorithm design. In addition, a methodology for performance prediction and template generation is proposed given the geometric and physical properties of CAD models. This methodology does not rely on forming images, and alleviates the computational burden of generating multiple speckle fields and redundant ray-tracing. This dissertation intends to show that the performance of template matching ATRs on SAL imagery can be accurately and rapidly estimated by analyzing the physical and geometric properties of CAD models.

Contents

List of Figures

List of Tables

Acknowledgment

I have had the honor and privilege to work with some amazing people during my graduate career. I would like to take this opportunity to thank the many people - far too many to list here - who have helped make this work possible.

First, I would like to thank my advisor, Dr. Raymer, for his guidance and mentorship. Dr. Raymer's enthusiasm for research is contagious, and his feedback and input were invaluable. I would like to thank my committee: Dr. Rigling, Dr. Velten, Dr. Garber, Dr. Prasad and Dr. Rizki for their time and feedback. Dr. Rigling had a key part in shaping this research; his feedback and suggestions added focus and direction for this project. Dr. Velten always found time to discuss research with me, and his decades of ATR experience helped make this project successful. Dr. Prasad's sage research advice is something I have used for years and will continue to do so. Dr. Garber always has constructive ideas and fascinating questions to turn good research into great research. Dr. Rizki's many years of experience of mentoring students and research is something I have benefited from since I was an undergraduate student, and I certainly continued to benefit as a PhD student.

This work would not have been possible without the support of many people at AFRL. This research started as a summer internship project in 2015, and unknown to me at the time, it would turn into my PhD topic. My mentors that summer were Mr. Rob Neuroth and Mr. Ed Zelnio. Rob got me started in the right direction and provided everything I needed to be successful. Rob also wrote an early version of the simulation tools used in this work. Ed's wisdom and knowledge about all things ATR has had and continues to have significant impact about how I view ATR and research in general. I would also like thank my branch, RYAT, for their support over the years. I would like to especially thank Ms. Clare Mikula and Dr. Olga Mendoza-Schrock for their leadership and for giving me the opportunity to work in the lab. While being a mentee myself, I was also given opportunities to be a mentor. I would like to thank Rose, Walker, and Brandon for being fantastic students and doing some great work.

I would like to thank Mr. Brett Keaffaber, Mr. Larry Barnes, Dr. Randy Depoy, Dr. Andrew Stokes, and Dr. Ed Watson for their help will all things LADAR and optics. They always had time to field my many questions and provide crucial feedback. I also owe thanks to Dr. Adam Nolan and Dr. Matthew Horvath with their help on all things performance prediction and the algorithms used in this dissertation. Additionally, I would like to thank the BiRG lab at Wright State for always being willing to lend an ear to my research problems, and for fruitful discussions on all things machine learning.

In addition to the support I got professionally and academically, I have been fortunate to have the support of many friends and family. Thank you, Mom and Dad, for your unwavering support. Without your love and encouragement I would not have made it this far into my graduate studies. Thank you, Grandma, for being such a positive influence in my life. I would like to thank Ginny, Joanna, and Travis for being the best big sisters and brother I could ask for. I would like to thank my in-laws, nieces, nephews, cousins, aunts, and uncles for always checking in on me and lending words of encouragement.

I would like to thank the many friends I have made during my academic career and elsewhere. If it was not for my computer science friends and study buddies: Paul, Brandon, Chris, Allison, and Gabby, I would not have continued into graduate school. I would like to thank one of my best friends, Lewis, not only for being a study buddy, but for always being willing to lend a helping hand and for his support over the years.

I owe a lot to "The Crew": Brittany, Tyler, and Matt for their many years of support and being the best friends I could ask for. I especially owe Matt a thank you, for not only being a great friend, but for always taking time to talk research.

I want to thank my friends and family at Abiding Christ Lutheran Church for their spiritual support and fellowship.

The person who was always there for me during the ups and downs of the PhD process was my wife, Juli. Her love and support kept me going in a way only she could do. She stuck with me through the highs and lows of this process, no matter what. Without her, this

dissertation would not have been possible.

Dedicated to

Juli

Introduction and Motivation

1.1 Automatic Target Recognition

Automatic Target Recogntion (ATR) is the act of automatically detecting and classifying targets of interest from collected sensor information [73]. ATR is a multi-discipline area and includes, but is not limited to, signal processing, image processing, artificial intelligence, statistics, and human performance. An ATR system goes through a multi-step process. These steps include and are not limited to:

- Sensing - collecting sensor data from a region of interest (RoI).
- Detection - select sub-regions that may contain targets of interest (ToI).
- Chipping - segregate the informative pixels from background and noise.
- Classification - declare the target as a certain type

An example of the ATR process is shown in Figure 1.1.

A variety of sensing modalities are leveraged in order to collect ATR data. Such modalities include Synthetic Aperture Radar (SAR), Synthetic Aperture Sonar (SAS), 3-D LADAR, Hyper Spectral Imaging (HSI), Wide Area Motion Imagery (WAMI), laser vibrometry, and infrared imagery. Examples of these modalities are shown in Figure 1.2.

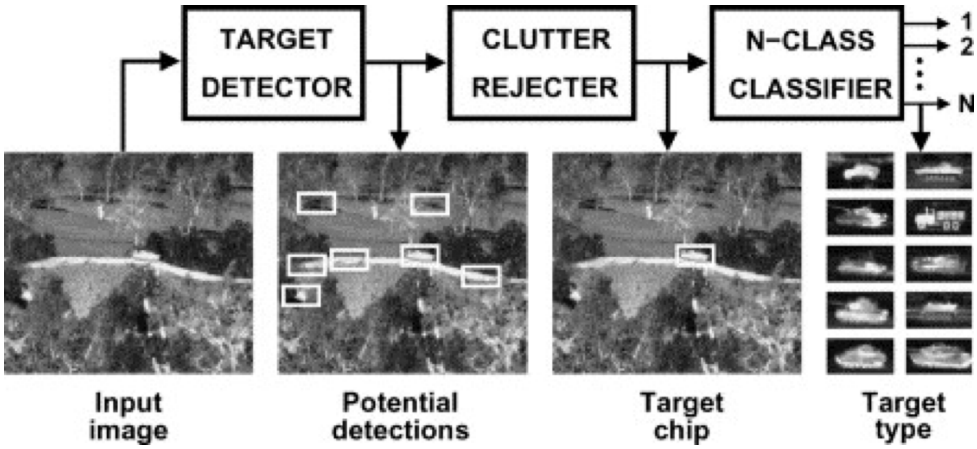


Figure 1.1: An example ATR Pipeline adapted from [15]. In this example, the collected sensor information is an image of a scene. Additional steps are taken such as clutter rejection.

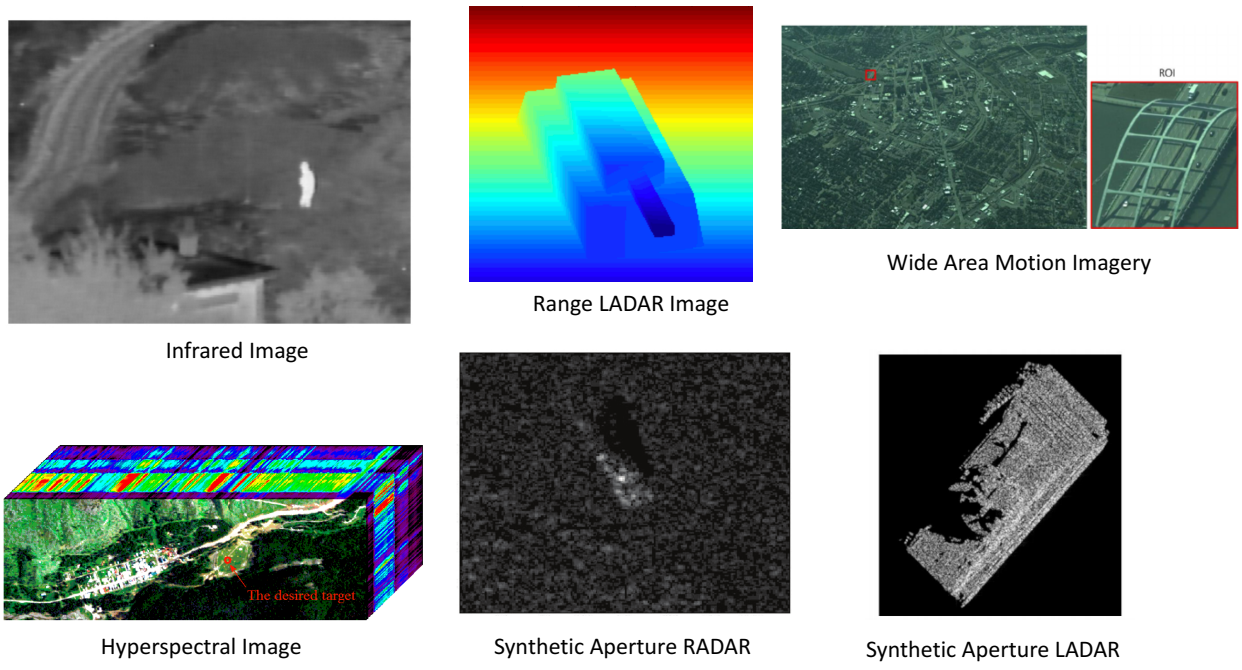


Figure 1.2: Examples of a variety of sensing modalities. Each modality has its own set of advantages and disadvantages, prevalent operating conditions, and use cases. Images from [23, 29, 25, 82, 2, 56] respectively.

Each modality has it's own set of strengths, weaknesses, and use cases. Each modality has a unique set of factors influencing the effectiveness of an ATR system. These factors are called operating conditions (OCs). The study of OCs and how they impact an ATR system are at the fore front of ATR research. OCs can be categorized into three main groups [73, 55]:

- Sensor OCs - factors that impact the the ability of a sensor to collect high quality data. Such OCs include sensor noise, phase errors, and motion compensation.
- Environment OCs - factors that modify the properties of the environment of both the sensor and target. Such OCs include weather, adversarial jammers, passive energy sources, clutter, foliage, and atmospheric noise.
- Target OCs - factors that alter the physical properties or targets of interest. Such OCs include camouflage netting, articulations, material properties, model number variations, decoys, and operating modes.

The OC space for any given sensor can be immeasurably large. Chapter 2 discusses strategies for overcoming this problem.

The modalities mentioned earlier in this section have a large body of knowledge discussing the sensor phenomenology, ATR algorithms, and sensor phenomenology. This dissertation focuses on Synthetic Aperture LADAR. For ATR purposes, SAL is relatively unexplored. As discussed in Chapter 2, the body of knowledge for SAL primarily consists of sensor design and phenomenology. The phenomenology and design are well studied and potential issues with designing an effective ATR on SAL data can be identified. Additionally, the state of the art in SAR ATR is discussed.

1.2 Towards Synthetic Aperture LADAR ATR

SAL is often viewed as the optical version of SAR [12, 37, 28]. A synthetic aperture in the radar domain allows the collection of imagery at resolution otherwise unobtainable by a physical aperture. The obtainable resolution in a SAR and SAL sensor is defined as:

$$res = \frac{\lambda}{2D} \tag{1.1}$$

where λ is the wavelength of the emitted pulse and D is the size of the aperture. In order to gain finer resolutions, the wavelength must shrink, or, the aperture size must increase. X-band SAR operates at $\lambda = .03m$. The Moving and Stationary Target Acquisition and Recognition (MSTAR) dataset was collected with X-band SAR, and $res = 1ft$ was achieved [2]. SAL has been shown to be feasible with $\lambda = 1.5\mu m$ [12, **?**]. In theory, a SAL sensor can collect imagery at a resolution that is $10,000$ times finer than a SAR sensor with the same aperture size.

SAR interactions with scatterers are more specular than SAL interactions. Longer wavelengths result in more mirror like behavior in scatterers. Small changes in aspect angle can result in vast differences in visual features of the targets. This specular nature is shown in Figure 1.3. The optical wavelengths of SAL result in more diffuse scattering. While diffuse scattering will result in more pixels on target in an image, SAL imagery is more susceptible to speckle phenomenology. Speckle patterns occur when scatterers are rough relative to the wavelength of the sensor. Speckle patterns can practically be modeled and studied as a random process [17]. In addition to speckle, SAL pulses are more susceptible to atmospheric noise than SAR pulses. Due to shorter wavelengths, small shifts in phase will cause distortions in formed SAL imagery [70, 18, 84].

This section discussed the differences in SAR and SAL phenomenology at a high level. The promising characteristics of SAL are finer resolutions with a small synthetic aperture and diffuse scattering. However, speckle and atmospheric noise present challenges

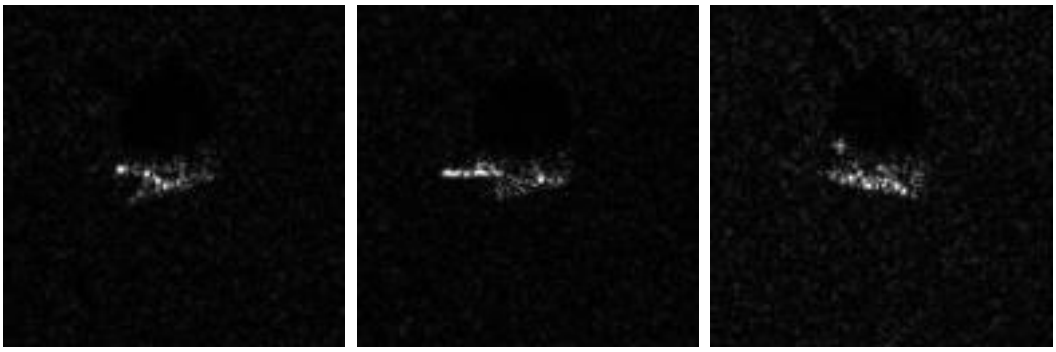


Figure 1.3: Three similar aspect views of the T-72 target from the MSTAR dataset. The middle image shows the turret part of the tank visible. The views before and after the middle view do not show the turret as visible.

for image quality. While the ideas discussed here are well studied, the implications in ATR design when switching from SAR to SAL have yet to be explored. The remainder of this document is organized as follows:

- Chapter 2 is a review of the state of the art in SAR ATR. Classification techniques, prediction performance ,operating conditions, and synthetic data strategies are discussed. A more through history and discussion of SAL phenomenology is presented.
- Chapter 3 presents the research gaps in the body of knowledge of SAL and ATR based on the review in Chapter 2. Target contributions and impact are listed.
- Chapter 4 presents the methods and results for a SAL ATR study as well as a proposed technique for SAL ATR performance prediction.
- Chapter 5 summarizes the results from Chapter 4 and discusses the effectiveness of the methods provided.

Literature Review

This chapter reviews the relevant body of work in SAR ATR and SAL phenomenology. In the SAR ATR literature, classification techniques, performance prediction methods, and synthetic data techniques are reviewed. Pertinent literature on SAL design, phenomenology, and synthetic data techniques are reviewed. In the following chapter, open research problems pertaining to SAL and ATR are identified.

2.1 Synthetic Aperture RADAR Automatic Target Recognition

The sections reviews classification techniques, performance prediction methods, and synthetic data techniques for SAR ATR.

2.1.1 Classification Techniques - Template Matching

The MSTAR data collection enables researches to study classification techniques on military vehicles. MSTAR was collected with X-Band SAR and 1 foot resolution. The original MSTAR release contained three targets: the T-72 tank, BMP2 infantry fighting vehicle, and BTR-70 armored personnel carrier. These targets are imaged in a full 360° sweep around the target at 15° and 17° elevation angles. Later, MSTAR was updated with additional tar-

Figure 2.1: Examples of MSTAR targets and photographs of the targets. MSTAR images contain target, background, and shadow information. Figure adapted from [85]

.

gets at varying articulations and clutter objects. MSTAR evaluations are commonly done on the ten-target dataset. Examples of the ten target dataset are shown in Figure 2.1.

With the release of MSTAR, the first publicly available classification techniques were developed. Alongside the MSTAR dataset release, a series of papers were written containing best practices and challenge problems for MSTAR evaluation as well as a baseline ATR [69, 67, 68, 55]. The baseline ATR presented in [69] is a template based classifier. Templates are constructed by forming a mean image of a target from a group of images contained within the same aspect bin. Then, each image is classified by finding the template that yields the lowest mean square error (MSE) on the test image. This MSE classifier also takes steps to shift the image around each template to correct against registration issues. Additionally, a mask is applied to the imagery in order to reduce the impact of background pixels. The images are power normalized to account for differences in relative power among different targets and look angles. This template matching algorithm was

a relatively simple approach compared to template matching techniques developed after MSTAR was released. The baseline accuracy achieved on the original 3 target dataset with 10 serial numbers was 90%. This simple MSE approach was extended and studied into a quantized form in [32].

The quantized grayscale matching (QGM) algorithm is presented in [36] and revisited in [32]. QGM is a quantization based template matching algorithm. QGM aims to exploit more features of images in addition to pixel intensity information. QGM exploits background and shadow information in addition to target information. The target, background, and shadow regions are quantized individually. Each template under the QGM scheme contains the empirical probabilities of each pixel realizing a certain quantile. Each image is classified as whichever class template yields the maximum log likelihood score summed across all pixels.

Multinomial Pattern Matching (MPM) algorithm is presented in [42]. The form of MPM presented in [42] is referred to as the scalar form, while the vector form is presented in [33]. Similar to QGM, each pixel in a template is the empirical probability of a pixel realizing certain quantile. After the probabilities are found, a penalty table is created. The penalty table contains the penalty of assigning a pixel in a test image a certain quantile. The penalties are based on the probability of a pixel realizing a certain quantile, as well as the amount of evidence used to find the probabilities. MPM also quantizes pixels in a fashion similar to the QGM approach. The MPM approach does not segregate pixels based on whether or not they are target, background, or shadow.

In addition to the steps each template matching algorithm takes, there are common preprocessing steps all template matching algorithms can take in order to improve performance:

- Pose estimation - SAR images will appear as rotated towards the sensor location. Pose estimation is a necessary step in template matching algorithms in order to rotate the pixels in to a consistent orientation. Additionally, template matching algo-

rithms may only compute similarity scores based on the template representative of the estimated pose. There are numerous pose estimation techniques validated on the MSTAR dataset that are accurate within 5° [87, 59, 78].

- Blurring and filtering - Noise manifests in SAR images due to sensor thermal noise, atmospheric noise, and varying background pixels. Noisy pixels can be suppressed via filters as well as other speckle reduction techniques [71].

2.1.2 Classification Techniques - Traditional Machine Learning

The previous section was an overview of template matching based ATRs on SAR data. Traditional machine learning approaches have also been leveraged to classify SAR images. Two popular approaches are support vector machines (SVMs) and deep neural networks (DNNs).

An SVM approach was taken in [10] on the MSTAR dataset. This approach does not bin templates based on pose unlike the template matching proposes discussed in the previous chapter. For 10° bins, this would result in 36 sub classifiers. The SVM approach treats each target as a class without the need to pose binning. On the 3 target MSTAR dataset, the approach in [10] achieved 92.8% accuracy compared to the 88.9% accuracy in the baseline MSE approach. The SVM approach in [80] achieved 97.7% accuracy on the 3 target MSTAR set. Additional SVM approaches also achieve similar accuracy scores [86, 79, 5]. SVMs have also been shown to be effective at rejecting out of library confusers [65, 74].

Neural networks are an effective choice for image recognition and computer vision problems. Neural networks are a promising approach to the general ATR problem as discussed in [63]. One of the benefits of an neural network based ATR is the explicit preprocessing other techniques rely on can be implicitly embedded in the network architecture. For SAR image recognition, convolutional neural networks (CNNs) are of particular inter-

est. A number of CNN architectures have been shown to be highly accurate on the MSTAR dataset, often reporting nearly 99% accuracy [19, 60, 54].

The SAR classification techniques presented here utilize the MSTAR dataset as a gold standard. Publicly available and physically measured SAR data is not widely available, and MSTAR has been the standard for over 20 years. While the data remains stagnant, classification techniques have evolved. The different schools of SAR image classification are all effective at classifying MSTAR images, given the users train on the sequestered train set and test on the sequestered test set. However, as will be discussed in Section 2.1.4, there are several issues with the MSTAR dataset which makes the classification task trivial. Additionally, MSTAR only encompasses a relatively small OC space. While later editions of MSTAR include articulations and other confusers, it is only a fraction of the grander OC space that can be engaged in the SAR ATR realm. The effectiveness of an ATR is tied to it's robustness towards OCs. In order to study a vast OC space, synthetic data and performance prediction techniques are used when real data is not accessible. The following section reviews techniques for performance prediction of SAR ATRs. Section 2.1.4 reviews best practices and research trends when utilizing synthetic imagery while studying ATR performance.

2.1.3 Performance Prediction and Operating Conditions

Performance Prediction is a sub-area of ATR in which the performance of ATRs are studied without explicit train and test phases on imagery. Performance prediction techniques alleviate the data burden that would otherwise be incurred when large OC spaces much be investigated. The idea of extended operating conditions (EOCs) were first introduced in [40]. An EOC is an OC that was not anticipated by the ATR designer. ATR algorithms can be trained or have preprocessing techniques in order to alleviate the negative affects of OCs. However, given the large OC space most sensors operate in, not all OCs and OC levels can be tested. EOCs pertinent to the MSTAR data collect were discussed in [55] and

[67].

Performance prediction can be divided into multiple approaches. For SAR ATR, there are three main approaches:

- **Synthetic Empirical Studies** - the analysis of ATR performance utilizing synthetic data. While computationally intensive, the advent of high performance computers (HPCs), GPUs, and cloud processing enable ATR studies on large amounts of synthetic imagery in large OC spaces. Examples of empirical studies can be seen in [48, 45, 53]

- **Geometric Studies** - SAR returns are highly dependent on the physical characteristics of the target. The material properties, aspect angle, scatterer size, scatterer orientation, and other aspects play a role in estimating the radar cross section (RCS) of a target. Geometric studies allow the prediction of performance by studying the elements of a target that contribute to a return, without forming the images directly. While this offers more rapid analysis than the empirical approach, the specular nature of SAR makes it difficult to study from a pure geometric standpoint. Examples of geometric studies and SAR performance prediction can be seen in [58, 57].

- **Probabilistic Studies** - This performance prediction methodology delves into the statistical makeup of algorithms and data in order to model classification performance under a variety of OCs. This type of approach aims to provide closed form models that predict target separability given algorithm parameters and OCs as input. These studies are often validated with empirical methods and a blend of synthetic and real data. An example of this approach on SAR template matching algorithms is given in [32, 33].

While each of the methodologies listed have their own schools of thought, they can all utilize synthetic data as a form of validation. Synthetic data alleviates the need to collect real data at all of the OCs of interest to an ATR system. While synthetic data is cheaper

and easier to obtain, synthetic image simulations can still be computationally burdensome. Additionally, simulations may contain imaging artifacts that will not manifest in a real data collect or fail to artifacts common to real data. The following section discuses synthetic data and how it is leveraged in SAR ATR studies. Challenges are identified, and some of the benefits of using synthetic data in tandem with MSTAR are discussed.

2.1.4 Synthetic Data and SAR ATR

Synthetic data enables ATR research given a wide swath of OCs. For large targets, the physical optical (PO) approaches involve shooting-and-bouncing rays (SBR) in order to estimate the radar cross section (RCS) of targets [61, 62, 31]. Compared to other approaches such as method of moments (MoM) and finite difference time domain (FDTD), the PO and SBR method is less computationally intensive, but also less accurate.

Given the reliance on synthetic data, there are a number of approaches used to make synthetic data more realistic. Additionally, there are approaches to make classification algorithms robust to training on synthetic data and testing on real data. Generative Adversarial Networks (GANs) have been used to transfer synthetic data to the real domain [47, 30, 14]. Ideally, pairing synthetic imagery with real imagery removes image artifacts introduced via the simulation code. Increased realism in synthetic data allows the ATR researcher to draw conclusions in the synthetic domain that are applicable to the real domain.

Another way to leverage synthetic imagery is to train algorithms on synthetic data and test on collected real data. The Synthetic and Measured Paired Labeled Experiment (SAMPLE) dataset provides a dataset to address this challenge problem [48]. Specifically, the challenge is to train on synthetic images of the MSTAR targets and test on the collected real imagery from the MSTAR collection. A baseline deep learning approach for addressing the problem is given in [72]. In addition to the baseline results, the authors address a key problem with the MSTAR data collect. Classification algorithms will perform well on the MSTAR dataset even when target pixels are completely masked. This means the

background and clutter pixels are correlated between the provided training and test sets. The challenge problem presented in [48] ensures the synthetic train set's clutter pixels do not correlate with the background. This is an example of incorporating synthetic data in an ATR study to alleviate potential issues with collected imagery.

2.2 Synthetic Aperture LADAR (SAL)

This section reviews the the design and phenomenology of SAL. First, the history of SAL from its conception to modern applications are discussed. Next, the phenomenology and challenges for potential SAL ATRs are identified and discussed.

2.2.1 SAL Design

The cross-range resolution of a SAL and SAR sensor is defined as:

$$res = \frac{\lambda}{2D} \tag{2.1}$$

where λ is the wavelength of the propagated signal, and D is the length of the synthetic aperture. Switching from RADAR to LADAR shrinks the wavelength by five orders of magnitude. The idea of utilizing a synthetic aperture with optical wavelengths was discussed in [49]. In this work, a small scale SAL was developed in the lab and imaged primitive targets. The theory behind SAL and physical sensors were further developed in the 1980's [4, 3, 44]. The first operational SAL was developed in [52] . Later, A 2-D SAL capable of imaging large military vehicles was designed in [28]. SAL sensor design and theory continued to be developed into the 2000's [51, 39, 12, 43].

A common theme in the design and application literature for SAL is to compare the capabilities of SAL to SAR. In theory, a SAL sensor can collect imagery at a resolution $10,000$ times finer than a SAR sensor with an equivalent synthetic aperture baseline. How-

ever with optical wavelengths come with a set of challenges. These advantages and disadvantages are summarized in [51]: *" A synthetic aperture ladar (SAL) could provide dramatic improvements in either resolution or, compared to synthetic aperture radar (SAR), the time needed to record an image, or both. The reduced imaging time results from the shorter time needed by the platform to traverse the synthetic aperture (SA) that produces the same resolution with a shorter wavelength. When the observation range reaches a thousand kilometers or more, no other method of imaging can offer centimeter-class resolution with a real aperture size no larger than a few meters...high-resolution SAL imaging from orbit is possible, but much more work needs to be done on this topic, because the atmosphere can degrade beam quality substantially at visible and infrared wavelengths."* - Lucke et al., "Synthetic Aperture LADAR: Fundamental Theory, Design Equations for a Satellite System, and Laboratory Demonstration" [51].

In the following section discusses the phenomenology of SAL and how it differs from SAR. Specifically, the aspects of SAL phenomenology that pose a challenge to image quality are addressed. Additionally, techniques for mitigating detrimental effects are discussed.

2.2.2 SAL Phenomenology

When a surface is rough compared to the wavelength of the energy interacting with a scatterer, the scatterer is diffuse. When the surface is smooth relative to the wavelength, the scatterer is specular. A diffuse surface can be described as matte-like and will reflect light in equal directions. A specular surface can be described as mirror-like, and light will reflect off the scatterer in a uniform direction. An example of the differences between a diffuse and specular surface is shown in Figure 2.2.

A key component of modeling specular returns is to take multi-bounce effects into account. SAR simulators model multi-bounce effects and can add to computation time based on the desired number of bounces required [31, 62]. For diffuse scattering exhibited

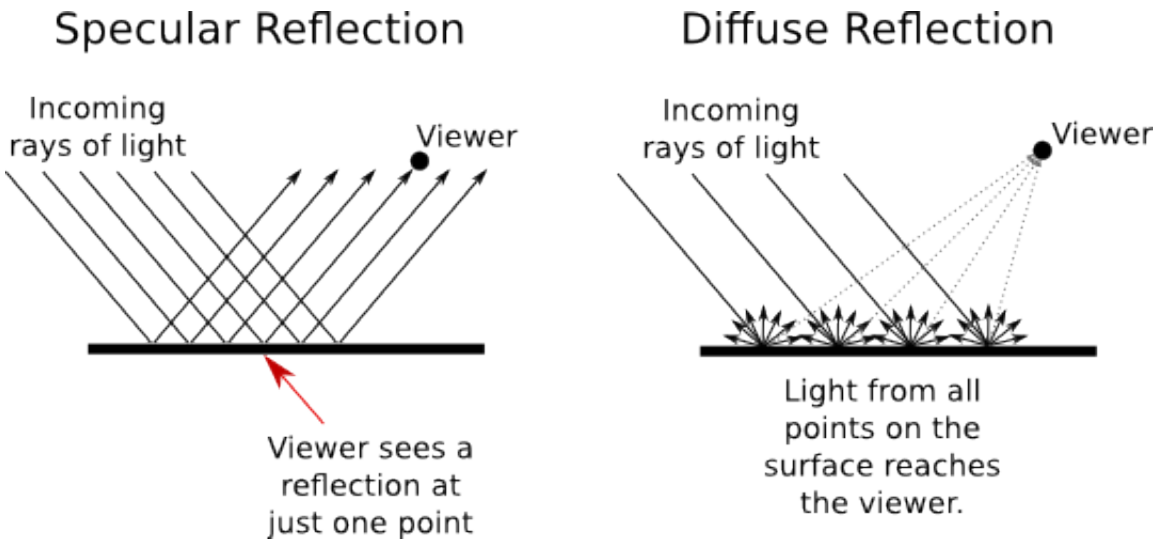


Figure 2.2: A description of diffuse versus specular scattering. Figure is adapted from [22]

in SAL, single bounce returns can be assumed [56]. The diffuse nature of SAL scattering also leads to more pixels on target since energy is more likely reflected back to the receiver, regardless of sensor position, than in the specular case. While more pixels on target is desirable for ATR purposes, the rough surfaces introduce a speckle phenomena. Speckle is the random perturbation in phase and amplitude of a received signal. Speckle manifests as a noise in a formed imagery. The statistical properties of speckle are thoroughly discussed in [26] and [17]. Examples of SAL images with speckle are shown in Chapter 4.

Removing speckle from imagery decreases the variance across pixels. There are two main approaches when removing speckle: speckle averaging and preprocessing techniques. Speckle averaging involves collecting n images of a target from nearly the same sensor orientation. The sensor must move enough to collect de-correlated speckle patterns. Speckle patterns start to de-correlate when the emitter and receiver move by $D/2$ [26]. For a SAL baseline, $D/2$ can is on the order of centimeters. The intensities of the n collected images are averaged together, resulting in a less noisy image. Examples of speckle averaging are shown in Chapter 4. Examples of de-noising images via speckle averaging are discussed in [9, 83, 8]. Processing techniques apply filters to speckled images to alleviate the effect. The benefit of these techniques over speckle averaging is multiple images are not needed.

In addition to traditional blurring techniques such as Gaussian blurring, other processing techniques are explored in [77, 6, 46, 16].

Atmospheric effects are more pronounced at optical wavelengths. Atmospheric turbulence causes beam break up as a wavefront propagates through the atmosphere. This beam break up can be combated by increasing the dwell time of the illuminated scene from the sensor [11]. A synthetic aperture effectively increases dwell time since multiple views of the target are collected in approximately the same location. While a synthetic aperture addresses the beam break-up problem, atmospheric noise can still cause distortions in SAL imagery. In simulation, atmospheres can be modeled by ray-tracing through phase screens [64]. Phase screens model the various layers of the atmosphere. The profiles of the atmosphere can be instantiated via a variety of models such as Hufnagel-Valley [35, 81] and Bufton wind screens [13]. The effects of the atmosphere on SAL image quality are discussed in [38, 7, 75]. Mitigating atmospheric effects in the image domain is essentially a focusing problem. Techniques for mitigating atmospheric effects in SAL imagery are discussed in [34, 18]. Atmospheric effects on imagery can be seen in Chapter 4.

Contributions

The previous chapter reviews classification techniques in SAR ATR, performance prediction, SAR synthetic data, and SAL research. As mentioned in Chapter 2, SAL has several promising characteristics that make it a suitable candidate for ATR research.

However, SAL has yet to be explored through the ATR lens. While the phenomenology and imaging challenges for SAL are well studied and documented, how they would impact ATR performances is unknown. Additionally, performance prediction techniques for SAL ATR have not been developed. The following contribution help fill in the gap between ATR and SAL:

1. **Verification of how the phenomenology of SAL impacts template matching parameters and performance.** The performance on SAR on template matching has been studies for decades, starting with initial empirical experiments and culminating in analytical performance modeling. Template matching algorithms are ideal for performance prediction studies. The training and testing mechanisms of these algorithms are defined by governing equations, thus, are considered clear-box algorithms. Additionally, as discussed in Chapter 2, these algorithms are competitive with other statistical machine learning methods such as SVMs and neural networks. As the scattering of targets shifts from specular to diffuse, different OCs come into play, and their interaction with algorithm parameters are unknown. An ATR study is proposed to quantify the differences between SAR and SAL in terms of template matching performance. Additionally, the impact phase errors and speckle averaging

has on parameters choices and performance are identified.

2. **A methodology for geometry based ATR performance prediction by analyzing physical properties of CAD models.** Barring speckle, the diffuse nature of SAL scattering gives a more direct representation of the physical make-up of the target. We propose a method for predicting the performance of SAL template matching algorithms by only analyzing the angle and material information of interrogated targets. In order to conduct empirical performance prediction studies with synthetic data, millions of images must be formed in order to ensure random draws of speckle as well as covering the entire OC space. Additionally, image formation with a SAL simulator requires ray-tracing to an imaging grid at multiple aperture locations. In order to motivate this CAD based performance prediction, the correlation between similarity trends in the image and CAD domain are shown. Next, ATR experiments are conducted entirely in the CAD domain and compared to the ATR results found in the image domain. The trends in performance, confusion, and classification in the CAD domain are similar to the trends in the SAL image domain.

3. **A methodology for generating SAL templates directly from CAD models without image generation.** Templates formed in the CAD domain are useful and give an accurate assessment of performance in the image domain. This dissertation also demonstrates that CAD based templates are suitable substitutes for image based templates when classifying SAL images. The performance of training on CAD based templates and testing on SAL images will yield similar results to traditional template formation. In addition to overall performance, the confusion and classification tendencies when training on CAD based templates and testing on imagery are similar to the tendencies found entirely in the image domain.

The following chapter fully implements contribution 1 previously mentioned. Evidence for the feasibility of contributions 2 and 3 are provided in the form of correlation

studies among the image domain and CAD domain. Then, ATR experiments are carried out to demonstrate the effectiveness of CAD based performance prediction. Chapter 5 summarizes the results and proposes future work.

Methods

4.1 Template Matching on SAL Images

This section describes how the shift from specular to diffuse returns impacts parameter choices for the quantized template matching algorithms Multinomial Pattern Matching (MPM) and Quantized Mean Square Error (QMSE). First, training and testing procedures are described for both algorithms. Second, all experiment parameters are listed. These parameters include targets, algorithm parameters, and operating conditions. Algorithm parameters are swept and and performance is analyzed for both SAR and SAL images under low-levels of imaging OCs. Next, the effects of quadratic phase error (QPE) and speckle averaging have on algorithm performance for SAL imagery are analyzed. Finally, factor interaction plots to show the significant interactions among operating conditions and algorithm parameters.

4.1.1 Image Quantization

Image quantization normalizes images with varying amplitudes into nQ discreet bins. Lower numbered quantiles contain the relatively dim pixels, while higher numbered quantiles contain the brighter pixels. The quantization process described here can also be found in [32, 42]. Image quantization consists of the following steps:

1. The quantile boundaries are determined by setting a threshold value (τ) such that any

pixel in the input image I below $minPixel = \frac{max(I)}{10^{7/20}}$ is set to 1.

2. The quantile boundaries $\boldsymbol{qBoundaries}$ are determined by finding the linear spacing in nQ steps between the $minPixel$ and $maxPixel = max(I)$.

3. For all pixels p in $I(p)$, create a quantized image $Iq(p)$ such that for all quantization levels (q)

$$\begin{cases} IQ(p) = 1 & I(p) < \boldsymbol{qBoundaries}(1) \\ IQ(p) = q & I(p) \geq \boldsymbol{qBounaries}(q-1) \quad \&\& \quad I(p) < \boldsymbol{qBounaries}(q) \\ IQ(p) = nQ & I(p) > \boldsymbol{qBoundaries}(nQ-1) \quad \&\& \quad I(p) \leq \boldsymbol{qBoundaries}(nQ) \end{cases}$$

The quantized image IQ now has pixel values between and including 1 and nQ. Examples of quantized images are shown in Figure 4.1.

4.1.2 Quantized Mean Square Error (QMSE)

The implementation of QMSE used in the following experiments is adapted from [32]. For a given target T and pose bin $\boldsymbol{\phi}_{bin}$ a QMSE template is the mean image of the quantized images in the template $\boldsymbol{QMSE}\boldsymbol{\mathcal{T}}_{T,\phi_{bin}}$.To form a template of a given target T and pose bin $\boldsymbol{\phi}_{bin}$:

$$\boldsymbol{QMSE}\boldsymbol{\mathcal{T}}(p)_{T,\boldsymbol{\phi}_{bin}} = \frac{1}{S} \sum_{\forall \phi \in \boldsymbol{\phi}_{bin}} \sum_{p=1}^{n} IQ_{T,\phi}(p) \tag{4.1}$$

where S is the total number of images used to form the template, p is the pth pixel of IQ, and n is the number of pixels in IQ. For QMSE templates, a template has the same size as the images used to form the template. For example, if IQ are all $[50 \times 50]$ pixels, then $\mathcal{T}_{QMSE}$ will also be $[50 \times 50]$ pixels. The pose bin vectors $\boldsymbol{\phi}_{bin}$ are determined prior

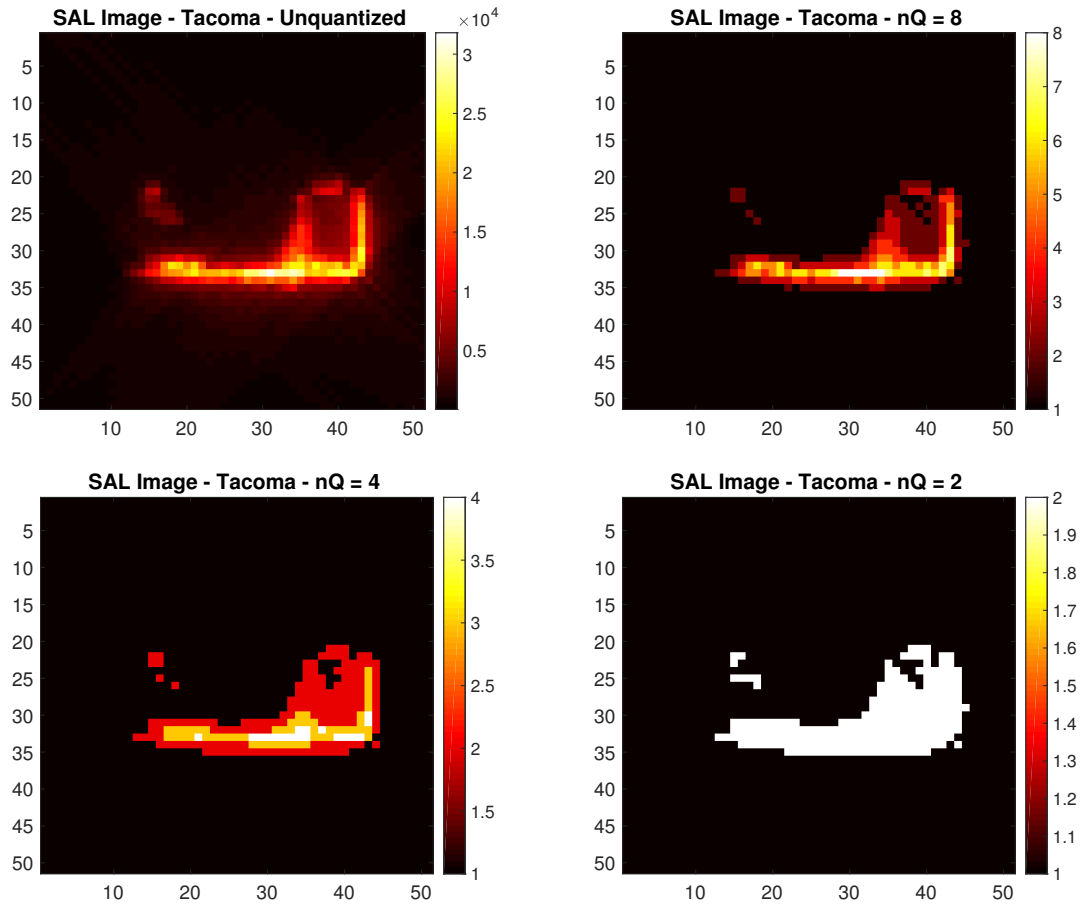


Figure 4.1: Examples of quantized SAL images.

to training. The following experiments are conducted with linearly spaced $\boldsymbol{\phi}_{bin}$ structures and have no over lapping aspect angles. For example, a full $360°$ collection of a target with $1°$ aspect increments and a desired $\boldsymbol{\phi}_{bin}$ size of $10°$, then $\boldsymbol{\phi}_{bin(1)} = [0°, 1° \ldots 9°]$, $\boldsymbol{\phi}_{bin(2)} = [10°, 11° \ldots 19°] \ldots \boldsymbol{\phi}_{bin(36)} = [350°, 351° \ldots 359°]$.

A test image is quantized to the same number of levels as the training templates. Each test image's pose is estimated and compared to all of the templates containing the estimated pose. The declared class is the template yielding the lowest MSE score:

$$c_{dec} = \underset{c \in C}{\arg\min} \left[\frac{1}{n} \sum_{p=1}^{n} (IQ(p) - \boldsymbol{QMSE}\boldsymbol{\mathcal{T}}(p,q)_{c,\phi_{est}})^2 \right] \tag{4.2}$$

where C is the collection all of the in-library class labels, n is the number of pixels in

the test image and template, and ϕ_{est} is the estimated target pose.

4.1.3 Multinomial Pattern Matching (MPM)

The implementation of MPM used in the following experiments is adapted from [42]. An MPM template $_{\boldsymbol{MPM}}\boldsymbol{\mathcal{T}}(p,q)_{T,\boldsymbol{\phi}_{bin}}$ is a table of penalty scores for assigning a pixel (p) a certain quantile (q). The penalty table is formed by the following steps as originally described in [42]:

1. For all quantized images in a template, find the empirical probabilities of a pixel realizing each quantile. The collection of images for each template is converted to a row-wise concatenation of multinomial random variables:

$$\boldsymbol{P}_{T,\boldsymbol{\phi}_{bin}} = \left[\frac{\boldsymbol{p}(1:nQ,1)}{N}, \frac{\boldsymbol{p}(1:nQ,2)}{N} \cdots \frac{\boldsymbol{p}(1:nQ,n)}{N}\right] \tag{4.3}$$

   where N is the number of images in the template, n is the number of pixels per image in the template, and $\boldsymbol{p}(p,q)$ is number of occurrences of pixel p realizing quantile q. Each sub vector $\boldsymbol{p}$ is an $[1 \times nQ]$ column vector. After concatenation, $\boldsymbol{p}(p,q)_{T,\boldsymbol{\phi}_{bin}}$ is an $[n \times nQ]$ matrix of probabilities.

2. In order to avoid divide by zero issues in later steps, a hedged version of $\boldsymbol{p}_{T,\boldsymbol{\phi}_{bin}}$ is created:

$$\boldsymbol{P}_{hedged}(q,p)_{T,\boldsymbol{\phi}_{bin}} = \frac{(N \times \boldsymbol{P}(q,p)_{T,\boldsymbol{\phi}_{bin}}) + v}{N + (nQ \times v)} \tag{4.4}$$

   where v is a small constant. For our experiments $v = 0.01$. The v constant also hedges against inaccurate probability estimates due to small sample size.

3. The quadratic penalty table is formed by:

$$_{\boldsymbol{MPM}}\boldsymbol{\mathcal{T}}(p,q)_{T,\boldsymbol{\phi}_{bin}} = \frac{(1-\boldsymbol{P}(p,q))^2 - \boldsymbol{\mu}(p)}{\boldsymbol{\sigma}(p)} \tag{4.5}$$

where

$$\boldsymbol{\mu}(p) = \sum_{q=1}^{nQ} \boldsymbol{P}_{hedged}(p,q)(1 - \boldsymbol{P}(p,q))^2 \tag{4.6}$$

and

$$\boldsymbol{\sigma}(p)^2 = \sum_{q=1}^{nQ} \boldsymbol{P}_{hedged}(p,q)(1 - \boldsymbol{P}(p,q))^4 - \boldsymbol{\mu}(p)^2 \tag{4.7}$$

The declared class of a test image is the penalty table that yields that lowest penalty score. For all quantized pixels in IQ find:

$$c_{dec} = \arg\min_{c \in C} \sum_{p=1}^{n} \boldsymbol{MPM}\boldsymbol{\mathcal{T}}(p, IQ(p))_{c,\boldsymbol{\phi}_{bin}} \tag{4.8}$$

Additional implementations of MPM are discussed in [33]. The scalar implementation described in this section and in [42] is used this work.

4.1.4 Experimental Setup

Experiments are conducted in two phases. Performance of both QMSE and MPM are given as a function of algorithm parameters (nQ, $\boldsymbol{\phi}_{bin}$) and pose error ϕ_{err}. First, baseline results are given on both SAR and SAL images on benign operating conditions (OCs). Second, results of SAL images on additional levels of quadratic phase error (QPE) and speckle averaging (SA) are given. Imaging parameters, operating conditions, and algorithm parameters are shown in Tables 4.1, 4.2, and 4.3.

For all experiments, the Avalon, Sentra, Tacoma, Camry, and Mazda MPV from the AFRL CV Data Domes are selected as targets of interest [21]. The Avalon, Sentra, and Camry models are all sedans with similar mostly convex features. The Tacoma model contains both convex and concave features. The bed of the Tacoma truck contains dihedral and trihedral features leading to multi-bounce features in the SAR imagery. The Mazda

Imaging Parameters	Description
$\lambda_{SAL} = 1.5\mu m$	The wavelength of the SAL signal propagated during ray-tracing. This wavelength was selected based on the experimental setups of discussed in previous SAL literature [28, 12, 51, 50]
$\lambda_{SAR} = 0.3m$	The wavelength of the SAR signals propagated during ray-tracing. This wavelength was selected based on the sensor parameters from the MSTAR data collect [69, 2].
$wr = wx = 10m$	The range (wr) and cross-range (xr) extent of the imaged scene. The level of $10m$ is large enough to encompass all of the selected targets.
$\delta_r = \delta_{wr} = 0.2m$	The range and cross-range resolution of the collected phase histories. The level of $\delta = 0.2m$ results in distinguishable features of the targets and results in images that are $[51 \times 51]$ in size, allowing manageable image sizes and computation sizes. A resolution of $\delta = 0.2m$ is theoretically easily achievable by a SAL sensor [12] and is $.1m$ finer resolution than the SAR MSTAR collection [2].
$imageGridScale = 0.7$	As discussed in [27], the imaging grid during used in back-projection should have smaller scaing than the desired resolution. A scaling of 0.7 alleviates any aliasing effects during image formation.
$el_{train} = 17^\circ$	The elevation angle of the sensor when forming the training data. This aligns with the elevation angle used in the MSTAR data collect.
$el_{test} = 15^\circ$	The elevation angle of the sensor forming the testing data.
$\boldsymbol{az} = [0^\circ, 2^\circ, \ldots 358^\circ]$	The center azimuth aperture positions for both the training and testing imagery. Each target is imaged from a full 360° view in 2° steps.

Table 4.1: Imaging parameters for MPM and QMSE experiments.

MPV model is mostly convex in features, but larger in size than the sedan models.

The following section presents the baseline results for MPM and QMSE for both SAR and SAL imagery. In order to directly quantify the differences in performance on MPM and QMSE between specular SAR returns and diffuse SAL returns, imagery is formed under benign imaging operating conditions. Performance is measured as a function of pose error and over a sweep of algorithm parameters. The results of each parameter is the mean F1-score of 10 independent runs.

Algorithm Parameters	Description
$nQ = [2, 4, 8, 16]$	The quantization levels swept for both MPM and QMSE. Images formed with the varying nQ levels are shown in Figure 4.1
$size(\boldsymbol{\phi}_{bin}) = [10°, 20°, 40°]$	The size of the pose bins per template. Larger bin sizes results in a smaller number of templates over all. For instance when $size(\boldsymbol{\phi}_{bin}) = 10°$ results in 36 templates per target. A bin size of $size(\boldsymbol{\phi}_{bin}) = 40°$ results in 9 templates per target.
$nDups = [1, 3, 5]$	The number of times an image at the same aspect angle in included in a template with a different draw of noise.

Table 4.2: Experimental algorithm parameters for MPM and QMSE.

Operating Conditions	Description
$\boldsymbol{\phi}_{err} = [0°, \pm 2°, \pm 4°, \pm 6°, \pm 8°, \pm 10°]$	The bounds of random pose error applied to images. Pose errors cause the image to be misaligned with the common facing of the templates.
$SA = [1, 2, 5, 20, 40, 80]$	Number of images used for speckle averaging on SAL images. The effects of speckle averaging are shown in figure
$QPE = [\frac{\pi}{2}, \pi, 2\pi]$	The amount of quadratic phase error (QPE) applied to SAL images. QPE manifests from atmospheric turbulence. Examples of images with varying levels of QPE are shown in Figure 4.11

Table 4.3: Extended operating conditions (OCs) applied to test imagery when evaluating MPM and QMSE.

4.1.5 Results - SAR and SAL Baseline

Figures 4.2 shows the baseline performance of the QMSE classifiers as a function of pose error. For both modalities, $nQ = 2$ and yields the highest F1-scores when pose error is low $size(\boldsymbol{\phi}_{bin}) = 10°$. In the SAR case, increasing the bin size when pose error increases improves performance over the smaller bin sizes. When the bin size is low and pose error is high, the probability of selecting a bin that does not contain the true pose of the image increases. For specular SAR imagery, neighboring templates may not contain similar features. In the SAL domain, diffuse returns increases the similarity across neighboring templates. An example of these scenarios are shown in Figure 4.6. Since there is no blurring effect $QPE = 0$, the locations of the pixels are not compromised. The true shapes of

the targets are maintained, thus, extra intensity information from higher nQ are not needed to classify the target. For relatively low noise levels and $SA = 80$ in the SAL domain, the expected variance for each possible illumination angle is minimized for $nQ = 2$ while higher levels of nQ results in higher variance. This behavior is shown in Figure 4.10.

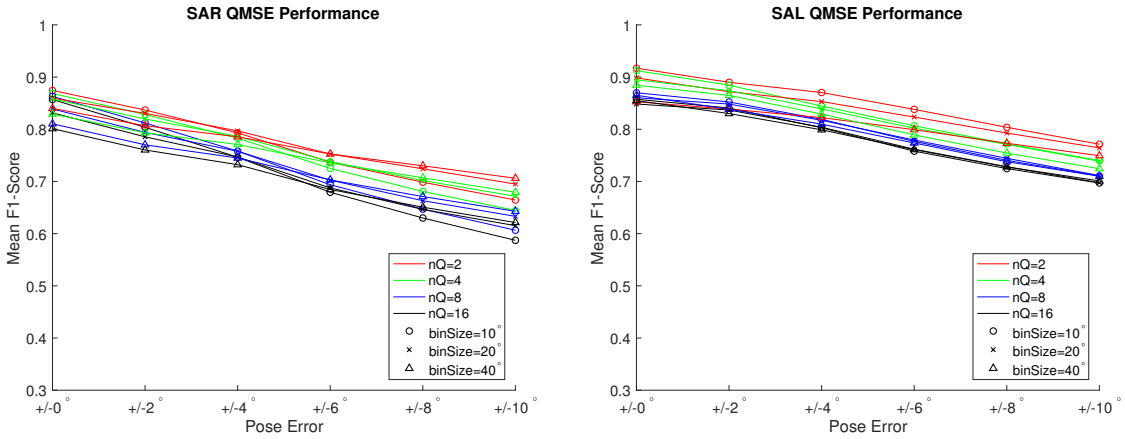


Figure 4.2: The performance (F1-Score) of the QMSE classifier on SAL and SAR imagery. Results are shown as a function of the pose error bounds images could experience. The best performing algorithm parameters for both modalities are $nQ = 2$ and $size(\boldsymbol{\phi}_{bin}) = 10°$ for small amounts of pose error. In the SAR case, wider pose bins yields a higher F1-Score when pose errors are more severe.

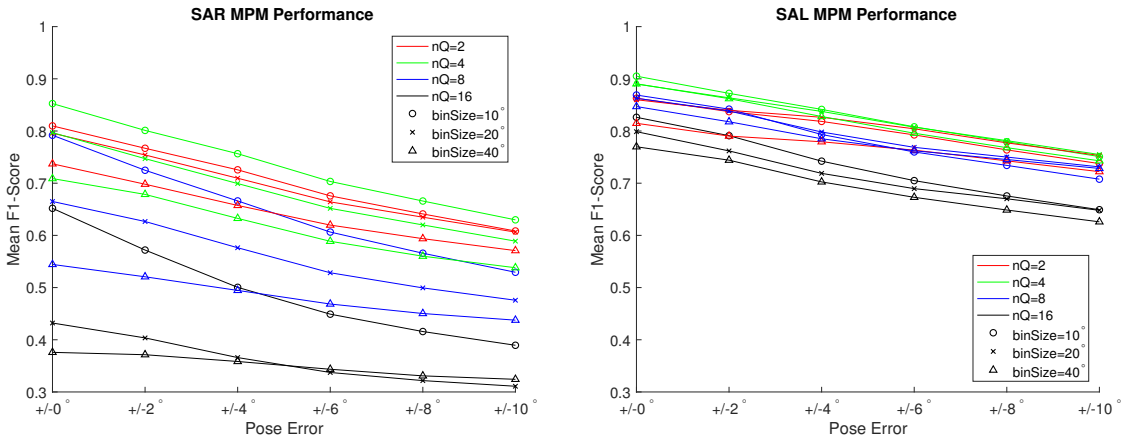


Figure 4.3: The baseline performance (F1-Score) of the MPM classifier on SAL and SAR imagery. Results are shown as a function of the bounds of pose error images could experience. The best performing parameters for all pose errors and modalities are $nQ = 4$ and $size(\boldsymbol{\phi}_{bin}) = 10°$. Contrary to the optimal QMSE parameters, MPM prefers extra intensity information for the pixels.

Figure 4.3 shows the baseline performance of the MPM classifier as a function of pose error. While the $nQ = 2$ parameter was optimal for QMSE, MPM yields the best performance when $nQ = 4$. For QMSE, the pixels in a template are ordinal. A raw distance is computed between the template pixels and a test image. The maximum error any pixel-to-template comparison can realize in the QMSE case is when the test pixel is $Iq(p) = nQ$ and the template pixel $_{\boldsymbol{QMSE}}\boldsymbol{\mathcal{T}}(p) = 1$ or vice versa. Since the lowest nQ level corresponds to off target pixels, QMSE realizes the maximum penalty when a target pixel overlaps a background pixel. For MPM, pixels in a template nominal. Penalties are calculated by estimating the likelihood a test pixel is from the corresponding multinomial distribution. The maximum penalty for an MPM comparison is not necessarily when an on target features overlaps a background pixel. The parameter $nQ = 4$ is more robust to potential information loss when pixels are quantized. The binary nature of $nQ = 2$ may decimate some intensity information that more nQ levels will capture. The penalty behavior and information loss for MPM quantization levels are shown in Figures 4.5 and 4.4. For the SAR MPM case, increasing pose bin size has a detrimental effect on performance. In the SAL case, the pose bin options converge as pose error gets worse.

This section demonstrated the performance of SAR and SAL imagery on the QMSE and MPM and identified key differences in algorithm parameters per modality. The following sections evaluate the MPM and QMSE algorithms on SAL imagery as a function of key operating conditions as discussed in Section 4.2.1. First, the effects of speckle averaging of SAL images on algorithm performance is evaluated. The effects of speckle averaging and algorithm parameters for both QMSE and MPM are discussed. Second, the effects of quadratic phase error on algorithm parameters are discussed. The speckle averaging and QPE OCs are evaluated on other benign OCs as presented in the previous section. Section 4.1.8 discusses the most significant factors impacting algorithm performance for both modalities. Additionally, significant interactions among OCs and algorithm parameters are

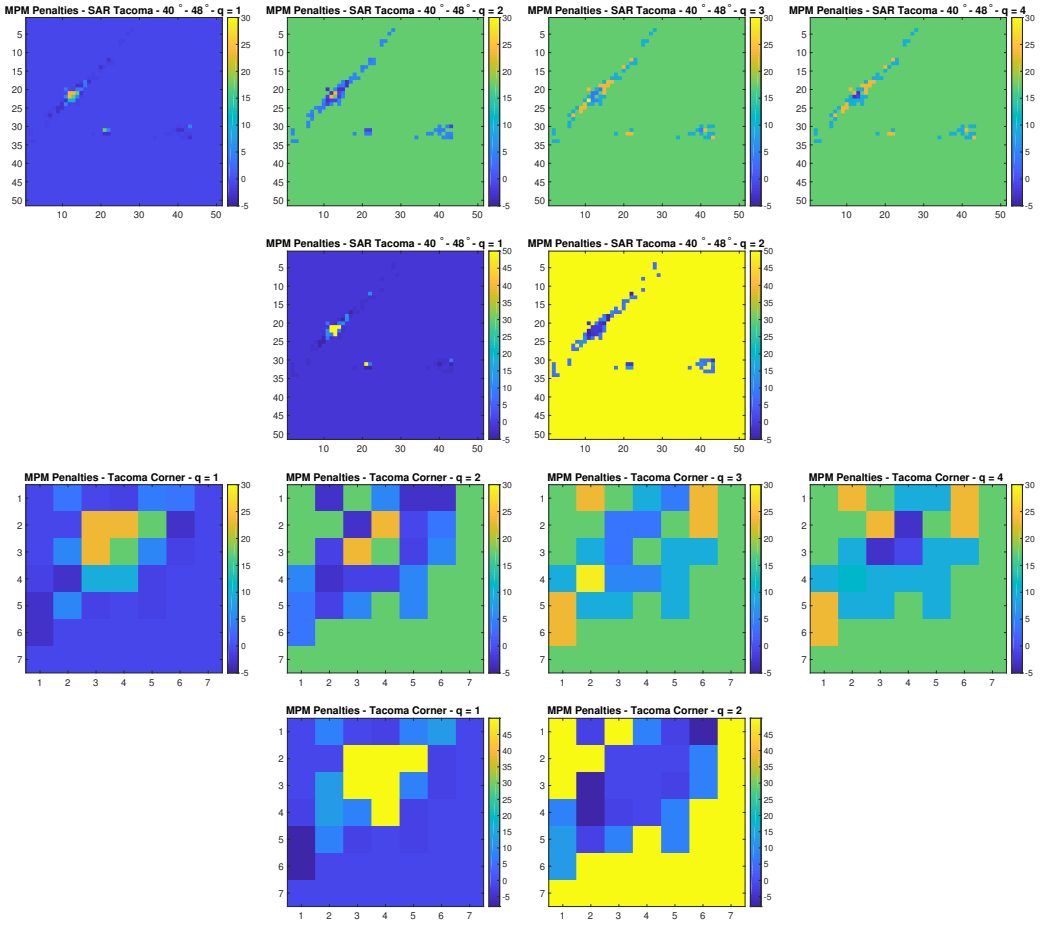


Figure 4.4: A visual representation of the trained penalty tables for the MPM algorithm. For the $nQ = 4$ case, the penalties for a given pixel do not necessarily shift in a ordinal fashion. Additionally, the maximum penalties do not occur when an on target pixel is confused for a background pixel. The $nQ = 2$ case masks nature of the penalties in the $nQ = 4$ case. Additionally, the maximum penalty for a penalty occurs in the $q = 2$ case when on on target pixel is labeled as the background. The first two rows of images contain the entire image of the target. The last two rows of images are the penalties of the pixels in the corner of the target where multi-bounce features will occur.

discussed.

4.1.6 Results - SAL and Speckle Averaging

This section discusses the performance of the QMSE and MPM algorithms on SAL imagery as a function of speckle averaging. As identified in Chapter 2, speckle noise is a key operating condition to investigate for diffuse returns. One methodology for combating

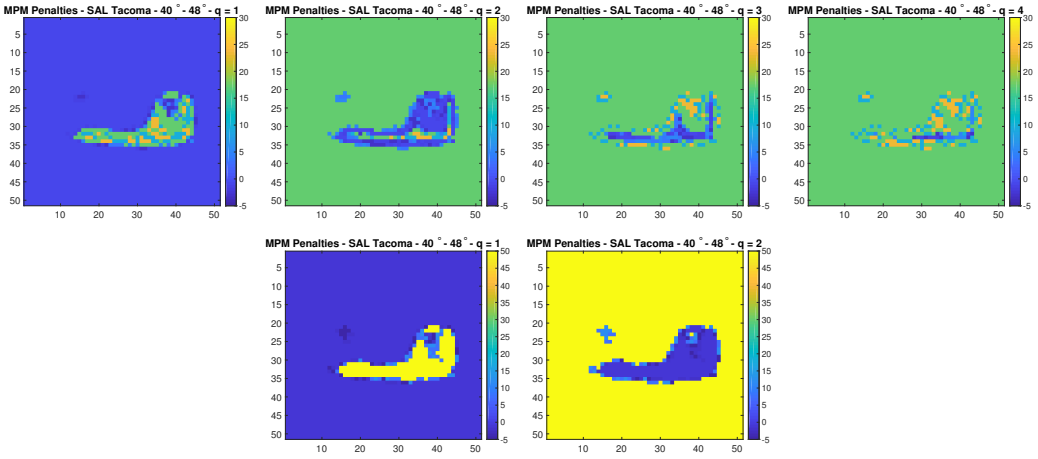


Figure 4.5: Example MPM penalties for the SAL case. Similar to the SAR case, the penalties for the $nQ = 4$ case do not necessarily shift in a linear fashion. The maximum penalties for pixels do not arise when a target pixel is confused with the background pixels. The $nQ = 2$ case masks information similar to the SAR case. While $nQ = 2$ reduces pixel to pixel variance, information about the intensity of the pixels is lost.

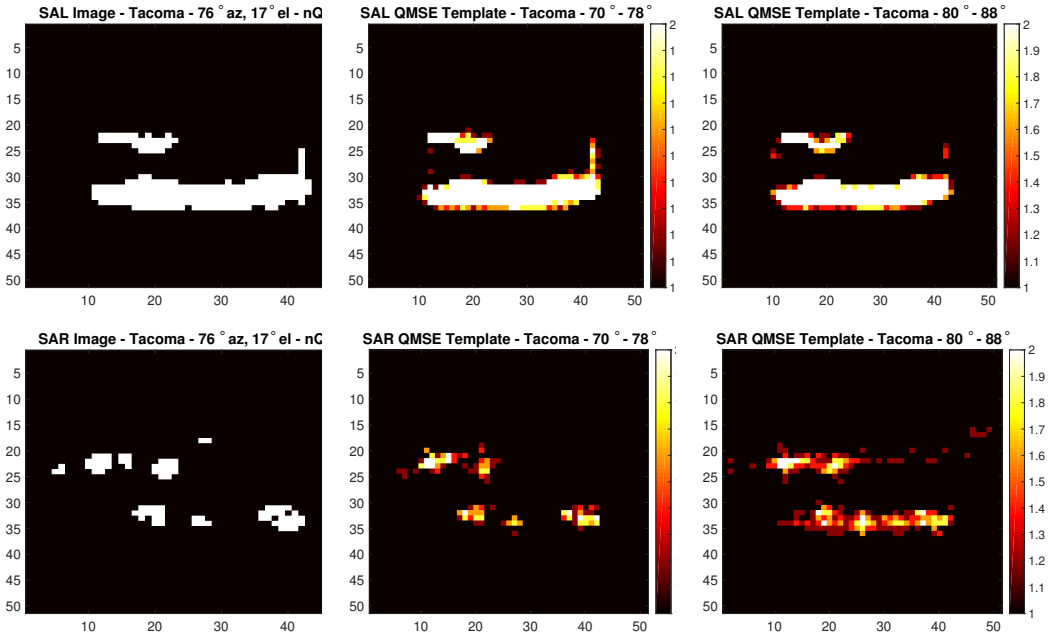


Figure 4.6: Examples of SAL and SAR QMSE templates and a test image. In the SAL case, the features across two neighboring templates contains similar features. Based on the results shown in Figure 4.2 neighboring SAL templates can serve as a proxy when pose errors cause the wrong template to be selected. In the SAR case, neighboring templates can have drastically different features.

speckle is speckle averaging. Speckle averaging is the process of taking n_{SA} images with uncorrelated speckle and finding the mean value of those pixels to form a less noisy image. The effects of speckle and speckle averaging are shown in Figure 4.7. The most significant gain in performance occurs from the increase from $n_{sa} = 1$ to $n_{sa} = 5$. The affect of increasing n_{sa} levels is shown in Figure 4.14a.

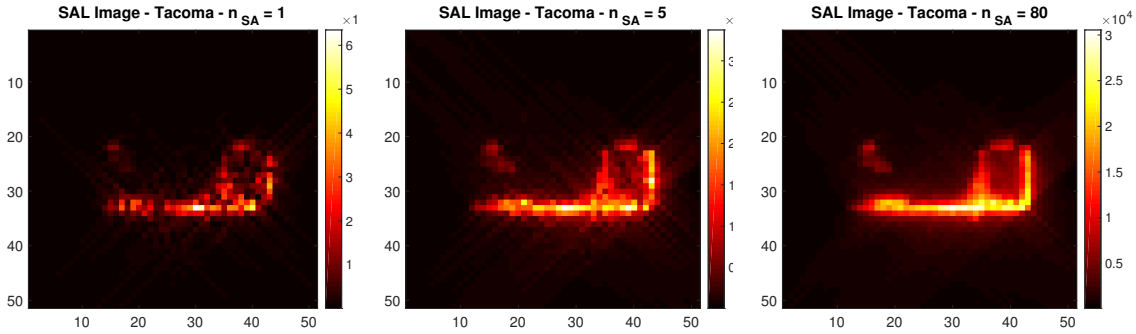


Figure 4.7: The effects of speckle averaging on SAL images. Speckle averaging results in less noisy images.

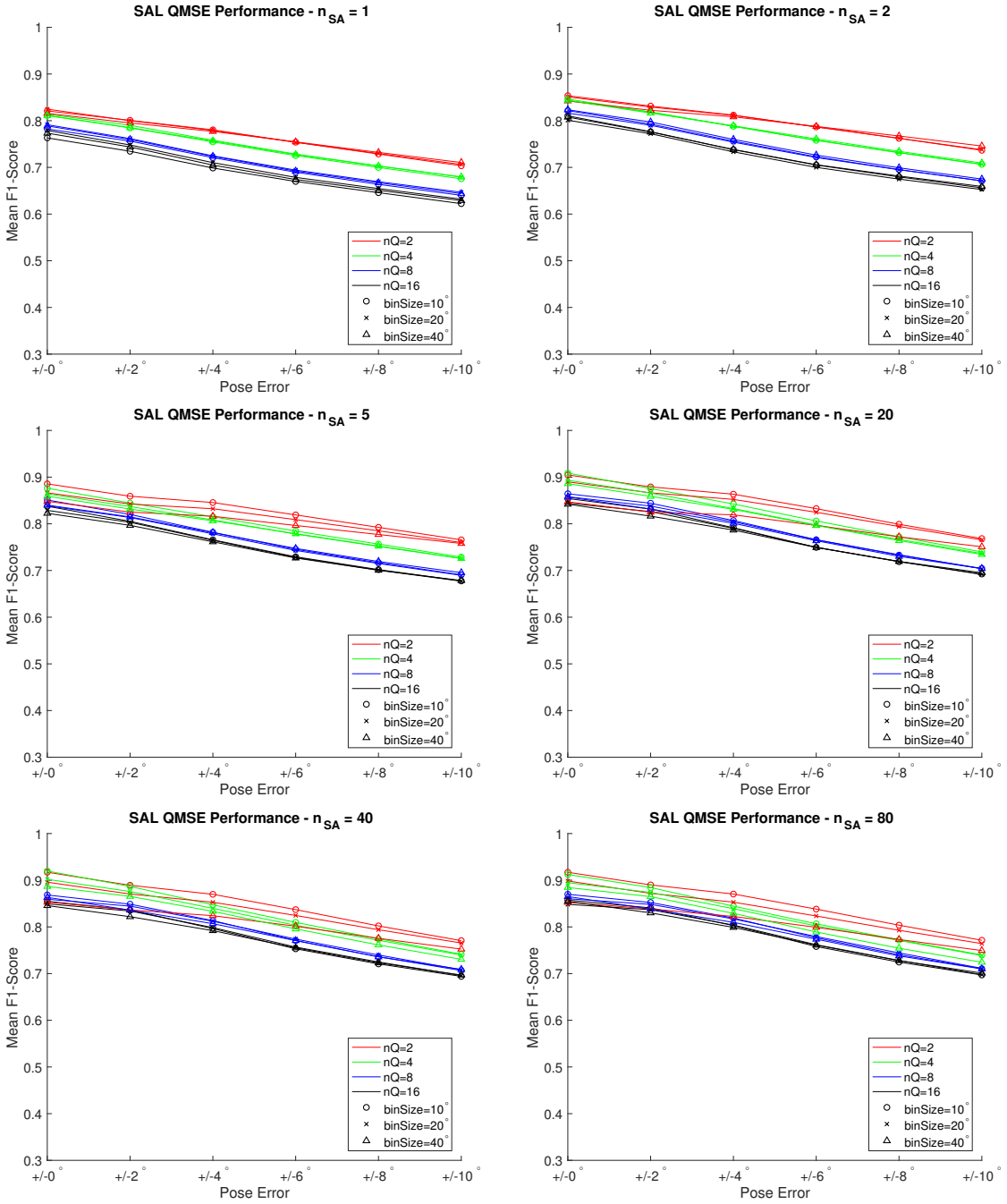


Figure 4.8: QMSE performance as a function of pose errors and number of images used for speckle averaging. The most significant gains in performance arise from increasing the lesser amounts of n_{SA}. Increases in performance beyond $n_{SA} = 5$ are not as significant. Diminishing returns in terms of pixel variance for $nQ = 2$ is described in Figure 4.10.

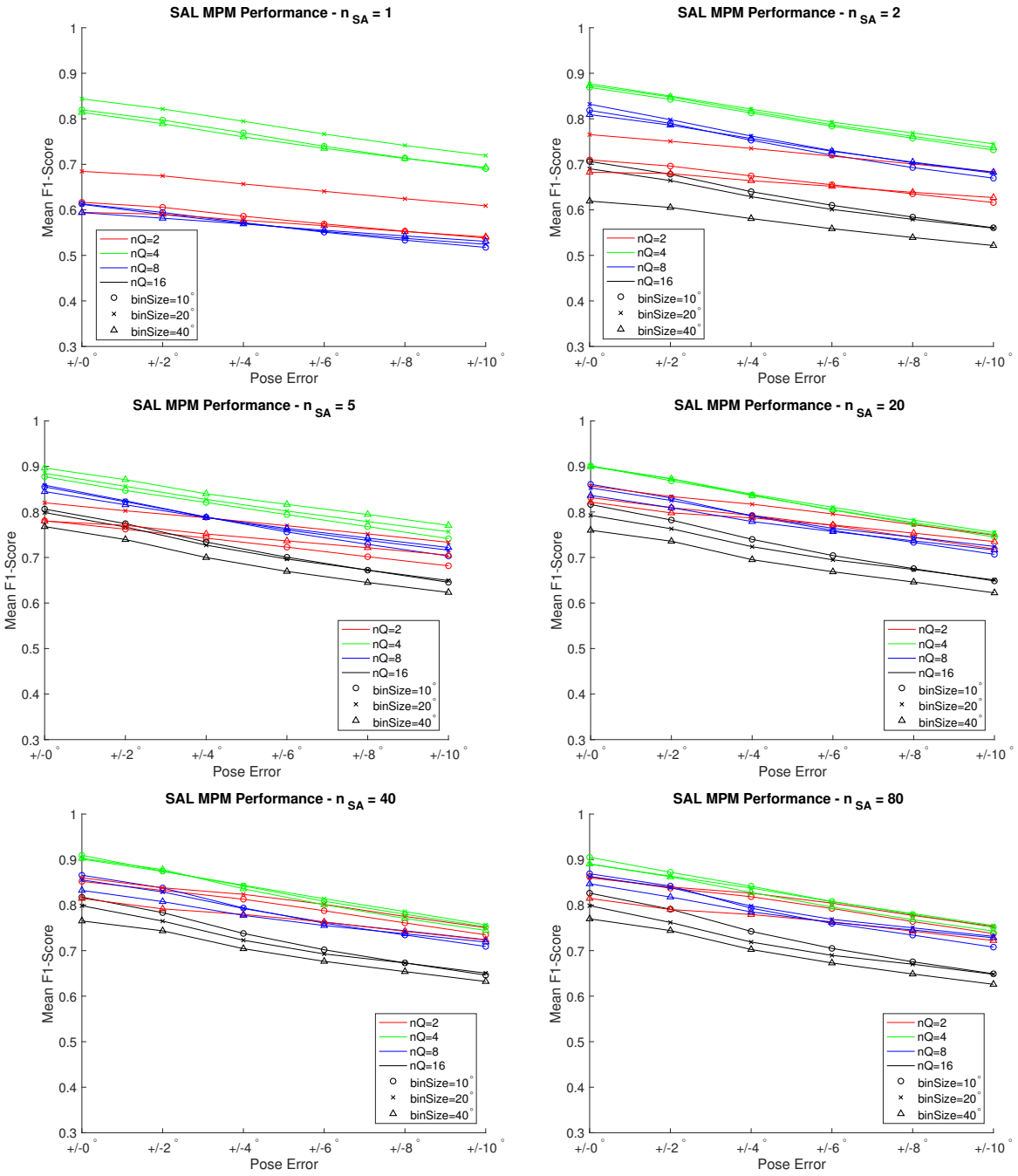


Figure 4.9: MPM performance on SAL images as a function of pose errors and images used for speckle averaging. For small amounts of $n_{SA} = 1, 2, 5$ larger bins are preferred. The intra-template variance can be reduced by increasing the number of images included in a template when n_{SA} is low.

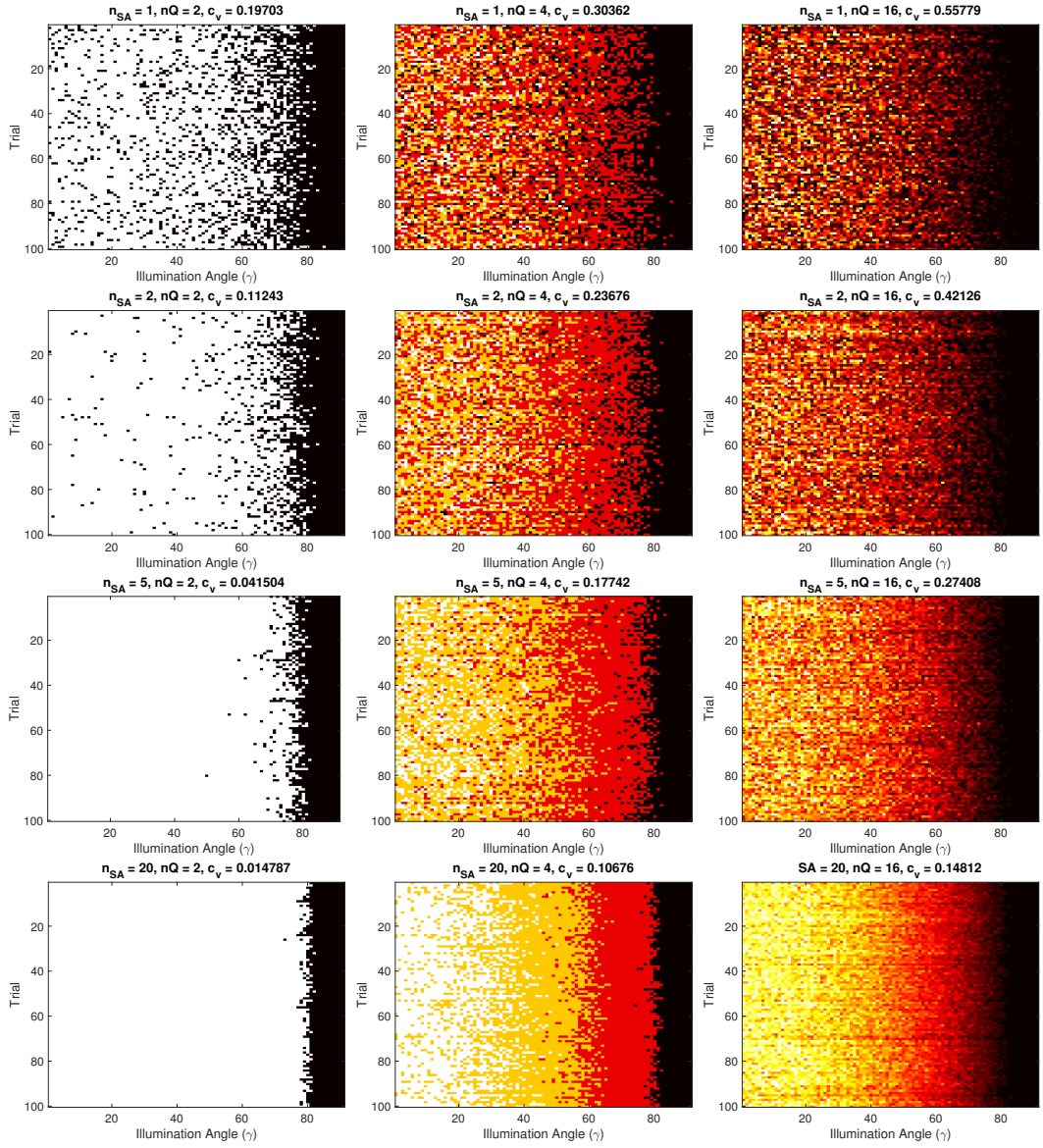
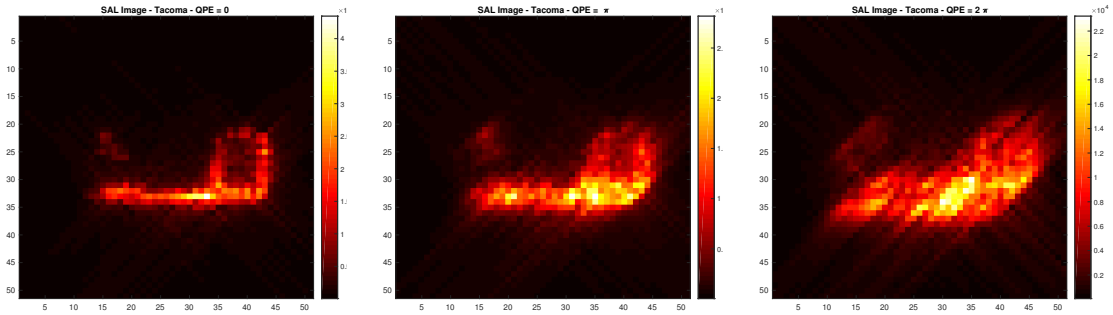


Figure 4.10: Empirical analysis of the coefficient of variance of pixels as a function of nQ level and n_{sa}. the coefficient of variance is defined as $c_v = \mu/\sigma$ where μ is the sample mean of each the trials and σ is the sample standard deviation. The $nQ = 2$ case benefits from rapid decrease in c_v as n_{sa} increases. Higher nQ levels give more intensity information, however, c_v also increases.

4.1.7 Results - SAL and Quadratic Phase Error

This section discusses the effects of QPE on MPM and QMSE performance with SAL imagery. QPEs cause blurring in an image. In the QMSE case, higher levels of nQ outperform the $nQ = 2$ level as phase errors increase. As shown in Figure 4.13, additional intensity



$QPE = 0$ $QPE = \pi$ $QPE = 2\pi$

Figure 4.11: The effects of quadratic phase error (QPE) on SAL imagery. QPE causes blurring effects in the image due to atmospheric turbulence.

information can aid in distinguishing targets. In the MPM case, $nQ = 4$ is the optimal nQ for all tested QPE levels. The bin size levels converge as QPE increases. This implies as QPE increases, neither widening nor shrinking pose bin size will aid in classification effectiveness.

4.1.8 Factor Analysis and ANOVA

This section discusses the interactions among the experimental parameters for the SAL ATR experiments. Figures 4.15a and 4.14a plot the interactions between the relevant SAR and SAL experimental parameters respectively. Interaction plots are a visual tool for assessing meaningful interactions among factors. Interaction occurs when trends among different levels are not parallel. An example of meaningful interaction is shown in Figure 4.14a. In the QMSE case, there is interaction between n_{sa} and nQ, as well as nQ and QPE. Conversely, there is not meaningful interaction between $poseError$ and all other factors. These interactions are also evident in the ATR results shown in the previous section. For example, the trends in performance shown in Figure 4.12 suggests increasing nQ levels will increase performance when QPE levels are also high in the QMSE case. This interaction can be

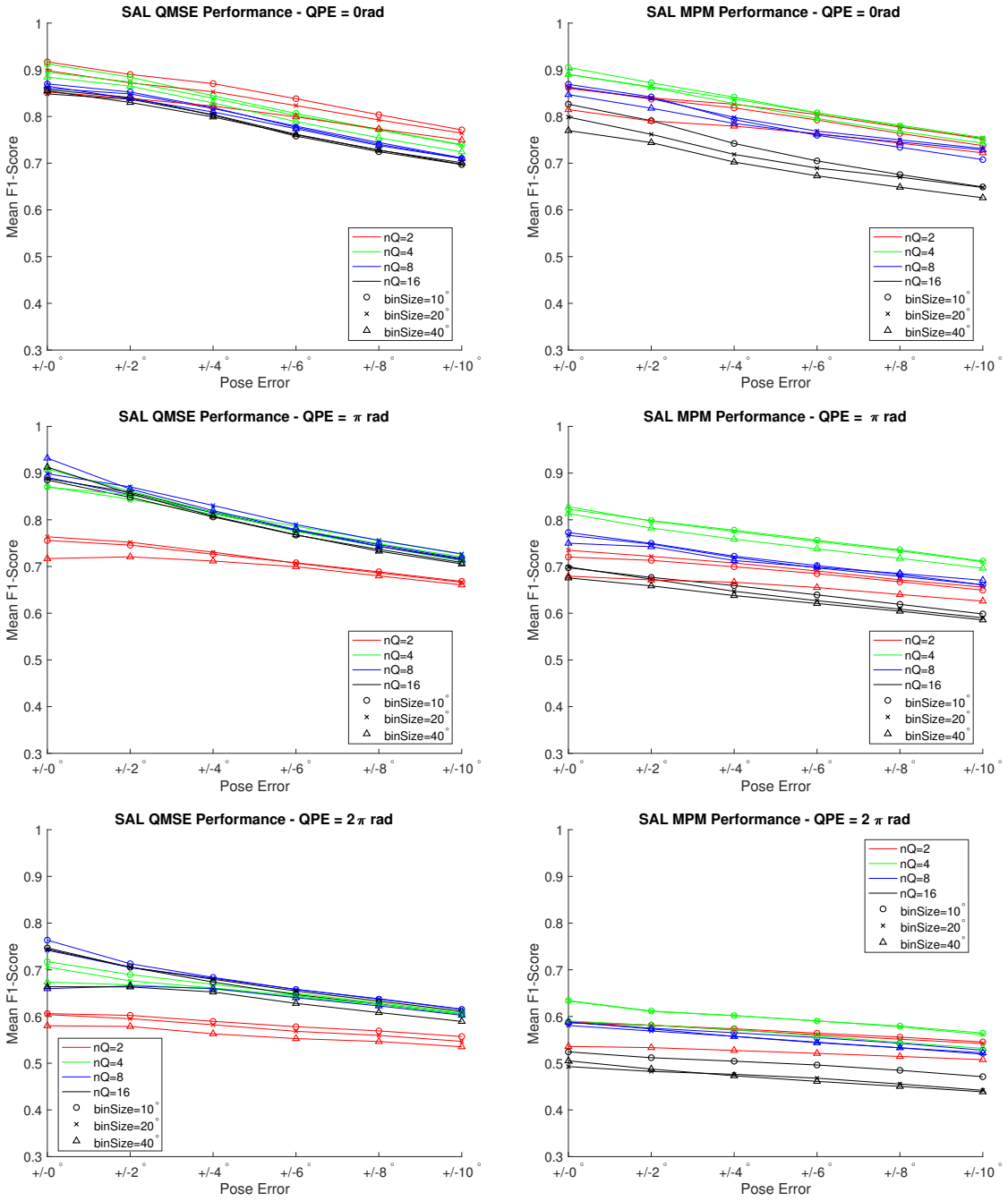


Figure 4.12: QMSE and MPM performance on SAL data at increasing levels of QPE. In the QMSE case, as QPE get more severe, higher levels of nQ increase performance. As discussed in previous sections, $nQ = 2$ was optimal due to pixel blurring effects. Under the effects of QPE, pixel locations are blurred, thus, more intensity information is needed to distinguish targets. This behavior is shown in Figure 4.13. For the MPM case, the best choice is $nQ = 4$. As pose errors get more severe, under large amount of QPE all bin sizes perform similarly.

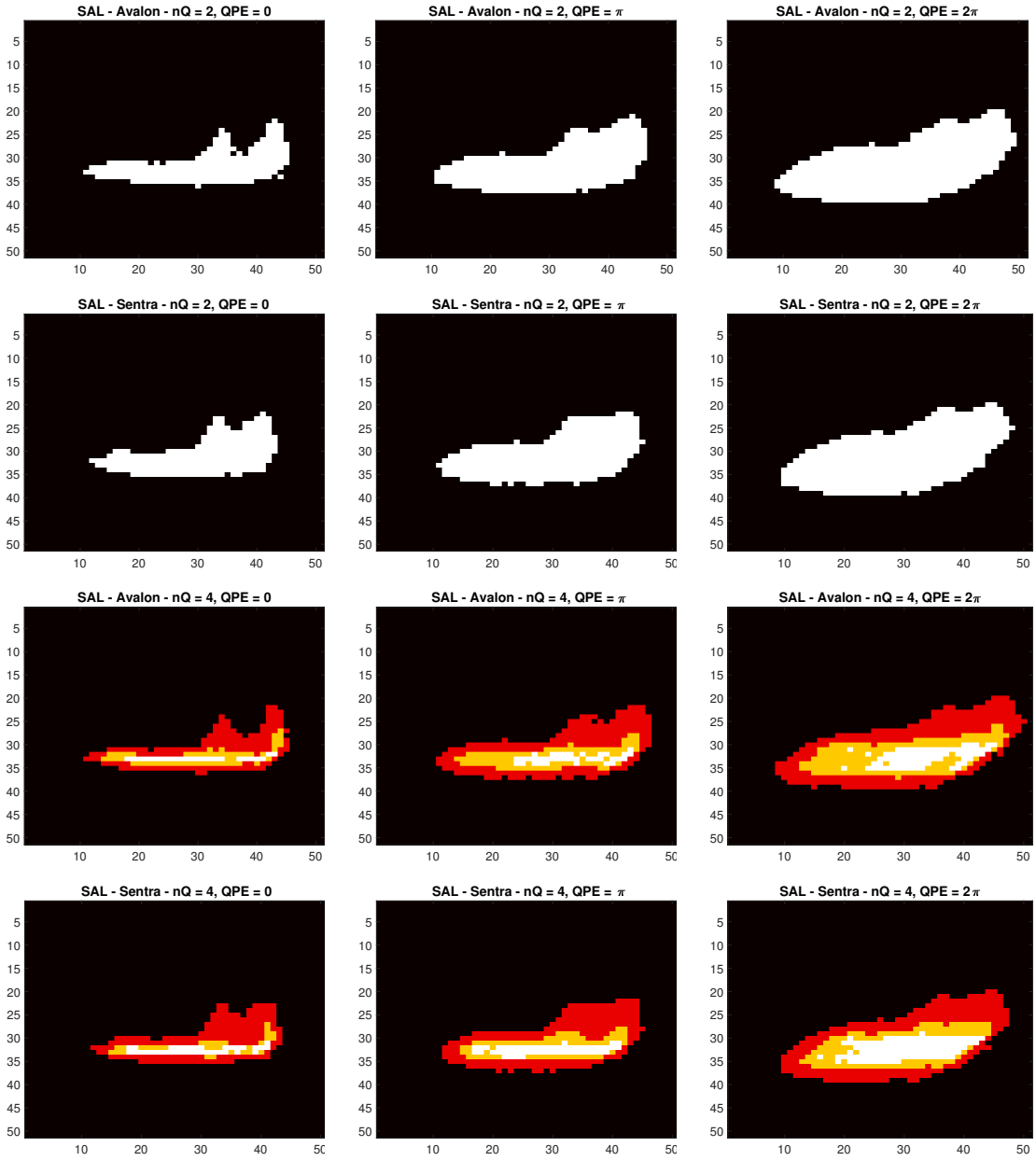
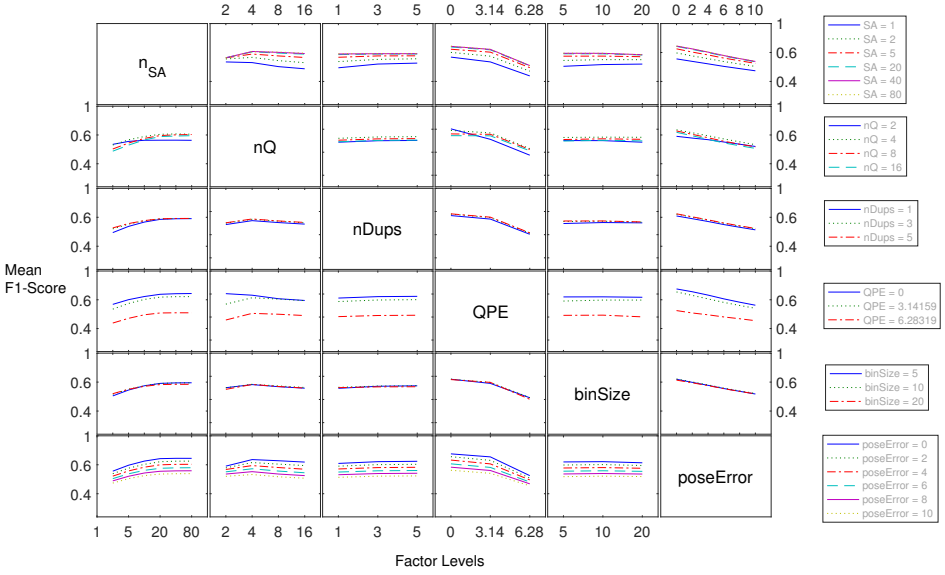


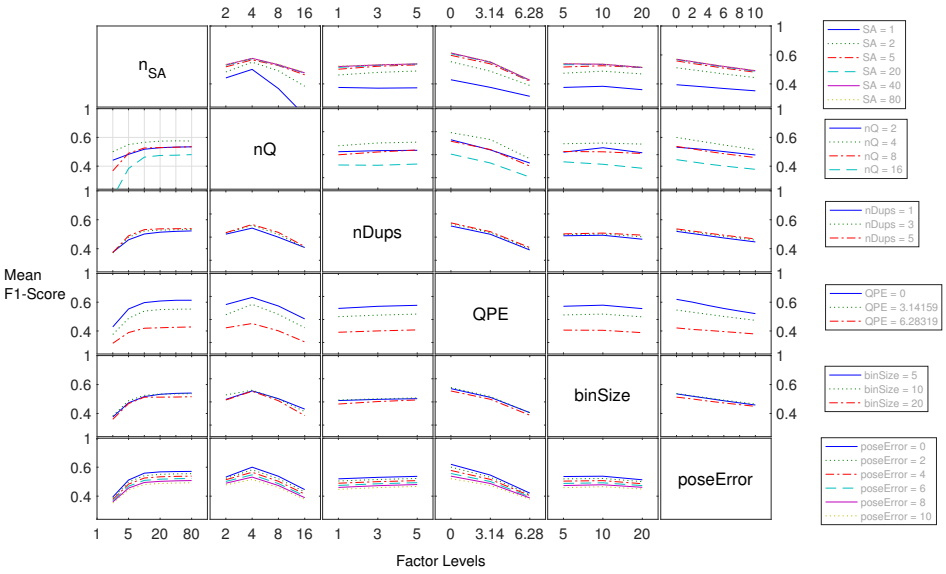
Figure 4.13: Examples of the QPE blurring effect on quantized images. At $nQ = 2$, features of the targets blend together. At $nQ = 4$, the shapes of the targets converge with varying intensity information inside the target's pixels.

confirmed by analyzing the factor interactions and ANOVA tables. ANOVA tables reveal the statistically significant factors which impact ATR performance. Tables A.1 and A.2 contain the ANOVA for SAR MPM and QMSE factors respectively. For both algorithms, the p value for the $nDups * poseError$ interaction is higher than $p > 0.05$ indicating that the $nDups * poseError$ interaction is rejected as significant at a 95% confidence interval. Tables A.3 and A.4 contain the ANOVA for MPM and QMSE respectively for the SAL case. In the MPM case, the $nQ * nDups$ interaction has $p = 0.05$ and indicates this interaction is rejected as significant at the 95% confidence interval.

SAL QMSE Factor Interactions

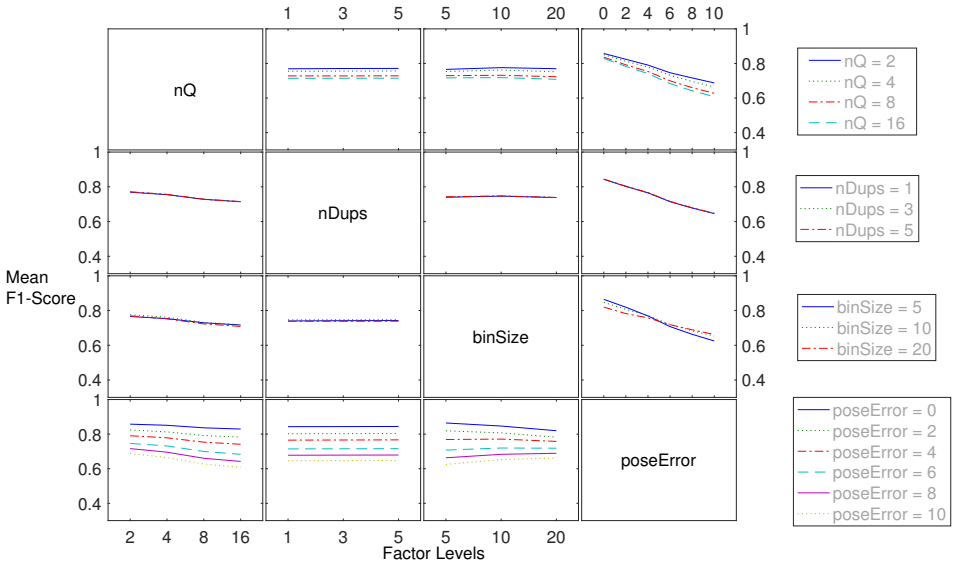


SAL MPM Factor Interactions

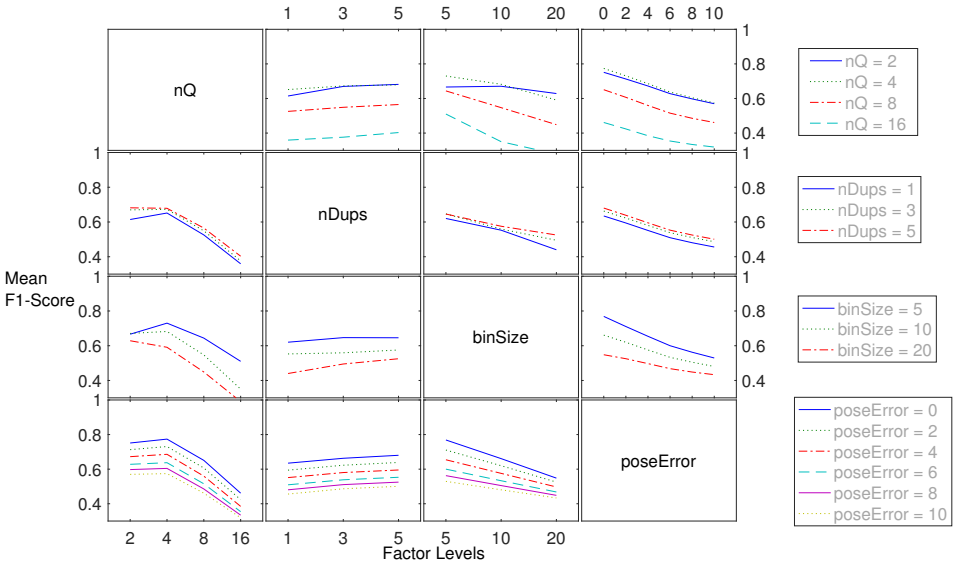


(a) Factor interactions for SAL QMSE and MPM performance. Each row represents a factor fixed at each of it's levels. Each line in the subplots represents the levels of each row factor. Each individual level is plotted against all other factors levels. Meaningful interactions can be gleaned from subplots where row factors are not parallel and contain large gaps in F1-score. Each algorithm has instances of unique and significant interactions. For example, there is discernible interaction between n_{SA} and $binSize$ for the SAL MPM case but little interaction in the QMSE case.

SAR QMSE Factor Interactions



SAR MPM Factor Interactions



(a) Factor interactions for QMSE and MPM on SAR imagery. For the QMSE case, the performance for various choices nQ are less severe than in the MPM case. In the MPM case, there is significant interaction between $poseError$ and $binSize$ but not in the QMSE case. The $nDups$ and $binSize$ factor does not have a measurable impact on MPM performance. Conversely, these factors impact QMSE performance.

4.2 CAD and SAL Image Similarity

This section describes how CAD physical properties are leveraged in order to estimate the similarity of SAL images. First, the key physical properties of the CAD models that are used to predict image similarity are discussed. Second, a methodology for transforming CAD physical properties into a 2-D space is introduced. Next, similarity trends within the CAD domain are compared to the similarity trends in the image domain via Spearman correlation. Finally, the correlation between the CAD representations and SAL image representations of targets are discussed.

The feasibility of the following approach is tested in the SAR domain. Key aspects of the SAR backscattering are identified. Additionally, the difficulty of capturing the specular nature of SAR is discussed.

4.2.1 SAL Backscattering and Key Physical Properties

The backscattering of a diffuse surface is modeled as:

$$\boldsymbol{E_{bs}} = \boldsymbol{A}_i \cos(\sqrt{(\boldsymbol{n} \cdot \boldsymbol{u}_i)}) \exp(jk_o r_i) \tag{4.9}$$

where $\boldsymbol{A}(i)$ is the random speckle value found at the surface, $\boldsymbol{n}$ is the normal of the surface, $\boldsymbol{u}_i$ is the direction of the incident field, k_o is the wavenumber of the sampled frequency, and r_i is the range from surface to sensor. Equation 4.9 is the backscatter model used in the first iteration of LaiderTracer [56]. This model assumes a perfectly Lambertian Bi-directional Reflectance Distribution Function (BRDF). The intensity of the backscattered field is proportional to the cosine of the illumination angle ($\gamma = \boldsymbol{u}_i \cdot \boldsymbol{n}$).

An update to LaiderTracer introduces material properties [66]. With the inclusion of material properties, the backscatter model becomes:

$$\boldsymbol{E_{bs}} = \boldsymbol{A}(i)\cos(\gamma)(f_m(\gamma))\exp(jk_o r_i) \tag{4.10}$$

where $f_m(\gamma)$ is the BRDF function for some material m measured at angle γ.

Each facet in a CAD model serves as a potential scatterer during ray-tracing in SAL simulation. Details about the CAD model file format used can be found in Appendix A. Each CAD model used in our experiments can be viewed as a triangular mesh. Examples of facets files from the civilian data domes can be seen in Figure 4.16.

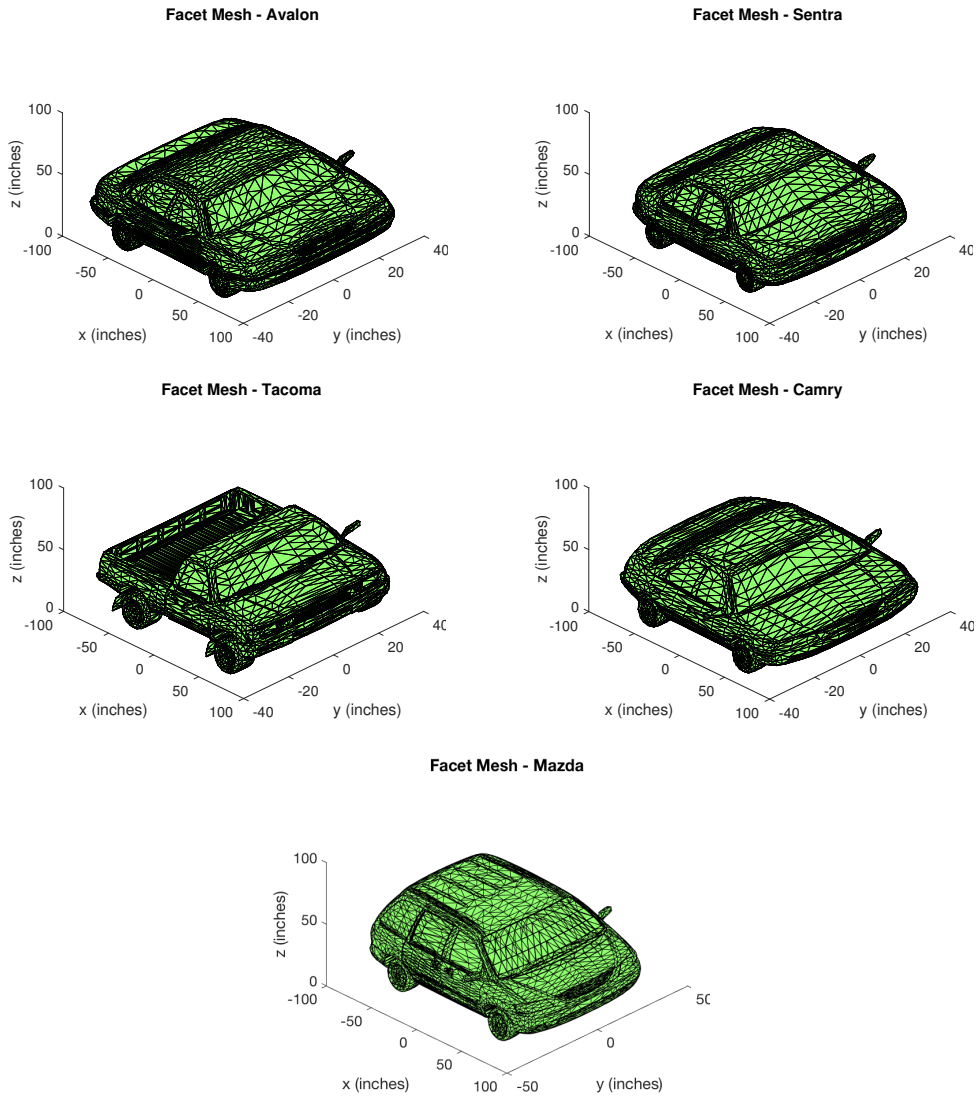


Figure 4.16: The CAD models for the civilian vehicles used in this work. Each CAD model is composed of triangular facets. The facets are composed of three vertices with x,y,z coordinates in inches.

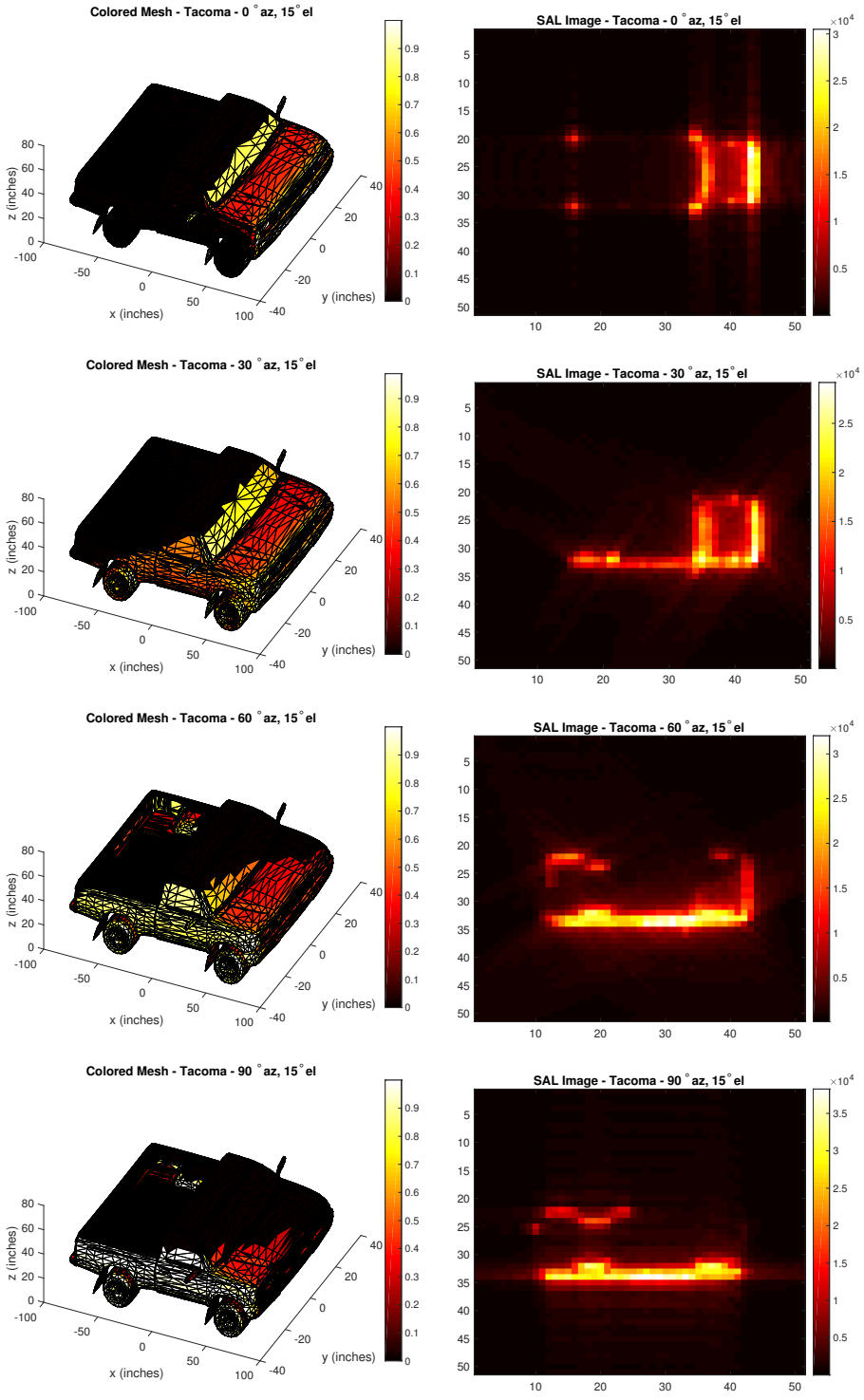


Figure 4.17: SAL images of the Tacoma and their corresponding CAD models filled in with the relevant physical properties. The SAL images are formed with the same imaging parameters described in 4.1. The CAD models are rotated to make the illuminated facets visible.

For each contributing scatterer, the backscatter field described by Equation 4.10 is coherently summed and concatenated at each aperture position to form a phase history. This process is described in full detail in [56, 66, 70]. The phase history is then processed by image formation algorithms such as back projection or the polar format algorithm [20].

The $cos(\gamma)$ and $f_m(\gamma)$ terms can be found with only the facet information from the CAD model and the position of the aperture. Each facet in the CAD model can be assigned the term:

$$fc(\gamma) = \cos(\gamma)(f_m(\gamma)) \tag{4.11}$$

Each facet visible from the sensor position is filled in with that facet's contribution to the far-field return found in Equation 4.11. Figure 4.17 shows meshes filled with the fc value assuming perfectly diffuse scatterers and the corresponding SAL image. In this case, a perfectly Lambertian surface is assumed. Given this assumption, the physical properties of each facet can be simplified:

$$fc(\gamma) = cos(\gamma) \tag{4.12}$$

Since the response is exactly the cosine of the illumination angle, the intensity of a facet will be bright as γ approaches 0° and dim as it approaches 90°. Visual inspection shows that the relative intensity of the filled in facets correspond to the intensities of the pixels in the SAL image. In general, the pixels are bright where there are a relatively large number of scatterers occupying the same space and normal to the sensor. The pixels tend to be more dim where there are relatively few scatterers occupying the same space, and the sensor is at a grazing angle relative to the facet.

4.2.2 Converting Facet Mesh to 2-D

Section 4.2.1 discussed the relevant physical properties of a CAD model that are a part of the SAL back-scattering model. Each visible facet in the CAD model is assigned a value

determined by Equation 4.11. Each facet in a CAD model is a triangle $\triangle \boldsymbol{abc}$ where each vertex $\boldsymbol{a}$, $\boldsymbol{b}$,$\boldsymbol{c}$ is a $1x3$ row vector containing the x, z, y location of that vertex in real space. Equation 4.11 extends the facet model. Each facet is a tuple of an x, y, z location and $fc(\gamma)$:

$$\boldsymbol{FM}(i) = [\{\boldsymbol{a}_i = [x_a, y_a, z_a], \boldsymbol{b}_i = [x_b, y_b, z_b], \boldsymbol{c}_i = [x_c, y_c, z_c]\}, fc(\gamma)] \tag{4.13}$$

The first step in converting each facet mesh $\boldsymbol{FM}$ object to a 2-D space is to convert it to a 3-D point cloud with the $fc(\gamma)$ serving as intensity information. Each facet is represented by it's center point rather than all three vertices:

$$\boldsymbol{FM_{mid}}(i) = \left[\frac{\boldsymbol{a}_i + \boldsymbol{b}_i + \boldsymbol{c}_i}{3}, \cos(\gamma)(f_m(\gamma))\right] \tag{4.14}$$

Each entry in the $\boldsymbol{FM_{mid}}$ matrix is a $1x4$ row vector where the first three entries are the x, y, z coordinates of the point and the fourth entry is the intensity information. The transformation from mesh to point cloud is shown in Figure 4.19.

In SAL image formation, an imaging grid is specifies the bounds of the scene as well as pixel size. In order to compare the 2-D representation of the point cloud to a SAL image of the same target, the pixel spacing and image size should be the same. The imaging grid is formed by specifying a range and cross-range extent (wr and wx) as well as a scene height value ($maxZ$). The scene height $maxZ$ must be larger than the max z coordinate in the facet model. Assuming the range and cross range resolutions are equivalent, the pixel locations in the x,y, and z dimensions are specified by:

$$\boldsymbol{xLoc} = linspace\left(-\frac{wr}{2}, \frac{wr}{2}, gs\right) \tag{4.15}$$

$$\boldsymbol{yLoc} = linspace\left(-\frac{wx}{2}, \frac{wx}{2}, gs\right) \tag{4.16}$$

$$\boldsymbol{zLoc} = linspace\Big(0, maxZ, gs\Big) \tag{4.17}$$

where $linspace(minCoord, maxCoord, gs)$ is the linear spacing from $minCoord$ to $maxCoors$ with step size gs. Ultimately, 2-D representations of the CAD information and a SAL image must occupy the same space. To allow this, gs is set to $gs = res \times ps$, where res is the desired range and cross-range resolution of the SAL imagery, and $gs < 1$ is a constant scaling value to ensure the pixel space of an image is smaller than a resolution cell. When forming 2-D images, pixel intensities are mapped to a 2-D grid and it is assumed that $\boldsymbol{zLoc} = 0$. Thus, the imaging grid is a 2-D matrix initialized with zeros:

$$\boldsymbol{IG}_{i \times j} = 0 \tag{4.18}$$

where i is the length of $\boldsymbol{xLoc}$ and j is the length of $\boldsymbol{yLoc}$. After image formation, the $\boldsymbol{IG}$ matrix is a 2-D complex image. The location vectors are queried find the x, y location of each pixel:

$$\boldsymbol{pixelLocation}(\boldsymbol{IG}(i,j)) = [\boldsymbol{xLoc}(i), \boldsymbol{yLoc}(j)] \tag{4.19}$$

More details on the imaging grid and how it is used in image formation is described in [20] and [27].

We now map $\boldsymbol{FM_{mid}}$ to the $\boldsymbol{IG}$ structure described in Equation 4.18. This transformation is composed of the following steps:

1. Render a separate 2-D point cloud ($\boldsymbol{PG}$) where all the x, y locations are the pair-wise mid points of $\boldsymbol{xLoc}$ and $\boldsymbol{yLoc}$ respectively. An example of $\boldsymbol{PG}$ is shown in Figure 4.18.

2. For each point in $\boldsymbol{PG}$, find all points within a specified radius $radius = gs$ in the $\boldsymbol{PM_{mid}}$ point cloud. Sum the $fc(\gamma)$ terms found at these points and assign them to the current point in $\boldsymbol{PG}$.

3. Finally, for each cell in $\boldsymbol{IG}(i, j)$, sum the intensity information for the points in $\boldsymbol{PG}$ that fall within the cell's bounds. Examples of SAL images and their corresponding $\boldsymbol{IG}$ representations are shown in Figure 4.20.

This section presented a method to transform the physical information of CAD models to the same dimension as SAL images. In following section, trends in cross target similarity in both the CAD and image domain are analyzed. First, the correlation scores of $\boldsymbol{IG}$ representations and SAL images are found. Second, similarity trends among $\boldsymbol{IG}$ representations and trends among SAL images are compared.

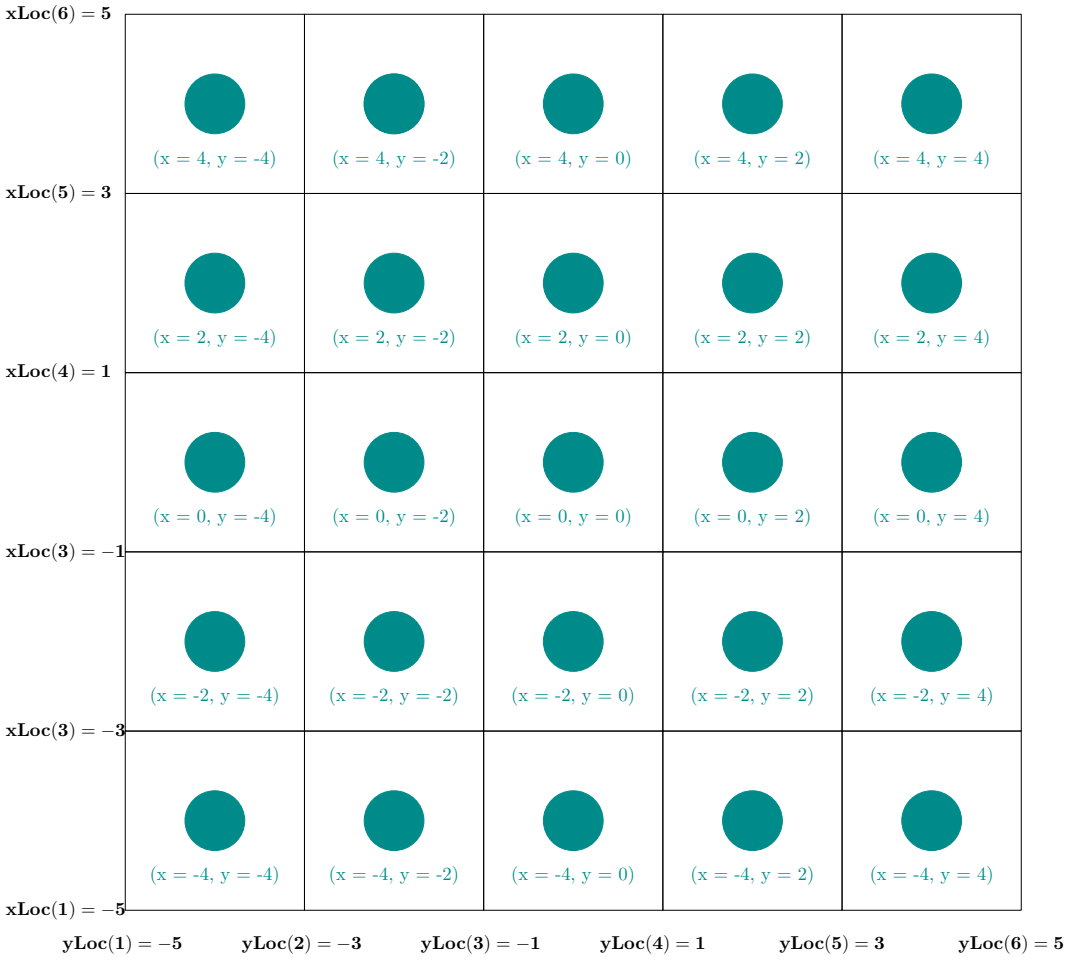


Figure 4.18: A top-down view of the imaging grid $\boldsymbol{IG}$ with mid-points specified in $\boldsymbol{PG}$. The location vectors $\boldsymbol{xLoc}$ and $\boldsymbol{yLoc}$ were formed with the parameters of $wr = wx = 10$ and $gs = 2$.

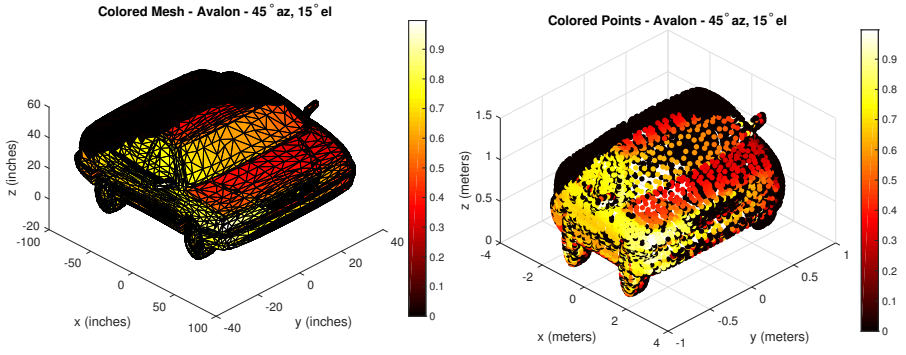


Figure 4.19: An example of converting a filled in mesh to an intensity point cloud. Each point in the point cloud represents the center point of each facet in the mesh. The process to convert mesh to point cloud is described by Equation 4.14

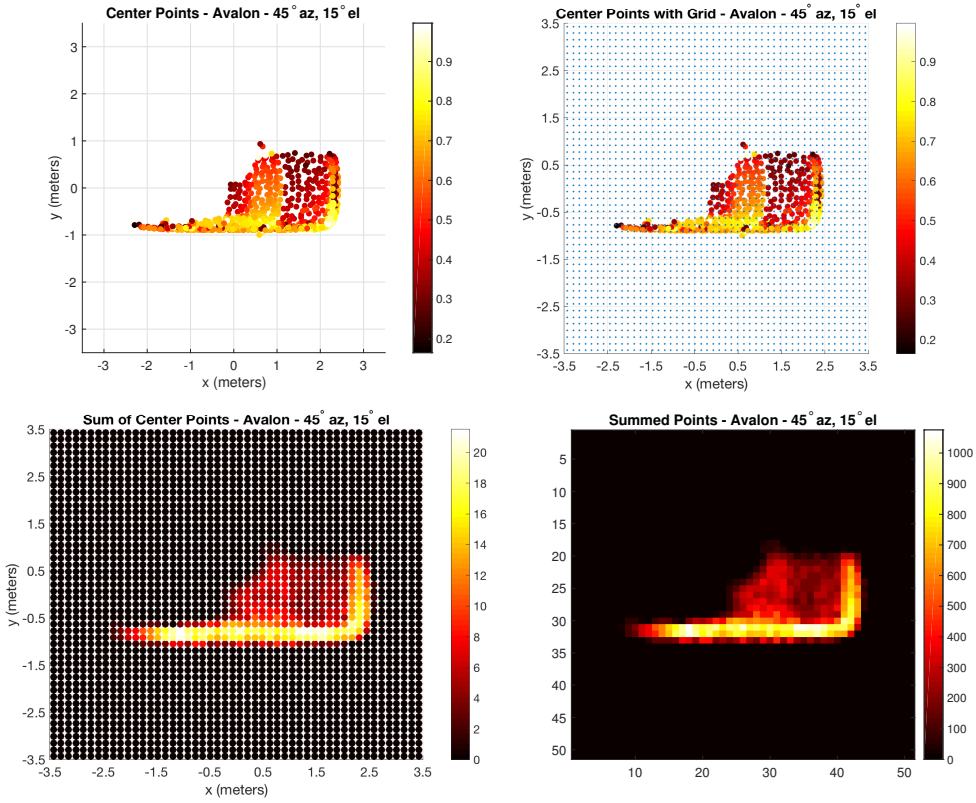


Figure 4.20: An example of converting center points to a 2-D grid. In this example, the point cloud is mapped to the same imaging grid properties in the template matching experiments in Section 4.1. Each point in the 2-D point cloud grid represents the center point of the 2-D image grid. All points within the specified radius of each grid point are summed. Each point in the summed point cloud are within the cells of a 2-D imaging grid in the x and y dimension. For each cell, all points within it's bounds are summed resulting in a 2-D representation.

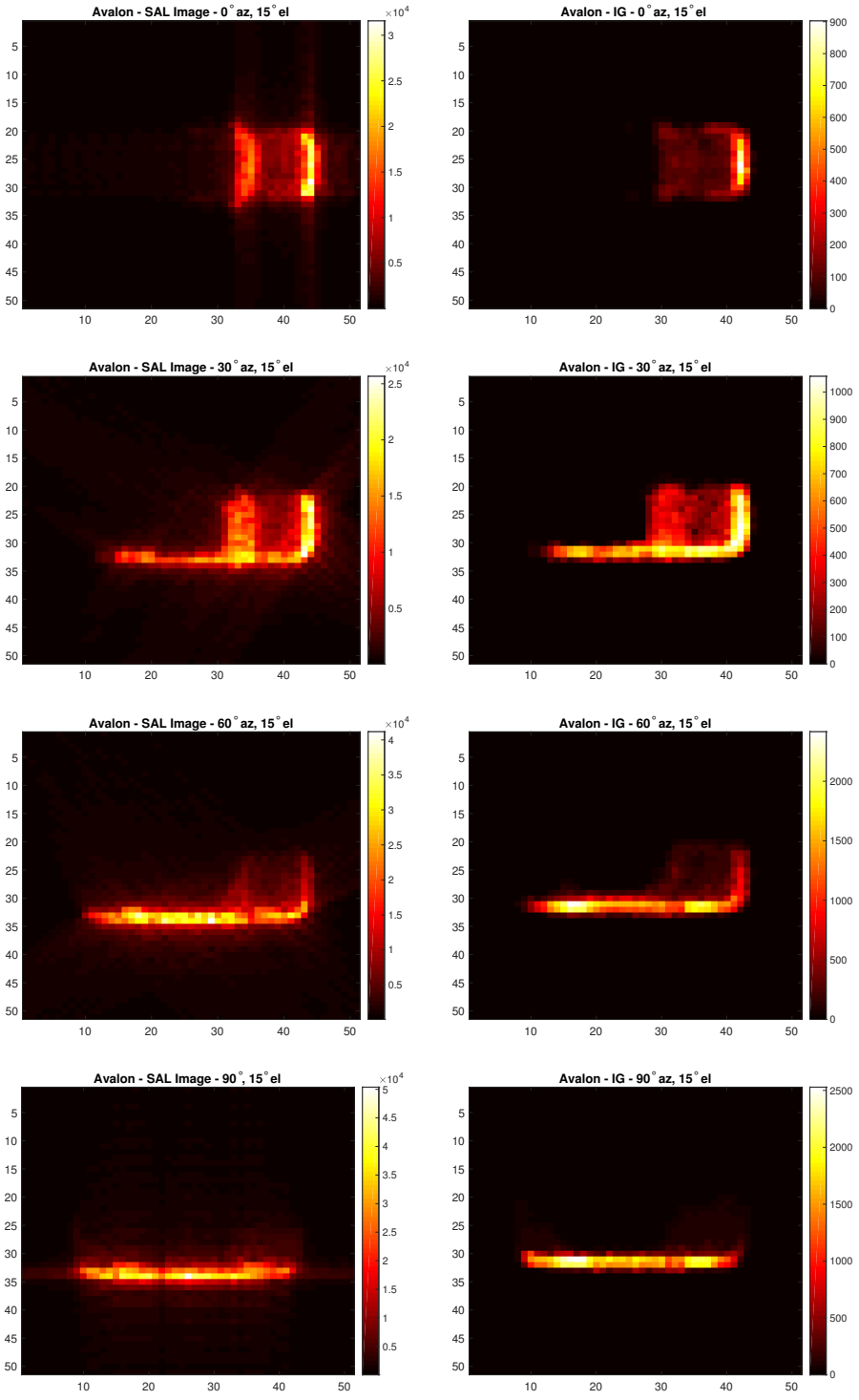


Figure 4.21: Example SAL Images of the Avalon model and it's corresponding ***IG*** representations formed with the imagine parameters found in Section 4.1. All five targets are shown in Appendix C.

4.2.3 Predicting SAL Image Similarity via CAD Analysis

This section demonstrates the trends in cross target $\boldsymbol{IG}$ correlate with cross target SAL image similarity. Similarity matrices aid in analysis [24]. A similarity matrix $\boldsymbol{S}$ is written as:

$$\boldsymbol{S_{AvB}}(i,j) = sim(A_i, B_j) \tag{4.20}$$

where sim is a similarity metric for objects A and B at modifications i and j respectively. For SAL image analysis, A and B are SAL images of two targets. For CAD analysis, A and B are the $\boldsymbol{IG}$ representations of targets A and B respectively.When $A = B$, sim computes the self-similarity. When $A \neq B$, sim computes the cross-similarity. The modifications i and j are azimuth positions of the sensor. Pearson correlation (r_p) is computed between the SAL images and $\boldsymbol{IG}$ representations:

$$r_p = \frac{\sum(A_i - \bar{A})(B_i - \bar{B})}{\sqrt{\sum(A_i - \bar{A})^2 \sum(B_i - \bar{B})^2}} \tag{4.21}$$

where $\bar{A}$ and $\bar{B}$ are the mean values of A and B respectively. The index i is the linear index of the values in A and B. Example similarity matrices are shown in Figure 4.22.

In order to determine the similarity trends in both the SAL image and CAD the following steps are taken:

1. For all targets of interest T and aspect angles $\boldsymbol{az} = [0°, 4°, 8° \cdots 360°]$ form the self-similarity matrices $\boldsymbol{S}_{AvA}(i,j) = sim(A_i, A_j) \forall i, j \in \boldsymbol{az}$ and cross-similarity matrices $\boldsymbol{S}_{AvT}(i,j) = sim(A_i, T_j) \forall i, j \in \boldsymbol{az}$.

2. Select A_i to be the test target A at aspect angle i. Select row i from the self-similarity matrix ($\boldsymbol{S}_{AvA}$) and the ith rows from all cross-similarity matrices $\boldsymbol{S}_{AvT}$. Concatenate the rows to form a ranking vector $\boldsymbol{imgRank}_{A_i}$. The vector $\boldsymbol{imgRank}_{A_i}$ contains the similarity scores of target A at aspect i versus all other targets and aspect angles.

3. Repeat steps 1 and 2 for the $\boldsymbol{IG}$ representations.

4. Find the Spearman correlation coefficient between $\boldsymbol{imgRank}_{A_i}$ and $\boldsymbol{IGRank}_{A_i}$ for all aspect angles i.

The Spearman correlation $r_s(\boldsymbol{imgRank}_{A_i}, \boldsymbol{cadRank}_{A_i})$ quantifies how similar the rankings are for target similarity in both the SAL image and CAD domain. If $r_s(\boldsymbol{imgRank}_{A_i}, \boldsymbol{cadRank}_{A_i}) \approx 1$, the rankings are very highly correlated. Spearman correlation is defined as:

$$r_s = 1 - \frac{6 \sum (d_i)^2)}{n(n^2 - 1)} \tag{4.22}$$

where d_i is the difference between the ith observation of two variables and n is the total number of observations.

The following experiments utilize same imaging parameters and targets as listed in Section 4.1. The selected aspect angles are $\boldsymbol{az} = [0°, 4°, 8° \ldots 356°]$. All targets are imaged at an elevation angle of $el = 17°$. Figure 4.23 shows the Spearman correlation scores for the CAD and SAL image rankings for all targets and aspect angles. All r_s scores are above 0.9 indicating the rankings of $\boldsymbol{IG}$ are strongly correlated with the rankings of their SAL image counterparts.

4.2.4 SAL Image and CAD Correlation

The previous section explores how similarity trends in the SAL image domain highly correlate with similarity trends in the CAD domain. This section explores how well SAL image and CAD representations correlate with each other. Let $A_{CAD,i}$ be the $\boldsymbol{IG}$ representation of a target A at aspect angle i. Let $A_{SAL,i}$ be the SAL image of target A as aspect angle i. Since both representations are in the same space as described in Section 4.2.2, Pearson correlation can be applied:

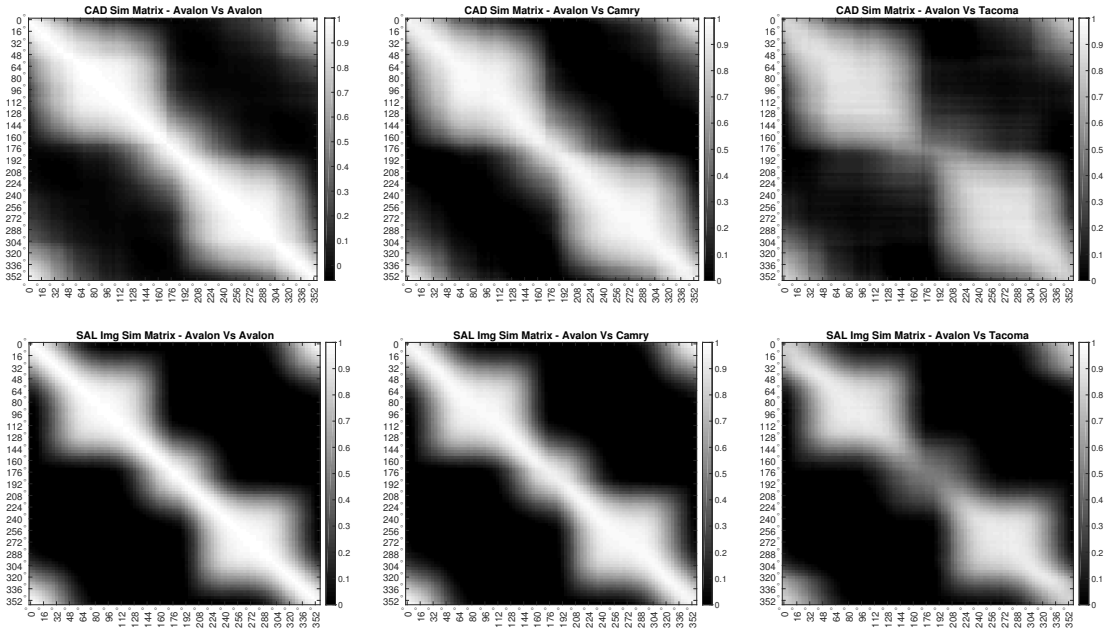


Figure 4.22: Example SAL image and CAD similarity matrices. The axes of each similarity matrix range from $0°$ to $356°$ in $4°$ increments. Three civilian vehicles from the CV data domes are selected, the Avalon, Camry, and Tacoma. The self-similarity matrix exhibits strong similarity close to the diagonal. The Avalon and Camry cross-similarity matrix exhibit relatively high similarity scores. The Avalon and Camry are both sedans and have similar geometrical structure. The Avalon and Tacoma cross-similarity matrix exhibits relatively low similarity scores. Scores are particularly low when comparing the rear of the vehicles ($i \approx j \approx 180°$). The Tacoma rear is a concave truck bed compared to a convex trunk on the Avalon.

$$\boldsymbol{S}(i,j) = r_p(A_{SAL}, A_{CAD}) \tag{4.23}$$

for all aspect angles i and targets j. Figure 4.24 shows the r_p scores for the same targets and aspect angles tested in the previous section.

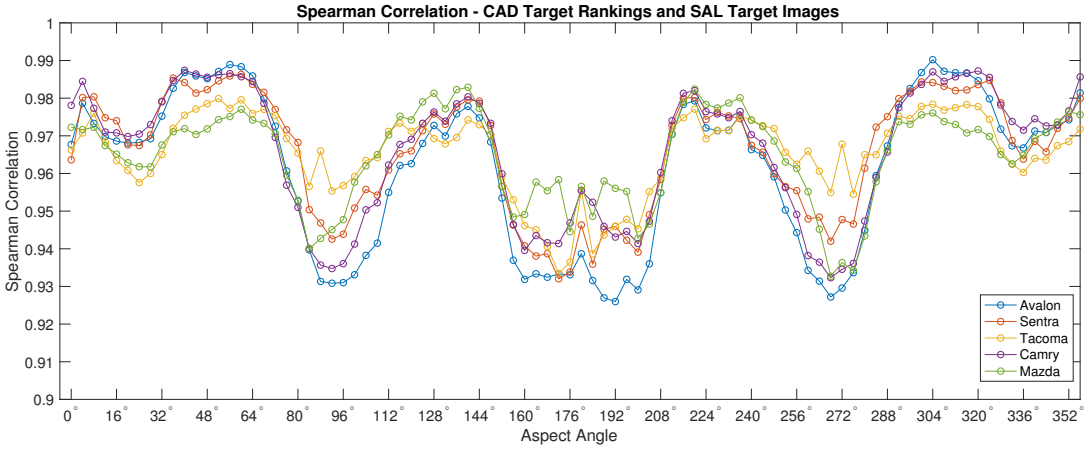


Figure 4.23: Spearman correlation scores between the rankings of CAD representations of targets and SAL images of the same targets and aspect angles. For these experiments, there are five targets and 90 different aspect angles. Thus, each ***imgRank*** and ***IGRank*** vector is a $[450x1]$ vector of rankings.

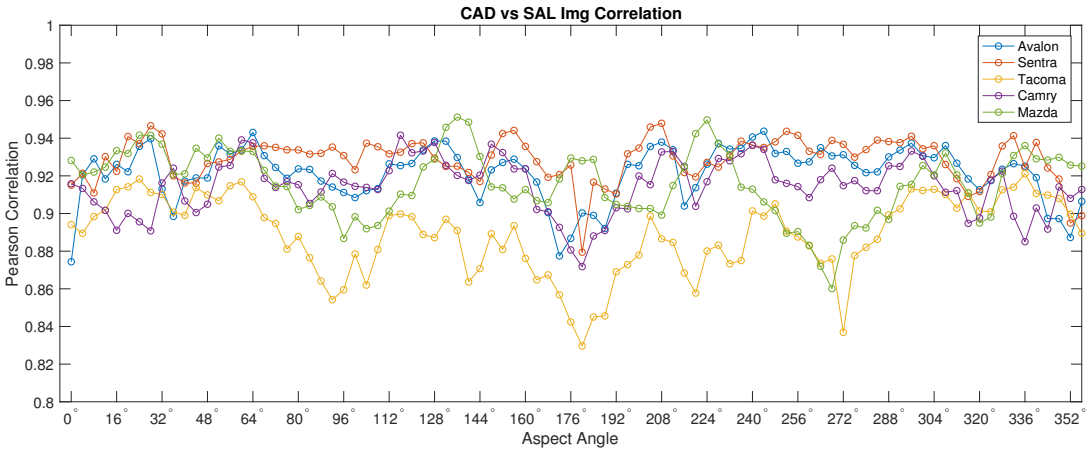


Figure 4.24: Pearson correlation between the SAL image representation of targets and their corresponding CAD representations. All correlation scores are above 0.8, indicating the CAD representations and SAL image representations are highly similar to each other. Of the vehicles listed, the Tacoma model has the lowest correlation scores. The Tacoma has a mix of concave and convex features, while the sedan models and Mazda model are mostly convex.

4.2.5 Slant Plane Adjustments to IG Model

The previous sections describe the process for converting vehicle meshes into the same dimension and structure as SAL images. The assumes all features will be mapped directly into the ground plane. In SAL image formation, images are formed in the slant plane. The

slant plane is the plane formed by the look direction of the ladar and the velocity vector of the platform. Features in a ground plane assumption will not appear in the same location as the slant plane. The approach described in the previous sections do not take the slant plane into account, and features in the CAD representations may not match with features in the image domain.

In order to account for the slant plane, rotations to the imaging grid and CAD point cloud must be made. The following rotations to the point cloud structures will insure the locations of the point cloud features will match the slant plane viewed by the sensor:

- The points in the imaging grid must be rotated to match the azimuth location of the sensor:

$$\boldsymbol{xLocR} = \boldsymbol{xLoc} * cos(az) - \boldsymbol{yLoc} * sin(az) \tag{4.24}$$

$$\boldsymbol{yLocR} = \boldsymbol{zLoc} * sin(az) - \boldsymbol{yLoc} * cos(az) \tag{4.25}$$

- The points in the imaging grid must be rotated to match the elevation angle of the sensor:

$$\boldsymbol{xLocR} = \boldsymbol{xLoc} * cos(el) + \boldsymbol{zLoc} * sin(el) \tag{4.26}$$

$$\boldsymbol{zLocR} = \boldsymbol{zLoc} * cos(az) - \boldsymbol{xLoc} * sin(el) \tag{4.27}$$

These adjustments are also made to the point cloud of the CAD model.

The point cloud formed in the previous section uses the center point of the facets that contribute to a SAL return. The center point assumption will be accurate if the facets are smaller than the resolution cells of the imaging grid. However if a facet is larger than a resolution cell the center point of the facet my not fall within the ray traced grid cell. The location where the ray would hit the facet can be used instead without significant computational burden. The approach for finding the intersection point of a ray and facet is shown in [76]. An example of the improved $\boldsymbol{IG}$ structure is shown in 4.25. The Spearman cor-

relation trends between CAD representation rankings and SAL image rankings are shown in Figure 4.26. The updated Pearson correlation scores between SAL images and CAD representations are shown in Figure 4.27.

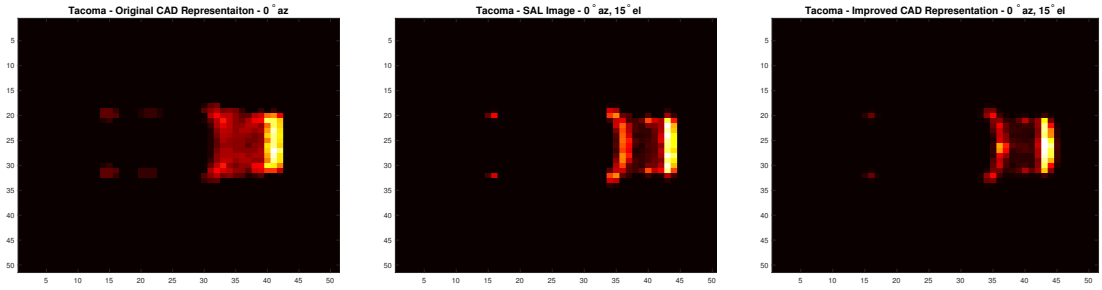


Figure 4.25: Example of the improved ***IG*** model compared to the original approach. Accounting for the slant plane ensures features appear in the same pixel locations as the image features. Additionally, the center point assumption will not be accurate for large facets. Utilizing the point where the ray intersects a facet helps ensure features are in the correct pixel location and captures the relative intensity.

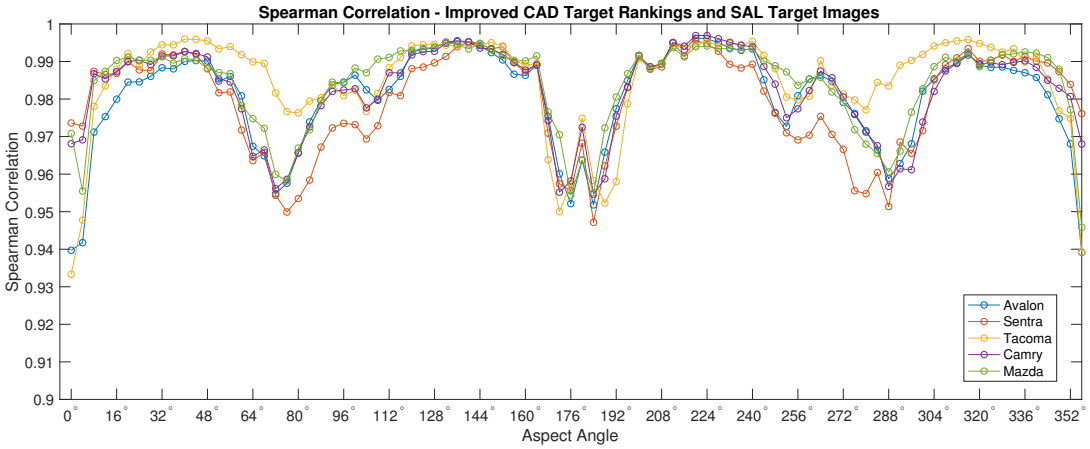


Figure 4.26: Spearman correlation scores between CAD rankings and SAL image rankings after accounting for the slant plane. Spearman correlation scores are overall higher after accounting for the slant plane affect.

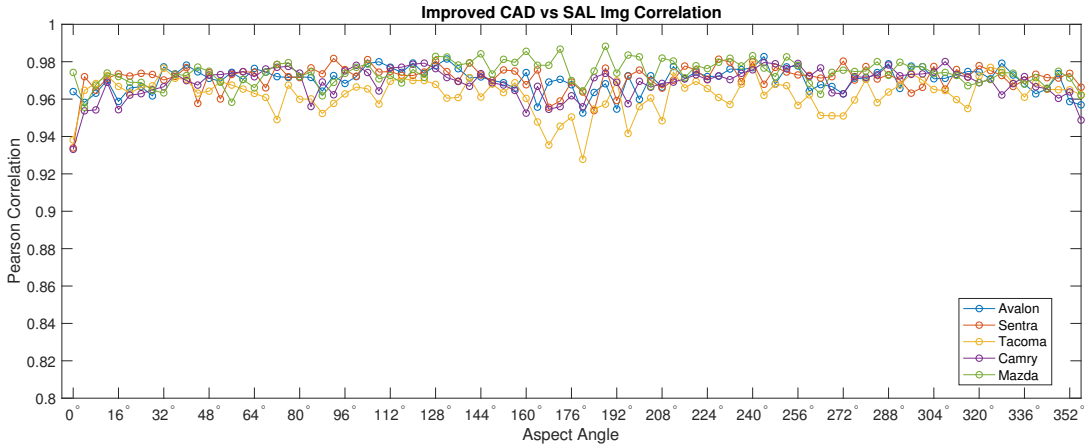


Figure 4.27: Pearson correlation between the SAL image representation of targets and their corresponding CAD representations after accounting for the slant plane effect. The correlation values between the SAL images and CAD representations with the ground plane assumptions range between .8200 and .9600. After accounting for the slant plane affect, correlation values range between .9200 and .9900.

4.2.6 IG Computational Resources Vs Ray-Tracing

In this section the computational benefits of utilizing the ***IG*** model are discussed. The runtime of collecting ***IG*** models is compared to the runtime of collecting SAL imagery with the same imaging parameters. In order to ensure a fair comparison between the ***IG*** model and forming imagery, LaiderTracer is used for both cases. To collect the ***IG*** model, the additional tasks LaiderTracer has to do in order to collect and image are deactivated, and only the routines needed to collect the ***IG*** model are used. When collecting the imagery, LaiderTracer performs only the tasks necessary to form the image, and does not form the ***IG*** model. This ensures that the trials times experience the same amount of overhead in terms of loading facet files and utilizing the same ray-tracing routine. The key difference being that the ***IG*** needs to shoot a single ray at each pixel location. In order to form a SAL image, rays are shot at each pixel location at each aperture location. Additionally, the timing tests are conducted on the same hardware. Table 4.4 shows the timing difference between collecting a full azimuth sweep of the CV targets in both the CAD and SAL domain. The same imaging parameters are the same as used for Section 4.1.

	IG Collection Time (seconds)	SAL Image Collection Time - nSA = 1 (seconds)
Avalon	968.4599	2727.6173
Sentra	666.8359	1816.9010
Tacoma	708.5000	1818.1374
Camry	565.6544	1772.8175
Mazda	1989.5977	3091.0246

Table 4.4: ***IG*** collection time versus SAL image collection time using LaiderTracer. The times listed is the average runtime to collect a full azimuth look of each listed vehicle in 2° increments for 10 trials. The imaging parameters used are the same parameters used in Section 4.1. Each azimuth look was computed in a serial fashion. The runs were conducted on a standard compute node on the Koehr Navy DSRC HPC. This compute node has a Intel Xeon Platinum 8168 CPU with a core speed on 2.7 GHz and 192 GB of usable memory. More details about the computation environment can be found at [1]

In order to conduct speckle averaging in the image domain, the interrogated target must be re-ray-traced and n_{SA} number of times. One method for alleviating the need to re-ray-trace the target is to store the pristine return information and then apply the speckle effect after the fact. As discussed in [70], LaiderTracer forms a sub-phase history for each pixel in the imaging grid. Of these pixels, the only ones kept are the ones that resulted in a ray and facet intersection. In order to store these sub-phase histories, the storage required is:

$$phStackSize = 16(bytes) \times length(\hat{\boldsymbol{k}}_o) \times length(\hat{\boldsymbol{\phi}}) \times nFacets \tag{4.28}$$

where $\hat{\boldsymbol{k}}$ is the wavenumber vector, $\hat{\boldsymbol{\phi}}$ contains the aperture positions of the sensor, and $nFacets$ is the number of facets intersected during ray-tracing. Phase histories are composed on complex numbers. In MATLAB, complex numbers are stored as 16 byte doubles. For each $nFacet$ sub phase history, a random speckle value is then applied. Then, the phase histories are summed to form a single $length(\hat{\boldsymbol{k}}_o) \times length(\hat{\boldsymbol{\phi}})$ phase history.

Before the ***IG*** structure is converted to 2-D space, a point cloud of the relevant facet information is stored. The point cloud rendered needs an x, y, z and intensity value for each

intersected facet. Thus, the storage needed for this structure is:

$$pcSize = 8(bytes) \times 4 \times nFacets \tag{4.29}$$

For the experiments presented so far in the image domain, the number of slow time samples is $length(\hat{\boldsymbol{\phi}}) = 51$ and $length(\hat{\boldsymbol{k}}_o) = 51$ given equal range and cross range resolution. At the resolution used, typically the number of facet intersected is on the order of 100. Table 4.5 shows compares the $psSize$ and $phStackSize$ space requirements assuming the phase history size used in these experiments.

nFacets	**500**	**600**	**700**	**800**	**900**	**1000**
pcSize (Megabytes)	0.0160	0.0192	0.0224	.0256	.0288	.0320
phStackSize (Megabytes)	20.8080	24.9696	29.1312	33.2928	37.5544	41.6160

Table 4.5: Example space requirements needed for the ***IG*** model and pristine phase history approach. These space requirements assume phase histories are 51×51 slow time and fast time samples. For the experiments conducted in this work, the number of facets intersected for a given ray-trace can range from 500 facets to 1000 facets.

4.3 CAD Based Performance Prediction for SAL ATR

This section discusses the feasibility and effectiveness of predicting the performance of template matching algorithms by training and testing on ***IG*** representations exclusively. The training and testing procedures of both MPM and QMSE on ***IG*** are identical to the training and testing procedures on SAL imagery. The same baseline algorithm and imaging parameters from Section 4.1 are used. This section first demonstrates that the overall F1-score of each combination of algorithm parameters can be predicted by training and testing only on ***IG*** representations. Second, this section shows that the trends in confusion in the image domain can be predicted by the confusion matrices formed in the CAD domain. Finally, we show that the QMSE and MPM scores assigned by CAD domain templates

rank CAD domain test samples similarly to the image domain. All confusion matrices formed for these experiments are shown in Appendix C. For all combinations of algorithm parameters, 10 independent training and test cycles are conducted in the image domain and are compared to the results of training and testing on the $\boldsymbol{IG}$ representations.

4.3.1 Results - F1-Scores

In this section the F1-scores found by testing and training in the CAD domain are compared to the F1-scores found by training and testing in the image domain. Figure 4.28 shows the trend in F1-scores for MPM and QMSE in both the CAD and image domain as pose errors get worse. The absolute error between the CAD and image domain F1-scores are provided. The F1-scores provided for the image domain is the mean F1-score from 10 independent train and test cycles for each parameter combination. The mean absolute difference between the F1-scores found in the CAD domain and SAL domain are provided for each possible parameter combination.

4.3.2 Results - Confusion Matrix Comparisons

The results in Section 4.3.1 shows that the F1-scores found in the image domain are similar to those found in the SAL domain. While the overall performance has been captured in the CAD domain, similar F1-scores can be found from confusion matrices with difference confusion and attractor class tendencies. Understanding how targets will be confused with other targets is a key part of ATR assessment. In this section we summarize the differences between the confusion matrices found in the CAD domain and SAL domain. The mean absolute difference between the CAD domain confusion matrix and SAL domain for all trials and pose errors. The mean of these errors are provided for each algorithm and parameter combination. Confusion errors for $nQ = 2$, $nQ = 4$, and $nQ = 8$, are shown in Figures 4.29, 4.30, and 4.31 respectively.

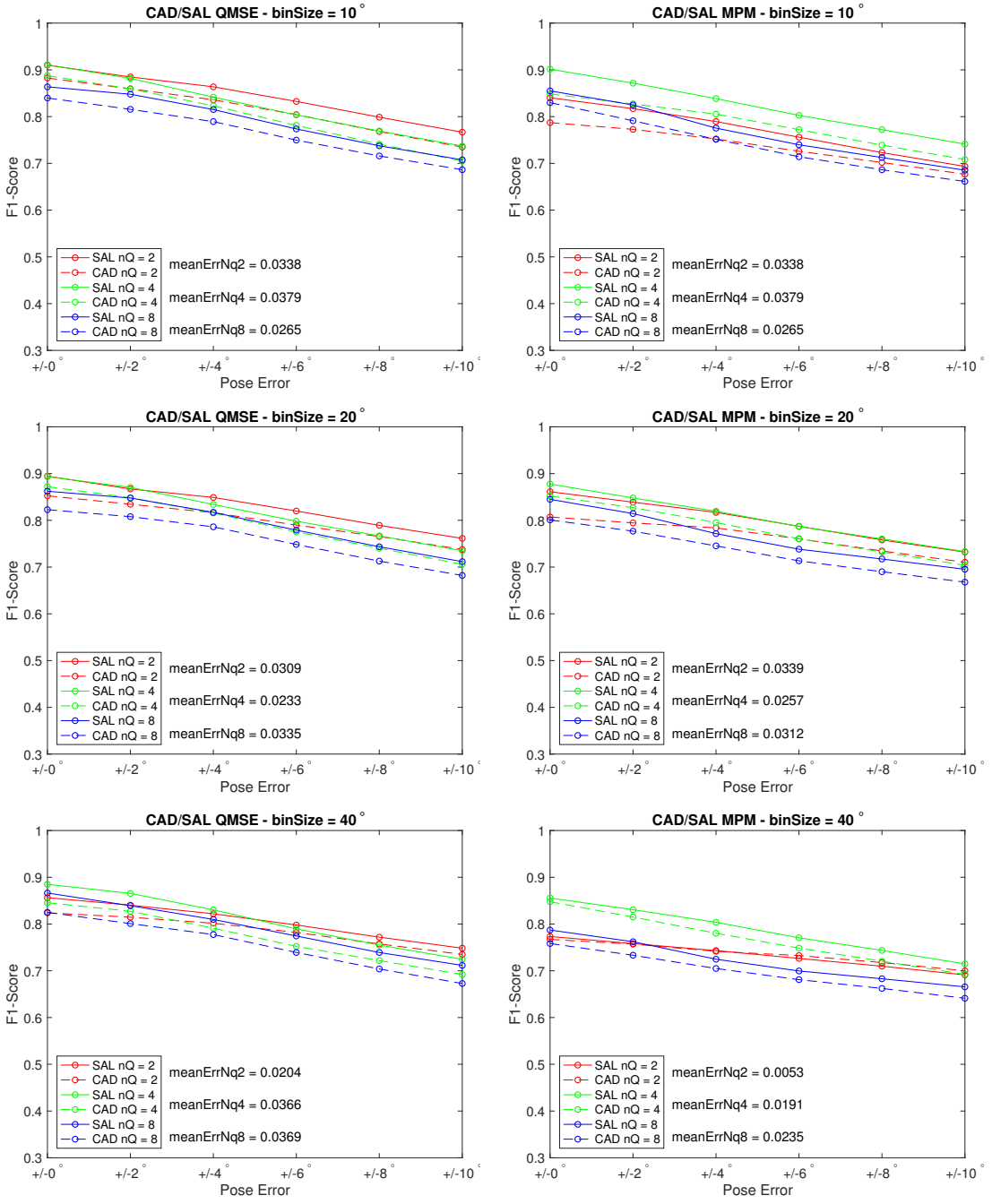


Figure 4.28: F1-scores for all parameter combinations and algorithms . The dashed lines represent the F1-scores found from the CAD domain. The solid lines represent the F1-scores from the image domain. Each point in the image domain is the mean F1-score for 10 trials. The CAD domain F1-scores follow similar trends in the image domain.

4.3.3 Results - Template Rankings

This section shows how well the scores assigned by templates formed in the CAD domain correlate with the scores assigned my SAL domain templates. While classifying test samples for both QMSE and MPM, the sample is scored against all templates from all classes and for the estimated pose. The scores assigned by each target template can be kept as a list of rankings and compared via Spearman correlation. If the Spearman correlation between two sets of rankings is 1.0000, then the samples were ranked identically. Tables 4.7 and 4.6 show the Spearman correlations between the SAL and CAD domain templates for each target and each combination of algorithm parameters.

	A	**S**	**T**	**C**	**M**
$nQ = 2, binSize = 10$	0.9197	0.9361	0.9412	0.9221	0.9313
$nQ = 2, binSize = 20$	0.9557	0.9551	0.9670	0.9575	0.9590
$nQ = 2, binSize = 40$	0.9600	0.9589	0.9676	0.9423	0.9520
$nQ = 4, binSize = 10$	0.9629	0.9552	0.9662	0.9623	0.9638
$nQ = 4, binSize = 20$	0.9630	0.9547	0.9670	0.9612	0.9656
$nQ = 4, binSize = 40$	0.9657	0.9482	0.9684	0.9674	0.9555
$nQ = 8, binSize = 10$	0.9522	0.9553	0.9624	0.9500	0.9600
$nQ = 8, binSize = 20$	0.9489	0.9537	0.9609	0.9551	0.9614
$nQ = 8, binSize = 40$	0.9426	0.9456	0.9653	0.9509	0.9460

Table 4.6: Spearman Correlation - CAD vs SAL Template Rankings - MPM

	A	**S**	**T**	**C**	**M**
$nQ = 2, binSize = 10$	0.9707	0.9700	0.9781	0.9623	0.9665
$nQ = 2, binSize = 20$	0.9718	0.9706	0.9771	0.9639	0.9698
$nQ = 2, binSize = 40$	0.9763	0.9796	0.9781	0.9691	0.9704
$nQ = 4, binSize = 10$	0.9722	0.9576	0.9779	0.9720	0.9747
$nQ = 4, binSize = 20$	0.9760	0.9525	0.9773	0.9740	0.9740
$nQ = 4, binSize = 40$	0.9737	0.9555	0.9797	0.9727	0.9773
$nQ = 8, binSize = 10$	0.9793	0.9629	0.9818	0.9792	0.9778
$nQ = 8, binSize = 20$	0.9804	0.9635	0.9831	0.9786	0.9796
$nQ = 8, binSize = 40$	0.9784	0.9630	0.9808	0.9747	0.9770

Table 4.7: Spearman Correlation - CAD vs SAL Template Rankings - QMSE

4.4 CAD Based Template Formation for SAL ATR

This section demonstrates the effectiveness of forming templates in the CAD domain and testing on the SAL domain. The ability to form templates in the CAD domain is more computationally efficient with the ***IG*** approach than forming images from ray-tracing and generating speckle fields. Section 4.3 demonstrates the ability to train and test in the CAD domain and reason about the classification trends when training and testing in the SAL image domain. This section aims to demonstrate that CAD domain representations of targets are a viable substitute for templates traditionally formed from images.

4.4.1 Results - F1-Scores

This section shows the difference in F1-scores found by training on CAD representation and testing on SAL imagery versus training and testing on SAL images. Figure 4.32 shows the F1-scores for both training scenarios for all combinations of algorithm parameters. Each F1-score when testing on images is the mean F1-score across 10 independent trials. The mean absolute error between both training scenarios are provided.

4.4.2 Results - Confusion Matrix Comparisons

This section provides a comparison of the confusion matrices for both training scenarios. Figures 4.33, 4.34, and 4.35 show the confusion errors between training on CAD and training on images for quantization levels $nQ = 2$, $nQ = 4$, and $nQ = 8$ respectively. The approach for forming these error matrices is the same as Section 4.3.2.

4.4.3 Results - Template Rankings

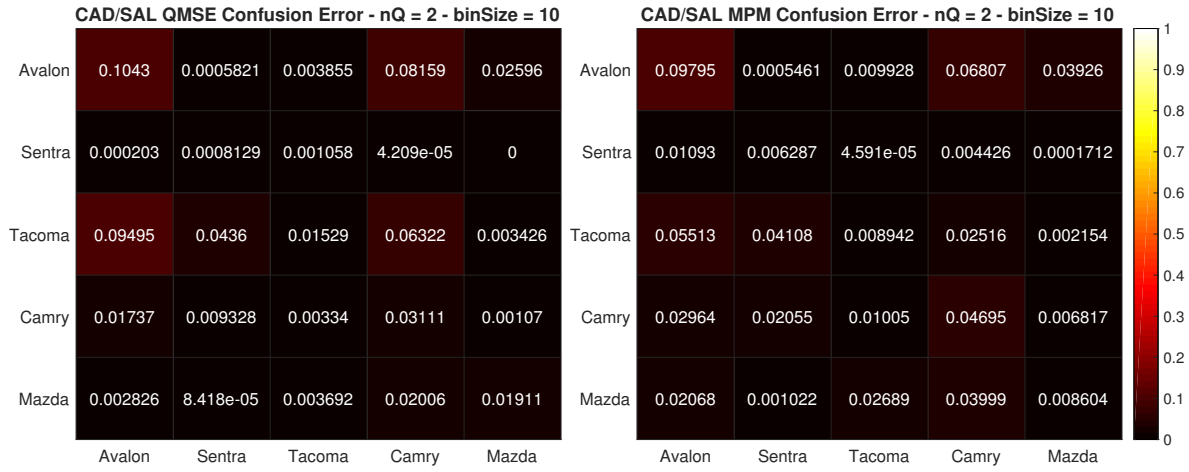
This section compares the CAD based template decisions and the image based template decisions when both classify SAL images. The rankings are found via Spearman correlation and follow the same approach as described in Section 4.3.3.

	A	**S**	**T**	**C**	**M**
$nQ = 2, binSize = 10$	0.9219	0.9464	0.9460	0.9282	0.9469
$nQ = 2, binSize = 20$	0.9589	0.9661	0.9704	0.9595	0.9646
$nQ = 2, binSize = 40$	0.9657	0.9620	0.9700	0.9481	0.9602
$nQ = 4, binSize = 10$	0.9684	0.9675	0.9657	0.9689	0.9671
$nQ = 4, binSize = 20$	0.9710	0.9669	0.9659	0.9692	0.9715
$nQ = 4, binSize = 40$	0.9678	0.9615	0.9651	0.9661	0.9629
$nQ = 8, binSize = 10$	0.9628	0.9726	0.9644	0.9619	0.9681
$nQ = 8, binSize = 20$	0.9630	0.9713	0.9650	0.9636	0.9702
$nQ = 8, binSize = 40$	0.9500	0.9623	0.9673	0.9563	0.9569

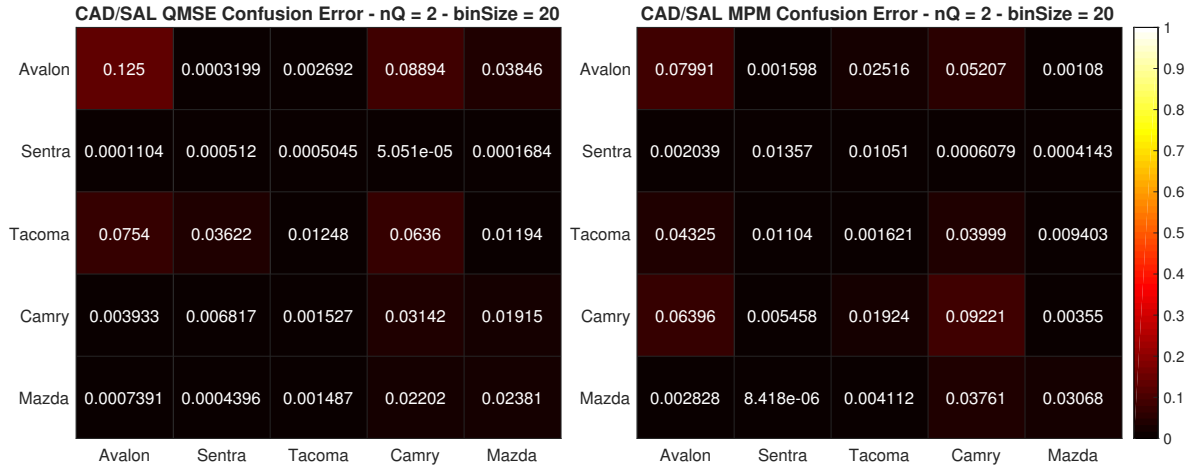
Table 4.8: Spearman Correlation - Train CAD Test SAL vs Train SAL Test SAL - MPM

	A	**S**	**T**	**C**	**M**
$nQ = 2, binSize = 10$	0.9752	0.9747	0.9769	0.9673	0.9726
$nQ = 2, binSize = 20$	0.9776	0.9768	0.9805	0.9702	0.9753
$nQ = 2, binSize = 40$	0.9799	0.9830	0.9802	0.9721	0.9754
$nQ = 4, binSize = 10$	0.9851	0.9638	0.9817	0.9839	0.9821
$nQ = 4, binSize = 20$	0.9875	0.9669	0.9845	0.9854	0.9823
$nQ = 4, binSize = 40$	0.9862	0.9687	0.9835	0.9857	0.9800
$nQ = 8, binSize = 10$	0.9888	0.9746	0.9830	0.9884	0.9854
$nQ = 8, binSize = 20$	0.9900	0.9768	0.9840	0.9887	0.9864
$nQ = 8, binSize = 40$	0.9876	0.9867	0.9835	0.9867	0.9830

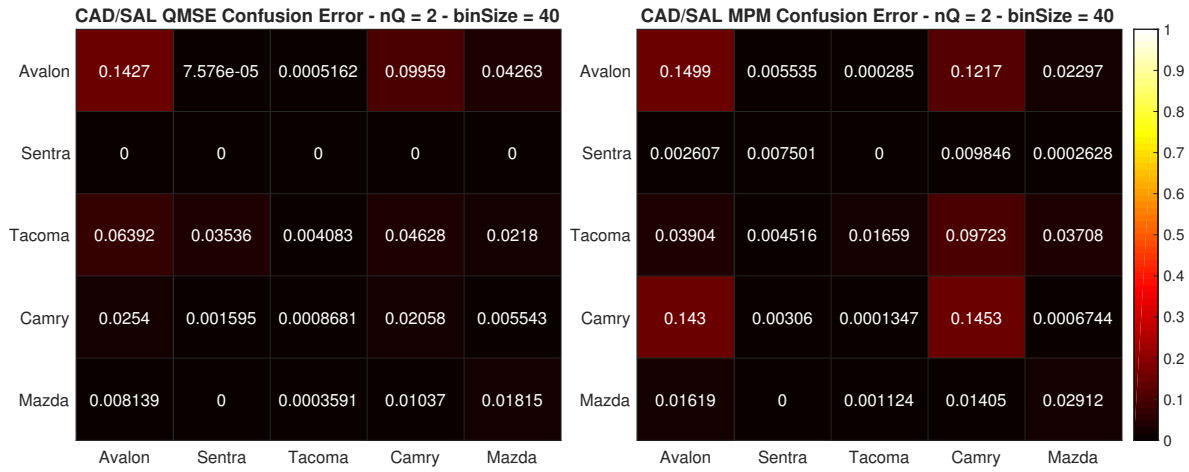
Table 4.9: Spearman Correlation - Train CAD Test SAL vs Train SAL Test SAL - QMSE



$meanErr = 0.0219$ $meanErr = 0.0232$

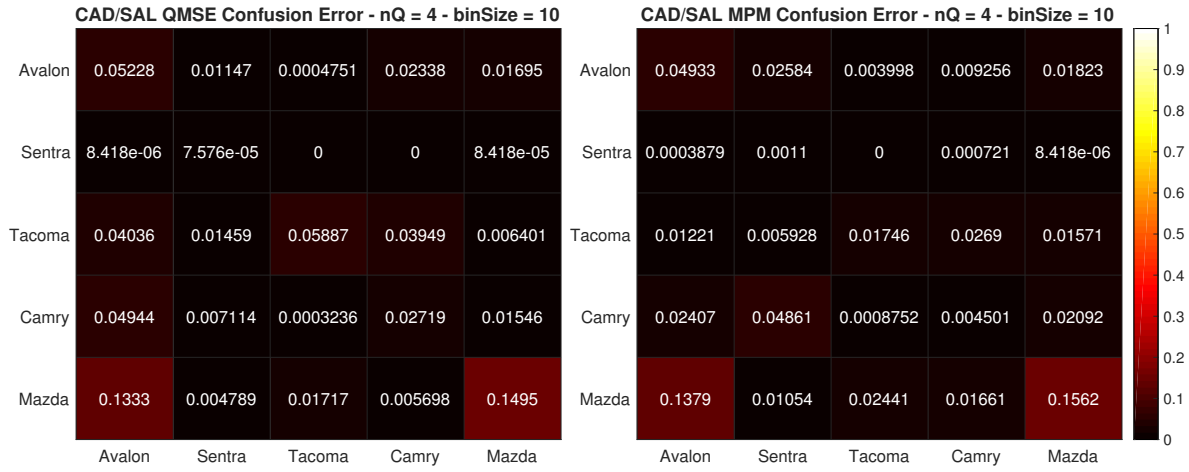


$meanErr = 0.0227$ $meanErr = 0.0221$

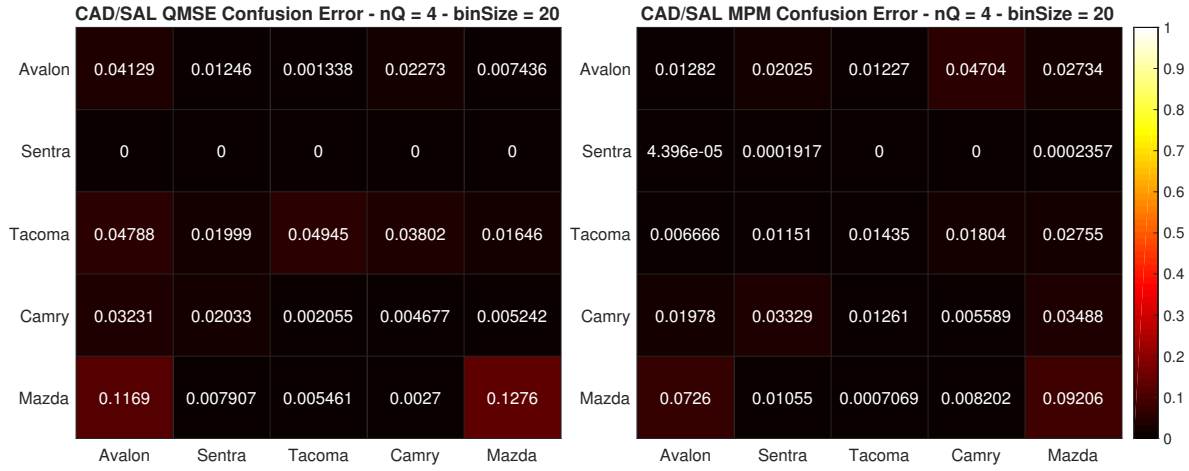


$meanErr = 0.0219$ $meanErr = 0.0347$

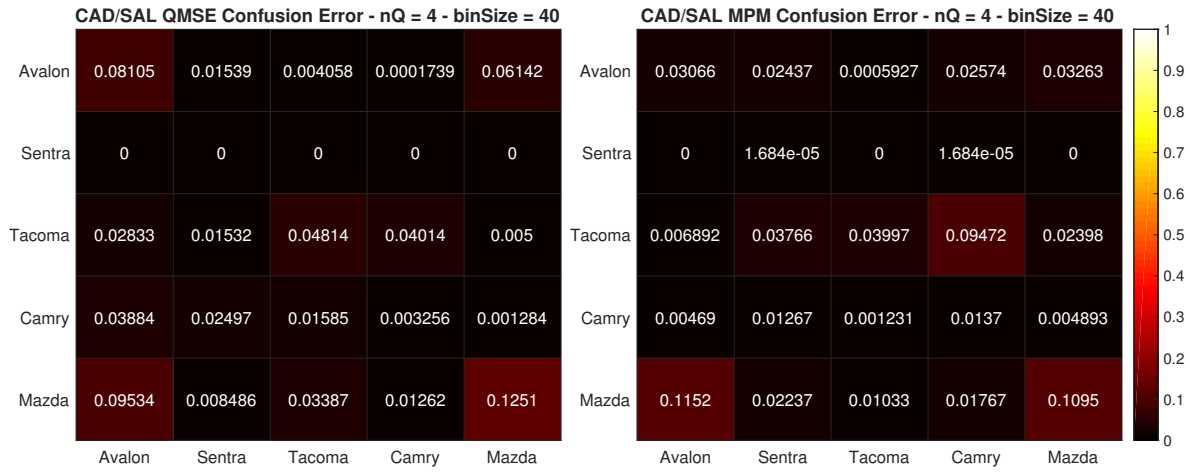
Figure 4.29: CAD and SAL Confusion Matrix Error - $nQ = 2$



$meanErr = 0.0270$ $meanErr = 0.0253$

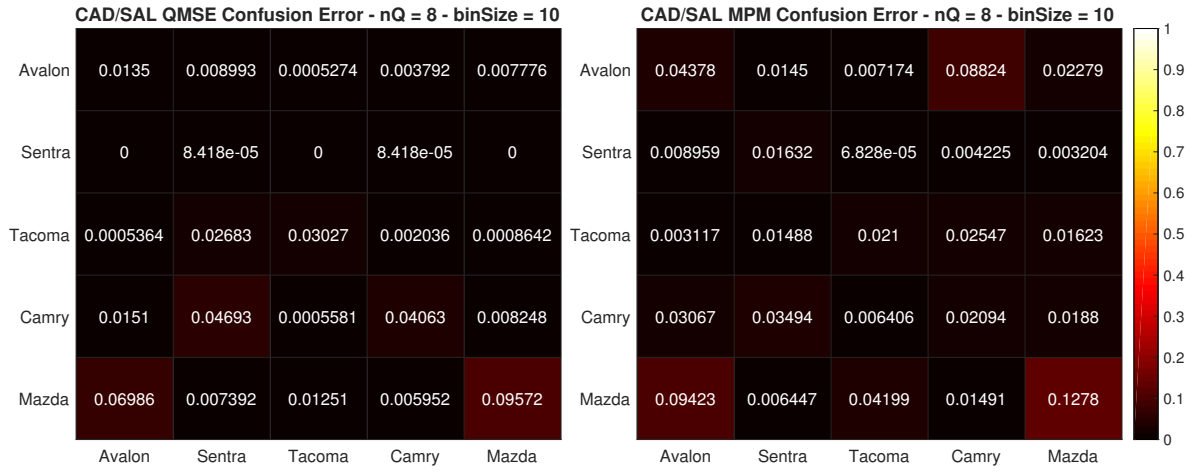


$meanErr = 0.0233$ $meanErr = 0.0195$

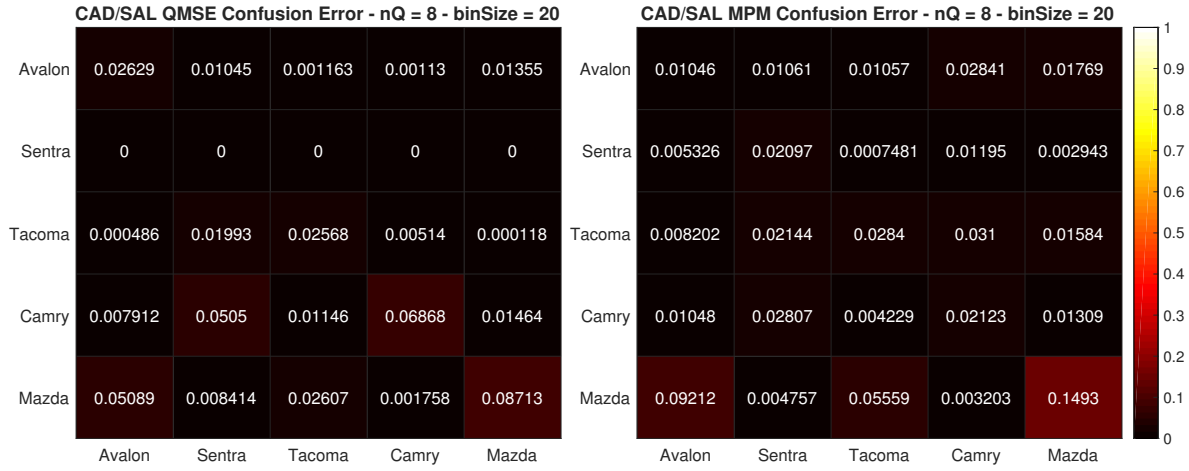


$meanErr = 0.0263$ $meanErr = 0.0252$

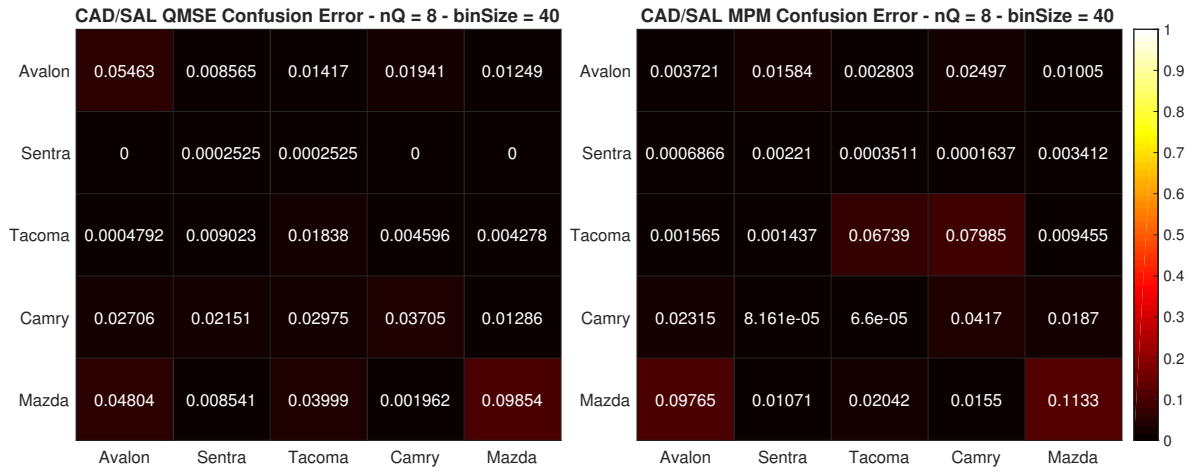
Figure 4.30: CAD and SAL Confusion Matrix Error - $nQ = 4$



$meanErr = 0.0159$ $meanErr = 0.0275$



$meanErr = 0.0173$ $meanErr = 0.0243$



$meanErr = 0.0189$ $meanErr = 0.0226$

Figure 4.31: CAD and SAL Confusion Matrix Error - $nQ = 8$

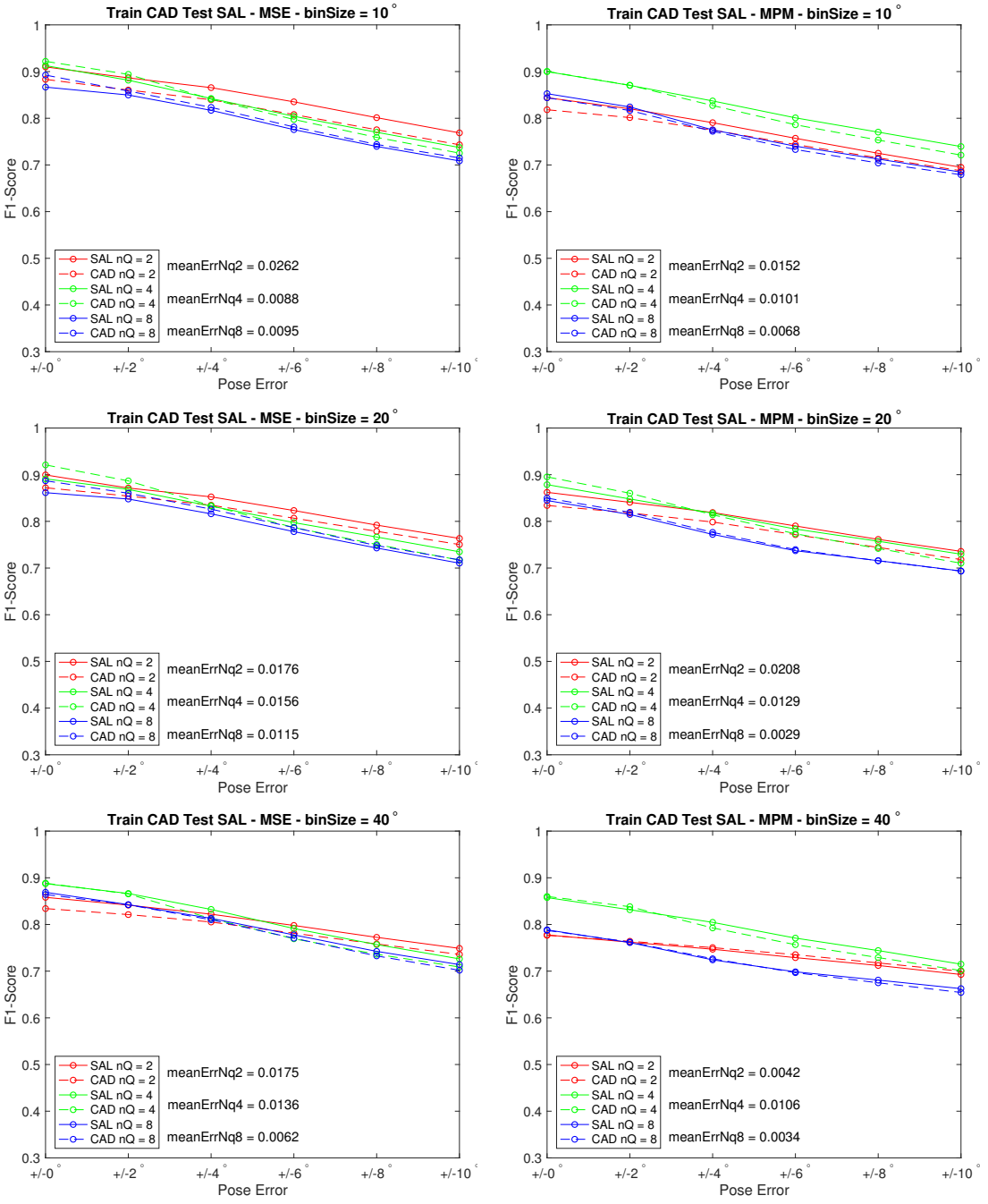
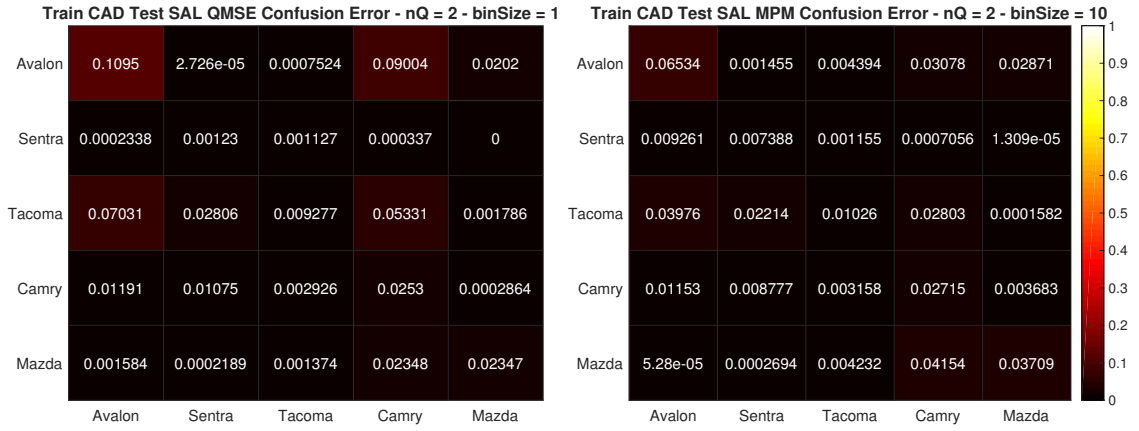
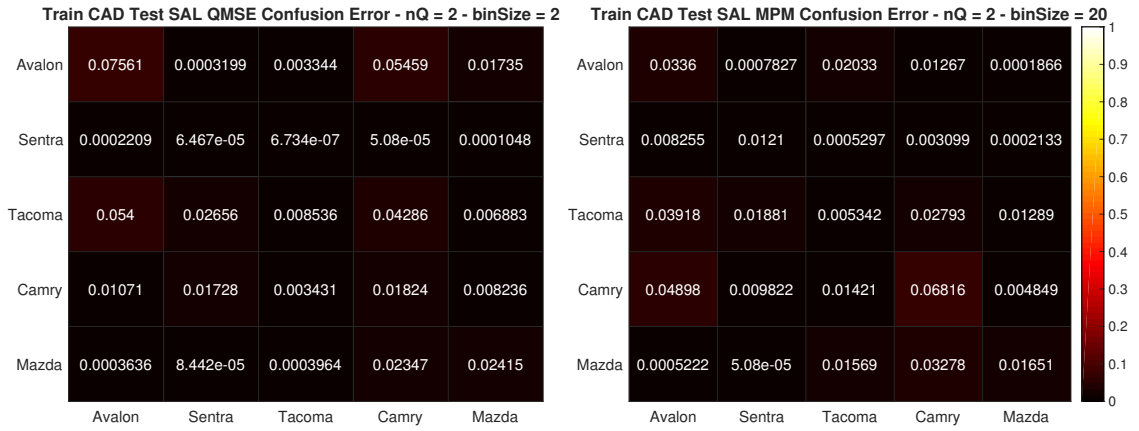


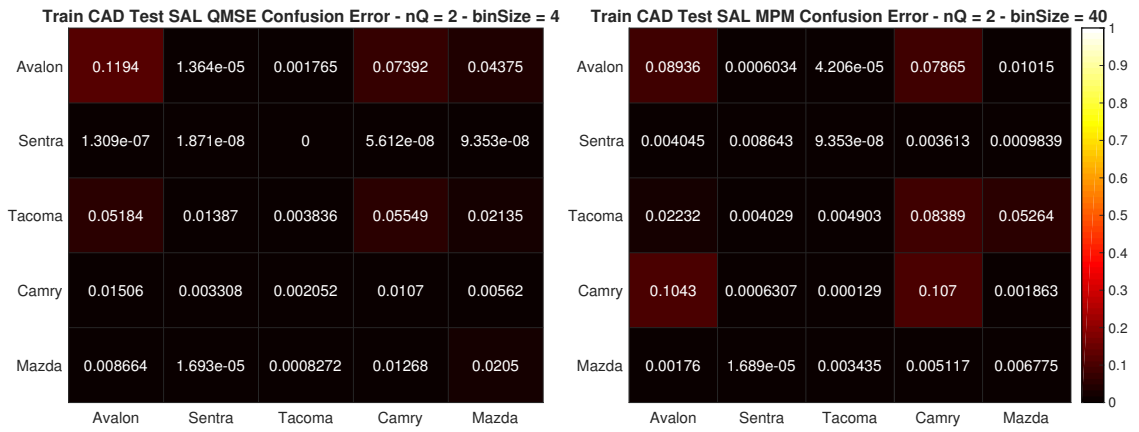
Figure 4.32: F1-scores when training on CAD representations and testing on images compared to F1-scores when training and testing on images. The dashed lines represent when templates were formed on CAD information. The solid lines represent when templates were formed on images.



$meanErr = 0.0315$ $meanErr = 0.0155$

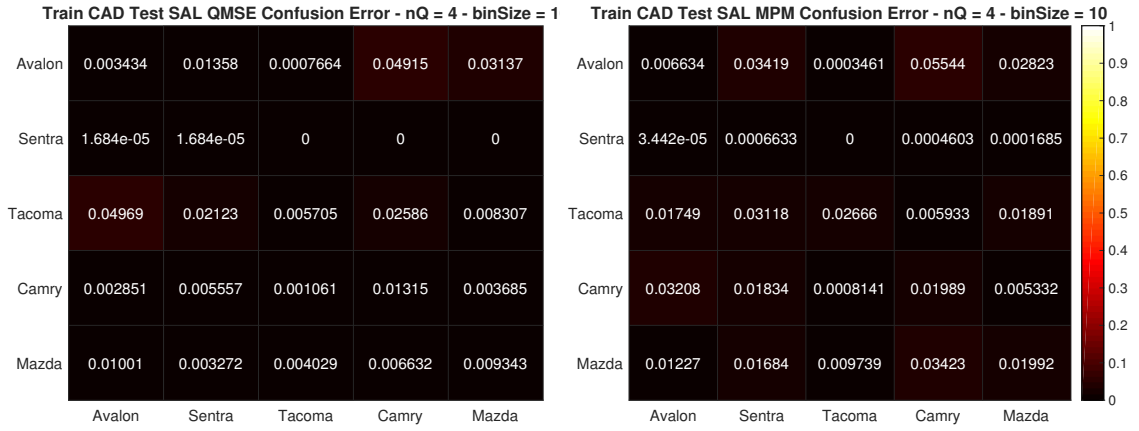


$meanErr = 0.0159$ $meanErr = 0.0163$

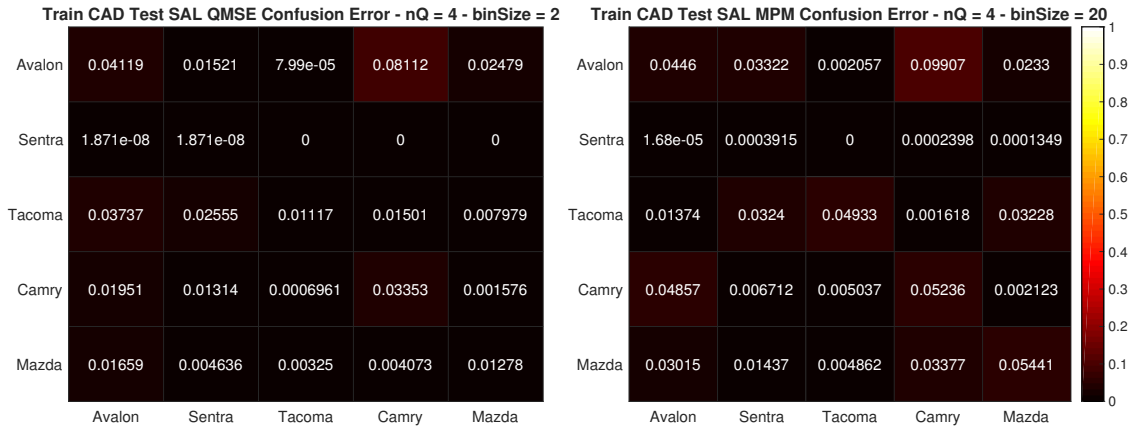


$meanErr = 0.0186$ $meanErr = 0.0238$

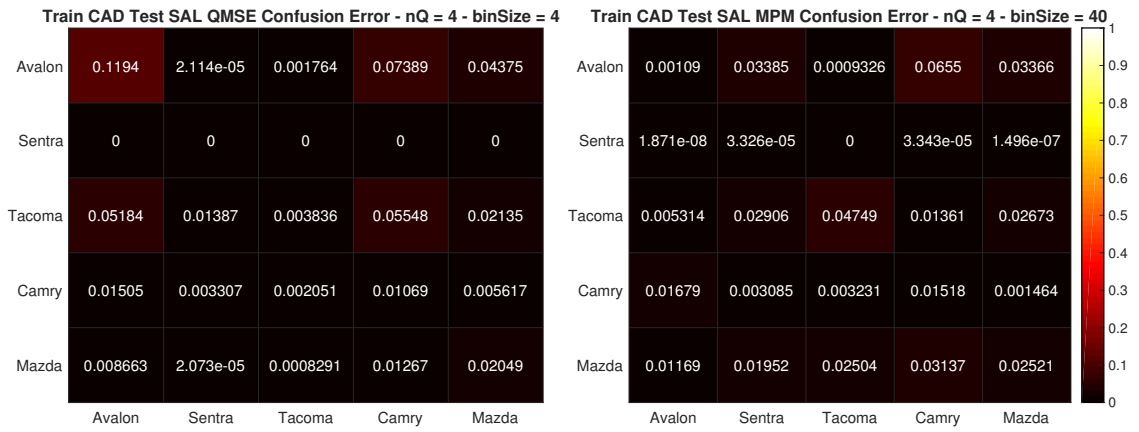
Figure 4.33: Train CAD Test SAL vs. Train SAL Test SAL - $nQ = 2$



$meanErr = 0.0107$ $meanErr = 0.0158$

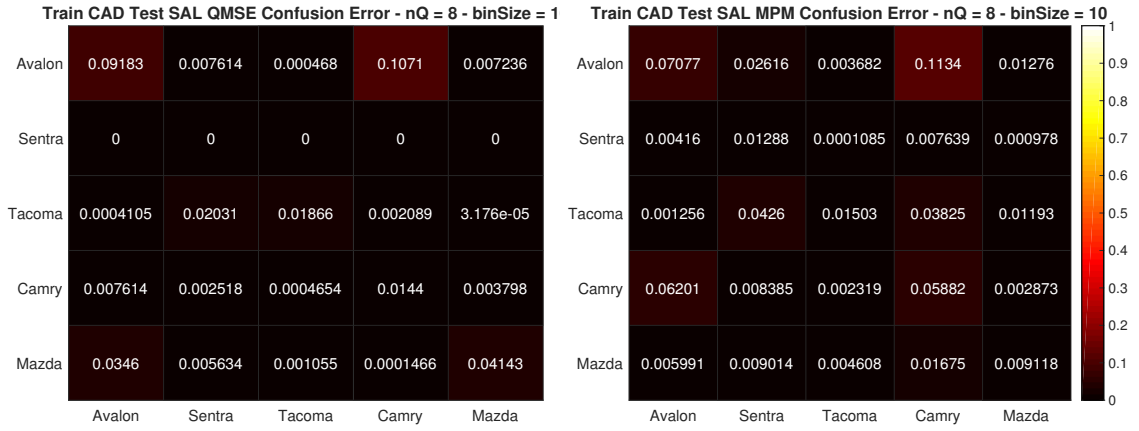


$meanErr = 0.0148$ $meanErr = 0.0234$

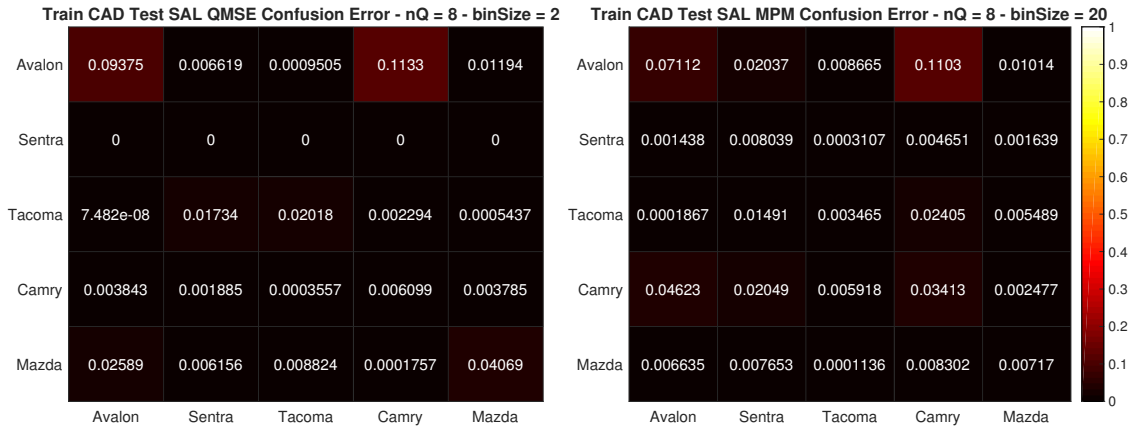


$meanErr = 0.0186$ $meanErr = 0.0164$

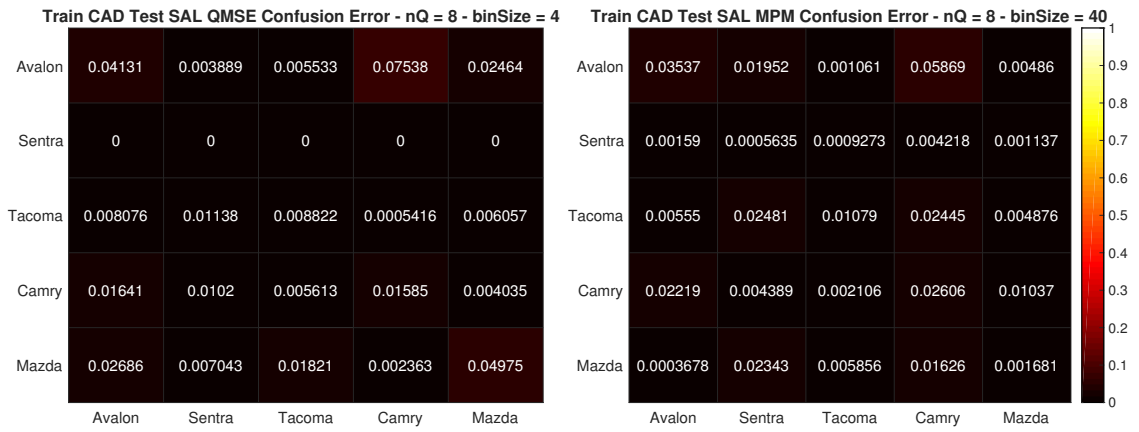
Figure 4.34: Train CAD Test SAL vs. Train SAL Test SAL - $nQ = 4$



$meanErr = 0.0147$ $meanErr = 0.0217$



$meanErr = 0.0146$ $meanErr = 0.0170$



$meanErr = 0.0137$ $meanErr = 0.0124$

Figure 4.35: Train CAD Test SAL vs. Train SAL Test SAL - $nQ = 8$

Discussion and Future Work

This chapter discusses the results found in Chapter 4 and proposes future work. The results from the template matching experiments are summarized and the key OC interactions are identified. Next, the performance prediction results from Section 4.3 are discussed. Finally, the feasibility of training on CAD based templates to classify SAL images is discussed.

5.1 Template Matching on SAL Images

Section 4.1 discusses the results of template matching classifiers on SAR and SAL imagery. As discussed in Section 4.1, $nQ = 2$ yields the best performance for the QMSE classifier when pixels are not blurred. As pose errors degrade, wider template bins increase SAR QMSE performance while SAL QMSE performs best on narrow bins despite pose errors. For both modalities, the optimal quantization scheme for MPM is $nQ = 4$. As previously discussed, MPM relies on additional intensity information to achieve it's best performance. As pose error gets worse, utilizing a different bin size or quantization level did not increase performance. For the SAL MPM case, there is relatively little difference in performance with respect to the selected bin size. For the SAR MPM case, there are large gaps amongst the parameter choices.

Speckle averaging and the effects of QPEs on template matching performance were explored in Section 4.1. At the resolution used to perform the experiments, there are diminishing returns on performance for speckle averaging after $n_{sa} = 5$. QPE causes features

to blur together. In the QMSE case, quantizing pixels to more levels increase performance as phase errors get worse. In the MPM case, additional quantization levels does not aid in performance. As phase errors get worse, $nQ = 4$ remains the best choice for quantization. Additionally, the choice in bin sizes converge.

Finally, all possible combinations of experiment parameters were tested. The statistical significance of all factors and factor interactions are explored. Additionally, interactions among OCs and algorithm parameters are shown. According to Figure 4.14a The significant interactions for SAL MSE are:

- $n_{SA} * nQ$ - at low levels of speckle averaging, low nQ levels are preferred. As speckle averaging increases, $nQ = 4$ becomes the best choice.
- $QPE * nQ$ - as QPE increases, higher levels of nQ are preferred.
- $poseError * nQ$ - as pose error gets worse, the effectiveness of $nQ = 2$ increases, but, $nQ = 4$ is still preferred.

The number of duplicate images in a template does not interact with the other factor parameters. Additionally, binSize does not have much interaction with other factors.

The significant interactions for SAL MPM are:

- $n_{SA} * nQ$ - while $nQ = 4$ is the optimal setting, $nQ = 2$ becomes the second best option as SA increases.
- $n_{SA} * binSize$ - the 10° bin size is the best choice when SA is low, then 20° is the optimal size when SA is at the medium levels, and then 10° is the optimal choice at high levels of speckle averaging.
- $binSize * nQ$ - at low nQ levels, 20° bin size is the optimal choice. As nQ increases, the bin size of 10° is optimal.

- $QPE{*}binSize$ - At low levels of QPE, the 20° bin size is optimal. As QPE increases, the size of 10° becomes the best choice.

- The nQ parameters has significant interactions with the other parameters when $nQ = 2$ and $nQ = 8$. The optimal setting is $nQ = 4$.

According to Figure 4.15a, the only meaningful interaction for SAR MSE is the interaction between binSize and poseError. As pose error increases, the higher levels of binSize are preferred. The significant interactions for SAR MPM are nDups * nQ and binSize * nQ. In both cases, $nQ = 4$ is the optimal choice until nDups and binSize increases to larger levels.

5.2 CAD Based Performance Prediction

This section discusses the ability to utilize ***IG*** CAD representations to predict the performance of MPM and QMSE on SAL imagery. Section 4.3.1 shows that the F1-scores found in the CAD domain are similar to those found in the SAL domain for all tested algorithm parameters. Overall trends in F1-score as pose errors get worse are similar in both domains. For instance, in the QMSE case the best overall performing quantization level in $nQ = 2$. However, $nQ = 4$ has similar or better performance when pose error is minimal. This trend still holds true in the CAD domain. When $binSize = 40^{\circ}$, $nQ = 4$ is the best performance quantization level until $nQ = 2$ becomes the best performing quantization level at $\pm 6^{\circ}$. This trend holds in the CAD domain as well. In the MPM case, the best performing quantization level is $nQ = 4$. This is shown in both the CAD domain and SAL domain. When $binSize = 20^{\circ}$, the performance between $nQ = 2$ and $nQ = 4$ converges as pose errors get worse in both the CAD and SAL domain.

While the overall performance of MPM and QMSE can be predicted by CAD domain analysis, it is possible to achieve similar F1-scores with dissimilar confusion matrices. Section 4.3.2 compares the CAD domain and SAL domain confusion matrices. The error

matrices provided contains the mean absolute error between CAD domain confusion matrix elements and SAL domain confusion matrix elements, across all pose errors. Then, the mean error of these error matrices is provided. While the over all error between the domains is low, there are instances where the CAD domain over predicts. For example, in the $nQ = 2$ case for both algorithms the CAD domain tends to correctly classify the Avalon more so than the image domain. In the $nQ = 4$ case, there are discrepancies for the Mazda target for both algorithms. This trend also remains for the $nQ = 8$ case. The error matrices provided in this section aim to summarize the differences between the CAD and SAL domain confusion matrices. A more robust analysis can be done via visual inspection of all confusion matrices. All confusion matrices can be found in Appendix C.

When classifying images, all test images are compared against the templates for all targets for the predicted pose. Thus, each target will have a list of scores assigned for all test samples. These lists of scores can be compared via Spearman correlation. Specifically, the CAD domain template scores can be compared to the SAL domain template scores. Tables 4.6 and 4.7 show the Spearman correlation scores between the CAD and SAL domain templates. These values indicate that scores assigned by templates across the CAD and SAL domain are strongly correlated. Thus, analyzing trends in template scores in the CAD domain will be useful for predicting the same trends in the image domain.

5.3 CAD Based Template Formation

This section discusses the feasibility of training on CAD based templates for classifying SAL images. Since the CAD domain representations can be generated with less computation burden than image based templates, the ability to generate templates with CAD representations is desirable. Section 4.4.1 shows that the F1-scores found entirely in the image domain are similar to those found when substituting the templates formed in the CAD domain. Additionally, the overall trends discussed in the previous section still hold

when substituting in CAD based templates. The F1-score trends when training on CAD based templates and testing on SAL imagery are better represented than when conducting the ATR entirely in the CAD space. This highlights a trade-off of prediction capabilities. Better estimates of F1-score can be found on CAD based templates, however, images will have to be formed and are more computationally burdensome than operating entirely on CAD representations. The confusion matrix analysis shown in Section 4.4.2 shows minimal difference between the confusion matrices found by training on CAD representations and training on SAL images. The higher errors occur in the same matrix elements as discussed in the previous section, however, to a lesser degree. Finally, Tables 4.9 and 4.8 show that the classification decisions of CAD based templates and image based templates are strongly correlated for all tested algorithm parameters.

5.4 Summary

This dissertation intended to show that performance prediction of SAL template matching ATRs can be predicted accurately and rapidly by analyzing the physical properties of CAD models. The $\boldsymbol{IG}$ representations presented in Section 4.2.2 gives representations of targets analogous to SAL images but in a more timely fashion as shown in Table 4.4.

These CAD based representations are highly correlated with their SAL image counterpart as shown in Figures 4.27 and 4.26. Due to these representations being strongly correlated, $\boldsymbol{IG}$ representations of targets can replace SAL images in templates and still yield similar results to the traditional image based templates. Section 4.3.1 shows the F1-scores predicted in the CAD domain are similar to those found in the image domain. Section 4.3.2 shows minimal error between confusion matrices found in the CAD domain versus the confusion matrices found in the SAL domain. Section 4.3.3 shows the scores assigned by CAD based templates on $\boldsymbol{IG}$ representations of targets are strongly correlated as the rankings in the image domain. These results provide evidence that CAD domain analysis of template

matching algorithm will accurately predict performance trends in the SAL image domain.

In addition to results shown in the previously discussed sections, additional results are shown from training on $\boldsymbol{IG}$ representations and testing on SAL images. Section 4.4.1 shows the F1-scores when training on $\boldsymbol{IG}$ are similar to the F1-scores when training on SAL images. Section 4.4.2 shows the confusion matrix error between templates formed on $\boldsymbol{IG}$ and traditional image based templates is minimal. Section 4.4.3 shows the ranking on SAL imagery for both CAD and image based templates are highly similar. These results provide evidence that CAD based templates are suitable substitutes for image based templates when conducting empirical performance prediction studies on SAL imagery.

5.5 Future Work

This dissertation conducted the first ATR experiments comparing performance and OC analysis for SAR and SAL. In order to extend the body of knowledge on the effects of switching from specular to diffuse returns on ATR performance, more OCs and algorithms can be explored. For instance, neural networks have been shown to be highly effective at classifying SAR imagery, but has not been explored on SAL imagery. In the SAL OC space, more work can be done to analyze atmospheric effects. This dissertation utilizes generic quadratic phase error. However, different parts of the world will have drastically different atmospheric profiles that may yield phase errors other than quadratic. The relationship between speckle and quantization levels was discussed in Chapter 4. The analysis between speckle and quantization levels was done empirically and only assuming Lambertian responses. A more theoretical approach to this type of behavior may lead to statistical methods for SAL ATR with respect to speckle. The effects of speckle and resolution can also be explored further. As resolution becomes finer, more speckle averaging must be done in order to mitigate the speckle effect. Chapter 4 shows that minimal speckle averaging is needed to make performance converge. However, the trade-off between speckle averaging

and resolution has yet to be explored.

The ATR community is always seeking out more sources of measured data. Currently, there is no publicly available SAL dataset. A gold standard measured SAL dataset will enable the ATR community to conduct MSTAR-like research on military targets [2]. A gold standard measured dataset will open the floodgates with respect to SAL ATR design. Besides just ATR assessment, a measured dataset will allow the ATR community to delve into understanding the measured and synthetic gap for SAL data. A measured SAL dataset will also help further verify the effectiveness of the CAD based performance prediction and template generation.

The methodology for CAD based performance prediction presented in this dissertation maps well for a variety of algorithm parameters on pristine imaging parameters. Further iterations of the IG model must be done in order to include the analysis of image domain OCs, such as speckle and atmospheric noise. The IG model converts a 3-D mesh to 2-D pixel space. It may be the case that the information lost in the conversion may be useful for ATR analysis. The conversion from 3-D mesh to 2-D pixel space was a natural step in order to form MPM and QMSE templates. However, non-template based ATR algorithms may be better suited learning features from the 3-D mesh in a different form. Transfer learning techniques may be useful in order to map CAD representations of targets to measured or synthetic SAL images.

At a broad level, SAL is another tool in a very large sensing technology toolbox. While SAL is a promising candidate for ATR, it will be important to understand its strengths and weaknesses with respect to other sensing modalities. The fusion capabilities of SAL with other sensors is a broad research area that can not only improve and advance SAL ATR, but ATR for other sensing technology as well.

Bibliography

[1] Navy dsrc - high performance computing systems. https://www.navydsrc.hpc.mil/hardware/index.html. Accessed: 2022-2-19.

[2] Sdms - mstar. https://www.sdms.afrl.af.mil/index.php?collection=mstar. Accessed: 2017-11-28.

[3] Carl C Aleksoff. Optical synthetic aperture techniques. Technical report, ENVIRONMENTAL RESEARCH INST OF MICHIGAN ANN ARBOR INFRARED AND OPTICS DIV, 1985.

[4] CC Aleksoff, JS Accetta, LM Peterson, AM Tai, A Klooster, KS Schroeder, RM Majewski, JO Abshier, and M Fee. Synthetic aperture imaging with a pulsed co2 tea laser. In *Laser Radar II*, volume 783, pages 29–41. International Society for Optics and Photonics, 1987.

[5] Georgios C Anagnostopoulos. Svm-based target recognition from synthetic aperture radar images using target region outline descriptors. *Nonlinear Analysis: Theory, Methods & Applications*, 71(12):e2934–e2939, 2009.

[6] Fabrizio Argenti, Alessandro Lapini, Tiziano Bianchi, and Luciano Alparone. A tutorial on speckle reduction in synthetic aperture radar images. *IEEE Geoscience and remote sensing magazine*, 1(3):6–35, 2013.

[7] Zeb Barber. Synthetic aperture ladar imaging and atmospheric turbulence. Technical report, Montana State Univ Bozeman Bozeman United States, 2016.

[8] Rejean Baribeau and Marc Rioux. Influence of speckle on laser range finders. *Applied optics*, 30(20):2873–2878, 1991.

[9] M Bashkansky and J Reintjes. Statistics and reduction of speckle in optical coherence tomography. *Optics Letters*, 25(8):545–547, 2000.

[10] Michael Lee Bryant and Frederick D Garber. Svm classifier applied to the mstar public data set. In *Algorithms for Synthetic Aperture Radar Imagery VI*, volume 3721, pages 355–361. International Society for Optics and Photonics, 1999.

[11] AL Buck. Effects of the atmosphere on laser beam propagation. *Applied optics*, 6(4):703–708, 1967.

[12] Joseph Buck, B W. Krause, A I. Malm, and C M. Ryan. Synthetic aperture imaging at optical wavelengths. In *Conference on Lasers and Electro-Optics/International Quantum Electronics Conference*, page PThB3. Optical Society of America, 2009.

[13] Jack L Bufton, Peter O Minott, Michael W Fitzmaurice, and Paul J Titterton. Measurements of turbulence profiles in the troposphere. *JOSA*, 62(9):1068–1070, 1972.

[14] Changjie Cao, Zongjie Cao, and Zongyong Cui. Ldgan: A synthetic aperture radar image generation method for automatic target recognition. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5):3495–3508, 2019.

[15] Sizhe Chen and Haipeng Wang. Sar target recognition based on deep learning. In *Data Science and Advanced Analytics (DSAA), 2014 International Conference on*, pages 541–547. IEEE, 2014.

[16] Thomas R Crimmins. Geometric filter for speckle reduction. *Applied optics*, 24(10):1438–1443, 1985.

[17] JC Dainty. The Statistics of Speckle Patterns. *Progress in Optics*, 14:1–46, 1977.

[18] Randy S Depoy and Arnab K Shaw. Algorithm to overcome atmospheric phase errors in SAL data. *Applied optics*, 59(1):140–150, 2020.

[19] Jun Ding, Bo Chen, Hongwei Liu, and Mengyuan Huang. Convolutional neural network with data augmentation for sar target recognition. *IEEE Geoscience and remote sensing letters*, 13(3):364–368, 2016.

[20] Armin W Doerry. Basics of polar-format algorithm for processing synthetic aperture radar images. *Sandia National Laboratories report SAND2012-3369, Unlimited Release*, 2012.

[21] Kerry E Dungan, Lee C Potter, Jason Blackaby, and John Nehrbass. Discrimination of civilian vehicles using wide-angle sar. *Algorithms for Synthetic Aperture Radar Imagery XV*, 6970:69700Z, 2008.

[22] David J Eck. *Introduction to Computer Graphics*. David J. Eck, 2016.

[23] Zi-Jun Feng, Xiao-Ling Zhang, Li-Yong Yuan, and Jia-Nan Wang. Infrared target detection and location for visual surveillance using fusion scheme of visible and infrared images. *Mathematical Problems in Engineering*, 2013, 2013.

[24] Jonathan Foote. Visualizing music and audio using self-similarity. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 77–80, 1999.

[25] Lianru Gao, Bin Yang, Qian Du, and Bing Zhang. Adjusted spectral matched filter for target detection in hyperspectral imagery. *Remote sensing*, 7(6):6611–6634, 2015.

[26] Joseph W Goodman. Some fundamental properties of speckle. *JOSA*, 66(11):1145–1150, 1976.

[27] LeRoy A Gorham and Linda J Moore. Sar image formation toolbox for matlab. In *Algorithms for Synthetic Aperture Radar Imagery XVII*, volume 7699, page 769906. International Society for Optics and Photonics, 2010.

[28] Thomas J Green, Stephen Marcus, and Barry D Colella. Synthetic-aperture-radar imaging with a solid-state laser. *Applied optics*, 34(30):6941–6949, 1995.

[29] Kristjan H Greenewald. Prediction of optimal bayesian classification performance for ladar atr. 2012.

[30] Liang Guo, Hongfei Yin, Xiaodong Zeng, Mengdao Xing, and Yu Tang. Analysis of airborne synthetic aperture ladar imaging with platform vibration. *Optik-International Journal for Light and Electron Optics*, 140:171–177, 2017.

[31] Michael Hazlett, Dennis J Andersh, Shung Wu Lee, Hao Ling, and CL Yu. Xpatch: a high-frequency electromagnetic scattering prediction code using shooting and bouncing rays. In *Targets and Backgrounds: Characterization and Representation*, volume 2469, pages 266–275. International Society for Optics and Photonics, 1995.

[32] Matthew Horvath and Brian Rigling. Performance prediction of quantized sar atr algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(1):189–204, 2016.

[33] Matthew S. Horvath and Brian D. Rigling. Multinomial pattern matching revisited, 2015.

[34] Zhili Hua, Hongping Li, and Yongjian Gu. Atmosphere turbulence phase compensation in synthetic aperture ladar data processing, 2007.

[35] Robert E Hufnagel. Propagation through atmospheric turbulence. *The Infrared Handbook*, 6:1–56, 1978.

[36] William W. Irving and Gil J. Ettinger. Classification of targets in synthetic aperture radar imagery via quantized grayscale matching, 1999.

[37] Thomas J. Karr. Synthetic aperture ladar for planetary sensing, 2003.

[38] Thomas J. Karr. Synthetic aperture ladar resolution through turbulence, 2003.

[39] Thomas J. Karr, John H. Glezen, and Henry E. Lee. Phase and frequency stability for synthetic aperture ladar, 2007.

[40] Eric R Keydel, Shung W Lee, and John T Moore. Mstar extended operating conditions: A tutorial. In *Aerospace/Defense Sensing and Controls*, pages 228–242. International Society for Optics and Photonics, 1996.

[41] Eugene F Knott, John F Schaeffer, and Michael T Tulley. *Radar cross section*. SciTech Publishing, 2004.

[42] Melissa L. Koudelka, John A. Richards, and Mark W. Koch. Multinomial pattern matching for high range resolution radar profiles, 2007.

[43] Brian Krause, Joseph Buck, Christopher Ryan, David Hwang, Piotr Kondratko, Andrew Malm, Andrew Gleason, and Shaun Ashby. Synthetic aperture ladar flight demonstration. In *CLEO: Science and Innovations*, page PDPB7. Optical Society of America, 2011.

[44] Thomas G Kyle. High resolution laser imaging system. *Applied optics*, 28(13):2651–2656, 1989.

[45] Ellen E Laubie, Brian D Rigling, and Robert P Penno. An empirical look at cross-target correlation in bistatic sar images. In *2018 IEEE Radar Conference (RadarConf18)*, pages 0531–0536. IEEE, 2018.

[46] Jong-Sen Lee. Speckle suppression and analysis for synthetic aperture radar images. *Optical engineering*, 25(5):255636, 1986.

[47] Benjamin Lewis, Jennifer Liu, and Amy Wong. Generative adversarial networks for sar image realism. In *Algorithms for Synthetic Aperture Radar Imagery XXV*, volume 10647, page 1064709. International Society for Optics and Photonics, 2018.

[48] Benjamin Lewis, Theresa Scarnati, Elizabeth Sudkamp, John Nehrbass, Stephen Rosencrantz, and Edmund Zelnio. A sar dataset for atr development: the synthetic and measured paired labeled experiment (sample). In *Algorithms for Synthetic Aperture Radar Imagery XXVI*, volume 10987, page 109870H. International Society for Optics and Photonics, 2019.

[49] TS Lewis and HS Hutchins. A synthetic aperture at 10.6 microns. *Proceedings of the IEEE*, 58(10):1781–1782, 1970.

[50] R. L. Lucke. Synthetic aperture ladar simulations with phase screens and fourier propagation. In *Aerospace Conference, 2004. Proceedings. 2004 IEEE*, volume 3, page 1798 Vol.3, March 2004.

[51] Robert L Lucke, Lee J Rickard, Mark Bashkansky, John Reintjes, and Eric E Funk. Synthetic Aperture LADAR (SAL): Fundamental Theory, Design Equations for a Satellite System, and Laboratory Demonstration. Technical report, DTIC Document, 2002.

[52] Stephen Marcus, Barry D Colella, and Thomas J Green. Solid-state laser synthetic aperture radar. *Applied optics*, 33(6):960–964, 1994.

[53] Linda J Moore, Brian D Rigling, Robert P Penno, and Edmund G Zelnio. Using phase for radar scatterer classification. In *Algorithms for Synthetic Aperture Radar Imagery XXIV*, volume 10201, page 102010J. International Society for Optics and Photonics, 2017.

[54] David AE Morgan. Deep convolutional neural networks for atr from sar imagery. *Proceedings of the Algorithms for Synthetic Aperture Radar Imagery XXII, Baltimore, MD, USA*, 23:94750F, 2015.

[55] John C Mossing and Timothy D Ross. Evaluation of sar atr algorithm performance sensitivity to mstar extended operating conditions. In *Aerospace/Defense Sensing and Controls*, pages 554–565. International Society for Optics and Photonics, 1998.

[56] Robert M. Neuroth, Brian D. Rigling, Edmund G. Zelnio, Edward A. Watson, Vincent J. Velten, and Todd V. Rovito. Asymptotic Modeling of Synthetic Aperture Ladar Sensor Phenomenology. *Proc. SPIE*, 9475:94750D–94750D–6, 2015.

[57] Adam Nolan, Brad Keserich, Andrew Lingg, and Steve Goley. Geometric saliency to characterize radar exploitation performance. In *Algorithms for Synthetic Aperture Radar Imagery XXI*, volume 9093, page 90930D. International Society for Optics and Photonics, 2014.

[58] Christopher Paulson, Edmund Zelnio, LeRoy Gorham, and Dapeng Wu. Using glint to perform geometric signature prediction and pose estimation. In *Algorithms for Synthetic Aperture Radar Imagery XIX*, volume 8394, page 83940R. International Society for Optics and Photonics, 2012.

[59] Jose C Principe, Dongxin Xu, and John W Fisher III. Pose estimation in sar using an information theoretic criterion. In *Algorithms for synthetic aperture radar imagery V*, volume 3370, pages 218–229. International Society for Optics and Photonics, 1998.

[60] Andrew Profeta, Andres Rodriguez, and H Scott Clouse. Convolutional neural networks for synthetic aperture radar classification. In *Algorithms for Synthetic Aperture Radar Imagery XXIII*, volume 9843, page 98430M. International Society for Optics and Photonics, 2016.

[61] Brian D. Rigling. Raider Tracer: a MATLAB-based Electromagnetic Scattering Simulator, 2007.

[62] Brian D. Rigling, Austin Mackey, Edward M. Friel, John W. Nehrbass, and Edmund G. Zelnio. Recent Improvements to the Raider Tracer Scattering Prediction Tool, 2014.

[63] Steven K Rogers, John M Colombi, Curtis E Martin, James C Gainey, Ken H Fielding, Tom J Burns, Dennis W Ruck, Matthew Kabrisky, and Mark Oxley. Neural networks for automatic target recognition. *Neural networks*, 8(7):1153–1184, 1995.

[64] Michael C Roggemann, Byron M Welsh, and Bobby R Hunt. *Imaging Through Turbulence*. CRC press, 1996.

[65] Jason D Roos and Arnab K Shaw. Probabilistic svm for open set automatic target recognition on high range resolution radar data. In *Automatic Target Recognition XXVII*, volume 10202, page 102020B. International Society for Optics and Photonics, 2017.

[66] Jacob W Ross, Brian D Rigling, and Edward A Watson. Analysis of speckle and material properties in laider tracer. In *SPIE Defense+ Security*, pages 1020102–1020102. International Society for Optics and Photonics, 2017.

[67] Timothy D. Ross, Jeff J. Bradley, Lannie J. Hudson, and Michael P. O'Connor. Sar atr: so what's the problem? an mstar perspective, 1999.

[68] Timothy D. Ross and John C. Mossing. Mstar evaluation methodology, 1999.

[69] Timothy D. Ross, Steven W. Worrell, Vincent J. Velten, John C. Mossing, and Michael L. Bryant. Standard sar atr evaluation experiments using the mstar public release data set, 1998.

[70] Rose M Rustowicz, Jacob W Ross, Lawrence J Barnes, and Brian D Rigling. Atmospheric Effects and Impact on Target Classification for Synthetic Aperture Ladar (SAL) imagery. In *SPIE Defense+ Security*. International Society for Optics and Photonics, 2018.

[71] Theresa Scarnati and Anne Gelb. Variance based joint sparsity reconstruction of synthetic aperture radar data for speckle reduction. In *Algorithms for Synthetic Aperture Radar Imagery XXV*, volume 10647, page 106470R. International Society for Optics and Photonics, 2018.

[72] Theresa Scarnati and Benjamin Lewis. A deep learning approach to the synthetic and measured paired and labeled experiment (sample) challenge problem. In *Algorithms for Synthetic Aperture Radar Imagery XXVI*, volume 10987, page 109870G. International Society for Optics and Photonics, 2019.

[73] Bruce J. Schachter. *Automatic Target Recognition*. SPIE, 2016.

[74] Matthew Scherreik and Brian Rigling. Multi-class open set recognition for sar imagery. In *Automatic Target Recognition XXVI*, volume 9844, page 98440M. International Society for Optics and Photonics, 2016.

[75] Bryce E Schumm and Matthew P Dierking. Wave optics simulations of synthetic aperture ladar performance through turbulence. *JOSA A*, 34(10):1888–1895, 2017.

[76] Peter Shirley and R Keith Morley. *Realistic ray tracing*. AK Peters, Ltd., 2008.

[77] Jianfeng Sun, Yu Zhou, Ya'nan Zhi, Enwen Dai, and Liren Liu. Laser speckle effect overcome using multi-receiver method in the synthetic aperture laser imaging ladar, 2012.

[78] Yijun Sun, Zhipeng Liu, Sinisa Todorovic, and Jian Li. Adaptive boosting for sar automatic target recognition. *IEEE Transactions on Aerospace and Electronic Systems*, 43(1):112–125, 2007.

[79] Wu Tao, Chen Xi, Ruang Xiangwei, and Niu Lei. Study on sar target recognition based on support vector machine. In *2009 2nd Asian-Pacific Conference on Synthetic Aperture Radar*, pages 856–859. IEEE, 2009.

[80] Céline Tison, Nadine Pourthié, and Jean-Claude Souyris. Target recognition in sar images with support vector machines (svm). In *2007 IEEE International Geoscience and Remote Sensing Symposium*, pages 456–459. IEEE, 2007.

[81] George C Valley. Isoplanatic degradation of tilt correction and short-term imaging systems. *Applied Optics*, 19(4):574–577, 1980.

[82] Adam WM van Eekeren, Jasper R van Huis, Pieter T Eendebak, and Jan Baan. Vehicle tracking in wide area motion imagery from an airborne platform. In *Electro-Optical and Infrared Systems: Technology and Applications XII; and Quantum Information Science and Technology*, volume 9648, page 96480I. International Society for Optics and Photonics, 2015.

[83] Andreas C Völker, Pavel Zakharov, B Weber, F Buck, and Frank Scheffold. Laser speckle imaging with an active noise reduction scheme. *Optics Express*, 13(24):9782–9787, 2005.

[84] Hugo Weichel. *Laser beam propagation in the atmosphere*, volume 3. SPIE press, 1990.

[85] Zhenyu Yue, Fei Gao, Qingxu Xiong, Jun Wang, Teng Huang, Erfu Yang, and Huiyu Zhou. A novel semi-supervised convolutional neural network method for synthetic aperture radar image recognition. *Cognitive Computation*, pages 1–12, 2019.

[86] Qun Zhao and Jose C Principe. Support vector machines for sar automatic target recognition. *IEEE Transactions on Aerospace and Electronic Systems*, 37(2):643–654, 2001.

[87] Qun Zhao, Dongxin Xu, and J Principe. Pose estimation of sar automatic target recognition. In *Proceedings of Image Understanding Workshop*, volume 11. Citeseer, 1998.

Appendix A
ANOVA of Template Matching Factors

Source	Sum Sq.	d.f.	Mean Sq.	F	p-val
nQ	28.8378	3	9.6126	23,022.57	0
nDups	0.7346	2	0.3673	879.74	0
binSize	8.1959	2	4.0980	9814.73	0
poseError	8.4194	5	1.6840	4032.96	0
nQ*nDups	0.1203	6	0.0200	48.03	0
nQ*binSize	2.2861	6	0.3810	912.56	0
nQ*poseError	0.1357	15	0.0091	21.66	0
nDups*binSize	0.3309	4	0.0827	198.15	0
nDups*poseError	0.0005	10	0.0001	0.13	0.9995
binSize*poseError	0.6497	10	0.0650	155.61	0
Error	0.8751	2096			
Total	50.5861	2159			

Table A.1: ANOVA Table for algorithm parameters and OC's impacting SAR MPM performance. A p value of 0.9995 for the nDups*poseError factor indicates there is not a significant interaction between those two factors.

Source	Sum Sq.	d.f.	Mean Sq.	F	p-val
nQ	1.0502	3	0.3501	59,114.81	0
nDups	0.0006	2	0.0003	53.12	0
binSize	0.0272	2	0.0136	2294.18	0
poseError	10.2356	5	2.04712	345,695.4	0
nQ*nDups	0.0002	6	0.0001	6.82	0
nQ*binSize	0.0092	6	0.0015	259.35	0
nQ*poseError	0.1057	15	0.0071	1190.35	0
nDups*binSize	0.0002	4	0.0001	10.37	0
nDups*poseError	0	10	0	0.21	0.9995
binSize*poseError	0.3372	10	0.0337	5695.04	0
Error	0.0124	2096			
Total	11.7787	2159			

Table A.2: ANOVA Table for algorithm parameters and OC's impacting SAR QMSE performance. A p value of 0.9995 for the nDups*poseError factor indicates there is not a significant interaction between those two factors.

Source	Sum Sq.	d.f.	Mean Sq.	F	p-val
SA	194.781	5	38.956	81,853.38	0
nQ	168.165	3	56.055	117,780.56	0
nDups	3.605	2	1.803	3787.37	0
QPE	297.414	2	148.707	312,458.01	0
binSize	4.504	2	2.252	4731.81	0
poseError	36.009	5	7.202	15,132.29	0
SA*nQ	61.888	15	4.126	8669.16	0
SA*nDups	1.293	10	0.129	271.58	0
SA*QPE	8.259	10	0.826	1735.38	0
SA*binSize	0.887	10	0.089	186.31	0
SA*poseError	1.322	25	0.053	111.07	0
nQ*nDups	1.391	6	0.232	487.18	0
nQ*QPE	2.727	6	0.454	954.97	0
nQ*binSize	4.63	6	0.772	1621.42	0
nQ*poseError	0.997	15	0.066	139.62	0
nDups*QPE	0.059	4	0.015	30.83	0
nDups*binSize	0.472	4	0.118	248.13	0
nDups*poseError	0.018	10	0.002	3.68	0
QPE*binSize	0.192	4	0.048	100.6	0 .0001
QPE*poseError	4.223	10	0.422	887.34	0
binSize*poseError	0.201	10	0.02	42.24	0
Error	18.405	38,673	0		
Total	812.426	38,837			

Table A.3: ANOVA Table for algorithm parameters and OC's impacting SAL MPM performance.

Source	Sum Sq.	d.f.	Mean Sq.	F	p-val
SA	49.075	5	9.8149	56,207.58	0
nQ	6.658	3	2.2193	12,709.47	0
nDups	1.539	2	0.7695	4406.67	0
QPE	196.074	2	98.037	561,433.84	0
binSize	0.243	2	0.1216	696.23	0
poseError	70.819	5	14.1638	81,112.38	0
SA*nQ	8.296	15	0.5531	3167.39	0
SA*nDups	1.369	10	0.1396	799.33	0
SA*QPE	0.4	10	0.04	229.32	0
SA*binSize	0.915	10	0.0915	523.94	0
SA*poseError	0.651	25	0.026	149.13	0
nQ*nDups	0.002	6	0.0003	1.87	0.0814
nQ*QPE	12.473	6	2.0788	11,904.94	0
nQ*binSize	0.324	6	0.0539	308.88	0
nQ*poseError	2.213	15	0.1475	844.93	0
nDups*QPE	0.025	4	0.0063	36.12	0
nDups*binSize	0.236	4	0.0589	337.48	0
nDups*poseError	0.036	10	0.0036	20.45	0
QPE*binSize	0.567	4	0.1416	811.13	0
QPE*poseError	3.354	10	0.3354	30.23	0
binSize*poseError	0.053	10	0.0053	30.23	0
Error	6.76	38,715	0.0002		
Total	362.107	38,979			

Table A.4: ANOVA Table for algorithm parameters and OC's impacting SAL MPM performance.

Appendix B
CAD Physical Properties and SAR Image Similarity

The previous section describes how key elements from the SAL back-scattering equation can be used to estimate the similarity of formed images. Additionally, the key CAD elements are highly correlated with the formed SAL images. The same approach on SAR phenomenology is discussed and the feasibility of mapping the approach from diffuse returns to specular returns is evaluated. The SAR E_{bs} term can be represented as the following:

$$E_{bs} = \frac{jk_o e^{k_o r}}{2\pi r} |\vec{n} \cdot \vec{u}_i| \int \int_A exp\{jk_o \overrightarrow{a} \cdot (\hat{u}_i - \hat{u}_o)\} d\vec{a} \tag{B.1}$$

Equation B.1 is known as the physical optics (PO) approach for calculating the far-field return of a scatterer. This form of the SAR PO-integral is implemented in RaiderTracer [61, 62]. Similar to the SAL E_{bs} formulation, the SAR E_{bs} relies on sensor and waveform information (k_o, r) as well as the geometry of the scatterers $\hat{u}, \hat{n}, A$. In this formulation, A represents the total area of the facet and $\hat{a}$ is a specific point on the surface. Thus, computing the SAR E_{bs} term involves computing a surface integral over the interrogated surface.

The double integral expression is a key difference between the SAR E_{bs} model and the SAL E_{bs} model. For specular returns, additional information is needed to accurately capture how energy will spread and reflect off of the facet. In addition to a more complicated scattering model, a key aspect of SAR modeling is capturing multi-bounce features.

Based on Equation B.1 the pertinent physical properties are:

- $|\hat{n} \cdot \hat{u}_i|$ - the cosine of the angle between the surface normal and incident ray
- A - the area of the scatterer
- $\vec{a}$ - a specific point on the scatterer
- $\hat{u}_i - \hat{u}_o$ - the vector difference between the incident field and the observation direction.

The $\vec{a}$ and $\hat{u}_i - \hat{u}_o$ terms reside within the integral expression. The double integral expression is written for an arbitrary shape and size. Computing this integral can be computationally expensive, thus, for polygonal facets the double integral expression can be written as [41]:

$$\int\int_A exp\{jk_o\overrightarrow{a} \cdot (\hat{u}_i - \hat{u}_o)\}d\vec{a} \approx \boldsymbol{I_o}_{(k_o,\hat{a}_m,\hat{u}_i,\hat{u}_o)} \tag{B.2}$$

where

$$\boldsymbol{I_o}_{(k_o,\hat{a}_m,\hat{u}_i,\hat{u}_o)} = \sum_{m=1}^{M}(\hat{p} \cdot \hat{a}_m)e^{ik_o\hat{r}_m\cdot(\hat{u}_i-\hat{u}_o)}\frac{sin(k_o\hat{a}_m \cdot (\hat{u}_i - \hat{u}_o)/2)}{(k_o\hat{a}_m \cdot (\hat{u}_i - \hat{u}_o)/2)} \tag{B.3}$$

where M is the number of sides in the polygon, $\hat{p}$ is a unit vector normal to $(\hat{u}_i - \hat{u}_o)$, $\hat{a}_m$ is a vector with length and orientation information of edge m, and $\hat{r}_m$ is the midpoint of each edge. The integrated area A is the electromagnetic size of the surface and not necessarily the physical size of the surface. In order to adequately capture the specular contribution of a facet, additional information is needed that was not represented in the diffuse case. The contribution of a facet to a specular return cannot be estimated only from the physical properties of the facet. The relationship between the orientation, shape, and size of the facet must be taken into consideration.

In the SAR case, a triangular facet contribution is given as:

$$\boldsymbol{fc}_{SAR,\gamma} = \boldsymbol{I_o}_{(k_o,\hat{a}_m,\hat{u}_i,\hat{u}_o)} \tag{B.4}$$

The wave number information k_o is a complex value containing phase and amplitude information. In order to apply the approach in the previous section to for IG, the complex values in $\boldsymbol{I_o}$ are summed and the magnitude of the sum is assigned to the facet. The $\boldsymbol{FM}_{mp}$ matrix is formed as:

$$\boldsymbol{FM}_{mp} = \left[\frac{\boldsymbol{a}_i + \boldsymbol{b}_i + \boldsymbol{c}_i}{3}, |sum(\boldsymbol{fc}_{SAR,\gamma})| \right] \tag{B.5}$$

where c is the center index of $\boldsymbol{fc}_{SAR,\gamma}$.The grid structure ($\boldsymbol{IG}_{CAD}$) from the previous section is formed in the same manner. The image and CAD correlation results are shown in Figure B.2. The CAD ranking and image ranking Spearman correlation results are shown in Figure B.1. The far-field estimates for the SAR case were formed with RaiderTracer and with the same pristine imaging conditions discussed in Section 4.1. The $\boldsymbol{IG}$ formation in the SAR case required evaluating the PO-integral due to the specular nature of SAR scattering. Only CAD physical characteristics were needed in the SAL case. The grid formation approach on SAR data yielded lower correlation values than in the SAL case.

Potential issues with the current $\boldsymbol{IG}$ formulation approach for specular returns include difficulties with multi-bounce features and specular artifacts seen in SAR imagery not captured in the $\boldsymbol{IG}$ structure. Additionally, in order to use terms directly from the E_{bs} equation, more sensor and orientation information is needed for the specular SAR case than in the diffuse SAL case.

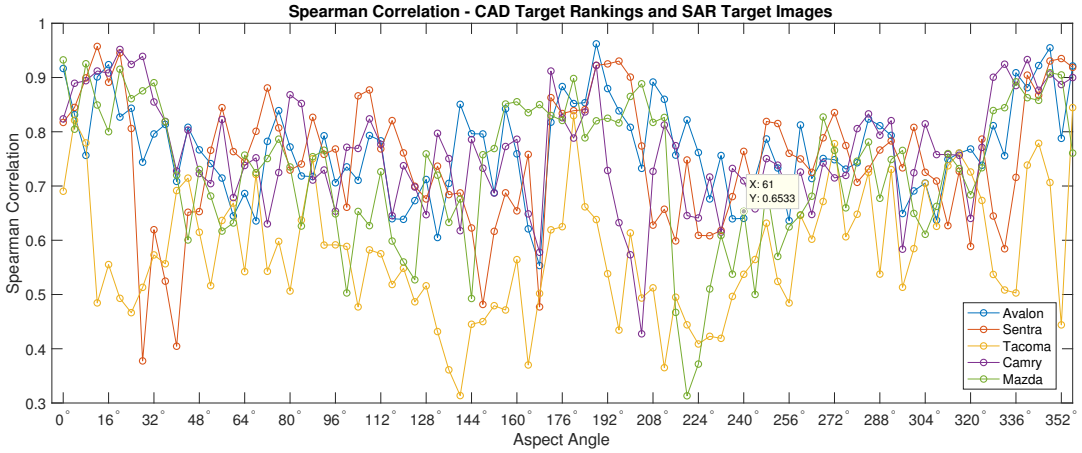


Figure B.1: Spearman correlation between the SAR image rankings and $\boldsymbol{IG}$ rankings of targets. The average r_s score for each target are 0.7920, 0.7774, 0.5810, 0.7927, 0.7860 for the Avalon, Sentra, Tacoma, Camry, and Mazda models respectively. The Tacoma model contains pronounced multi-bounce features.

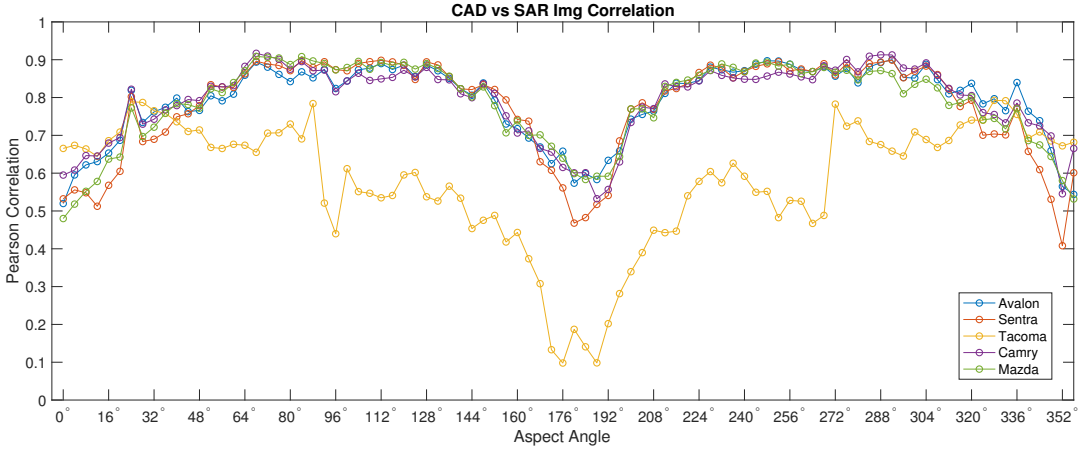
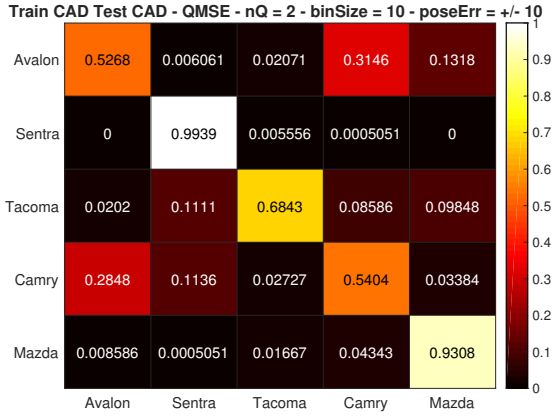


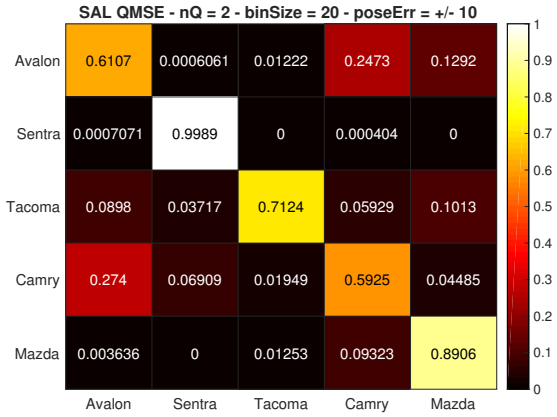
Figure B.2: Pearson correlation between SAR images and $\boldsymbol{IG}$ representations. The average r_p scores are 0.7685, 0.7471, 0.5781, 0.7638, 0.7295 for the Avalon, Sentra, Tacoma, Camry, and Mazda models respectively. The sedan and Mazda models contain mostly single-bounce features while the Tacoma contains pronounced multi-bounce features. The weak correlation between $\boldsymbol{IG}$ and SAR images in the Tacoma case indicate the methodology does not adequately capture specular and multi-bounce responses.

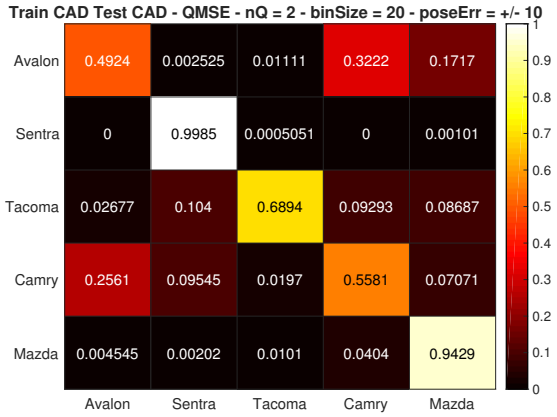
Appendix C
Confusion Matrices

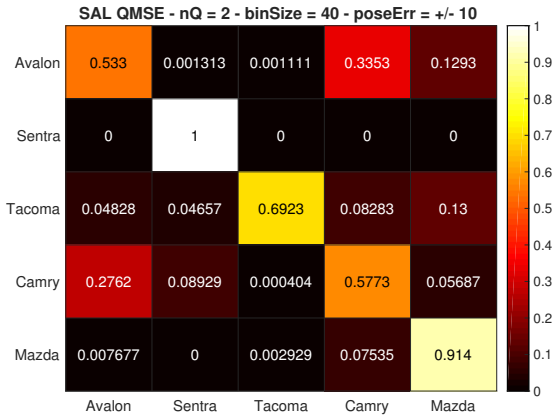
This section contains the confusion matrices formed in the experiments described in Section 4.3. The left column of matrices are formed entirely in the image domain, and is the average of the confusion matrices found from 10 trials. In Section C.1 the right column of matrices are the confusion matrices found by testing and training on CAD representations. In Section C.2 the right column of matrices are the confusion matrices found by training on CAD representations and testing on SAL images. The matrices are separated by algorithm parameters (quantization level and bin size), and the algorithm itself. The matrices allow for a visual inspection of confusion trends between traditional training and testing on imagery, training and testing entirely in the CAD domain, and training on CAD and testing on SAL images. For space considerations, the matrices listed are only for the worse case scenario of pose error.

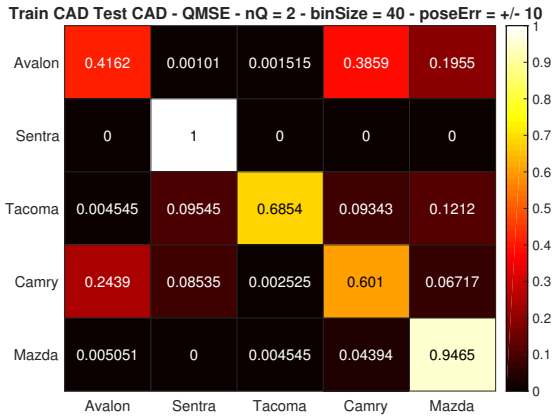
C.1 CAD vs SAL Image Confusion Matrices



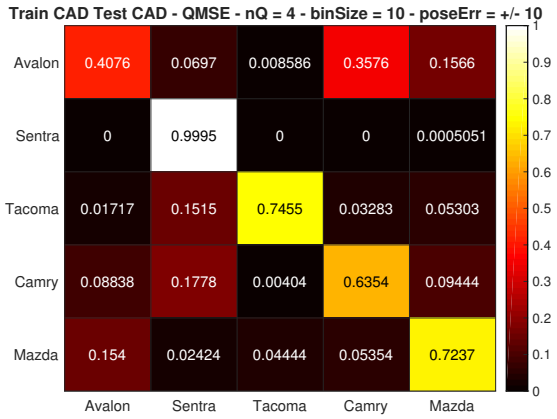


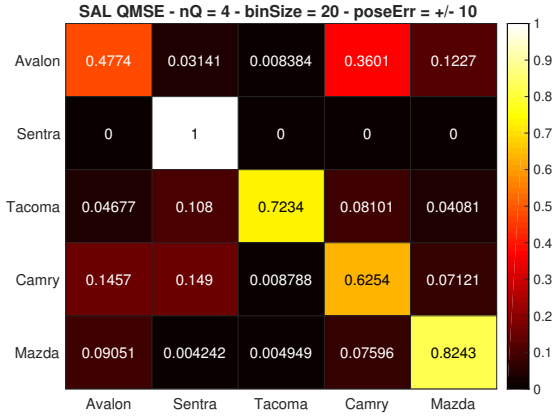


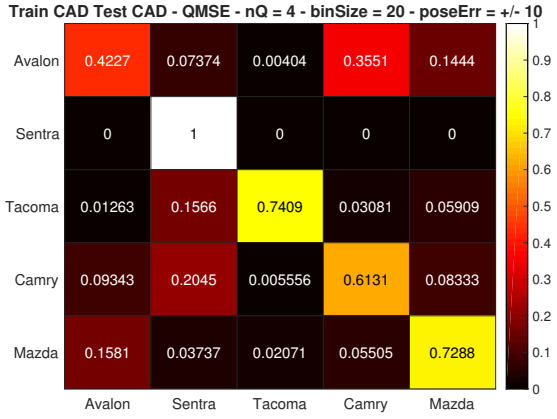


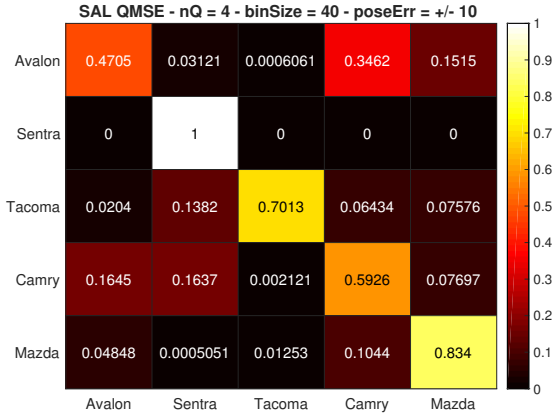


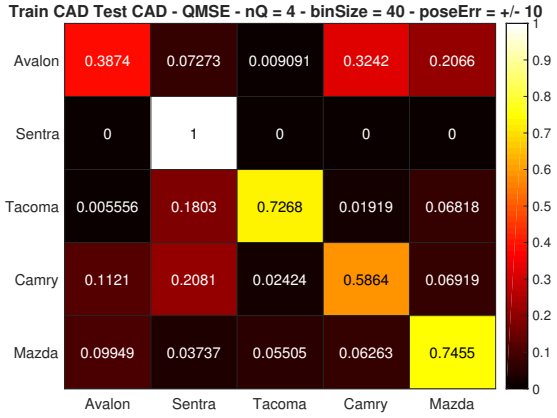

SAL QMSE - nQ = 4 - binSize = 10 - poseErr = +/- 10
Avalon 0.4881 0.03333 0.007677 0.3544 0.1165
Sentra 0.000101 0.9999 0 0 0
Tacoma 0.04768 0.1161 0.7152 0.07657 0.04455
Camry 0.1546 0.1456 0.006162 0.6308 0.06283
Mazda 0.08707 0.004747 0.01222 0.06545 0.8305
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

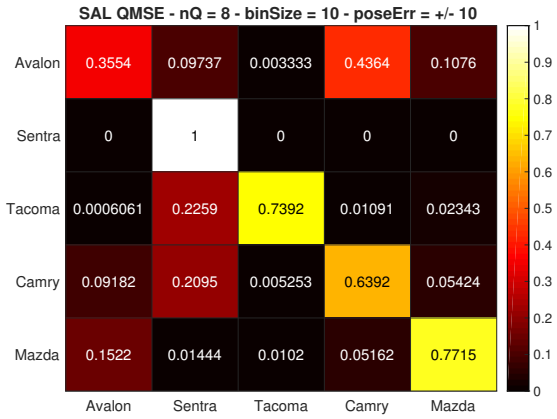

Train CAD Test CAD - QMSE - nQ = 4 - binSize = 10 - poseErr = +/- 10
Avalon 0.4076 0.0697 0.008586 0.3576 0.1566
Sentra 0 0.9995 0 0 0.0005051
Tacoma 0.01717 0.1515 0.7455 0.03283 0.05303
Camry 0.08838 0.1778 0.00404 0.6354 0.09444
Mazda 0.154 0.02424 0.04444 0.05354 0.7237
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

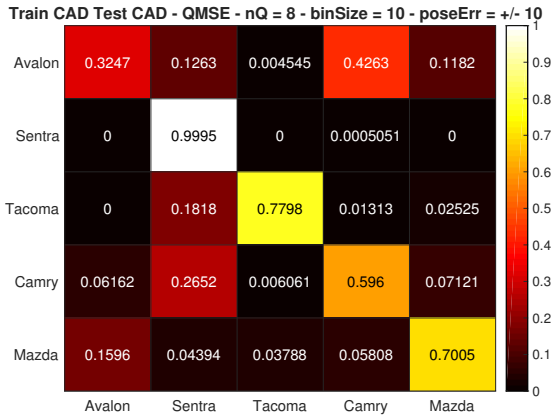

SAL QMSE - nQ = 4 - binSize = 20 - poseErr = +/- 10
Avalon 0.4774 0.03141 0.008384 0.3601 0.1227
Sentra 0 1 0 0 0
Tacoma 0.04677 0.108 0.7234 0.08101 0.04081
Camry 0.1457 0.149 0.008788 0.6254 0.07121
Mazda 0.09051 0.004242 0.004949 0.07596 0.8243
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

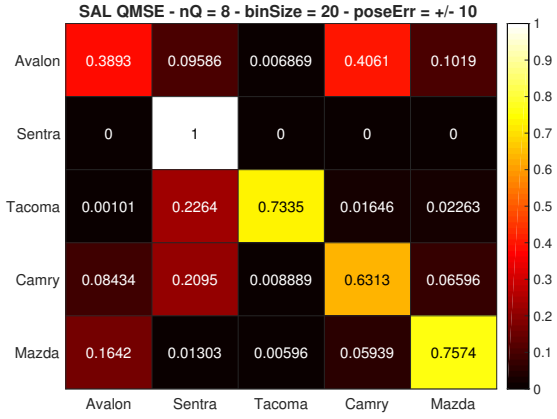

Train CAD Test CAD - QMSE - nQ = 4 - binSize = 20 - poseErr = +/- 10
Avalon 0.4227 0.07374 0.00404 0.3551 0.1444
Sentra 0 1 0 0 0
Tacoma 0.01263 0.1566 0.7409 0.03081 0.05909
Camry 0.09343 0.2045 0.005556 0.6131 0.08333
Mazda 0.1581 0.03737 0.02071 0.05505 0.7288
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

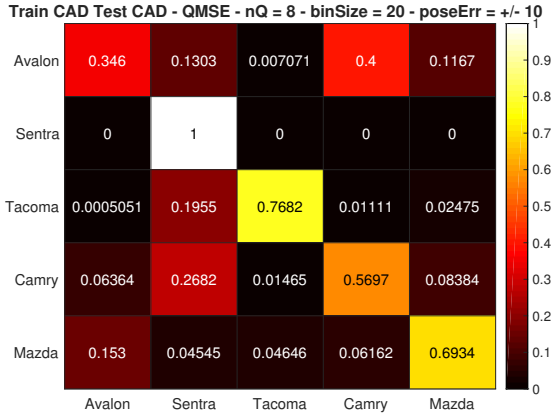

SAL QMSE - nQ = 4 - binSize = 40 - poseErr = +/- 10
Avalon 0.4705 0.03121 0.0006061 0.3462 0.1515
Sentra 0 1 0 0 0
Tacoma 0.0204 0.1382 0.7013 0.06434 0.07576
Camry 0.1645 0.1637 0.002121 0.5926 0.07697
Mazda 0.04848 0.0005051 0.01253 0.1044 0.834
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

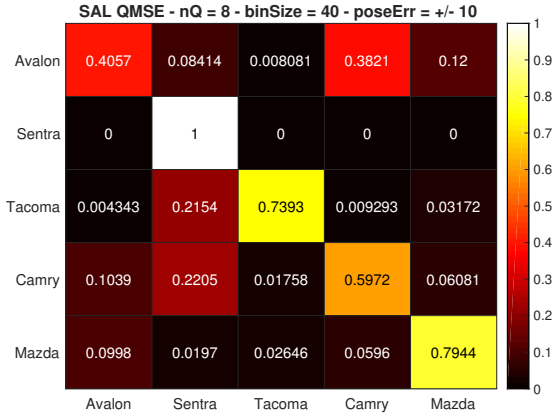

Train CAD Test CAD - QMSE - nQ = 4 - binSize = 40 - poseErr = +/- 10
Avalon 0.3874 0.07273 0.009091 0.3242 0.2066
Sentra 0 1 0 0 0
Tacoma 0.005556 0.1803 0.7268 0.01919 0.06818
Camry 0.1121 0.2081 0.02424 0.5864 0.06919
Mazda 0.09949 0.03737 0.05505 0.06263 0.7455
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

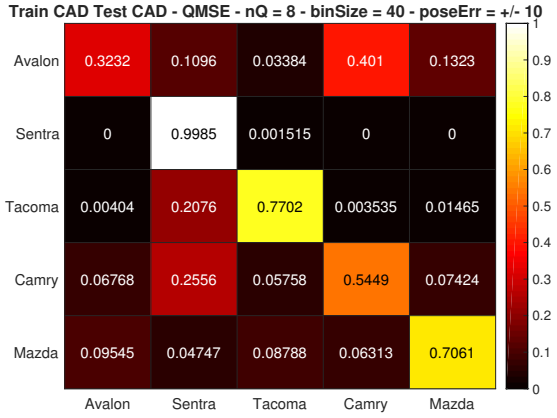

SAL QMSE - nQ = 8 - binSize = 10 - poseErr = +/- 10
Avalon 0.3554 0.09737 0.003333 0.4364 0.1076
Sentra 0 1 0 0 0
Tacoma 0.0006061 0.2259 0.7392 0.01091 0.02343
Camry 0.09182 0.2095 0.005253 0.6392 0.05424
Mazda 0.1522 0.01444 0.0102 0.05162 0.7715
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0


Train CAD Test CAD - QMSE - nQ = 8 - binSize = 10 - poseErr = +/- 10
Avalon 0.3247 0.1263 0.004545 0.4263 0.1182
Sentra 0 0.9995 0 0.0005051 0
Tacoma 0 0.1818 0.7798 0.01313 0.02525
Camry 0.06162 0.2652 0.006061 0.596 0.07121
Mazda 0.1596 0.04394 0.03788 0.05808 0.7005
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

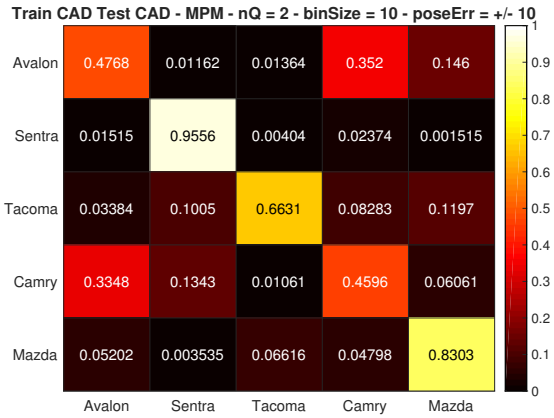

SAL QMSE - nQ = 8 - binSize = 20 - poseErr = +/- 10
Avalon 0.3893 0.09586 0.006869 0.4061 0.1019
Sentra 0 1 0 0 0
Tacoma 0.00101 0.2264 0.7335 0.01646 0.02263
Camry 0.08434 0.2095 0.008889 0.6313 0.06596
Mazda 0.1642 0.01303 0.00596 0.05939 0.7574
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0


Train CAD Test CAD - QMSE - nQ = 8 - binSize = 20 - poseErr = +/- 10
Avalon 0.346 0.1303 0.007071 0.4 0.1167
Sentra 0 1 0 0 0
Tacoma 0.0005051 0.1955 0.7682 0.01111 0.02475
Camry 0.06364 0.2682 0.01465 0.5697 0.08384
Mazda 0.153 0.04545 0.04646 0.06162 0.6934
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

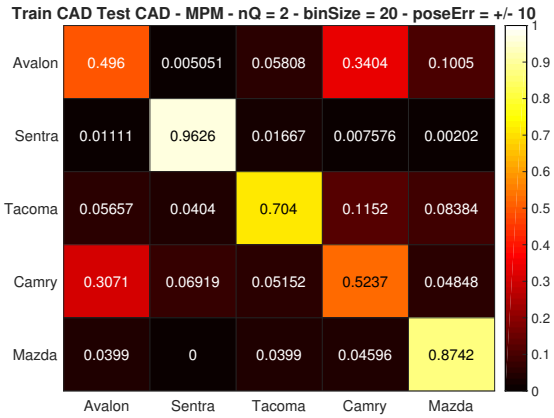

SAL QMSE - nQ = 8 - binSize = 40 - poseErr = +/- 10
Avalon 0.4057 0.08414 0.008081 0.3821 0.12
Sentra 0 1 0 0 0
Tacoma 0.004343 0.2154 0.7393 0.009293 0.03172
Camry 0.1039 0.2205 0.01758 0.5972 0.06081
Mazda 0.0998 0.0197 0.02646 0.0596 0.7944
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0


Train CAD Test CAD - QMSE - nQ = 8 - binSize = 40 - poseErr = +/- 10
Avalon 0.3232 0.1096 0.03384 0.401 0.1323
Sentra 0 0.9985 0.001515 0 0
Tacoma 0.00404 0.2076 0.7702 0.003535 0.01465
Camry 0.06768 0.2556 0.05758 0.5449 0.07424
Mazda 0.09545 0.04747 0.08788 0.06313 0.7061
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

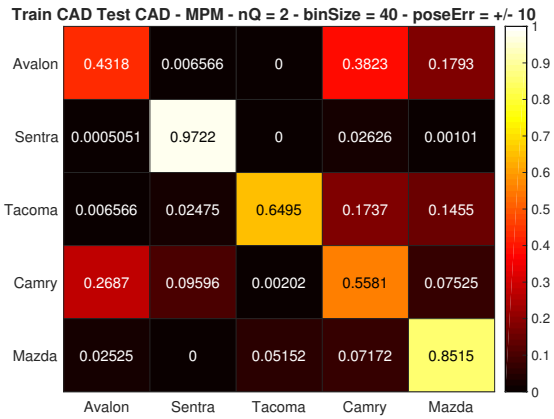

SAL MPM - nQ = 2 - binSize = 10 - poseErr = +/- 10
Avalon 0.5607 0.008182 0.02586 0.327 0.07828
Sentra 0.03646 0.9293 0.003232 0.03071 0.000303
Tacoma 0.08707 0.04414 0.6732 0.08172 0.1138
Camry 0.3438 0.09727 0.02596 0.5005 0.03242
Mazda 0.03586 0 0.05394 0.1239 0.7863
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

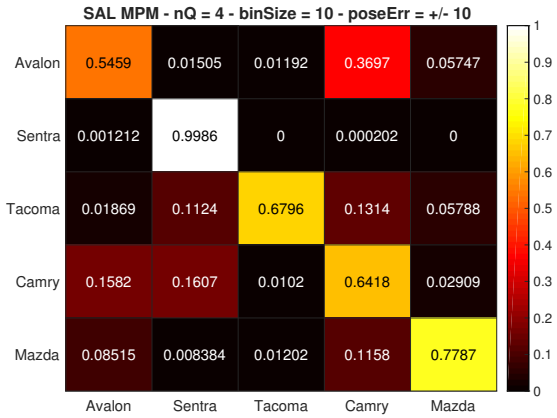

Train CAD Test CAD - MPM - nQ = 2 - binSize = 10 - poseErr = +/- 10
Avalon 0.4768 0.01162 0.01364 0.352 0.146
Sentra 0.01515 0.9556 0.00404 0.02374 0.001515
Tacoma 0.03384 0.1005 0.6631 0.08283 0.1197
Camry 0.3348 0.1343 0.01061 0.4596 0.06061
Mazda 0.05202 0.003535 0.06616 0.04798 0.8303
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

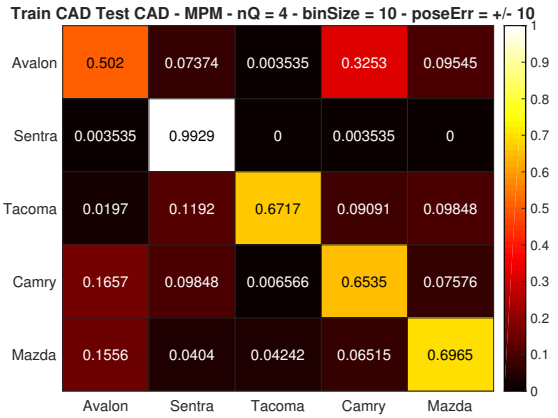

SAL MPM - nQ = 2 - binSize = 20 - poseErr = +/- 10
Avalon 0.5404 0.0007071 0.0297 0.3206 0.1086
Sentra 0.001919 0.9899 0.00404 0.003838 0.000303
Tacoma 0.08667 0.02323 0.7048 0.08798 0.09727
Camry 0.2546 0.05879 0.03222 0.6 0.05434
Mazda 0.02505 0 0.03131 0.1135 0.8301
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

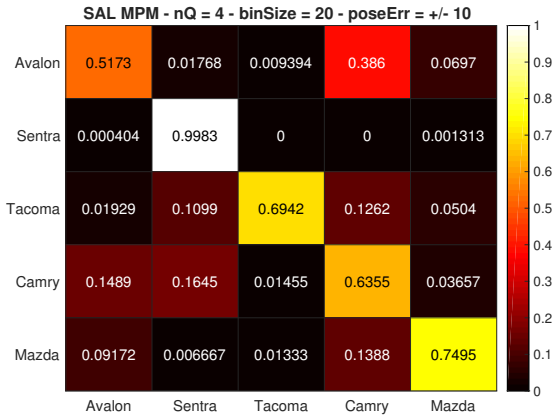

Train CAD Test CAD - MPM - nQ = 2 - binSize = 20 - poseErr = +/- 10
Avalon 0.496 0.005051 0.05808 0.3404 0.1005
Sentra 0.01111 0.9626 0.01667 0.007576 0.00202
Tacoma 0.05657 0.0404 0.704 0.1152 0.08384
Camry 0.3071 0.06919 0.05152 0.5237 0.04848
Mazda 0.0399 0 0.0399 0.04596 0.8742
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

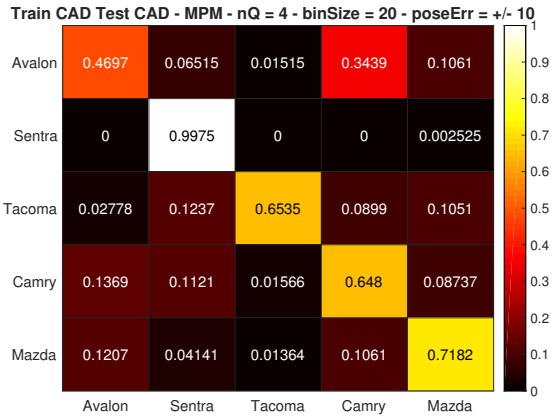

SAL MPM - nQ = 2 - binSize = 40 - poseErr = +/- 10
Avalon 0.5551 0.001414 0.0008081 0.3041 0.1386
Sentra 0.006667 0.9743 0 0.01889 0.000101
Tacoma 0.04505 0.0298 0.6506 0.09909 0.1755
Camry 0.3977 0.08475 0.00202 0.4525 0.06303
Mazda 0.04717 0 0.05222 0.09778 0.8028
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0


Train CAD Test CAD - MPM - nQ = 2 - binSize = 40 - poseErr = +/- 10
Avalon 0.4318 0.006566 0 0.3823 0.1793
Sentra 0.0005051 0.9722 0 0.02626 0.00101
Tacoma 0.006566 0.02475 0.6495 0.1737 0.1455
Camry 0.2687 0.09596 0.00202 0.5581 0.07525
Mazda 0.02525 0 0.05152 0.07172 0.8515
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

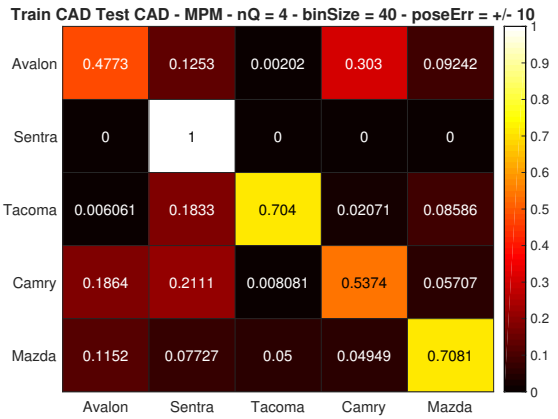

SAL MPM - nQ = 4 - binSize = 10 - poseErr = +/- 10
Avalon 0.5459 0.01505 0.01192 0.3697 0.05747
Sentra 0.001212 0.9986 0 0.000202 0
Tacoma 0.01869 0.1124 0.6796 0.1314 0.05788
Camry 0.1582 0.1607 0.0102 0.6418 0.02909
Mazda 0.08515 0.008384 0.01202 0.1158 0.7787
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

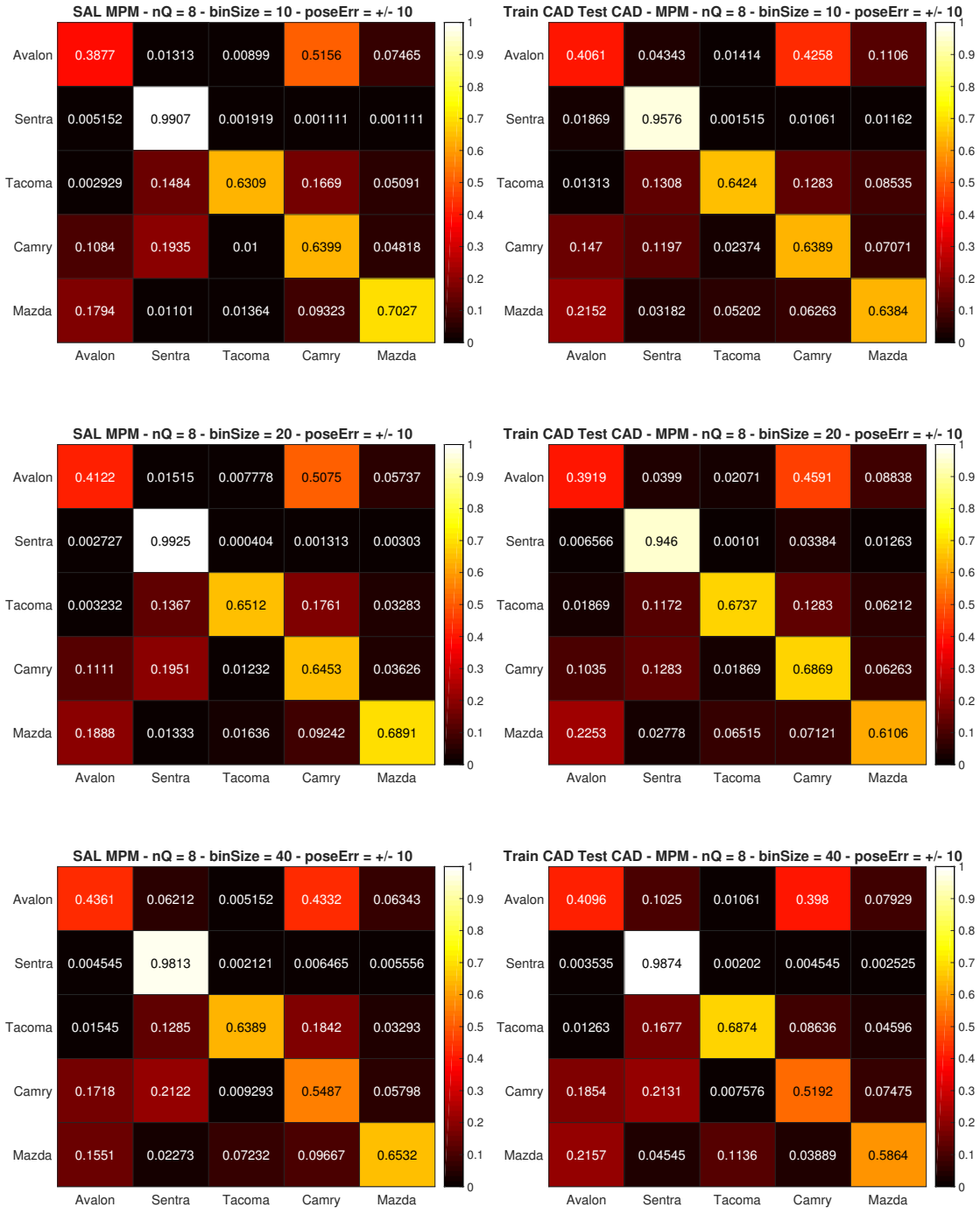

Train CAD Test CAD - MPM - nQ = 4 - binSize = 10 - poseErr = +/- 10
Avalon 0.502 0.07374 0.003535 0.3253 0.09545
Sentra 0.003535 0.9929 0 0.003535 0
Tacoma 0.0197 0.1192 0.6717 0.09091 0.09848
Camry 0.1657 0.09848 0.006566 0.6535 0.07576
Mazda 0.1556 0.0404 0.04242 0.06515 0.6965
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0


SAL MPM - nQ = 4 - binSize = 20 - poseErr = +/- 10
Avalon 0.5173 0.01768 0.009394 0.386 0.0697
Sentra 0.000404 0.9983 0 0 0.001313
Tacoma 0.01929 0.1099 0.6942 0.1262 0.0504
Camry 0.1489 0.1645 0.01455 0.6355 0.03657
Mazda 0.09172 0.006667 0.01333 0.1388 0.7495
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0


Train CAD Test CAD - MPM - nQ = 4 - binSize = 20 - poseErr = +/- 10
Avalon 0.4697 0.06515 0.01515 0.3439 0.1061
Sentra 0 0.9975 0 0 0.002525
Tacoma 0.02778 0.1237 0.6535 0.0899 0.1051
Camry 0.1369 0.1121 0.01566 0.648 0.08737
Mazda 0.1207 0.04141 0.01364 0.1061 0.7182
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

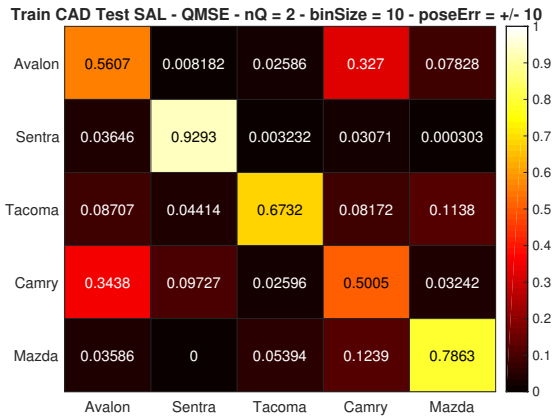

SAL MPM - nQ = 4 - binSize = 40 - poseErr = +/- 10
Avalon 0.5345 0.04303 0.003939 0.3672 0.05131
Sentra 0 0.9998 0 0.000202 0
Tacoma 0.00798 0.1034 0.6899 0.134 0.06465
Camry 0.1988 0.1944 0.01232 0.5556 0.03889
Mazda 0.05313 0.005253 0.05808 0.1302 0.7533
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

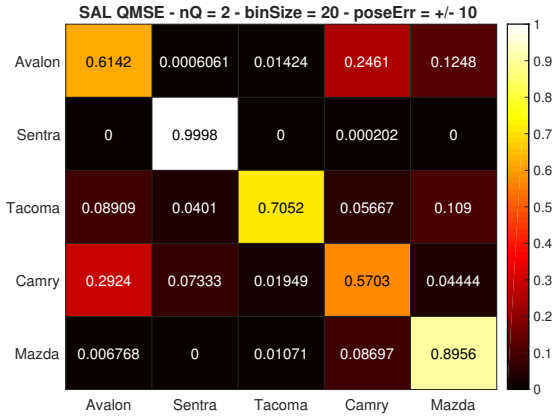

Train CAD Test CAD - MPM - nQ = 4 - binSize = 40 - poseErr = +/- 10
Avalon 0.4773 0.1253 0.00202 0.303 0.09242
Sentra 0 1 0 0 0
Tacoma 0.006061 0.1833 0.704 0.02071 0.08586
Camry 0.1864 0.2111 0.008081 0.5374 0.05707
Mazda 0.1152 0.07727 0.05 0.04949 0.7081
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

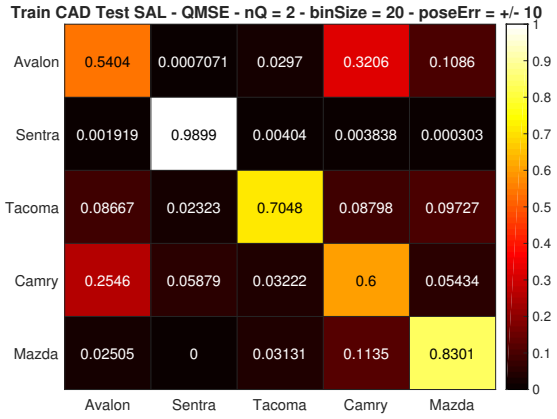


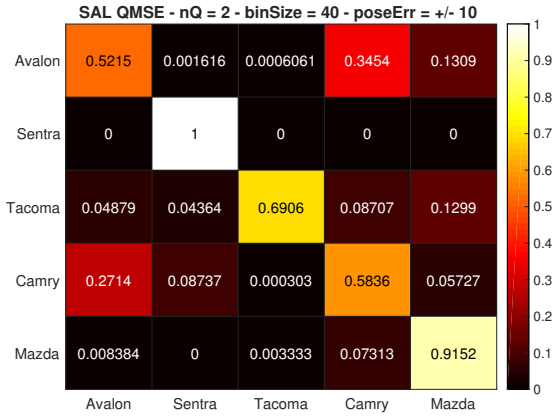
C.2 Train CAD Test SAL vs Train SAL Test SAL

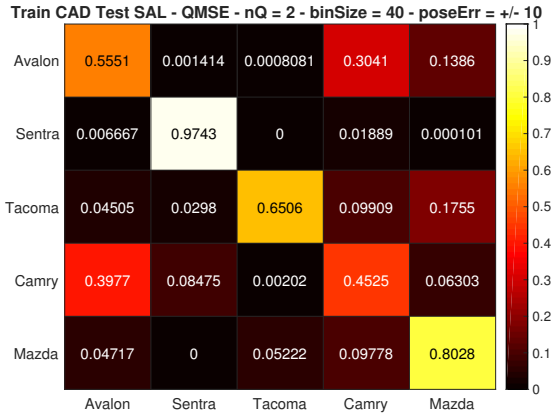

SAL QMSE - nQ = 2 - binSize = 10 - poseErr = +/- 10
Avalon 0.6497 0.004545 0.02535 0.2345 0.08586
Sentra 0.0006061 0.9958 0.003131 0.0005051 0
Tacoma 0.1043 0.04525 0.7078 0.04424 0.09838
Camry 0.2986 0.0803 0.02444 0.5739 0.02273
Mazda 0.009596 0 0.01394 0.08737 0.8891
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0


Train CAD Test SAL - QMSE - nQ = 2 - binSize = 10 - poseErr = +/- 10
Avalon 0.5607 0.008182 0.02586 0.327 0.07828
Sentra 0.03646 0.9293 0.003232 0.03071 0.000303
Tacoma 0.08707 0.04414 0.6732 0.08172 0.1138
Camry 0.3438 0.09727 0.02596 0.5005 0.03242
Mazda 0.03586 0 0.05394 0.1239 0.7863
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

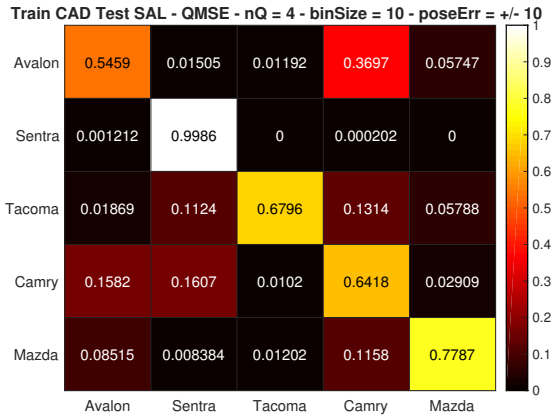

SAL QMSE - nQ = 2 - binSize = 20 - poseErr = +/- 10
Avalon 0.6142 0.0006061 0.01424 0.2461 0.1248
Sentra 0 0.9998 0 0.000202 0
Tacoma 0.08909 0.0401 0.7052 0.05667 0.109
Camry 0.2924 0.07333 0.01949 0.5703 0.04444
Mazda 0.006768 0 0.01071 0.08697 0.8956
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

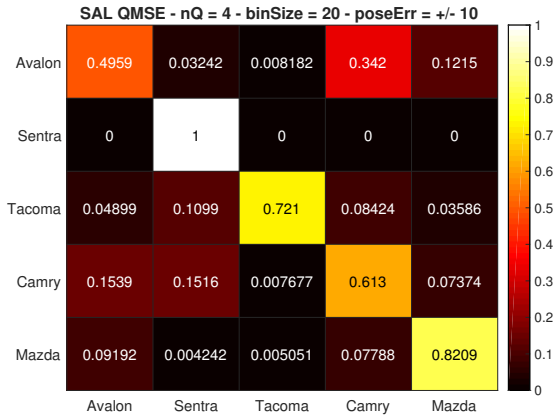

Train CAD Test SAL - QMSE - nQ = 2 - binSize = 20 - poseErr = +/- 10
Avalon 0.5404 0.0007071 0.0297 0.3206 0.1086
Sentra 0.001919 0.9899 0.00404 0.003838 0.000303
Tacoma 0.08667 0.02323 0.7048 0.08798 0.09727
Camry 0.2546 0.05879 0.03222 0.6 0.05434
Mazda 0.02505 0 0.03131 0.1135 0.8301
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

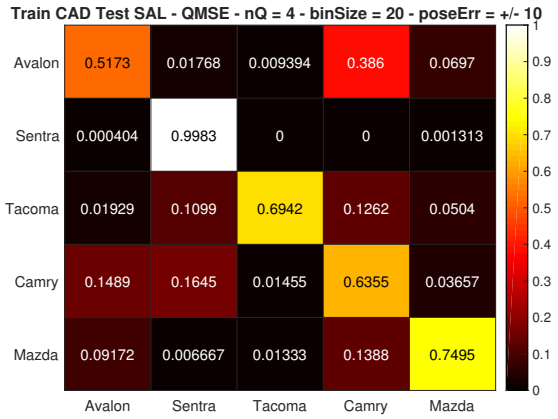

SAL QMSE - nQ = 2 - binSize = 40 - poseErr = +/- 10
Avalon 0.5215 0.001616 0.0006061 0.3454 0.1309
Sentra 0 1 0 0 0
Tacoma 0.04879 0.04364 0.6906 0.08707 0.1299
Camry 0.2714 0.08737 0.000303 0.5836 0.05727
Mazda 0.008384 0 0.003333 0.07313 0.9152
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0


Train CAD Test SAL - QMSE - nQ = 2 - binSize = 40 - poseErr = +/- 10
Avalon 0.5551 0.001414 0.0008081 0.3041 0.1386
Sentra 0.006667 0.9743 0 0.01889 0.000101
Tacoma 0.04505 0.0298 0.6506 0.09909 0.1755
Camry 0.3977 0.08475 0.00202 0.4525 0.06303
Mazda 0.04717 0 0.05222 0.09778 0.8028
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

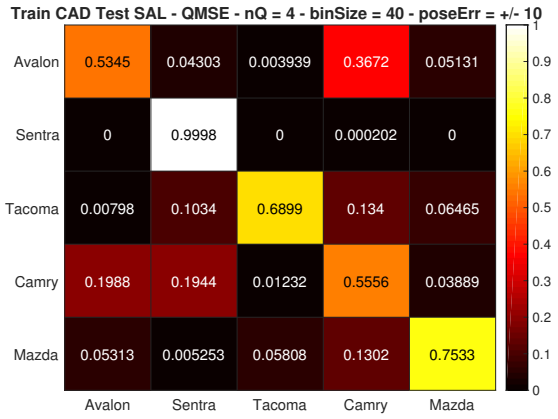

SAL QMSE - nQ = 4 - binSize = 10 - poseErr = +/- 10
Avalon 0.4915 0.03404 0.006768 0.3506 0.1171
Sentra 0 1 0 0 0
Tacoma 0.03879 0.1182 0.7149 0.08202 0.04606
Camry 0.1632 0.152 0.006162 0.6143 0.06424
Mazda 0.08333 0.003434 0.006869 0.06889 0.8375
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0


Train CAD Test SAL - QMSE - nQ = 4 - binSize = 10 - poseErr = +/- 10
Avalon 0.5459 0.01505 0.01192 0.3697 0.05747
Sentra 0.001212 0.9986 0 0.000202 0
Tacoma 0.01869 0.1124 0.6796 0.1314 0.05788
Camry 0.1582 0.1607 0.0102 0.6418 0.02909
Mazda 0.08515 0.008384 0.01202 0.1158 0.7787
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

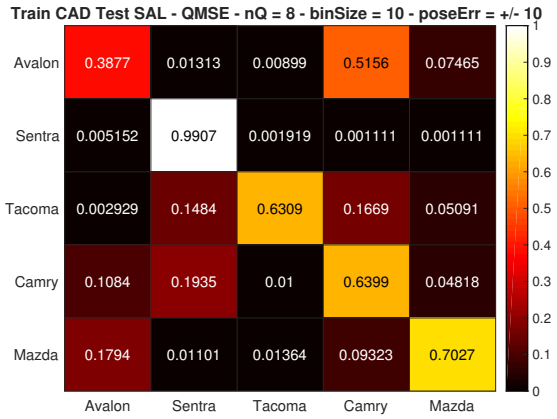

SAL QMSE - nQ = 4 - binSize = 20 - poseErr = +/- 10
Avalon 0.4959 0.03242 0.008182 0.342 0.1215
Sentra 0 1 0 0 0
Tacoma 0.04899 0.1099 0.721 0.08424 0.03586
Camry 0.1539 0.1516 0.007677 0.613 0.07374
Mazda 0.09192 0.004242 0.005051 0.07788 0.8209
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

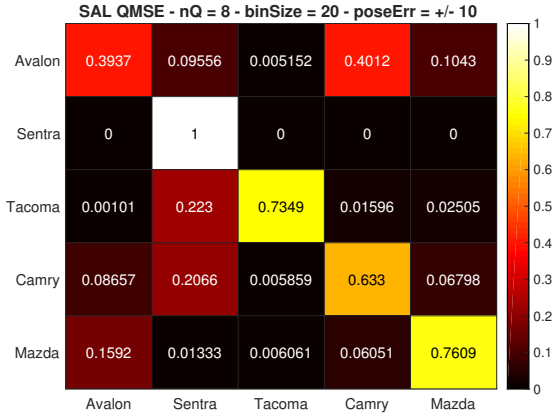

Train CAD Test SAL - QMSE - nQ = 4 - binSize = 20 - poseErr = +/- 10
Avalon 0.5173 0.01768 0.009394 0.386 0.0697
Sentra 0.000404 0.9983 0 0 0.001313
Tacoma 0.01929 0.1099 0.6942 0.1262 0.0504
Camry 0.1489 0.1645 0.01455 0.6355 0.03657
Mazda 0.09172 0.006667 0.01333 0.1388 0.7495
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

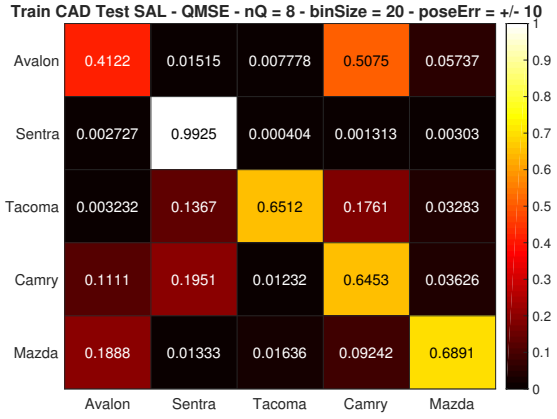

SAL QMSE - nQ = 4 - binSize = 40 - poseErr = +/- 10
Avalon 0.4731 0.03556 0.000303 0.3489 0.1421
Sentra 0 1 0 0 0
Tacoma 0.0201 0.1412 0.6977 0.06212 0.07889
Camry 0.1812 0.1641 0.0008081 0.5781 0.07576
Mazda 0.04919 0.0008081 0.01263 0.1035 0.8338
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

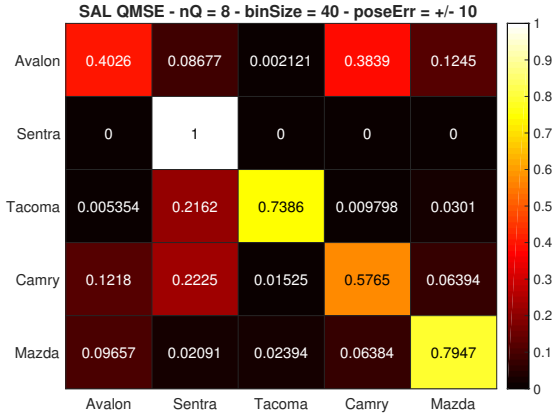

Train CAD Test SAL - QMSE - nQ = 4 - binSize = 40 - poseErr = +/- 10
Avalon 0.5345 0.04303 0.003939 0.3672 0.05131
Sentra 0 0.9998 0 0.000202 0
Tacoma 0.00798 0.1034 0.6899 0.134 0.06465
Camry 0.1988 0.1944 0.01232 0.5556 0.03889
Mazda 0.05313 0.005253 0.05808 0.1302 0.7533
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

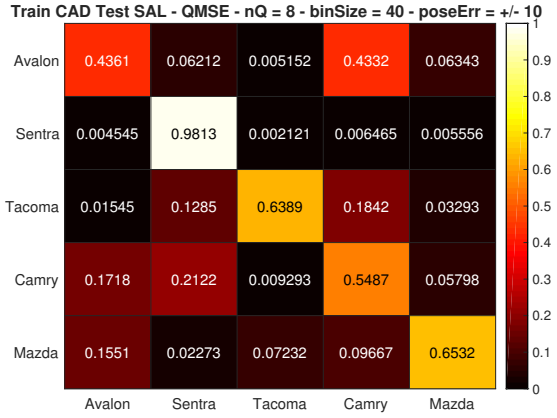

SAL QMSE - nQ = 8 - binSize = 10 - poseErr = +/- 10
Avalon 0.3551 0.1017 0.001616 0.4297 0.1119
Sentra 0 1 0 0 0
Tacoma 0.0005051 0.2264 0.7321 0.01313 0.02788
Camry 0.09424 0.2142 0.001818 0.6341 0.05556
Mazda 0.1558 0.01485 0.006364 0.05283 0.7702
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0


Train CAD Test SAL - QMSE - nQ = 8 - binSize = 10 - poseErr = +/- 10
Avalon 0.3877 0.01313 0.00899 0.5156 0.07465
Sentra 0.005152 0.9907 0.001919 0.001111 0.001111
Tacoma 0.002929 0.1484 0.6309 0.1669 0.05091
Camry 0.1084 0.1935 0.01 0.6399 0.04818
Mazda 0.1794 0.01101 0.01364 0.09323 0.7027
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

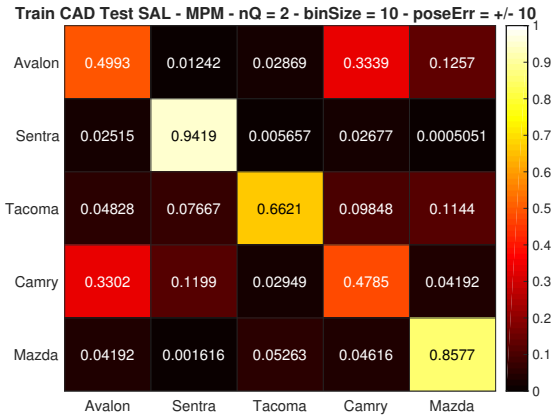

SAL QMSE - nQ = 8 - binSize = 20 - poseErr = +/- 10
Avalon 0.3937 0.09556 0.005152 0.4012 0.1043
Sentra 0 1 0 0 0
Tacoma 0.00101 0.223 0.7349 0.01596 0.02505
Camry 0.08657 0.2066 0.005859 0.633 0.06798
Mazda 0.1592 0.01333 0.006061 0.06051 0.7609
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0


Train CAD Test SAL - QMSE - nQ = 8 - binSize = 20 - poseErr = +/- 10
Avalon 0.4122 0.01515 0.007778 0.5075 0.05737
Sentra 0.002727 0.9925 0.000404 0.001313 0.00303
Tacoma 0.003232 0.1367 0.6512 0.1761 0.03283
Camry 0.1111 0.1951 0.01232 0.6453 0.03626
Mazda 0.1888 0.01333 0.01636 0.09242 0.6891
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

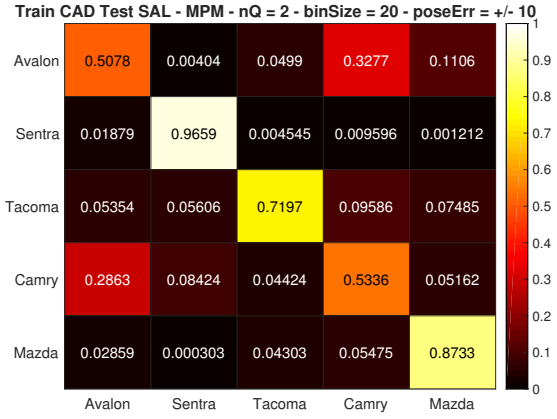

SAL QMSE - nQ = 8 - binSize = 40 - poseErr = +/- 10
Avalon 0.4026 0.08677 0.002121 0.3839 0.1245
Sentra 0 1 0 0 0
Tacoma 0.005354 0.2162 0.7386 0.009798 0.0301
Camry 0.1218 0.2225 0.01525 0.5765 0.06394
Mazda 0.09657 0.02091 0.02394 0.06384 0.7947
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

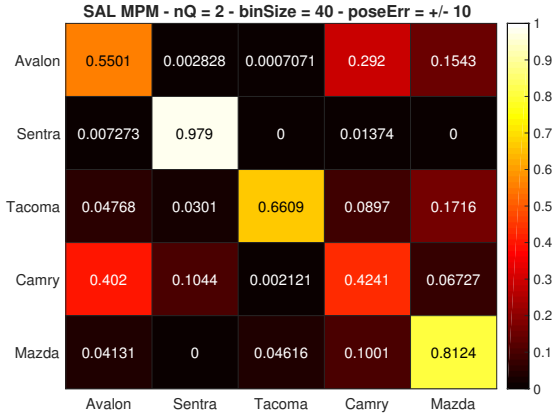

Train CAD Test SAL - QMSE - nQ = 8 - binSize = 40 - poseErr = +/- 10
Avalon 0.4361 0.06212 0.005152 0.4332 0.06343
Sentra 0.004545 0.9813 0.002121 0.006465 0.005556
Tacoma 0.01545 0.1285 0.6389 0.1842 0.03293
Camry 0.1718 0.2122 0.009293 0.5487 0.05798
Mazda 0.1551 0.02273 0.07232 0.09667 0.6532
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

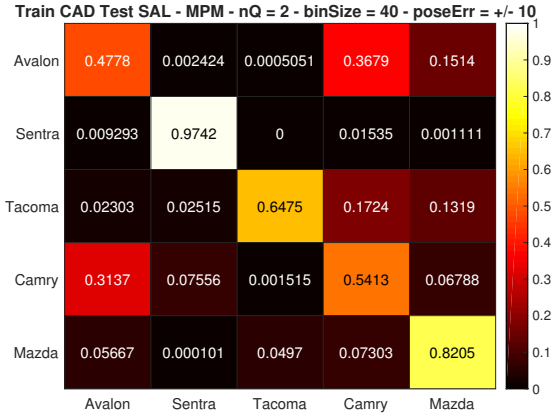

SAL MPM - nQ = 2 - binSize = 10 - poseErr = +/- 10
Avalon 0.553 0.01081 0.03222 0.3105 0.09343
Sentra 0.02879 0.9407 0.005455 0.02475 0.000303
Tacoma 0.09687 0.04303 0.6793 0.05899 0.1218
Camry 0.3339 0.1077 0.03081 0.4876 0.04
Mazda 0.03434 0.000101 0.0601 0.1134 0.792
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

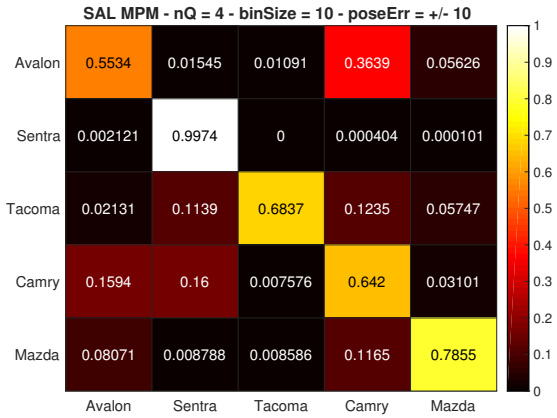

Train CAD Test SAL - MPM - nQ = 2 - binSize = 10 - poseErr = +/- 10
Avalon 0.4993 0.01242 0.02869 0.3339 0.1257
Sentra 0.02515 0.9419 0.005657 0.02677 0.0005051
Tacoma 0.04828 0.07667 0.6621 0.09848 0.1144
Camry 0.3302 0.1199 0.02949 0.4785 0.04192
Mazda 0.04192 0.001616 0.05263 0.04616 0.8577
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

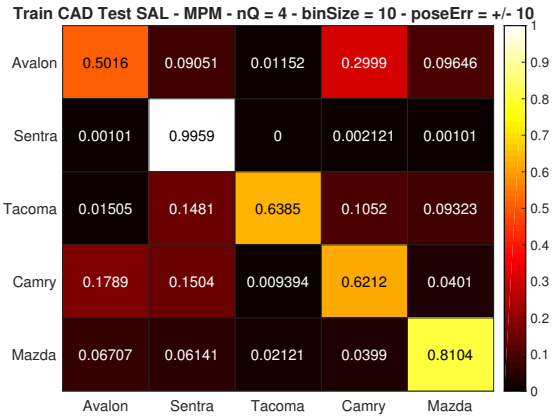

SAL MPM - nQ = 2 - binSize = 20 - poseErr = +/- 10
Avalon 0.5284 0.001414 0.03727 0.3349 0.09798
Sentra 0.005253 0.9823 0.004747 0.007677 0
Tacoma 0.08535 0.02172 0.7106 0.09354 0.08879
Camry 0.2681 0.06081 0.03949 0.583 0.04859
Mazda 0.01949 0.000101 0.03394 0.1216 0.8248
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

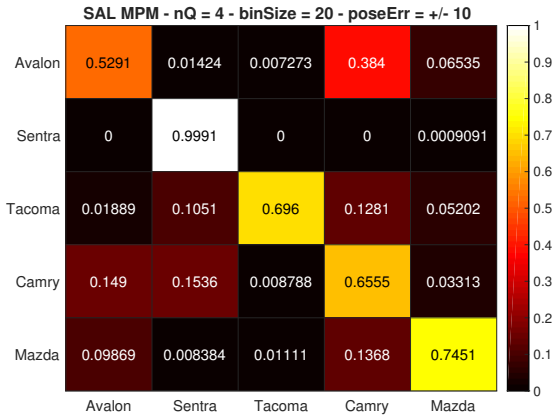

Train CAD Test SAL - MPM - nQ = 2 - binSize = 20 - poseErr = +/- 10
Avalon 0.5078 0.00404 0.0499 0.3277 0.1106
Sentra 0.01879 0.9659 0.004545 0.009596 0.001212
Tacoma 0.05354 0.05606 0.7197 0.09586 0.07485
Camry 0.2863 0.08424 0.04424 0.5336 0.05162
Mazda 0.02859 0.000303 0.04303 0.05475 0.8733
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

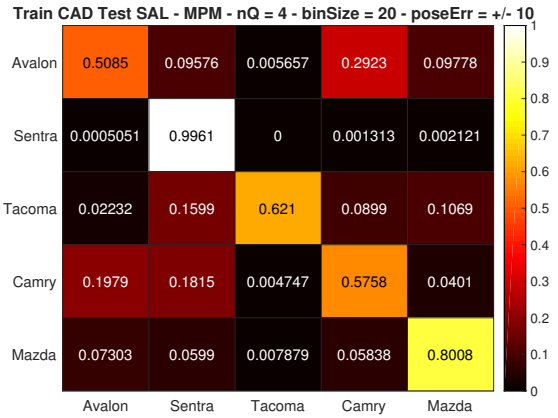

SAL MPM - nQ = 2 - binSize = 40 - poseErr = +/- 10
Avalon 0.5501 0.002828 0.0007071 0.292 0.1543
Sentra 0.007273 0.979 0 0.01374 0
Tacoma 0.04768 0.0301 0.6609 0.0897 0.1716
Camry 0.402 0.1044 0.002121 0.4241 0.06727
Mazda 0.04131 0 0.04616 0.1001 0.8124
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0


Train CAD Test SAL - MPM - nQ = 2 - binSize = 40 - poseErr = +/- 10
Avalon 0.4778 0.002424 0.0005051 0.3679 0.1514
Sentra 0.009293 0.9742 0 0.01535 0.001111
Tacoma 0.02303 0.02515 0.6475 0.1724 0.1319
Camry 0.3137 0.07556 0.001515 0.5413 0.06788
Mazda 0.05667 0.000101 0.0497 0.07303 0.8205
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

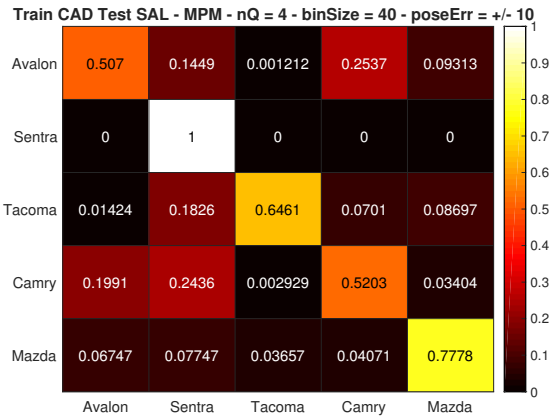

SAL MPM - nQ = 4 - binSize = 10 - poseErr = +/- 10
Avalon 0.5534 0.01545 0.01091 0.3639 0.05626
Sentra 0.002121 0.9974 0 0.000404 0.000101
Tacoma 0.02131 0.1139 0.6837 0.1235 0.05747
Camry 0.1594 0.16 0.007576 0.642 0.03101
Mazda 0.08071 0.008788 0.008586 0.1165 0.7855
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

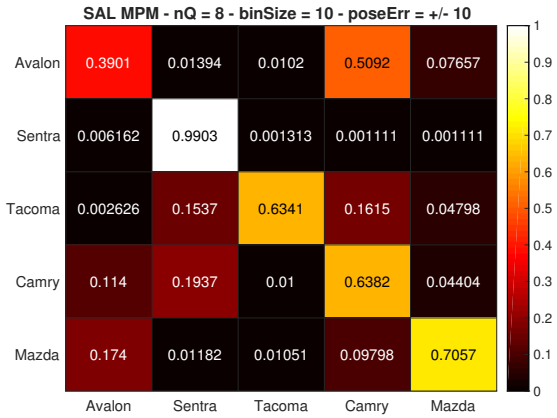

Train CAD Test SAL - MPM - nQ = 4 - binSize = 10 - poseErr = +/- 10
Avalon 0.5016 0.09051 0.01152 0.2999 0.09646
Sentra 0.00101 0.9959 0 0.002121 0.00101
Tacoma 0.01505 0.1481 0.6385 0.1052 0.09323
Camry 0.1789 0.1504 0.009394 0.6212 0.0401
Mazda 0.06707 0.06141 0.02121 0.0399 0.8104
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

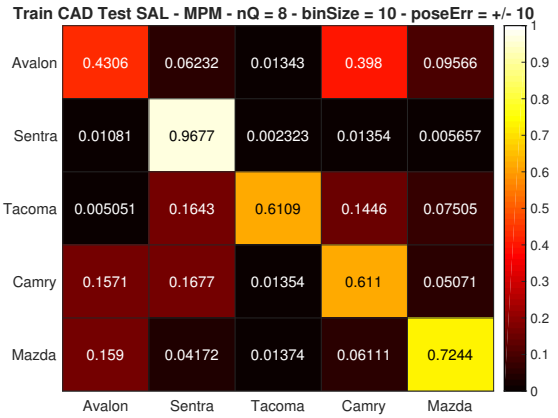

SAL MPM - nQ = 4 - binSize = 20 - poseErr = +/- 10
Avalon 0.5291 0.01424 0.007273 0.384 0.06535
Sentra 0 0.9991 0 0 0.0009091
Tacoma 0.01889 0.1051 0.696 0.1281 0.05202
Camry 0.149 0.1536 0.008788 0.6555 0.03313
Mazda 0.09869 0.008384 0.01111 0.1368 0.7451
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

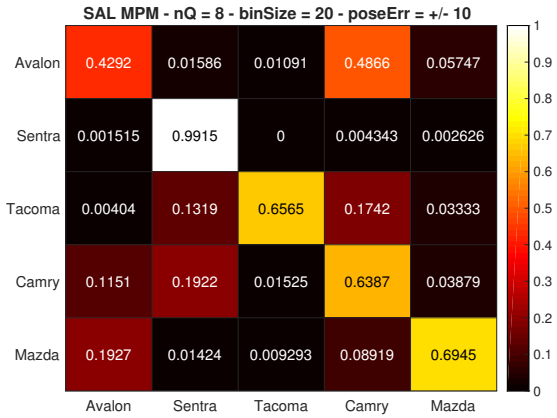

Train CAD Test SAL - MPM - nQ = 4 - binSize = 20 - poseErr = +/- 10
Avalon 0.5085 0.09576 0.005657 0.2923 0.09778
Sentra 0.0005051 0.9961 0 0.001313 0.002121
Tacoma 0.02232 0.1599 0.621 0.0899 0.1069
Camry 0.1979 0.1815 0.004747 0.5758 0.0401
Mazda 0.07303 0.0599 0.007879 0.05838 0.8008
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

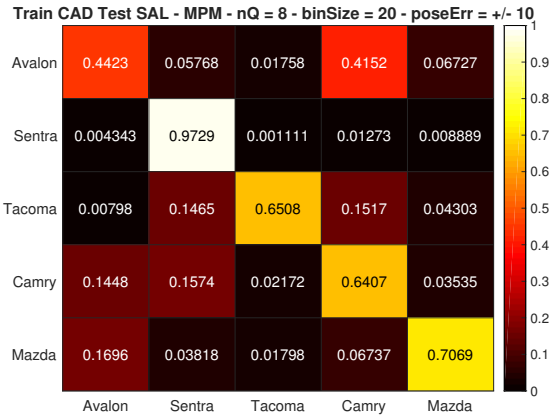

SAL MPM - nQ = 4 - binSize = 40 - poseErr = +/- 10
Avalon 0.5468 0.04556 0.004444 0.3503 0.05293
Sentra 0 1 0 0 0
Tacoma 0.0104 0.09737 0.6941 0.1346 0.06343
Camry 0.209 0.2002 0.01172 0.5388 0.0403
Mazda 0.05919 0.005051 0.05879 0.1194 0.7576
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0


Train CAD Test SAL - MPM - nQ = 4 - binSize = 40 - poseErr = +/- 10
Avalon 0.507 0.1449 0.001212 0.2537 0.09313
Sentra 0 1 0 0 0
Tacoma 0.01424 0.1826 0.6461 0.0701 0.08697
Camry 0.1991 0.2436 0.002929 0.5203 0.03404
Mazda 0.06747 0.07747 0.03657 0.04071 0.7778
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0

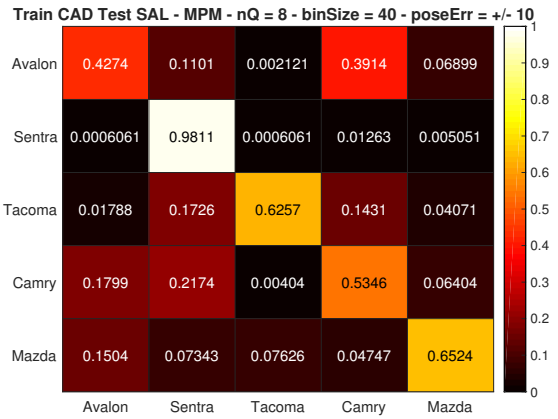

SAL MPM - nQ = 8 - binSize = 10 - poseErr = +/- 10
Avalon 0.3901 0.01394 0.0102 0.5092 0.07657
Sentra 0.006162 0.9903 0.001313 0.001111 0.001111
Tacoma 0.002626 0.1537 0.6341 0.1615 0.04798
Camry 0.114 0.1937 0.01 0.6382 0.04404
Mazda 0.174 0.01182 0.01051 0.09798 0.7057
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0


Train CAD Test SAL - MPM - nQ = 8 - binSize = 10 - poseErr = +/- 10
Avalon 0.4306 0.06232 0.01343 0.398 0.09566
Sentra 0.01081 0.9677 0.002323 0.01354 0.005657
Tacoma 0.005051 0.1643 0.6109 0.1446 0.07505
Camry 0.1571 0.1677 0.01354 0.611 0.05071
Mazda 0.159 0.04172 0.01374 0.06111 0.7244
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0


SAL MPM - nQ = 8 - binSize = 20 - poseErr = +/- 10
Avalon 0.4292 0.01586 0.01091 0.4866 0.05747
Sentra 0.001515 0.9915 0 0.004343 0.002626
Tacoma 0.00404 0.1319 0.6565 0.1742 0.03333
Camry 0.1151 0.1922 0.01525 0.6387 0.03879
Mazda 0.1927 0.01424 0.009293 0.08919 0.6945
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0


Train CAD Test SAL - MPM - nQ = 8 - binSize = 20 - poseErr = +/- 10
Avalon 0.4423 0.05768 0.01758 0.4152 0.06727
Sentra 0.004343 0.9729 0.001111 0.01273 0.008889
Tacoma 0.00798 0.1465 0.6508 0.1517 0.04303
Camry 0.1448 0.1574 0.02172 0.6407 0.03535
Mazda 0.1696 0.03818 0.01798 0.06737 0.7069
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0


SAL MPM - nQ = 8 - binSize = 40 - poseErr = +/- 10
Avalon 0.4307 0.0597 0.003939 0.4459 0.0598
Sentra 0.005253 0.9806 0.002828 0.004545 0.006768
Tacoma 0.01455 0.1194 0.6462 0.1875 0.03242
Camry 0.1499 0.2064 0.009596 0.5806 0.05354
Mazda 0.1611 0.0197 0.07646 0.09232 0.6504
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0


Train CAD Test SAL - MPM - nQ = 8 - binSize = 40 - poseErr = +/- 10
Avalon 0.4274 0.1101 0.002121 0.3914 0.06899
Sentra 0.0006061 0.9811 0.0006061 0.01263 0.005051
Tacoma 0.01788 0.1726 0.6257 0.1431 0.04071
Camry 0.1799 0.2174 0.00404 0.5346 0.06404
Mazda 0.1504 0.07343 0.07626 0.04747 0.6524
Avalon Sentra Tacoma Camry Mazda
1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0