ATR by means of Polarimetric ISAR Images and multi-view 3D InISAR

“Second Progress Report”

Document Reference: **Second Progress Report**
Deliverable: **D2**

Title: **Second Progress Report**

Contractor: **CNIT**

Prepared by:
Marco Martorella,
Elisa Giusti
Ajeet Kumar
Francesco Mancuso
Giulio Meucci
Alberto Lupidi
Selenia Ghio

Work Package: **WP2**
WP2.1: Pol-InISAR image formation algorithm implementation,
WP2.2: Pol-InISAR image formation algorithm performance analysis,
WP2.3: Development of an automatic classifier that makes use of Pol-InISAR images.

Version: **0.2**

Classification: **Unclassified**

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 15-06-2022	**2. REPORT TYPE** Progress Report (Year #2)	**3. DATES COVERED** *(From - To)* 15-06-2021 to 15-06-2022

4. TITLE AND SUBTITLE
Title: ATR by means of Polarimetric ISAR Images and multi-view 3D InISAR
Sub-title: Second Progress Report

5a. CONTRACT NUMBER
N62909-20-1-2062

5b. GRANT NUMBER
GRANT13048417

5c. PROGRAM ELEMENT NUMBER
1000012606

6. AUTHOR(S)
Marco Martorella,, Elisa Giusti, Ajeet Kumar, Francesco Mancuso, Giulio Meucci, Alberto Lupidi, Selenia Ghio

5d. PROJECT NUMBER
N/A

5e. TASK NUMBER
N/A

5f. WORK UNIT NUMBER
N/A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
CNIT - Radar and Surveillance Systems National Laboratory
Galleria Gerace 14, 56124 Pisa, Italy

8. PERFORMING ORGANIZATION REPORT NUMBER
D2

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research Global
86 Blenheim Crescent
Ruislip MX HA4 7HB
United Kingdom

10. SPONSOR/MONITOR'S ACRONYM(S)
ONRG - London

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
N/A

12. DISTRIBUTION/AVAILABILITY STATEMENT
A - Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
None

14. ABSTRACT

According to the project program of work, this second progress report includes a summary of the technical activities performed during the project second year. More specifically, it includes:

- A review of fully-polarimetric 3D InISAR algorithms and their implementation through pseudocodes.
- A performance analysis of the 3D InISAR algorithms. In this regard two different approaches have been proposed and compared, namely the Coherence-based and the Span-based. Pros and cons of both approaches have been highlighted using either simulated and real data.
- The design and a preliminary implementation (for the purpose of a preliminary validation) of two ATR algorithms, which will be fully tested during the project's third year.

15. SUBJECT TERMS
Radar imaging, 3D imaging, polarimetric radar, classification, recognition

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Marco Martorella
UU	UU	UU	UU	75	**19b. TELEPHONE NUMBER** *(Include area code)* +39 347 5848897

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

According to the project program of work, this second progress report includes a summary of the technical activities performed during the project second year. More specifically, it includes:

- A review of fully-polarimetric 3D InISAR algorithms and their implementation through pseudo-codes.
- A performance analysis of the 3D InISAR algorithms. In this regard two different approaches have been proposed and compared, namely the Coherence-based and the Span-based. Pros and cons of both approaches have been highlighted using either simulated and real data. Simulated data, in fact, offer the possibility to numerically quantify the 3D reconstruction accuracy but have some limitations as they cannot reproduce faithfully the polarimetric target scattering mechanism at different aspect angles. On the other hand real data are realistic (even though acquired in a controlled geometry) but cannot be used to numerically quantify the reconstruction accuracy.
- The design and a preliminary implementation (for the purpose of a preliminary validation) of two ATR algorithms, which will be fully tested during the project's third year.

Chapter 2

Pol-InISAR image formation algorithm implementation

A complete description of Polarimetric Interferometry Inverse Synthetic Aperture Radar (Pol-InISAR) 3D imaging method for the detection of non-cooperative targets is presented in this Chapter. The block diagram of general steps of the Pol-InISAR-based 3-D imaging is described in Fig. 2.1. The detailed description of the steps are added in the section below.

2.1 General description of the proposed Pol-InISAR based 3D Imaging

The block diagram of Fig. 2.1 shows general Pol-InISAR based 3D target formation steps where N, $\forall N > 1$ number of Pol-ISAR datasets, acquired from N different points in space, can be utilized to form the 3D point cloud corresponding to the target. In the work of this report, we outlined the procedure for two specific cases: N=2 and N=3, corresponding to the Tank-72 and boat simulator ISAR data conditions, respectively. The Tank-T72 data-set has been acquired using different elevation angles. The target rotates on a turntable and as a consequence, two vertical antennas are enough for the application of 3D InISAR. Hence, $N = 2$ in this case. The antenna vertical baseline is suitable for the application of the proposed approach.

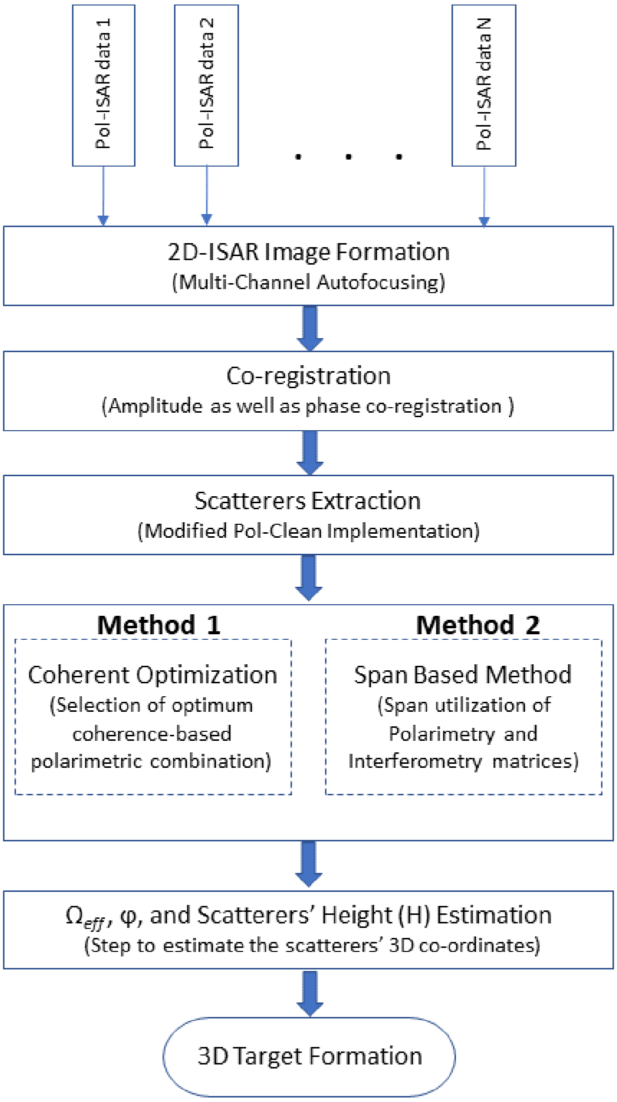


Figure 2.1: General Block Diagram describing the steps of 3-D target formation using Pol-InISAR based proposed methods.

2.1.1 2D ISAR image formation

One can observe from Fig. 2.1 that 3D imaging of non-co-operative targets starts with a 2D ISAR image formation. We are assuming here that the reader of this report has already knowledge of 2D ISAR algorithms and image formation techniques. Conventional 2D ISAR approaches have been largely investigated in the literature and therefore this part will not be analyzed here. More details can be found in [1].

2.1.2 Co-registration

The step of co-registration for the acquired images is essential to perform at either image- or signal-level to mitigate the effect of image mismatching created by the mismatch in target motion characteristics and wave path information collected at the different point of acquisitions in space [2]. The complete details of the image co-registration and its utility in 3D target formation can be found in [2–4]. The following steps are implemented to perform the co-registration [3].

- Select one reliable scatterer, possibly with the highest intensity (brightest scatterer).
- All the images, collected from different points in space are required to be translated or shifted in such a way that the selected scatterer becomes new center of the images at zero Doppler and range coordinates.
- The phase term of the selected scatterer is estimated and further subtracted or removed from the entire images.

2.1.3 Scatterers Extraction

To reduce the computation burden significantly by reducing the data size without losing any vital information, the scattering center extraction through the polarimetry CLEAN is required to be performed. This particular step has significant importance in ISAR imaging-based real time automatic target recognition (ATR), as it is difficult to handle the large data size in real time. In the work of this report, a modified Polarimetry CLEAN (Pol-CLEAN) is proposed and implemented. The modified Pol-CLEAN technique tries to use the polarimetric features as an additional information to separate scattering centers that are close in space, leading to a more reliable estimation of their complex amplitudes. The complete description of the implemented modified polarimetry CLEAN is added in Section 2.2.

2.1.4 Coherence Optimization and Span Based Methods

The availability of full-polarimetry data facilitates to obtain the interferogram information from all possible linear combinations of polarization states. Further among these combinations, the one with more reliable interferogram phase value can be selected. In this report, we outline two methods based on the selection of polarimetric combination: the coherence optimization-based method and the Span-based method. In coherence-optimization-based method, the optimum polarimetric combination is being selected based on the criteria of maximum coherence. While, in Span based method, the Span combination of polarimetry and interferometry matrices are being utilized by considering that the Span is representation of the total power that can improve the SNR ratio. The detailed description of coherence-optimization-based and Span-based methods are added in Section 2.3 and Section 2.4, respectively.

2.1.5 Scatterers' Height Estimation

The last vital step in the Pol-InISAR 3D imaging method is the height estimation (see Fig. 2.1) which is separately described here for two-acquisitions and three-acquisitions conditions. For the condition of two-acquisitions, the height of scatterers (H_{2A}) can be calculated by using the following equation.

$$H_{2A} = \frac{cR_0\Delta\theta_{12}}{4\pi f_0 d_{_V}} \tag{2.1}$$

where θ_{12} is the interferometry phase between the acquired data of two antennas: antenna 1 & antenna 2, as shown in Fig. 3.4. This formula has been used with the Tank-T72 data-set. This formula can be applied for moving targets having the target effective rotation vector aligned with the baseline axis (please refer to Fig. Fig. 3.1 for the definition of effective rotation vector). Terrestrial targets observed from an airborne systems may fall in this category. In the L-type antenna arrangement based three-acquisitions condition (as depicted in Fig. 3.1), the height ($H_{3\mathrm{A}}$) estimation can be performed through the following equation [1, 3].

$$H_{3\mathrm{A}} = \frac{c}{4\pi f_0} R_0 \left(\frac{\Delta\theta_V}{d_{_V}} \cos\phi - \frac{\Delta\theta_H}{d_{_H}} \sin\phi \right) \tag{2.2}$$

$$V_{_C} = \frac{R_0\Omega}{2\pi} \left(\frac{\Delta\theta_H}{d_H} \cos\phi + \frac{\Delta\theta_V}{d_V} \sin\phi \right) \tag{2.3}$$

where c is the speed of light, f_0, $d_{_H}$, and $d_{_V}$ are the radar-geometry parameters indicating central frequency, horizontal baseline, and vertical baseline. The angles $\Delta\theta_V$ and $\Delta\theta_H$ are the interferometry angles between antennas: V_ξ & C_ξ, and H_ξ & C_ξ, respectively. The angle ϕ in (2.2) is the angle between y_3 and Ω_{eff}, as shown in Fig. 3.1. As suggested in [1, 3], the ϕ angle and Ω indicating orientation and modulus of Ω_{eff}, can be calculated through (2.3), by means of the least square error approach, [1].

2.2 Polarimetric CLEAN: Conventional and Modified version

Polarimetric CLEAN

The Polarimetric CLEAN [5] exploits the polarimetric domain by selecting the Pauli's channel (Eq. 2.4) that contains the brightest scattering centre in the polarimetric ISAR image $\boldsymbol{I}(\tau, \nu)$:

$$(\bar{\tau}, \bar{\nu}, \bar{i}) = \arg \max_{(\tau,\nu,i)} \left\{ \left| I^{(i)}(\tau, \nu) \right| \right\} \tag{2.4}$$

Where $i \in \{1, 2, 3\}$, $\tau \in \{1, 2, \ldots, M\}$, $\nu \in \{1, 2, \ldots, N\}$ and M and N are the numbers of bins in range and cross-range. The estimation of the scatterer's PSF (Point Spread Function) is done in the $\bar{i}$-*th* polarimetric channel, by minimising the image energy after every scattering centre removal. The whole procedure stops when all the scattering centres have been removed from the image (i.e., when the residual energy goes under a threshold that is usually defined in such a way to avoid false detection - cancellation residues, noise).

Modified Polarimetric CLEAN

The Modified Polarimetric CLEAN [6] (Algorithm 1) searches for the coordinates of the brighest scatterer using the energy of the polarimetric ISAR image. It performs the estimation of the scatterer's PSF in a new image $P(\tau, \nu)$, obtained by projecting the polarimetric ISAR image onto the polarimetric vector $\boldsymbol{k}_C$ associated to the maximum of the energy (i.e., an image where the SNR is maximum for that scattering centre):

$$\begin{aligned} P(\tau, \nu) &= \left| \langle \boldsymbol{I}(\tau, \nu), \boldsymbol{k}_C \rangle \right| \\ &= \left| \boldsymbol{k}_C^{\dagger} \boldsymbol{I}(\tau, \nu) \right| \\ &= \left| \boldsymbol{k}_C^{\dagger} \boldsymbol{I}(\tau_m, \nu_n) \right|, \qquad \begin{cases} 1 \leq m \leq M \\ 1 \leq n \leq N \end{cases} \end{aligned} \tag{2.5}$$

where $\langle \cdot, \cdot \rangle$ is the inner product evaluated between each pixel of the image and the polarimetric vector $\boldsymbol{k}_C$ (Figure 2.3), M and N are the number of range and cross-range bins. After this projection, the PSF is estimated and the scatterer is subtracted from the image. In Figure 2.2, is shown the block diagram of the procedure: the process described above is done in the leftmost orange sub-block and it stops when a specific condition is met (i.e., when the residual energy is less than a certain threshold). The second part of the algorithm (right side of Figure 2.2) uses the information about the

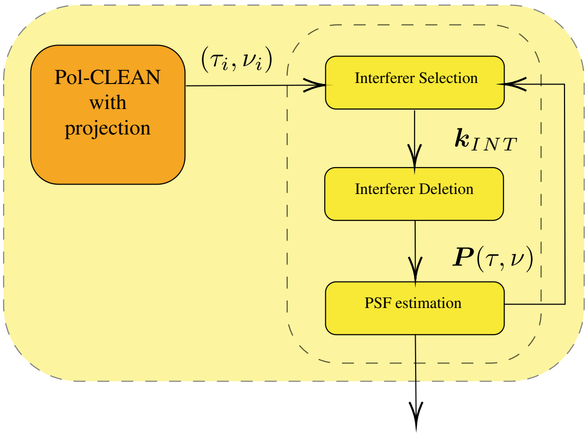


Figure 2.2: Modified Polarimetric CLEAN block diagram

position of the scattering centres in the image to repeat the extraction operation, but this time it tries to minimise the interference due to nearby scatterers before estimating the PSF. This is achieved through a process of interfering scatterer identification and deletion. It works as follow:

1. a scatterer from the list is selected as a candidate for the extraction,
2. the strongest interfering scattering centre is searched evaluating two metrics: the interfering power and the distance from the candidate scattering centre. The distance metric is defined as

$$d_{i,j} = \frac{1}{|x_j - x_i|} \cdot \frac{1}{|y_j - y_i|} \tag{2.6}$$

   where i refers to the candidate scatterer and j refers to the interfering one, and

   - if a interfering scattering centre is found, the amplitude of that scattering centre is deleted from the image

   $$P(\tau, \nu) = |\boldsymbol{I}(\tau, \nu)| - |\langle \boldsymbol{I}(\tau, \nu), \boldsymbol{k}_{INT} \rangle| \tag{2.7}$$

   - or, if there are no interfering scattering centre, the SNR is maximized using the candidate polarimetric vector, and

   $$P(\tau, \nu) = |\langle \boldsymbol{I}(\tau, \nu), \boldsymbol{k}_C \rangle| \tag{2.8}$$

3. the estimation of the the candidate scattering centre PSF characteristics is done in order to remove it from the image.

The second block procedure stops when the vector of coordinates passed by the previous block is

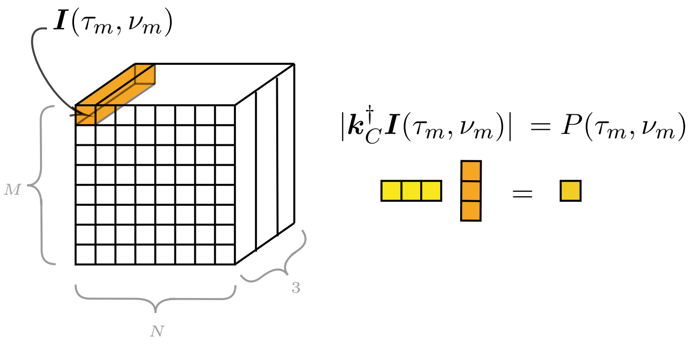


Figure 2.3: Graphical representation of data structure and the pixel-by-pixel projection method. The resulting image $P(\tau, \nu)$ is an $M \times N$ matrix

empty. The outputs of the Modified Pol-CLEAN are the coordinates of the extracted scattering centres and their estimated complex amplitudes.

Algorithm 1: Modified Polarimetric CLEAN

```
Data: polarimetric ISAR image $\boldsymbol{I}(\tau,\nu)$
Result: coordinates and complex amplitudes of the scattering centres
/* Pol-CLEAN with Projection */
set energy threshold;
$E \leftarrow \sum_{i=1}^{3} \left|I^{(i)}(\tau,\nu)\right|^2$;
while $E \geq threshold$ do
    $(\bar{\tau},\bar{\nu}) \leftarrow \arg\max_{(\tau,\nu)} \{E\}$;
    $\boldsymbol{k}_C \leftarrow \boldsymbol{I}(\bar{\tau},\bar{\nu})$;
    $P(\tau,\nu) \leftarrow |\langle \boldsymbol{I}(\tau,\nu), \boldsymbol{k}_C \rangle|$;
    $\hat{\boldsymbol{I}}_{PSF}(\tau,\nu) \leftarrow$ PolPSFestimate($P(\tau,\nu)$);
    $\boldsymbol{I}(\tau,\nu) \leftarrow \boldsymbol{I}(\tau,\nu) - \hat{\boldsymbol{I}}_{PSF}(\tau,\nu)$;
    $E \leftarrow \sum_{i=1}^{3} \left|I^{(i)}(\tau,\nu)\right|^2$;
end while
/* Pol-CLEAN with Interferer Suppression */
ind $\leftarrow$ 1;
N $\leftarrow$ number of extracted scattering centres;
while $ind \leq N$ do
    $\boldsymbol{k}_C \leftarrow \boldsymbol{I}(\tau_{ind},\nu_{ind})$;
    $\boldsymbol{k}_{INT} \leftarrow$ InterfererSearch();
    if there is an interferer then
        $P(\tau,\nu) \leftarrow |\boldsymbol{I}(\tau,\nu)| - |\langle \boldsymbol{I}(\tau,\nu), \boldsymbol{k}_{INT} \rangle|$;
    else
        $P(\tau,\nu) \leftarrow |\langle \boldsymbol{I}(\tau,\nu), \boldsymbol{k}_C \rangle|$;
    end if
    $\hat{\boldsymbol{I}}_{PSF}(\tau,\nu) \leftarrow$ PolPSFestimate($P(\tau,\nu)$);
    $\boldsymbol{I}(\tau,\nu) \leftarrow \boldsymbol{I}(\tau,\nu) - \hat{\boldsymbol{I}}_{PSF}(\tau,\nu)$;
    remove the extracted scattering centre from the list;
end while
```

2.3 Coherence optimization based method

The single-pol based conventional 3D InISAR system acquires propagation delay phase (ϕ_p) that allows to calculate single-channel based interferogram information only. Contrary to this, the full-polarimetry based 3D InISAR system acquires scattering information phase (ϕ_s) also along with the propagation delay phase. By combining the ϕ_p and ϕ_s phase values along with the other information associated with full-polarimetry, the interferogram from all possible linear combinations of polarization states can be obtained. Further among these combinations, the optimum one can be selected. The polarimetric combination achieving the highest possible coherence reduces any chance of uncertainties in the interferogram phase [7], and therefore can generate most reliable height estimation leading to the optimum 3D target reconstruction. Hence, in this work, we selected the optimum polarimetric combination which obtains the maximum possible coherence.

The coherence optimization based selection of optimum polarimetric combination was initially investigated for natural scatterers in [8, 9]. In general, these type of scatterers exhibit common polarimetric features. Contrarily, in the case of man-made complex targets, each scatterers can exhibit proper polarimetric behavior and therefore the corresponding processing should be applied to each scatterer that composes the target. In this paper, the full-polarimetry ISAR based coherent optimization approach is being implemented on the real Tank-72 ISAR data and boat simulator ISAR data, both exhibiting complex target structure scenarios. The T72 and boat simulator datasets exhibit two-acquisitions and three-acquisitions based InISAR radar scenarios where the two and three Pol-ISAR datasets are acquired from two and three different points in space, respectively. The procedure for the selection of coherence optimization based optimum polarimetric combination is different for these two different conditions. For two-acquisitions condition, the single-baseline coherence optimization approach is useful, while for more than two-acquisitions condition, multiple-baseline coherence optimization can be used. Both of these methods are described below in detail.

2.3.1 Two-acquisition based coherent optimization approach

The full-pol data acquired by the two antennas situated at two different points in space can be expressed in terms of the scattering vectors k_1 and k_2, as follows.

$$k_1 = [{S_{\mathrm{HH}}}^1 + {S_{\mathrm{VV}}}^1 \;\; {S_{\mathrm{HH}}}^1 - {S_{\mathrm{VV}}}^1 \;\; 2{S_{\mathrm{HV}}}^1]^T \tag{2.9}$$

$$k_2 = [{S_{\mathrm{HH}}}^2 + {S_{\mathrm{VV}}}^2 \;\; {S_{\mathrm{HH}}}^2 - {S_{\mathrm{VV}}}^2 \;\; 2{S_{\mathrm{HV}}}^2]^T \tag{2.10}$$

Further, by performing the outer product formed from k_1 and k_2, the 6×6 polarimetric interferometry matrix ($[T_6]$) can be generated as

$$[T_6] = \left\langle \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \cdot \begin{bmatrix} {k_1}^* & {k_2}^* \end{bmatrix} \right\rangle = \begin{bmatrix} T_{11} & \Omega_{12} \\ \Omega_{12}^\dagger & T_{22} \end{bmatrix}, \tag{2.11}$$

The matrices T_{11} and T_{22} contain polarimetric information, whereas the matrix Ω_{12} contains polarimetric as well as interferometry information. By jointly utilizing these matrices, the projection vector/vectors can be derived that helps to project k_1 and k_2 into two new polarimetric combinations which maximize the coherence between two acquisitions. The most general approach based on two projection vectors was established by Cloude *et al.* in [8], however, as the projection vector is complex quantity, utilizing two different projection vectors (w_1 and w_2) for two channels can induce an undesired phase shift in the interferogram in case $\arg(w_1^\dagger w_2) \neq 0$ [9, 10]. This erroneous phase shift can be avoided by using a single or unique projection vector based new approach, established by modifying the conventional optimization theory [11]. This new approach converge to the following 3×3 eigenvalue problem through which the unique projection vector (w) can be estimated.

$$[S]^{-1}[R]w = vw \tag{2.12}$$

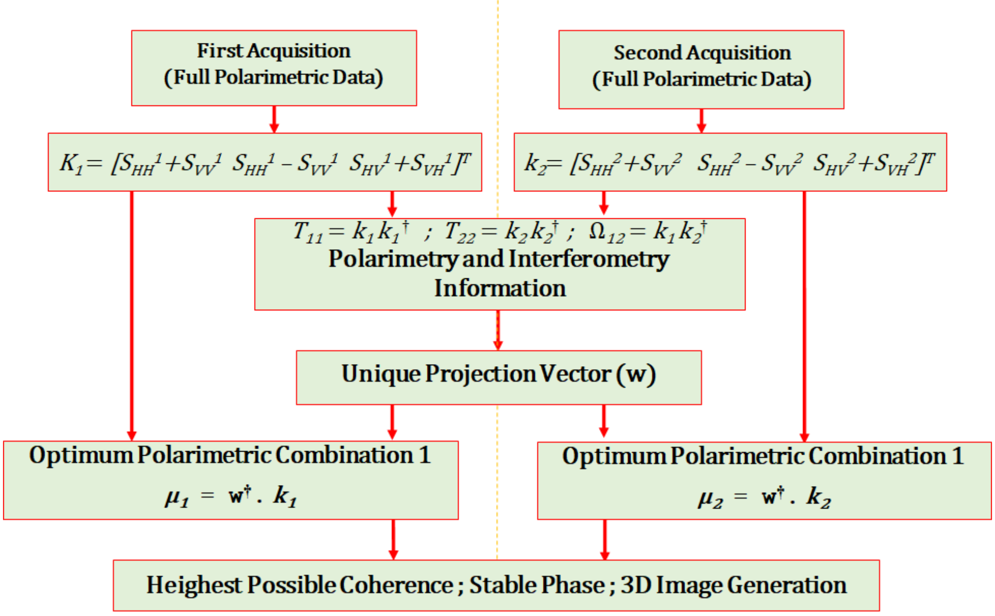


Figure 2.4: Two-acquisition based Coherent optimization approach

where $[S] = [T_{11}] + [T_{22}]$ and $[R] = [\Omega_{12}] + [\Omega_{12}]^{\dagger}$. Equation (2.12) provides three eigenvectors among which the one corresponding to the highest eigenvalue v needs to be selected as an unique projection vector w. Further, by using w, the scattering coefficients for two different acquisitions can be estimated as $\mu_1 = w^{\dagger}k_1$ and $\mu_2 = w^{\dagger}k_2$. Consequently, the coherence (γ_w) and phase ($\arg(\gamma_w)$) can be calculated as

$$\gamma_w = \frac{\langle \mu_1 \mu_2^* \rangle}{\sqrt{\langle \mu_1 \mu_1^* \rangle \langle \mu_2 \mu_2^* \rangle}} = \frac{w^{\dagger}[\Omega_{12}]w}{\sqrt{(w^{\dagger}[T_{11}]w)(w^{\dagger}[T_{22}]w)}} \tag{2.13}$$

$$\arg(\gamma_w) = \arg(\langle \mu_1 \mu_2^* \rangle) = \arg(w^{\dagger}[\Omega_{12}]w). \tag{2.14}$$

The coherence γ_w obtained using (2.13) is optimized coherence and the phase $\arg(\gamma_w)$ calculated using (2.14) is the optimized phase, leading to a better and reliable 3D reconstruction. The overview of the method can be looked into the block diagram shown in Fig. 2.4.

This aforementioned approach is suitable for the scenario when full-polarimetry data acquired in two acquisitions from two different point in space, generating PolInISAR information, such as for the case of Tank 72 data implemented in this paper. However, to make this unique projection vector based method suitable for the condition of more than two acquisitions, an extended version of this method, i.e. multiple-acquisition based optimization approach is described below in details.

2.3.2 Multiple-acquisition based coherence optimization approach

When more than two acquisitions from different points in space are available, such as for the boat simulator data implemented in this work, the acquired full-pol information can be expressed in terms of scattering vector as,

$$k_i = [{S_{\text{HH}}}^i + {S_{\text{VV}}}^i \;\; {S_{\text{HH}}}^i - {S_{\text{VV}}}^i \;\; 2{S_{\text{HV}}}^i]^T \tag{2.15}$$

where $i \in (1, 2, ..., \text{N})$ for N number of acquisitions. In this paper, we implemented multiple-acquisition based coherence optimization approach only for three acquisitions, so hereafter $\text{N} = 3$ is considered. For this condition, the combined polarimetric interferometry matrix can be generated as,

$$[T_9] = \left\langle \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} \cdot \begin{bmatrix} k_1^* & k_2^* & k_3^* \end{bmatrix} \right\rangle = \begin{bmatrix} T_{11} & \Omega_{12} & \Omega_{13} \\ \Omega_{12}^{\dagger} & T_{22} & \Omega_{23} \\ \Omega_{13}^{\dagger} & \Omega_{23}^{\dagger} & T_{33} \end{bmatrix} \tag{2.16}$$

The matrices T_{11}, T_{22}, and T_{33} contain polarimetric information associated with the three different antennas, whereas Ω_{12}, Ω_{13}, and Ω_{23} contain polarimetry as well as interferometry information associated with antennas: first - second, first - third, and second - third, respectively. Similar to the two-acquisition based optimization approach, in this three-acquisition condition, the sum of coherence parameter, i.e. $\sum |\gamma_{ij}|$ is taken as the optimization criterion to evaluate the unique projection vector (w), where γ_{ij} can be expressed as

$$\gamma_{ij} = \mathrm{w}^{\dagger} \Pi_{ij} \mathrm{w} \tag{2.17}$$

In (2.17), the parameters Π_{ij} and w can be estimated as

$$\mathrm{w} = \frac{\sqrt{T_e} w}{w^{\dagger} \sqrt{T_e} w}; \;\; \Pi_{ij} = T_e{}^{-1/2} \Omega_{ij} T_e{}^{-1/2}; \;\; T_e = \frac{1}{3} \sum_{1}^{3} T_{ii} \tag{2.18}$$

The process of unique projection vector (w) estimation is summarized in the iterative based approach described in the Algorithm 1. Further with the estimated w, the full-pol information of the three antennas V_{ξ}, C_{ξ}, and H_{ξ} can be projected to estimate the corresponding scattering coefficients as $\mu_1 = w^{\dagger} k_1$, $\mu_2 = w^{\dagger} k_2$, and $\mu_3 = w^{\dagger} k_3$. These three scattering coefficients can generate optimum $\sum |\gamma_{ij}|$ as well as optimum interferometric phases leading to better and reliable 3D reconstruction. The overview of the method can be looked into the block diagram shown in Fig. 2.5.

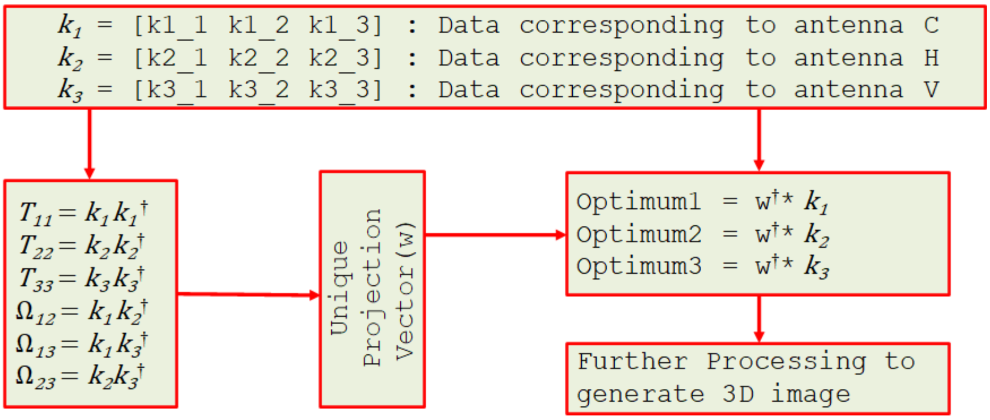


Figure 2.5: Three-acquisition based coherent optimization approach

```
Algorithm 2: The algorithm for three-acquisition based coherence optimization
  Data: matrices $\Pi_{ij}$ and $T_e$
  Result: unique projection vector $w$
  /* Step 1 */
  Initializing eigenvalue: $\lambda_{\text{max}} = 0$;
  Initializing optimal phase shift: $\theta_{ij} = \arg(\texttt{trace}(\Pi_{ij}))$;
  /* Step 2 */
  Calculate Hermitian Matrix:
      $$\mathrm{H} = \sum_{i=1}^{3} \sum_{j=1 \neq i}^{3} \Pi_{ij} e^{-\iota\theta_{ij}}$$
  Calculate the eigenvalues and eigenvectors solving $\mathrm{Hw} = \lambda \mathrm{w}$.
  /* Step 3 */
  repeat
      Update maximum eigenvalue $\lambda_{\text{max}} = \lambda^{u}_{\text{max}}$ and eigenvector w by selecting the largest
        estimated eigenvalue, and the corresponding eigenvector, respectively;
      Estimate the optimal phase $\theta^{u}_{ij} = \arg(\mathrm{w}^{\dagger}\Pi_{ij}\mathrm{w})$ and again with $\theta^{u}_{ij}$, calculate the updated
        $\mathrm{H}^{u}$ from:
          $$\mathrm{H}^{u} = \sum_{i=1}^{3} \sum_{j=1 \neq i}^{3} \Pi_{ij} e^{-\iota\theta^{u}_{ij}}$$
      Solve $\mathrm{H}^{u}\mathrm{w} = \lambda \mathrm{w}$ and update the parameters $\lambda^{u}_{\text{max}}$ and w by selecting maximum
        eigenvalue and corresponding eigenvector;
  until $\lambda^{u}_{max} - \lambda_{max} \geq eps$;
  /* Step 4 */
  Calculate the unique projection vector by
      $$w = \frac{T_e^{-\frac{1}{2}}\mathrm{w}}{\mathrm{w}^{\dagger} T_e^{-\frac{1}{2}}\mathrm{w}}$$
```

2.3.3 Limitations and future research regarding coherence optimization based method

- The consideration of single (unique) projection vector for all the antenna to avoid additional undesired phase shift in interferogram, adds a constraint, namely that the same polarimetric vector, w is optimum for all the N antennas, which may be not the case depending on the geometry. In the future, We will try to remove this constraint by taking general case of three different projection vector (for three antennas) and simultaneously mitigate the effect of additional undesired phase shift.

- We have integrated a coherent-optimization approach to achieve maximum coherence, however, by looking in the coherence results, one can observe that even though the coherence values are improved, there is a scope of improving it further. We will try to modify the presented method to further improve the coherence value.

2.4 Span Based Approach

Span based method utilizes the Span information of polarimetry and interferometry matrices for 3D imaging of target. In this approach, the polarimetric information of one channel or antenna is taken as a reference and the polarimetric information of other channels are projected onto the reference one. This operation corresponds to the evaluation of the polarimetric span of the matrices T_{11} and Ω_{12} in the case of 2 antennas, and T_{11}, Ω_{12}, and Ω_{13} in the case of three antennas. By assuming that the polarimetric information seen by the all antennas is the same, this operation cancels the scattering phase information and evaluates the residual interferometry phase. Further, this interferometry phase is implemented to form the 3D image of the target. The procedures of Span based method, for the case of two and three antenna conditions, are shown in Figs. 2.7, and 2.6, respectively.

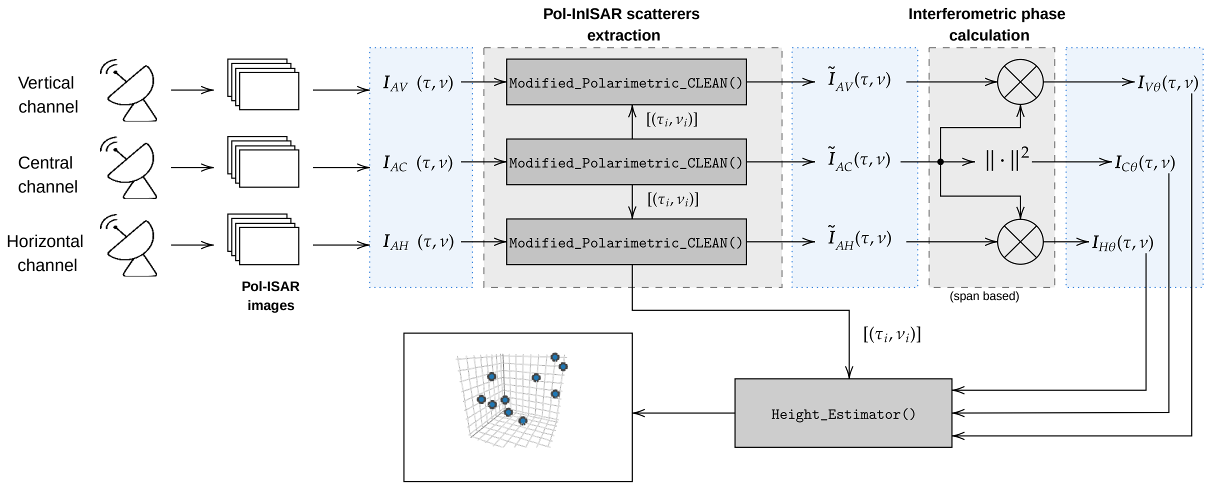


Figure 2.6: Processing chain flowchart of Span-based method (3 antenna condition)

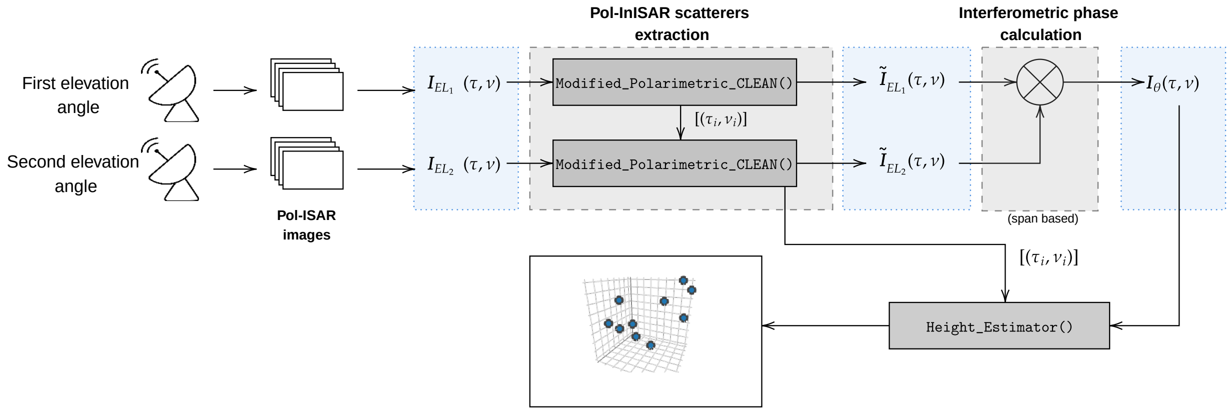


Figure 2.7: Processing chain flowchart of Span-based method (2 antenna condition)

2.4.1 Limitations and future research regarding Span based method

- The Span based method is established on the basis that the polarimetric information seen from all the antennas are same. In the simulated data, the condition is always well satisfied, but in real data scenarios, the violation of this condition can occur that can append significant amount of error.
- if compared with the Coherence based method, the Span based method utilizes the information contained in two and three interferometric, $\Omega_{i,j}$, and polarimetric, T_{ii}, matrices among four and nine available matrices for the condition of three antennas and two antennas based acquisitions, respectively. The other unused matrices can be utilized in future to improve further the result outcomes.

Chapter 3

Pol-InISAR image formation algorithm performance analysis

Validation of the proposed Pol-InISAR based proposed methods are carried out by using simulated as well as a real polarimetric ISAR datasets. The implemented simulated polarimetric ISAR data is a combination of point-like scatterers jointly constituting a complex boat type structure. Furthermore, the implemented real polarimetric ISAR data of a full-size Tank-72 (T72) is captured under Moving and Stationary Target Acquisition and Recognition (MSTAR) program by Georgia Technology Research Institute (GTRI). The Radar Geometry and the complete description of both the used datasets are added in the section below.

3.1 Datasets Description

3.1.1 Radar Geometry and Dataset description of Boat Simulator

The radar geometry of the simulator is shown in Fig. 3.1. The assumed geometry is the most popular 3D InISAR imaging geometry that contains three antennas located in the same plane orthogonal to the line of sight (LOS) and simultaneously lying on two orthogonal baselines as depicted in Fig. 3.1. The concept is to utilize three 2D complex ISAR images obtained from the three antennas V_ξ, C_ξ, and H_ξ, possessing similar range- and cross-range pattern but the different phases [12]. Further, decomposing the phase information can lead to the three-dimensional position of the scatterers. R_0 in Fig. 3.1 indicates the radar-target distance and is aligned with ξ_2 and $\Omega_T(t)$, where $\Omega_T(t)$ is the angular rotation vector which is applied to the center O of the target. The projection of $\Omega_T(t)$ onto

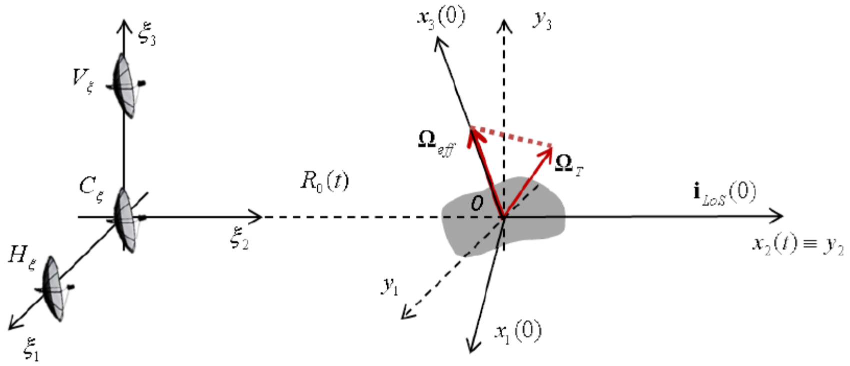


Figure 3.1: Radar target geometry of boat full-polarimetry ISAR data simulation.

the plane orthogonal to the LoS defines the effective rotation vector $\Omega_{eff}(t)$. The IPP is the plane orthogonal to $\Omega_{eff}(t)$. In this in-house developed boat simulator, we use 24 scatterer points situated in such a way that it forms a complex Astice (marine) boat like structure as depicted in Fig. 3.2. The benefit of using point-like simulator is that the point-to-point comparison between the actual and the reconstructed points is possible, as carried out in the next chapter (Section 3.2). The real image of the target, and the corresponding CAD model are shown in Fig. 3.2(a) and 3.2(b), respectively, whereas, the implemented scatterer points based boat simulator is shown in 3.2(c). The real boat is capable of generating six different scattering mechanisms of Table 3.1 [13, 14].

The angle θ_i in Table 3.1 indicates orientation angle of scattering structure. To make the tar-

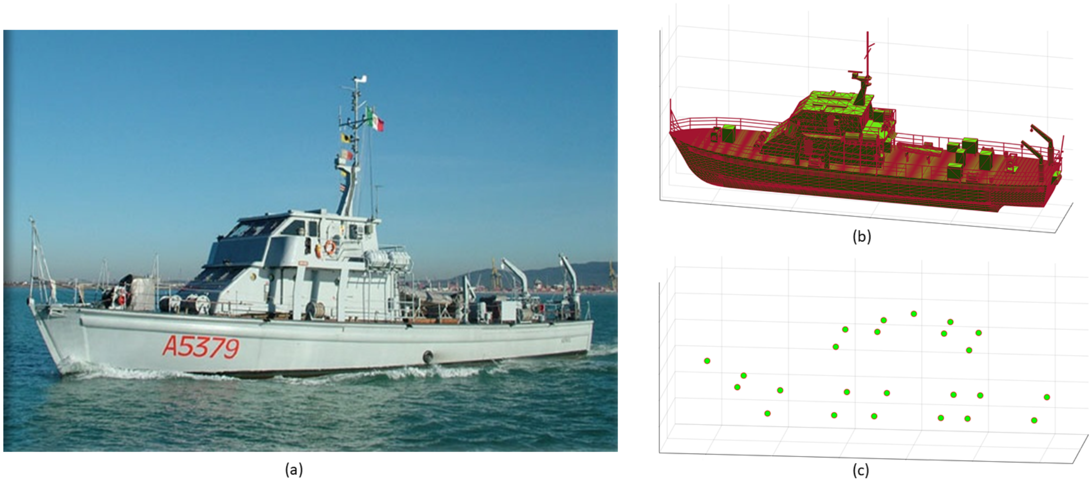

(a) (b) (c)

Figure 3.2: Image of the target (Astice boat) under observation (a) Original target image (b) CAD model corresponding to the target (c) Simulator corresponding to the target.

Table 3.1: Possible scattering mechanism generation from a real Boat.

Scattering mechanism	Sinclair matrix
Trihedral	$S_T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
Dihedral	$S_D = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Vertical dipole	$S_{VD} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$
Horizontal dipole	$S_{HD} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$
Helix	$S_H = \frac{1}{2} \begin{bmatrix} 1 & j \\ j & -1 \end{bmatrix}$
θ_i rotated dipole	$S_{D\theta_i} = \begin{bmatrix} \cos(\theta_i)^2 & \cos(\theta_i)\sin(\theta_i) \\ \cos(\theta_i)\sin(\theta_i) & \sin(\theta_i)^2 \end{bmatrix},$ $\theta_i = \{10°, 15°, 20°, 25°, 30°, 35°, 40°, 45°, 50°, 60°\}$

get constituting more realistic scatterers, the scattering matrix of each 24 scatterers are randomly selected within the aforementioned six scattering mechanisms. Among these six scattering mechanism types, the surface (S), dihedral (Di), and oriented-dihedral (ODi) type scattering generations are prevalent which occur due to boat-deck, boat-dihedrals, or boat-sea-dihedral. The part of antenna, tower, and guardrails of boat are made up of dipole like structures and hence can contribute in H, V, or oriented dipole scattering mechanisms. The four dipoles in combination can generate helix scattering mechanism too [14, 15]. Also, the complex structures on the boat can contribute in multiple-bounces generating multiple-interaction or volume scattering mechanism [14]. The complete information on the parameters, associated with the implemented boat simulator are summarized in Table 3.2. The target moves along a straight line with velocity $v = 20m/s$ and direction specified by the roll, pitch, yaw angles. In this simulation the scatterers are ideal and isotropic scatterers. It is also assumed that the scattering mechanisms do not change among the three antennas. In other words we are assuming that the scatterer do not change with the antenna viewing angles. The antennas are wuite close each others, the baselines are set equal to $1m$ and the target distance is $3km$, meaning that the aspect angle difference is roughly $\alpha \simeq 0.019°$.

Table 3.2: Simulation parameters associated with the implemented boat simulator.

Parameters	**Value**
Polarization	HH, HV, VH, VV
Career Frequency	10 GHz
Frequency Range	9.85 GHz - 10.15 GHz
Frequency Step	1.17 MHz
Frequency Bandwidth	300 MHz
No. of Transmitted Frequency	256
No. of Sweeps	128
Target Velocity	20 m/s
[Roll, Pitch, Yaw]	$[10°, 20°, 60°]$
[Horizontal & Vertical Baseline]	$[1m\ , 1m\ ]$

3.1.2 Radar geometry and dataset description of real Tank-72 data

The target T72 is a Soviet-designed main battle tank of dimension 9.54 (length)×3.59 (width)×2.23 (height) [16]. The actual image of T72 is shown in Fig. 3.3. The real T72 ISAR data was collected by performing turn-table experiment similar to the one depicted in Fig. 3.4. The parameters θ_{az}, θ_{el}, and R_0, mentioned in Fig. 3.4 indicate nominal azimuth angle, elevation angle, and radar-target distance, respectively. The complete dataset is publicly available and can be accessed through Air Force Research Laboratory (AFRL) website [17]. The whole data is stored in 29 different folders

Figure 3.3: Image of the target (Tank-72) under observation.

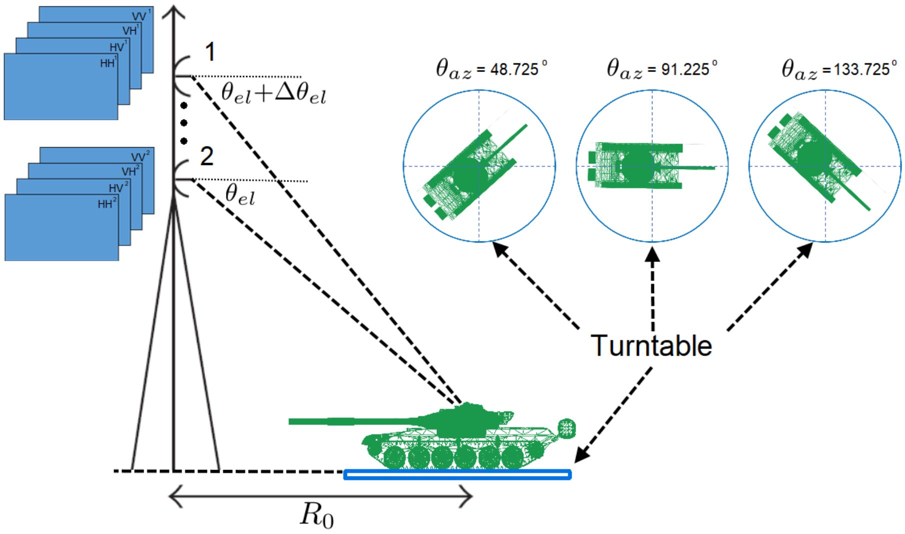


Figure 3.4: Radar target geometry of Tank-72 full-polarimetry ISAR data acquisition.

in such a way that each folder corresponds to a specific depression angle based records. In each folder, 85 different full-polarimetric information are stored for the complete azimuth variation of 3.9°, stepped in 85 times with the step-size of 0.05°, around the central azimuth angle of 89.231°. With the central frequency of 9.6 GHz, the frequency stepped with 3 MHz and covered the total bandwidth of 660 MHz. The range and cross-range resolution of this data is 1 foot × 1 foot. In this paper, 6 different elevation angle based T72 ISAR datasets are used that forms 5 elevation pairs generating five interferograms for a fixed azimuth angle. Further, three different azimuth angles of "same side" are being selected that forms combinedly 15 interferogram pairs. Even though, the T72 ISAR dataset is publicly available for complete 360°azimuth rotation, we implemented the "same side" based azimuth angles only (as depicted in Fig.3.4) by considering the fact that, while detecting non-cooperative targets, either the target will be going away or coming closer, without making complete rotation. The further description of the radar parameters associated with this T72 data is summarized in Table 3.3.

3.2 Performance analysis by using boat simulator

As the scatterers extracted from CLEAN method are being used for the further processing through Coherence-based and Span-based methods to form 3D image, it is worth evaluating first the performance of the proposed Modified Pol-CLEAN. In the next subsection, the polarimetric features evaluation based analysis of the Modified Pol-CLEAN is carried out. Then, the performance of

Table 3.3: Radar parameters of the Publicly available Tank-72 GTRI datasets

Parameter	**Value**
Polarization	HH, HV, VH, VV
Central Frequency	9.6 GHz
Frequency Range	9.27 GHz - 9.93 GHz
Frequency Step	3 MHz
No. of Transmitted Frequency	221
No. of Sweeps	79
Total Azimuth Angle Coverage	0°- 360°
Azimuth Increment Rate	0.05°
Total Aspect Angle for Each File	3.9
Total Elevation Angle Coverage	29.3°- 30.7°
Elevation Incremental Rate	0.14°
Resolution (Range × Cross-Range)	1 foot × 1 foot
Effective Baseline Coverage	0.15m - 4.23m

either the coherence-based and span-based 3DInISAR algorithms are evaluated in the next subsections.

3.2.1 Polarimetric features evaluation based analysis of the Modified Pol-CLEAN

An assessment on the quality of extracted scatteres' polarimetric vectors is performed by comparing the estimated polarimetric vectors with the model (reference) polarimetric vectors. The metric used to compare them is the **cosine similarity**, i.e. the cosine of the angle between a couple of vectors:

$$S_C^{(j)}\left(\boldsymbol{p}_M^{(j)},\boldsymbol{p}_R^{(j)}\right)=\frac{\boldsymbol{p}_M^{(j)}\cdot\boldsymbol{p}_R^{(j)}}{\left\|\boldsymbol{p}_M^{(j)}\right\|\left\|\boldsymbol{p}_R^{(j)}\right\|}, \tag{3.1}$$

where $\boldsymbol{p}_M^{(j)}$ is the polarimetric vector of the j-*th* scatterer of the model and $\boldsymbol{p}_R^{(j)}$ is the polarimetric vector of the j-*th* scatterer of the reconstruction. Our case is slightly different w.r.t. the "classic" metric, because our vectors are complex-valued. The cosine similarity in this case is itself a complex number, so its magnitude will be the matching metric needed. We defined a set of fifteen polarimetric vectors (Table 3.1), and every polarimetric vector of this set has been assigned to a point of the model (there are more scatterers than vectors, so the same vector could be assigned to more than one scattering centre). Herein, the two different polarimetric CLEAN approaches: Polarimetric CLEAN and Modified Polarimetric CLEAN have been compared. The complete theoretical description on these methods, and the methodological differences between them are highlighted in Section 2.2. Both algorithms use Pauli's decomposition [18] (Eq. 3.2) to represent the polarimetric ISAR image, in order to highlight the physical properties of the scattering mechanisms induced by scatterers and to reduce the size of the images. It also helps to better show the polarimetric data using RGB representation.

$$\boldsymbol{k} = \frac{1}{\sqrt{2}}\left[S_{HH} + S_{VV}, S_{HH} - S_{VV}, 2S_{HV}\right]^{T} \tag{3.2}$$

In Fig. 3.5 there is the Pauli RGB ISAR image of the target. The white markers represent the positions found by the extraction procedure. The performance evaluation of the Pol-CLEAN has been carried out with and without-noise conditions of boat simulator.

Noise-free simulation

We ran the reconstruction algorithm ten times, and for every launch the set of polarimetric vectors have been assigned to the points of the model in random way. Every run produces two point clouds, one per method. Once that reconstructions are ready, an assignment algorithm is needed to associate every point of the reconstruction to a point of the model. Therefore we have a target model and twenty reconstructions, registered w.r.t. the model, and the cosine similarity can be calculated. This simulated data and this simple target has been used because makes it possible the point-to-point association between the point cloud and the reference point-like target model. In more populated and complex models, there is an higher probability that the algorithm can't find all the points that

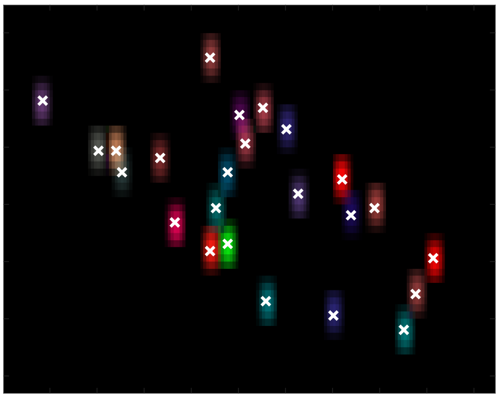


Figure 3.5: Pauli RGB ISAR image of the simulated target

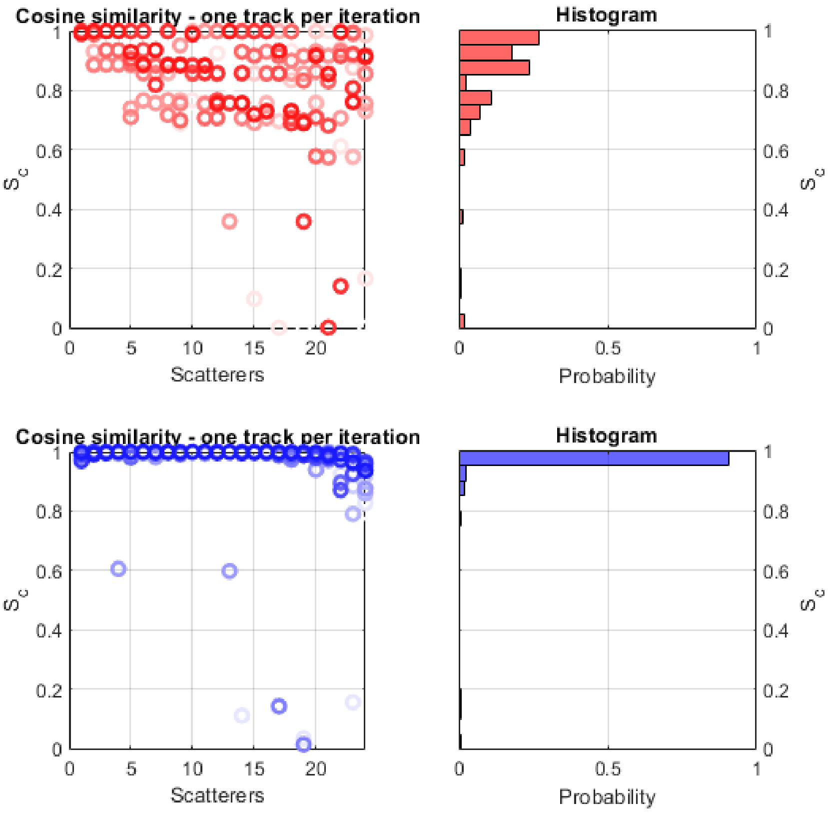


Figure 3.6: Cosine similarity comparison between Pol-CLEAN (in red) and Modified Pol-CLEAN (in blue).

compose the target. In Figure 3.6 there is the comparison between the two methods. The first row refers to the Pol-CLEAN, while the second one is the result of the Modified Pol-CLEAN. The dotted plots represent the similarity values obtained for each extracted scatterer. For every iteration (i.e., simulation, extraction of all the scattering centers in the image) a number of points equal to the number of extracted scatterers is plotted, and the color fades with the number of iterations. Already here it is visible that the similarity values obtained on the Modified Pol-CLEAN are ”squeezed” near one, the maximum. Only few points fall at lower values. This behavior is also visible in the histogram representation. All the values obtained among all iterations are serialized into a row vector and the histogram is evaluated. Again, the histogram of the Modified Pol-CLEAN is more skewed to high similarity values compared to the Pol-CLEAN’s one, and very few low value’s bins contain something. It must be also pointed that some simulations carried out using the Pol-CLEAN didn’t find all the points of the model, thus in the overall computation one vector was a little bit longer than the other.

Noisy simulation

The results showed until now are obtained neglecting the effect of noise, but in a real-world scenario's application noise must be considered. The metrics chosen to compare the two polarimetric extraction procedures are:

- the mean cosine similarity

$$\bar{S}_C = \frac{1}{N_s} \sum_{j=1}^{N_s} S_C^{(j)} \left(\boldsymbol{p}_M^{(j)}, \boldsymbol{p}_R^{(j)} \right), \tag{3.3}$$

  where $S_C^{(j)} \left(\boldsymbol{p}_M^{(j)}, \boldsymbol{p}_R^{(j)} \right)$ is defined in the Eq. 3.1, and

- the mean error on scatterer internal degree of freedom α estimation with

$$\alpha_{err}^{(j)} = \left| \hat{\alpha}_R^{(j)} - \alpha_M^{(j)} \right|, \tag{3.4}$$

  where $\hat{\alpha}_R^{(j)}$ is the estimated scatterer internal degree of freedom of the reconstruction's j-*th* scatterer and $\alpha_M^{(j)}$ is the scatterer internal degree of freedom of the model's j-*th* scatterer.

In Eq. 3.3 and Eq. 3.4, N_s is the number of scatterers in the reconstruction. These metrics are calculated after the scatterers association procedure, during which every scatterer from the reconstruction is associated to a scatterer from the model. The scatterer internal degree of freedom α defines the observed scattering mechanism:

$$\boldsymbol{k} = \left[\cos(\alpha) e^{j\delta}, \sin(\alpha)\cos(\beta) e^{j\gamma}, \sin(\alpha)\sin(\beta) e^{j\phi} \right]^T, \tag{3.5}$$

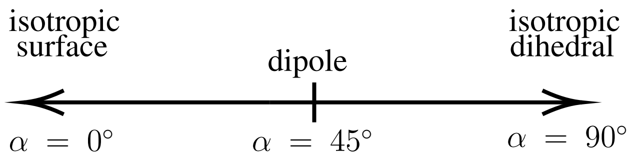


Figure 3.7: Scatterer internal degree of freedom interpretation

where $\boldsymbol{k}$ is the unitary polarimetric vector, β is the orientation of the scatterer w.r.t the LoS and δ, γ and ϕ are the phases in each Pauli's channel. The simulation is carried out by varying the SNR of the polarimetric ISAR image, thus a certain noise power is fixed and the additive white Gaussian noise is added directly in the image domain. We decided to adjust the SNR in the image domain because the decorrelation effect due to thermal noise has an impact directly on the noise of estimated interferometric phase [19]. The Cramer-Rao bound (Figure 3.8) on the phase variance σ_θ^2 is given by:

$$\sigma_\theta^2 = \frac{1}{2N_L}\frac{1-|\gamma|^2}{|\gamma|^2} \tag{3.6}$$

where N_L is the number of independent looks and γ is the complex correlation coefficient. The modulus of γ is the interferometric coherence and it depends on the signal-to-noise ratio:

$$|\gamma| \propto \frac{SNR}{SNR+1}. \tag{3.7}$$

The first analyzed metric, as said in the previous section, is the cosine similarity between the polarimetric vectors of the model and the estimated polarimetric vectors of the reconstruction. The obtained results are plotted in Figure 3.10(a). It is possible to observe that the similarity values obtained with the Pol-CLEAN are already very good, the average similarity is about 0.9 for every value of SNR. With the Modified Pol-CLEAN however there is an improvement, because the average similarity is pratically 1 (i.e., the maximum value).

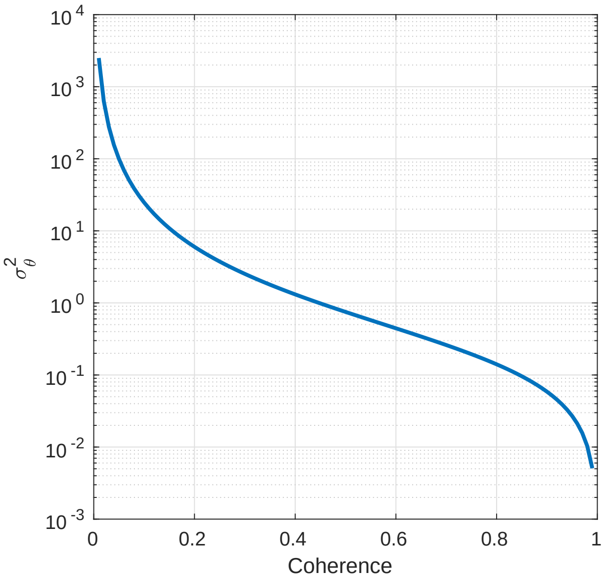
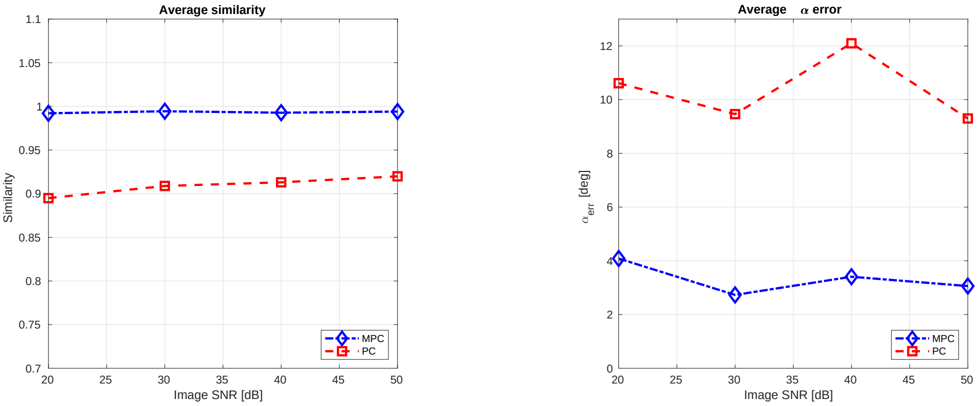


Figure 3.8: Cramer-Rao bound on σ_θ^2 estimation for $N_L = 2$



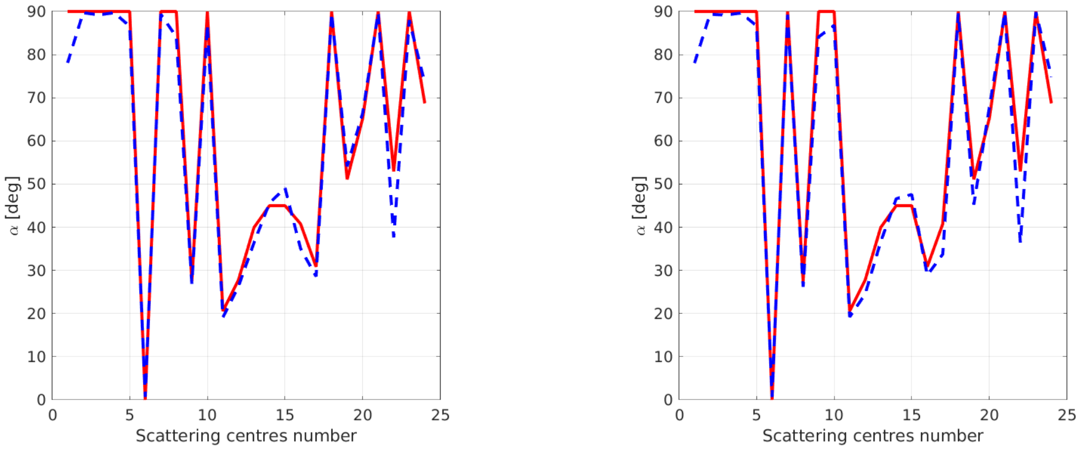
(a) (b)

Figure 3.9: Blue curve refers to Modified Pol-CLEAN (MPC), while red curve is the Pol-CLEAN (PC): (a) Comparison between average cosine similarity of the estimated polarimetric vectors w.r.t. the model polarimetric vectors (b) Comparison between the average estimation error on α

The second analyzed metric is the scatterer internal degree of freedom α. The obtained results are plotted in Figure 3.10(b). Here we can observe that there is an improvement in the capability of the system to properly identify the scattering mechanism associated with each scattering centre of the reconstruction, due to the lower error obtained with the Modified Pol-CLEAN. In Figure **??** there is the detailed result of α estimation for every scatterer of the target with the Modified Pol-CLEAN at the highest SNR used for the simulation.

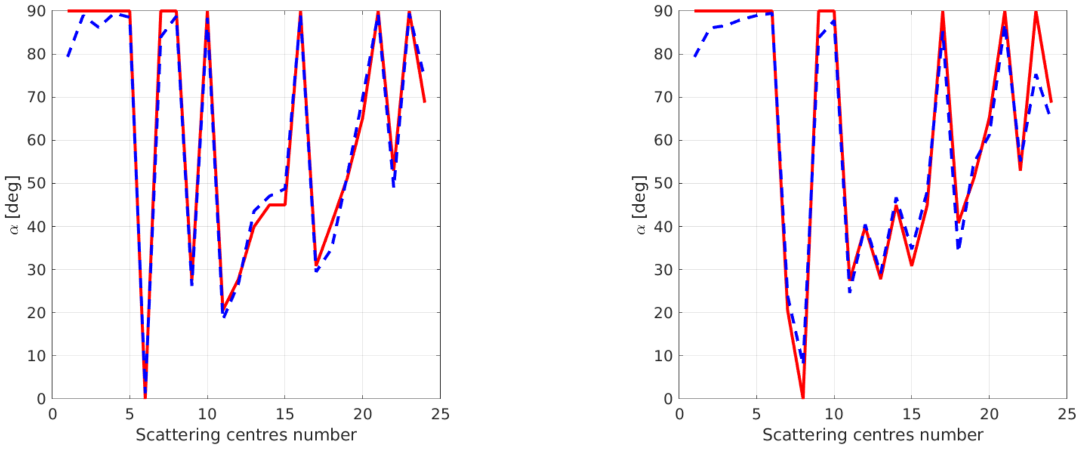
Table 3.4: Comparison of Single-pol InISAR and Full-pol InISAR coherence results for boat simulator data.

	Single-Pol Coherence (HH Polarization)	**Full-Pol Coherence (Coherence-based)**	**Full-Pol Coherence (Span-based)**	**Similarity Index**
SNR $\rightarrow \infty$	0.9265	0.9393	0.9352	0.9988
SNR = 40dB	0.9261	0.9367	0.9324	0.9979
SNR = 30dB	0.8963	0.9329	0.9284	0.9956
SNR = 20dB	0.8479	0.9139	0.9183	0.9786



(a) $SNR = 50$ dB

(b) $SNR = 40$ dB



(c) $SNR = 30$ dB

(d) $SNR = 20$ dB

Figure 3.10: Values of α of the model's scattering centres - red curve - and values of estimated α of the reconstruction's scattering centres - dashed blue curve -, obtained with Modified Polarimetric CLEAN. The red curves are different in these four plots because by adding the noise the scattering centres extraction order may change.

3.2.2 Performance evaluation based on coherence estimation and 3D imaging capability

The coherence values obtained by using boat simulator data, for the case of single-pol InISAR and full-pol InISAR are generated and added in Table 3.4 for comparison. The coherence results are generated for with- and without-noise scenarios to make the performance analysis more robust. In with-noise condition, the white Gaussian noise (N) is added directly to the data in the ISAR image domain. If assuming the signal level S at the ISAR image domain prior to the noise addition, the added noise (N) maintains signal to noise ($\frac{S}{N}$) ratio to 40 dB, 30 dB, and 20 dB, for three different conditions of noise addition. By comparing the results of Table 3.4, one can clearly see the improvement in the mean coherence value using full-pol data in comparison with the single-pol. In without-noise or ideal signal receiving condition, the improvement is not extensive while for the real or noisy scenarios, as for SNR 40 dB, SNR 30 dB, and SNR 20 dB, added in the 3.4, a significant improvement in the coherence level by using full-polarimetry data is observed comparing with single-polarimetry. The performance of full-pol Coherence-based and Span-based methods are quite similar here and both generates coherence level much better than single-pol, specially in noisy conditions.

The 3D results generated for without- and with-noise scenarios are shown in Fig. 3.11 - Fig. 3.22. The images shown in Fig. 3.11(a)-(d), Fig. 3.14(a)-(d), Fig. 3.17(a)-(d), and Fig. 3.20(a)-(d), indicate the single-pol 3D InISAR based results superimposing with the original reference model points, whereas, Fig. 3.12(a)-(d), Fig. 3.15(a)-(d), Fig. 3.18(a)-(d), and Fig. 3.21(a)-(d) indicate the full-pol coherence-method based results, and Fig. 3.13(a)-(d), Fig. 3.16(a)-(d), Fig. 3.19(a)-(d), and Fig. 3.22(a)-(d) indicate full-pol Span-method based results, for the case of no-noise, SNR 40dB, SNR 30dB, and SNR 20dB, respectively. In all of these images, the reconstructed 3D points are indicated by the blue filled circles, and the actual reference simulator points are indicated by the red hollow circles. By comparing these images one can conclude that the obtained reconstructed 3D points of full-pol InISAR results are closer to the actual reference simulator points in comparison with the obtained reconstructed 3D points of single-pol InISAR results. Hence, through the visual interpretation, one can conclude that the proposed approach of full-pol 3D InISAR performs better than the single-pol 3D InISAR. However, for the quantitative performance evaluation, we estimated two parameters: mean distance error (MDE) and Root mean Square Error (RMSE). MDE results in overall displacement of reconstructed points in comparison with the reference simulator. The formula used to calculate MDE is as follows.

$$\text{MDE} = \sum_{t=1}^{m} \frac{\sqrt{(R_x^t - M_x^t)^2 + (R_y^t - M_y^t)^2 + (R_z^t - M_z^t)^2}}{m}, \tag{3.8}$$

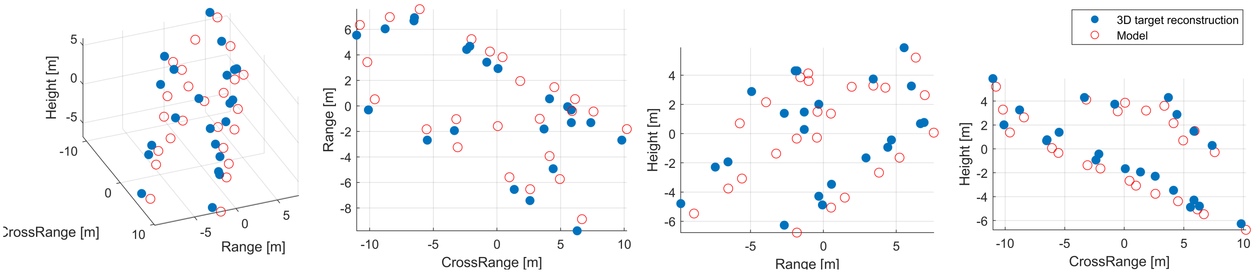
where (R_x^t, R_y^t, R_z^t) and (M_x^t, M_y^t, M_z^t) are the t^{th} point's (x, y, z) coordinate of reconstructed results and reference model, respectively. Another standard error evaluation parameter, i.e. RMSE is calculated through (4.10) to further quantify the mismatch between the reconstructed and original (reference) 3D point cloud.

$$\mathrm{RMSE} = \sqrt{\mathrm{mean}(||\mathbf{R} - \mathbf{M}||_F^2)}, \tag{3.9}$$

where $||.||_F$ is Frobenius norm and, **R** and **M** are the two matrices of dimension $3 \times t$, and each row of **R** and **M** indicate the t^{th} point's (x, y, z) coordinate of reconstructed results and reference model, respectively. The MDE and RMSE calculated for the condition of full-pol 3D InISAR (for both coherence and Span-based methods) and single-pol 3D InISAR, are directly added in the figures to have a clear visibility in comparing the results. On can see by comparing the RMSE and MDE results, that the both proposed full-pol InISAR methods outperforms the single-pol InISAR. Further, as the Span-based method established based on the assumption that all the polarimetric acquisitions are similar in nature, we tried to validate this situation by calculating Young's similarity factor [20], through the following formula.

$$r([k_1],[k_2]) = \frac{|k_1^{\dagger} k_2|^2}{||k_1||_2^2 ||k_2||_2^2} \tag{3.10}$$

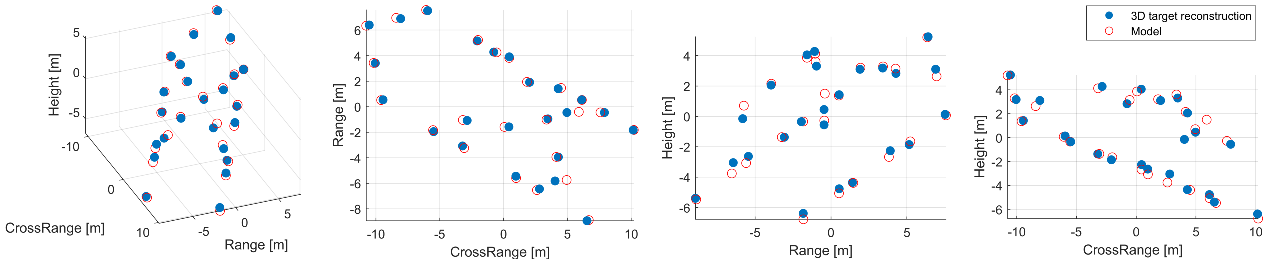
where k_1 and k_2 are the polarimetric scattering vectors of two antennas, and $||.||_2^2$ indicates the square sum of the absolute values of the components of the vector. The mean similarity value added in Table 3.4 reveals that the polarimetric information, captured by the three antennas (H, V, and C) are almost similar as $r([k_1],[k_2]) \approxeq 1$. It is an ideal condition for Span-based method and consequently the performance of this method is optimum here.



(a) (b) (c) (d)

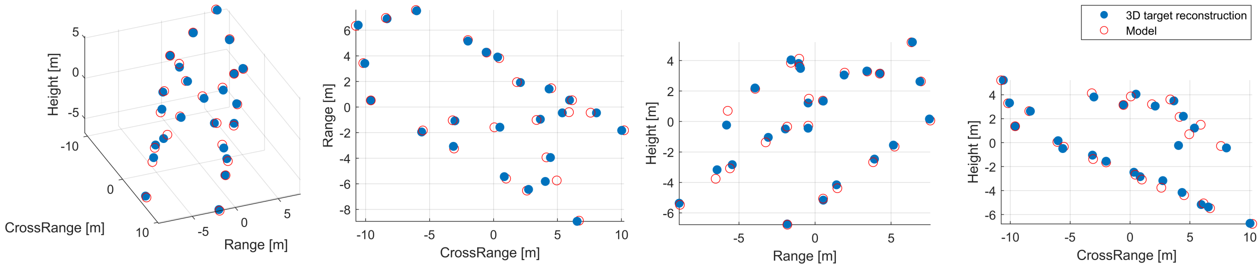
Figure 3.11: Single-pol InISAR 3D results of boat simulator for without-noise condition (a) 3D view(b) XY-plane view (c) YZ-plane view (d) XZ-plane view. The error obtained for this condition are MDE = 1.9613 ; RMSE = 1.5690

Both approaches performs similarly. Both MDE and RMSE assume similar values. Differences in their values may be due to imprecise coregistration between the reference target model and the target reconstruction. However, an exception in the results can be seen for the 20dB condition as the performance of Span-based seems to perform a little bit better than the Coherence-based one. The



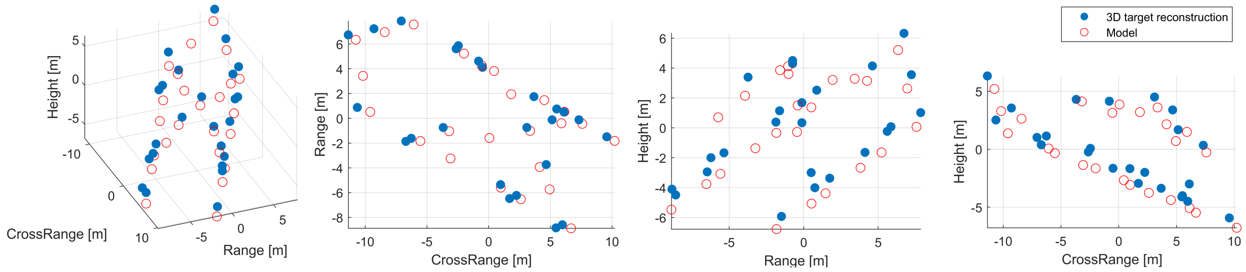
(a) (b) (c) (d)

Figure 3.12: Coherence optimization method based Full-pol InISAR 3D results of boat simulator for without-noise condition (a) 3D view (b) XY-plane view (c) YZ-plane view (d) XZ-plane view. The error obtained for this condition are MDE = 0.4004 ; RMSE = 0.2946



(a) (b) (c) (d)

Figure 3.13: Span method based Full-pol InISAR 3D results of boat simulator for without-noise condition (a) 3D view (b) XY-plane view (c) YZ-plane view (d) XZ-plane view. The error obtained for this condition are MDE = 0.3155 ; RMSE = 0.2343



(a) (b) (c) (d)

Figure 3.14: Single-pol InISAR 3D results of boat simulator for with-noise (SNR = 40dB) condition (a) 3D view (b) XY-plane view (c) YZ-plane view (d) XZ-plane view. The error obtained for this condition are MDE = 2.9815; RMSE = 2.7236

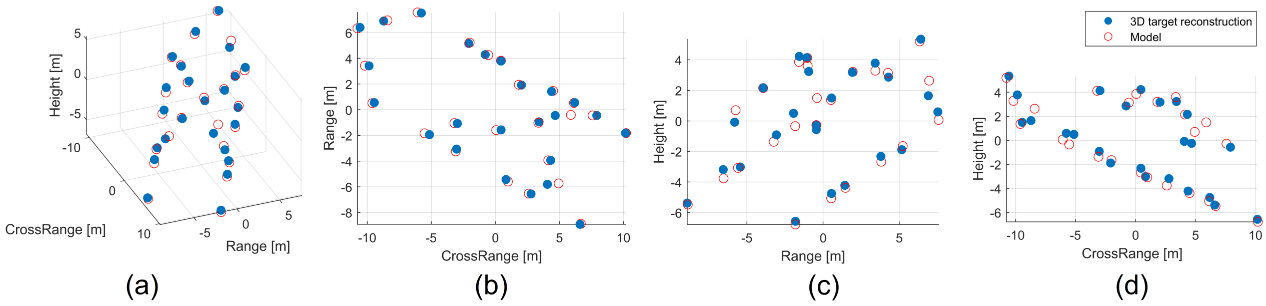


Figure 3.15: Coherence optimization method based Full-pol InISAR 3D results of boat simulator for with-noise (SNR 40dB) condition (a) 3D view (b) XY-plane view (c) YZ-plane view (d) XZ-plane view. The error obtained for this condition are MDE = 0.5027; RMSE = 0.3840

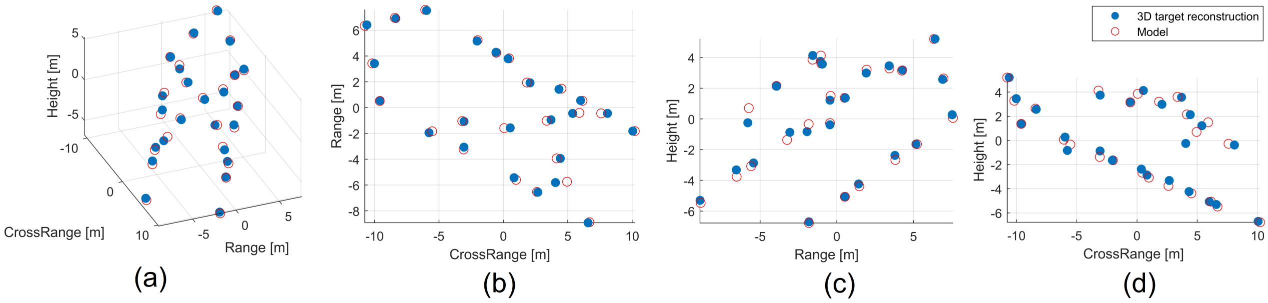


Figure 3.16: Span method based Full-pol InISAR 3D results of boat simulator for with-noise (SNR 40dB) condition (a) 3D view (b) XY-plane view (c) YZ-plane view (d) XZ-plane view. The error obtained for this condition are MDE = 0.3394 ; RMSE = 0.2466

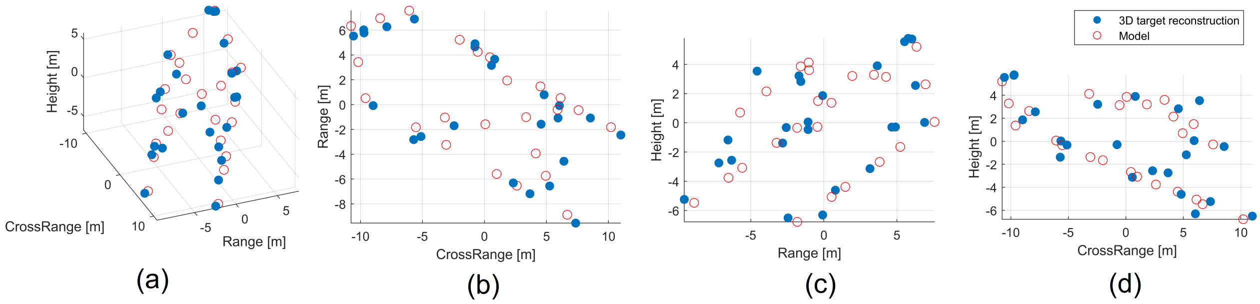
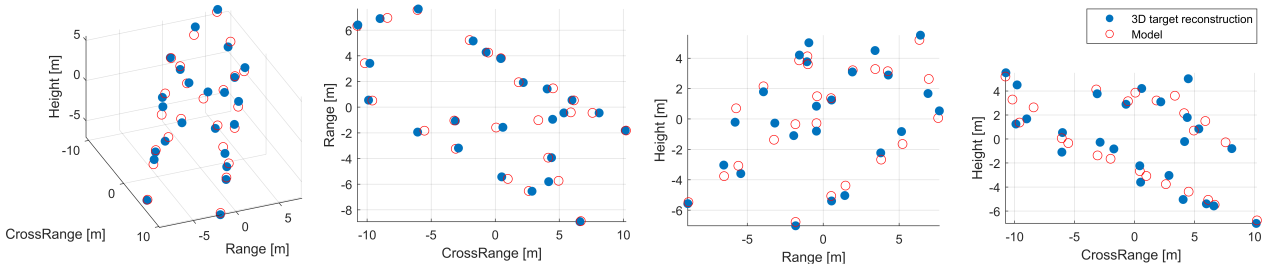
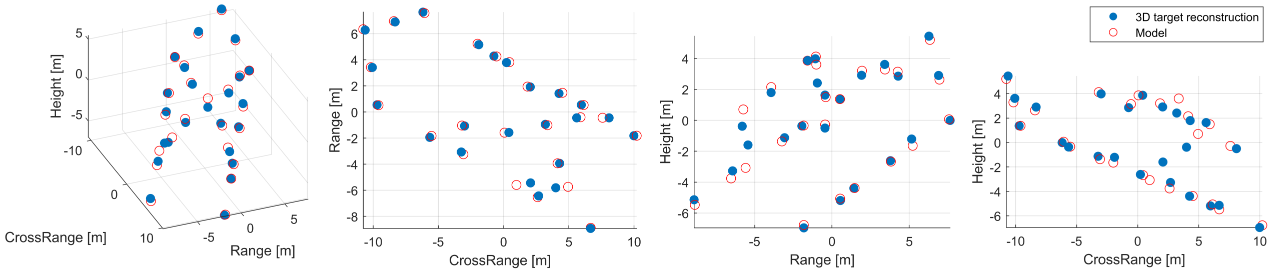


Figure 3.17: Single-pol InISAR 3D results of boat simulator for with-noise (SNR = 30dB) condition (a) 3D view (b) XY-plane view (c) YZ-plane view (d) XZ-plane view. The error obtained for this condition are MDE = 2.1962; RMSE = 1.9076



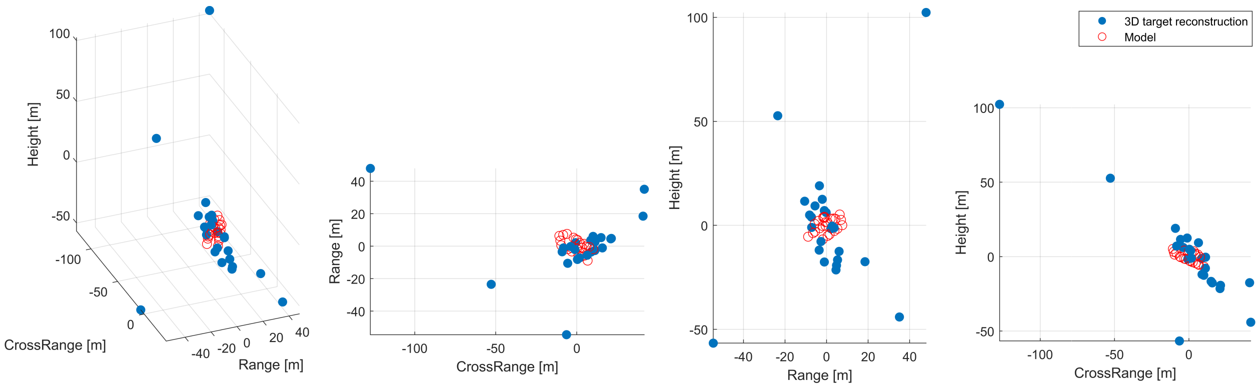
(a) (b) (c) (d)

Figure 3.18: Coherence optimization method based Full-pol InISAR 3D results of boat simulator for with-noise (SNR 30dB) condition (a) 3D view (b) XY-plane view (c) YZ-plane view (d) XZ-plane view. The error obtained for this condition are MDE = 0.6930; RMSE = 0.4607)



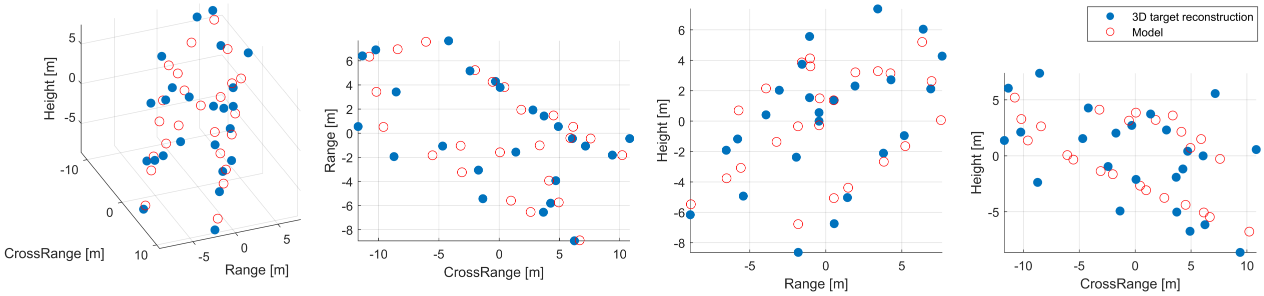
(a) (b) (c) (d)

Figure 3.19: Span method based Full-pol InISAR 3D results of boat simulator for with-noise (SNR 30dB) condition (a) 3D view (b) XY-plane view (c) YZ-plane view (d) XZ-plane view. The error obtained for this condition are MDE = 0.4274; RMSE = 0.3065



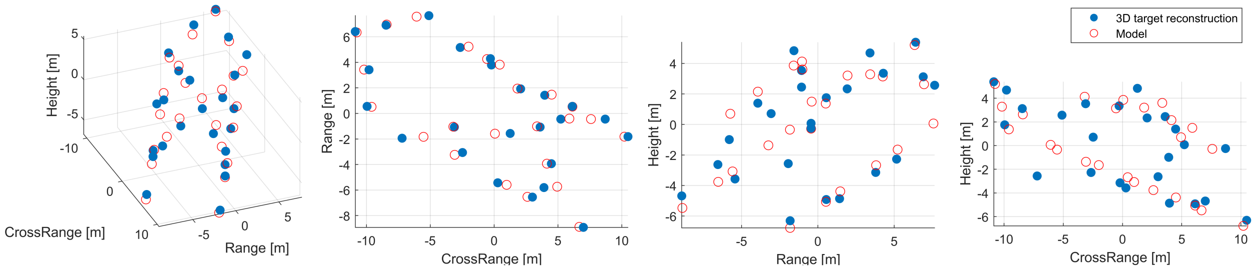
(a) (b) (c) (d)

Figure 3.20: Single-pol InISAR 3D results of boat simulator for with-noise (SNR = 20dB) condition (a) 3D view (b) XY-plane view (c) YZ-plane view (d) XZ-plane view. The error obtained for this condition are MDE = 26.3630 ; RMSE = 25.5849



(a) (b) (c) (d)

Figure 3.21: Coherence optimization method based Full-pol InISAR 3D results of boat simulator for with-noise (SNR 20dB) condition (a) 3D view (b) XY-plane view (c) YZ-plane view (d) XZ-plane view. The error obtained for this condition are MDE = 2.3660 ; RMSE = 1.6048



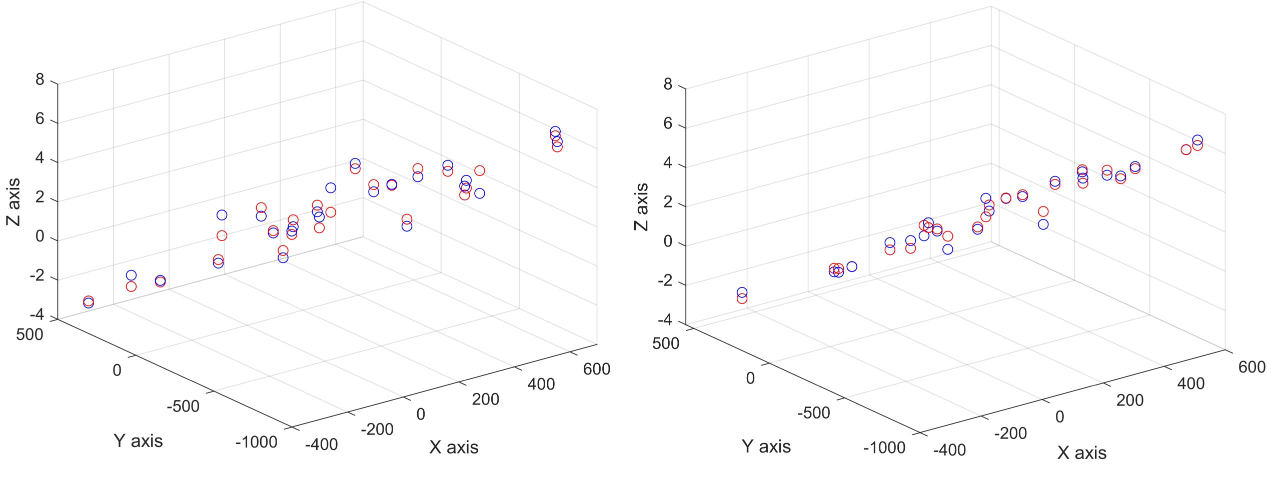
(a) (b) (c) (d)

Figure 3.22: Span method based Full-pol InISAR 3D results of boat simulator for with-noise (SNR 20dB) condition (a) 3D view (b) XY-plane view (c) YZ-plane view (d) XZ-plane view. The error obtained for this condition are MDE = 1.1026 ; RMSE = 0.7550

better understand the cause of this difference in the results we have analyzed more deeply how the estimator in Eq. (2.3) performs. It is important to note that Eq. (2.3) represents the equation of a 2D plane in a 3D space, namely

$$Z = X \cdot \Omega cos(\phi) + Y \cdot \Omega sin\phi \tag{3.11}$$

where $Z = \nu_C$, $X = \frac{R_0 \Delta\theta_H}{2\pi d_H}$ and $Y = \frac{R_0 \Delta\theta_V}{2\pi d_V}$ and both Ω and ϕ need to be estimated. According to this equation, the measured values of Z, X, Y of each scatterer identify a 2D plane. Then the algorithm estimates the unknown coefficients of this plane, namely $\Omega cos(\phi)$ and $\Omega sin(\phi)$. Figure HHH shows the scatterers in the X, Y, Z plane. More specifically the blue dots are the measured values of X, Y, Z extracted from the ISAR image and the interferograms, while the red dots represent the 2D plane estimated by the algorithms, namely $\hat{Z} = X \cdot \hat{\Omega} cos(\hat{\phi}) + Y \cdot \hat{\Omega} sin(\hat{\phi})$. The right-end image refers to the coherence-based approach, while the left-end image refers to the span-based approach. As it possible to note, the blue points well fit a 2D plane in both methods. However the estimated 2D plane slightly differs among the two and seems to better fits the blues dots for the SPAN based algorithm. This depends may depends on the interferometric phase accuracy. This, in turn affects the estimate of the scatterer height according to Eq. (2.2). Apart from this visual interpretation, quantitative assessment has also been performed by estimating the mean distance departure of the reconstructed points with the original model plane. For the Coherence- and Span-based methods, the mean distance values are 0.4106 and 0.3540, respectively.



(a) Coherence-based (SNR 20dB)

(b) Span-based (SNR 20dB)

Figure 3.23: Coherence-based and Span-based methods comparison on the accuracy in image plane reconstruction. Red and blue circles indicates model plane points and reconstructed points, respectively.

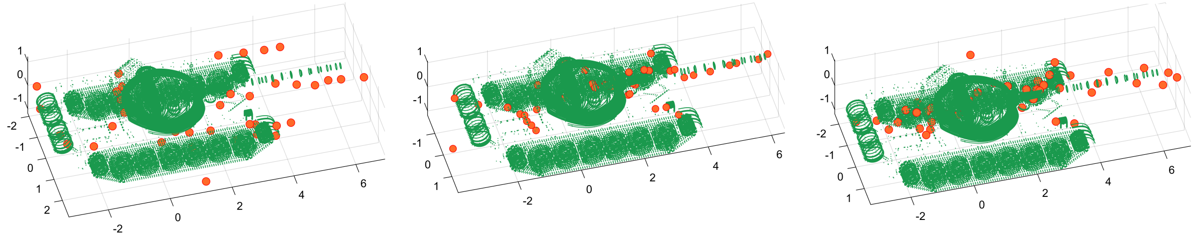
3.3 Performance analysis by using real Tank-72 GTRI data

The coherence values obtained by using real Tank 72 GTRI dataset, for the case of single-pol and full-pol (Coherence- and Span-based) methods, are added in Table 3.5. By looking in this table, one can observe that the coherence values obtained by using full-pol Coherence-based method are the optimum one and it shows a significant improvement over the conventional single-pol InISAR method. However, the Span-based full-pol method doesn't perform well in this real Tank 72 data condition. The reason behind the substandard performance of Span-based method is explained through the polarimetric similarity evaluated between the polarimetric channels forming interferogram.The polarimetric similarity results obtained between two acquisitions in Tank 72 are obtained through Eq. (3.10) and added in Table 3.5.

Table 3.5: Comparison of single-pol InISAR with Full-pol Coherence- and Span-based Pol-InISAR methods by implementing real Tank-72 dataset.

Azimuth angle	Elevation angles pairs	Single-Pol InISAR	Full-Pol InISAR Coherence Based	Full-Pol InISAR Span Based	Similarity Index
48.725°	29.5754°, 29.7162°	0.9132	0.9454	0.8729	0.6054
	29.7162°, 29.8603°	0.9115	0.9484	0.8693	0.7365
	29.8603°, 29.9994°	0.9339	0.9752	0.9023	0.4882
	29.9994°, 30.1409°	0.9233	0.9455	0.8693	0.8019
	30.1409°, 30.2801°	0.9196	0.9627	0.8984	0.5965
91.225°	29.5754°, 29.7162°	0.9316	0.9755	0.9006	0.7033
	29.7162°, 29.8603°	0.9265	0.9752	0.8645	0.5206
	29.8603°, 29.9994°	0.9228	0.9731	0.8779	0.6038
	29.9994°, 30.1409°	0.9166	0.973	0.8767	0.7733
	30.1409°, 30.2801°	0.9234	0.9742	0.8503	0.5489
133.725°	29.5754°, 29.7162°	0.9224	0.9781	0.8687	0.7305
	29.7162°, 29.8603°	0.9174	0.9677	0.8717	0.3896
	29.8603°, 29.9994°	0.9206	0.9317	0.8694	0.7734
	29.9994°, 30.1409°	0.9038	0.9441	0.8857	0.6220
	30.1409°, 30.2801°	0.9179	0.9621	0.9102	0.5068

From the results, it can be seen that the obtained similarity index values are relatively lower that indicates that the polarimetric informations obtained between two elevation pairs are not very similar. This violets the assumption taken in establishing Span-based Pol-InISAR method. Due to



(a) SIngle-pol (HH)

(b) Full-pol Coherence-based

(c) Full-pol Span-based

Figure 3.24: Single-pol, full-pol Coherence-based, and full-pol Span-based methods comparison based on 3D image formation.

this reason only, the coherence performance of Span-based method is not good, as can be seen from the Table 3.5. With this low level of coherence- and similarity-values, it is likely to get incorrect 3D imaging results from Span-based method. We tried to verify this by generating 3D point cloud for one combination of elevation pair [29.5754°, 29.7162°] at azimuth angle 91.225°. We selected this combination because single-pol, full-pol coherence- and full-pol Span-based obtain coherence values higher than 0.9 (as can be seen from Table3.5). The corresponding results for single-pol, full-pol Coherence-based, and full-pol Span-based methods are added in Fig 3.24(a), 3.24(b), and 3.24(c), respectively. One can observe that the 3D result obtained using Span-based method is not covering well the CAD model, while full-pol Coherence based method performs best among all three and overlaps well with the CAD model. Additionally, these results looks underpopulated and requires more scatterer points to clearly recognize the target. Therefore, the 3D reconstructed image with more populated points is further generated by fusing the 15 different elevation-pair and azimuth combinations based 3D results in one. We tried to obtain the multi-aspect full-pol 3D InISAR results using Coherence-based method, which is further compared with the single-pol InISAR 3D image.

3.3.1 Multi-aspect full-pol In-ISAR based 3D imaging

The fully-polarimetric data make it possible to have response in co- as well as cross-pol channel and through that the Pauli RGB image can be obtained. For, 48.725°, 91.225°, and 133.725° azimuth conditions (at $\theta_{el} = 29.5754$), the false RGB images are shown in Fig. 3.25(a), (b), and (c), respectively, where red, blue, and green colors indicate the polarimetric channels $\frac{1}{\sqrt{2}}|HH - VV|$, $\frac{1}{\sqrt{2}}|HH + VV|$, and $\sqrt{2}|HV|$, respectively. One can observe the existence of all the three colors in the images, that proves the presence of different class of scatterers in the target generating different scattering mechanisms. And to cover-up all various kinds of scatterers, it is important to utilize the complete polarimetric information inherent in full-pol ISAR data, instead of single-pol ISAR data.

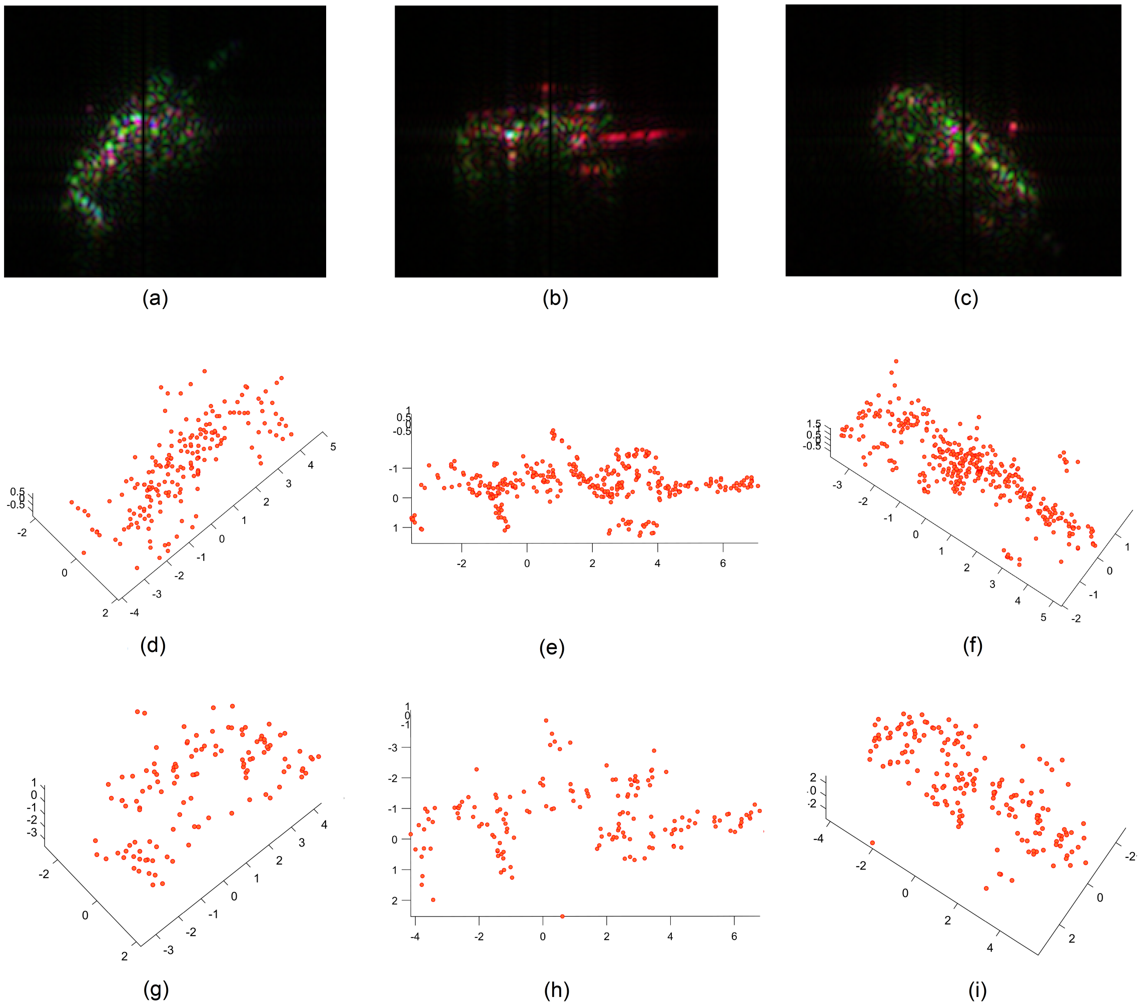
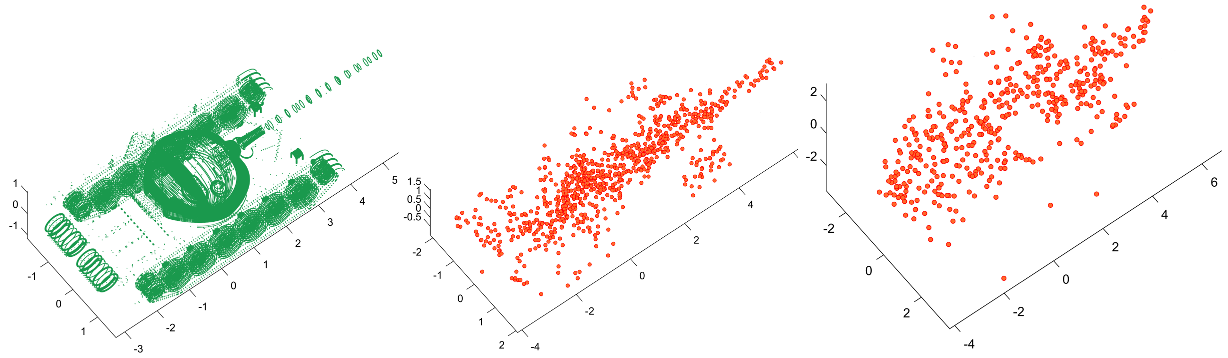


Figure 3.25: The results obtained using real Tank-72 data, where (a), (b), & (c) indicate Pauli decomposition false RGB images for azimuth conditions 48.725°, 91.225°, and 133.725°, respectively. (d), (e), & (f) indicate the full-pol InISAR based 3D imaging results and (g), (h), & (i) indicate the single-pol InISAR based 3D imaging results of Tank-72 ISAR data, obtained for the three azimuth angle combinations of 48.725°, 91.225°, and 133.725°, respectively.

Further, different parts of the Tank can be apparently seen in different azimuth angle conditions of 48.725°, 91.225°, and 133.725°, respectively. Therefore, to better detect the Tank in its full shape, it is important to combine all the azimuth aspects together.

The 3D reconstructed point clouds corresponding to the Tank-72 target, are shown in Figs. 3.25(d)-(f) and 3.25(g)-(i), generated by using full-pol and single-pol InISAR based conditions, respectively, and further, the combined results are shown in Fig. 3.26(b) and (c) for full-pol and

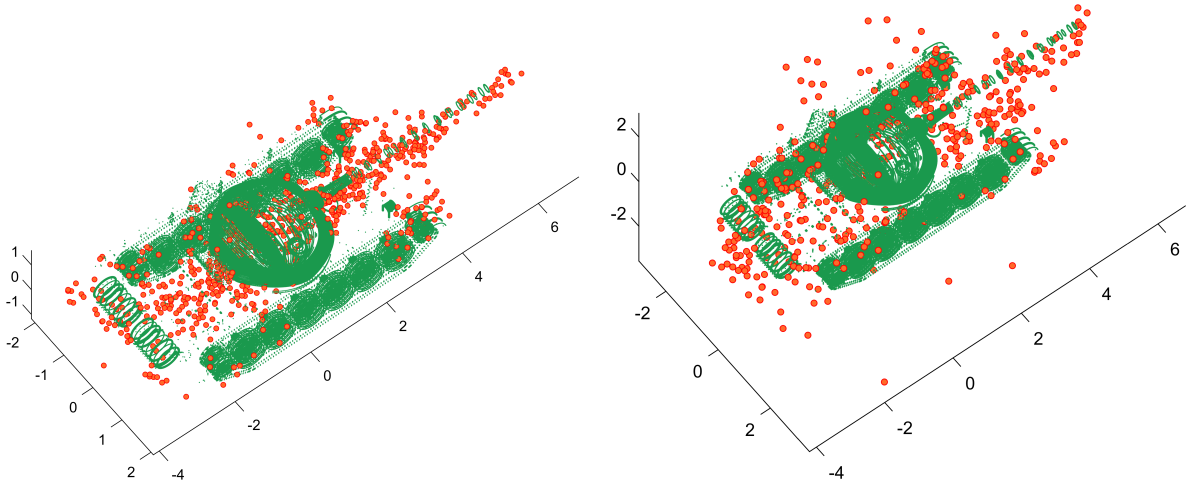


(a) (b) (c)

Figure 3.26: (a) Tank-72 CAD Model (down-sampled point-cloud) (b) Full-pol InISAR 3D result (c) Single-pol InISAR 3D result.

single-pol conditions, respectively. The CAD model of Tank-72 is shown in Fig. 3.26(a) for reference. By comparing these images, one can observe that the full-pol result is better reconstructing the target in comparison with the single-pol.

The 3D point cloud, generated by using full-pol and single-pol InISAR methods, as shown in Fig. 3.26(b) and (c), respectively, are superimposed on the Tank-72 CAD model, and the respective superimposed results are shown in Fig. 3.27(a) and (b). Better alignment of point cloud with the CAD model can be seen for the full-pol InISAR based results in comparison with single-pol.



(a)

(b)

Figure 3.27: Superimposition of the results obtained using (a) Full-pol 3D InISAR (b) Single-pol 3D InISAR, over target (Tank-72) CAD model.

Chapter 4

ATR Algorithms

This chapter presents the two ATR algorithm we have designed. Their performance will be deeply analysed in the next project year as declared in the project technical proposal. The two algorithms are pretty different in nature. The first is a sort of ”template matching” algorithm and directly compare the reconstructed 3D target image with 3D target template. This algorithm requires an alignment step, which can affect the recognition performance. However it does not need to be trained, but it needs a-priori information about the target, as for example the target geometrical model. The second approach falls into the Machine Learning category and it basically uses Deep Learning (DL) to recognize the target. This algorithm has been proposed for 3D cloud of points and originally designed for LIDAR sensors. It however needs to be trained and therefore needs a training set.

4.1 Template-matching algorithm

In [21] a first attempt to use 3D InISAR results to recognise targets has been proposed. Other ideas may come from the phased array radar systems typically used for example for concealed weapon detection and recognition. However, this part of the literature typically refers to a SAR geometry, which is completely known a priori and which uses higher frequencies, therefore, producing quite different radar images from those that will be obtained when using the proposed 3D InISAR approach. In fact, as shown in Section 2.1 the output of the 3D InISAR algorithm is a cloud of point in the 3D space and it is improperly defined as "image". This makes the 3D InISAR output quite different from standard radar images and it requires a change of perspective. Regardless of the sensor and the radar imaging algorithm used, the output of the proposed 3D InISAR algorithm appears more similar to the output of other sensors, e.g. LIDAR, typically used in computer vision. This is the main reason why we investigated the state of the art of LIDAR and computer vision. Authors in [21] do not propose a full-on ATR algorithm but this paper can be considered as a first attempt to build an ATR algorithm based on 3D InISAR. Following the results in [21], ATR can be accomplished by matching the 3D point-like reconstructions and the target 3D models. The 3D target models may be either a geometrical model, as for example a CAD model of the target, or, if available, a 3D e.m. points cloud generated by an e.m. simulator. Then, a 3-D point-like reconstruction of an unknown target has to be compared with a set of target models. A "match score" based approach can be used to recognize the target class as the one more "similar" to the 3D point-like target reconstruction [22]. The "match score" index should therefore be a measure of the similarity between the 3D point-like target reconstruction and the target models in the system database. Despite the definition of the target models, the alignment between the 3D point-like target reconstruction and the target models is the most important step before the classification. In fact, when used in real scenarios, the 3-D InISAR algorithm generates a 3-D cloud of points that may appear rotated (around any of the three principal axes) and translated with respect to the target model, due to both the target orientation with respect to the radar and its own motions during the observation time. Being the target non-cooperative, both the target orientation with respect to the radar and its own motions are not known a-priori. This may complicate the alignment phase. Hence, we need an automatic alignment algorithm that, according to an optimization criterion, aligns the cloud of points with the target model in an effective and satisfactory way. The alignment method proposed in [21] is summarized in subsection 4.1.1.

This section deals with the description of the 3D model-based classification algorithm. A model-based ATR algorithm falls into the template matching category as it take advantage of pre-designed models, often 3D CAD models, to be compared with the radar observations in order to produce a classification label that is based on some pre-defined metrics. In particular, the proposed method block scheme is illustrated in $Figure 14$ and includes four major steps that will be explained in the following:

- Alignment,
- Data Feature Extraction
- Performance Index Evaluation
- Decision Rule.

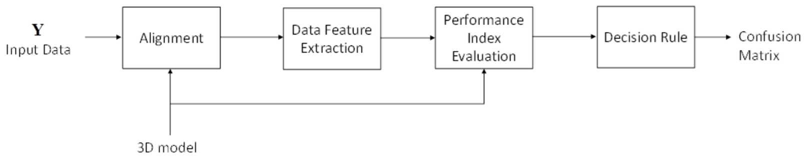


Figure 4.1: 3D model based classification algorithm block scheme

The input of the algorithm is given by a cloud of points in a 3D space, each one corresponding to a scatterer extracted from the received radar observation. In particular, the 3D space is defined by the direction of range ($\hat{i}_{y2}$), cross range ($\hat{i}_{y1}$) and the height orthogonal to the range- cross range plane ($\hat{i}_{y3}$). The received target extracted point coordinates are stored in a matrix used as input for the alignment step. As stated in [23], the alignment is the most important step in 3D classification since the classification is based on a measure of similarity between the received radar observation and the model. For this reason, the received observation must be aligned in order to produce accurate results. The alignment algorithm used here includes two steps:

- A coarse alignment, performed by the Principal Component Analysis (PCA)
- A finer alignment, performed by the Iterative Closest Point (ICP) estimation

It is worth pointing out that the alignment algorithm has been developed for clouds with a relatively high number of points, which is not the case of the 3D InISAR. As a consequence, some modifications to both the PCA and the ICP have been proposed in [21] to adapt these techniques to the radar case. Details of this modified algorithm have been provided by J. Cai et al in [21]. The result of the alignment process is given by a matrix representing the rotated target points that provide the minimum RMSE (Root Mean Square Error) in terms of Euclidean distance between the target points and the model points. Such RMSE can be used as a performance index to determine the target class according to its minimum value.

The authors of [21] focus their attention on the development of the alignment algorithm. The 3-D point-like reconstruction alignment is an important research field in computer vision and pattern recognition with broad applications, such as object detection and recognition. Generally speaking, the alignment problem is composed of two steps: a coarse alignment aiming at obtaining an initial estimation that is used as cueing input to the second step, namely the fine alignment. Obviously, the alignment algorithm is based on affine transformations, namely only rotations and translations. No scaling operations are admitted that would modify the shape of the reconstructed target. Also, is often useful to subsample the CAD in order to speed up the procedure, given that we can maintain the original point cloud density for each sub-volume of the CAD.

4.1.1 PCA and ICP

The PCA is used to perform the coarse alignment. However, a PCA-based method is typically used to align two clouds of points with a similar number of points. Therefore, an improved PCA-based algorithm is proposed. Moreover standard PCA is ambiguous as it cannot determine the axes pointing direction. In other words, PCA can estimate the axes orientation in the 3D space but not their pointing direction. Therefore resulting in a ambiguity of $180°$. Let P_i and Q_j be the set of points forming respectively the 3D InISAR reconstruction and the target model of the target, where $i = 1, \cdots, N_R$ and $j = 1, \cdots, N_M$ and $N_M \ll N_R$. Then, the PCA calculates the covariance matrices of the two clouds of points, $[C_P]_{(3\times 3)}$ and $[C_Q]_{(3\times 3)}$, and applies an Eigenvalue Decomposition (EVD) to obtain the eigenvalues and eigenvectors of the covariance matrices as follows:

$$C_P = U_P S_P U_P^T \tag{4.1}$$

$$C_Q = U_Q S_Q U_Q^T \tag{4.2}$$

where the eigenvectors U_P and U_Q represent the principal orientation of the point clouds, P_i and Q_j, respectively, and the respective eigenvalues. Therefore, the coarse rotation and translation matrices can be found as follows:

$$R_0 = U_P U_q^{-1} T_0 = \bar{Q} - R_0 \times \bar{P} \tag{4.3}$$

$$T_0 = \bar{Q} - R_0 \times \bar{P} \tag{4.4}$$

where $\bar{Q}$ and $\bar{P}$ are the centroids of Q_j and P_i, respectively. The point cloud after the coarse alignment is:

$$P_i^{'} = R_0 \times P_i + T_0 \tag{4.5}$$

However, as said previously, EVD is not able to recover the sign of each eigenvector, and this may lead to an incorrect alignment if considering that man-made targets exhibit a certain degree of symmetry with respect to one or more axes. Mathematically, there is no way to avoid this ambiguity. However, our purpose is to align P_i with Q_j, and therefore only the signs of U_P need to be adjusted while the others can be fixed as a template, (also the vice versa is possible).

Then, the Euclidean distances and the root mean square error (RMSE) are calculated between all combined pairs of point clouds. The calculated Euclidean distances can also be used to eliminate outliers, namely those scatterers that have a distance to any point of the model that is greater than a preset threshold. Therefore, the coarse alignment result is chosen as the one with the minimum RMSE. Is also important to note that the eigenvalues found by the PCA could be used as estimates of a target extent along its main axes.

The ICP algorithm is the most classical point cloud registration algorithm, which exploits an iterative method based on the nearest point to minimize the error function of the Euclidean distance between two point clouds. Firstly, for each point of P_i, the corresponding point in Q_j is found by searching for the closest one based on a distance measure, for example the Euclidean distance. Secondly, based on the corresponding pairs, the optimal rigid transformation, namely a rototranslation, is found by minimizing a distance-based objective function. If the Euclidean distance is chosen, the results are as follows:

$$E(R,T) = \frac{1}{N_R} \sum_{i=1}^{N_R} \|Q_i - (RP_i + T)\|_2 \tag{4.6}$$

Although the ICP algorithm is quite simple and intuitive, if two point clouds are far from each other or if their poses are very different, the ICP is likely to fall into a local optimum. That is the reason why a coarse alignment is needed. Accurate correspondence pairs are able to improve the precision and accelerate the convergence speed of the alignment. The k-d tree exploited in the coarse alignment step is also implemented to search the correspondence pairs in the accurate alignment step. The k-d tree is a generalization of binary tree and is able to recursively search the nearest neighbours with high efficiency. To quantitatively analyse the alignment performance of the ICP algorithm, the RMSE can be calculated at each iteration of the ICP algorithm:

$$RMSE^{(k)} = \sqrt{\frac{1}{N_R}\sum_{i=1}^{N_R} \|Q_i^{'(k)} - (RP_i^{'(k+1)} + T)\|_2^2} \tag{4.7}$$

According to the ICP algorithm, at a generic kth iteration, a new cloud of points $Q_i^{'}$ consisting of the closest points of Q_j to $P_i^{'}$ is obtained. Then, R and T are calculated (by using the SVD or Quaternions methods) and the new set $P_i^{'(k+1)}$ is calculated. Then the procedure is reiterated again by finding the new set $Q_i^{'(k+1)}$ until a stop condition is reached, namely $k = k_{max}$ or $E_k(R,T) - E_{k-1}(R,T) < \epsilon$. Both $E_k(R,T)$ and $RMSE^{(k)}$ may be used as a similarity degree between the target model and the radar target reconstruction, and therefore used as a mean of comparison among different target models to find the target class.

4.1.2 3D Polarimetric template matching

The use of the geometrical information, while fundamental, could be improved with the addition of the polarimetric information. The scatterers extracted from the interferometric radar data contain not only the relative position information, but also the information regarding the complex scattering data contained in the four polarimetric channels. However, it is not advised to add the polarimetric channel dimension directly to the coordinate vector to calculate the new RMSE, because polarimetric information is subject to more variation with respect to relative position vector because it depends strongly on the aspect angle as viewed by the radar. For this reason, the polarimetric information is used to calculate a secondary RMSE, which measures only the affinity in the polarimetric response. Once the reliability of the feature extraction is assesses (for example by analysing the signal to noise ratio level, variability of the polarimetric response or others metrics), it is possible to combine the Polarimetric RMSE with the 3D geometry RMSE with a linear combination, weighting differently the information, in order to obtain a final RMSE which contains the final metric. The main limitation of this approach is that the 3D model (CAD or database of point cloud) would need to contain the polarimetric information for comparison, and an extensive measurement campaign in a controlled environment as an anechoic chamber is often needed, to account both for amplitude values and aspect angle variability.

The complex amplitudes of the various scatterers extracted via the scatterer selection algorithm are stored in memory along with the coordinates of the scatterers. The scatterer matrix S is a $4xN$

matrix where 4 are the polarimetric channels (the cross-polar channels are not equal because the geometry is not monostatic) and N is the number of extracted scatterers.

Also, for each point in the CAD model, a polarimetric amplitude should be associated together with point coordinates.

For each point of the target extracted point cloud, the nearest point in space in the CAD point cloud is found, and the correspondent complex amplitude is memorized, to obtain a matrix C, again with dimension $4xN$.

The $RMSE_{pol}$ can then be calculated as follows:

$$RMSE_{POL} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{4}|C(k,n) - S(k,n)|^2} \tag{4.8}$$

The final metric can be evaluated in two ways

- By averaging the geometrical and polarimetric RMSE

- By computing a weighted average with respect to the weight of geometrical RMSE:

$$RMSE_{TOT} = \frac{RMSE_{GEO} + \alpha RMSE_{POL}}{1+\alpha} \tag{4.9}$$

  with a varying between 0 and 1 (if $\alpha = 1$ we fall back in case 1). Values over 1 are not advised.

Also, another strategy is to use the polarimetric RMSE only to evaluate in-class variability AFTER the main class is decided using the geometrical RMSE (which, anyway, includes polarimetry to select the optimal scatterers).

4.1.3 Example with real data

For clarity purpose, is possible to present an example with real polarimetric data to show how the algorithm works and the metric that it produces. For this assessment we used the MSTAR T-72 tank dataset, which is a 2D polarimetric radar dataset which is composed by several ISAR images of the tank observed from several azimuth angles and two different elevation angles. To produce the 3D interferometric data, we need two images from to slightly different azimuth angles or elevation angles, in order to simulate the return that an interferometric radar would see. In this way, is possible to extract the 3D data used to assess the relative positions of the scatterers. The polarimetric information can be extracted directly by the radar data by selecting the polarimetric complex amplitude at the defined coordinate.

As a model, a T-72 CAD model is freely available in both .stl (stereolitography) and .ply (point cloud) format, as shown in 4.2

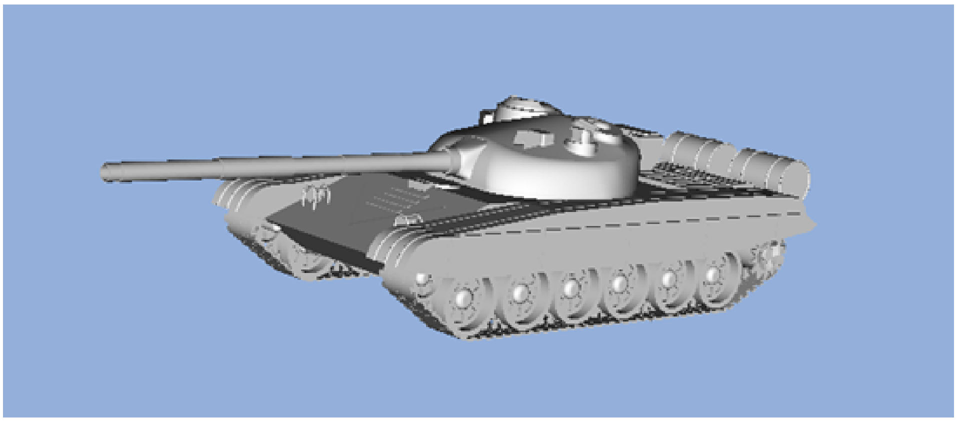


Figure 4.2: T-72 tank CAD model

4.3 and 4.4 show the magnitude ISAR images of the two slightly different views of the tank, for the HH channel.

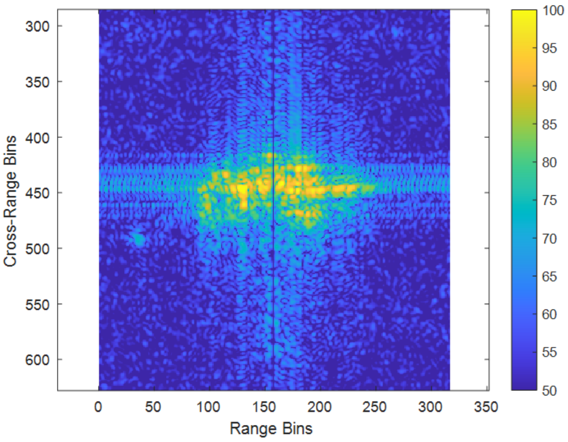


Figure 4.3: T-72 ISAR master image

Fig. 4.5 shows the RGB Pauli amplitude image, where different colors represent a different scattering behaviour. For example, scatterer in red, relative to the cannon, have a double-bounce

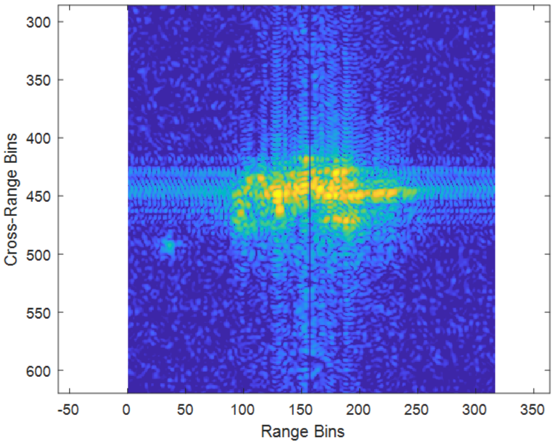


Figure 4.4: T-72 ISAR slaveimage

behaviour. The white circles represent the position of the scatterers extracted by the Pol-Clean algorithm for the 2D image.

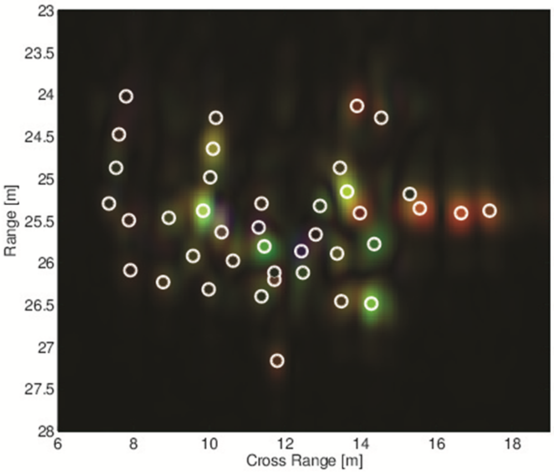


Figure 4.5: T-72 ISAR RGB image with extracted scatterers by POL-Clean

As explained in section 2.3 is possible also to use the polarimetric information to extract the optimal 3D point cloud from the interferometric data. The result is show in 4.6 and 4.7 extracting 50 scatterers from the dataset in the XY plane and YZ plane. The two images represent the back view and the top view of the 3D point cloud. Note how the dataset is tilted, because the algorithm can't control the image plane of the dataset.

The next step is matching the data cloud point with the CAD cloud point via the two step algo-

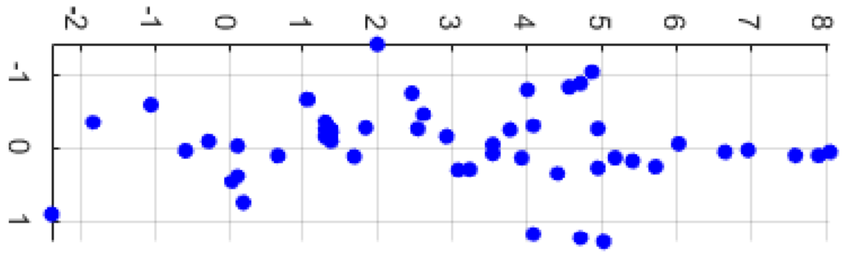


Figure 4.6: T-72 target cloud point, XY plane

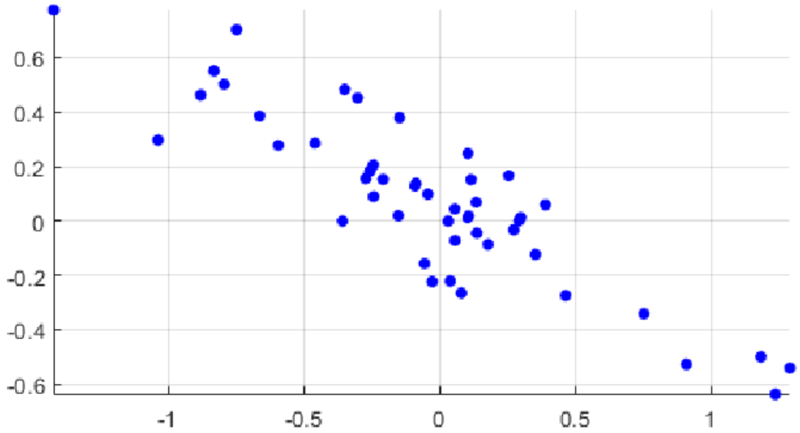


Figure 4.7: T-72 target cloud point, YZ plane

rithm presented in section 4.1.1.

The algorithm can match very well the two cloud points sets. 4.8 and 4.9 show the results of the PCA/ICP two step alignment procedure with polarimetric optimization. In the XY plan the alignment can match also the cannon, which is an important feature, and match with a good approximation also the planar size. Points in the XZ plane are somewhat more squeezed on the horizontal plane, but still the match quite well the silhouette.

The final objective of the first part of the procedure is however not to obtain the best visual match, but to produce a metric (in this case, the RMSE) which is used as a performance index to be compared between different dataset. As one would expect, the RMSE increases with increased

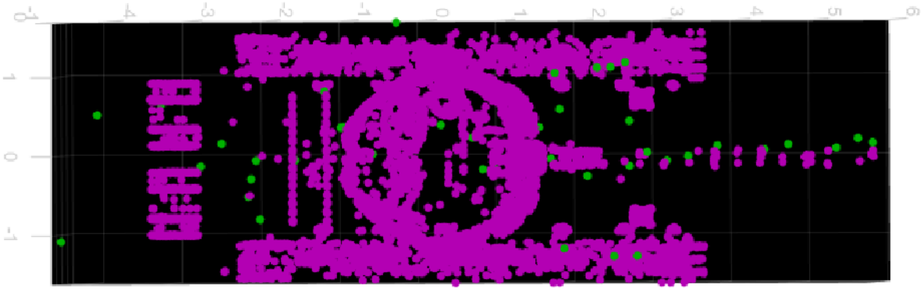


Figure 4.8: T-72 target/CAD cloud point matching, XY plane

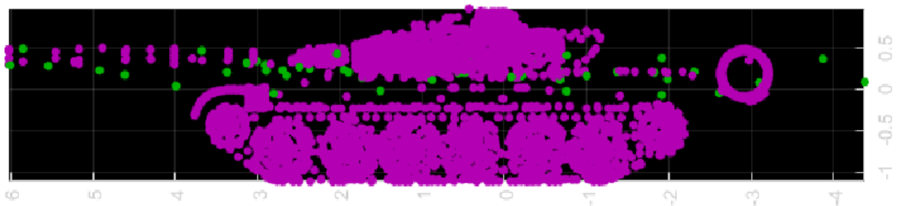


Figure 4.9: T-72 target/CAD cloud point matching, YZ plane

variance of the coordinates of the detected scatterers, as shown in Fig. 4.10. In the ideal case of no estimation error, the minimum RMSE is about 0.125. This is an important factor to asses the mismatch with wrong classes.

In Fig. 4.11 the algorithm is forced to match the 3D point cloud of the radar with the CAD of a P-92 plane, with comparable size, without coordinate error As expected, the algorithm finds a result but tries to align the tank cloud point along the wings. While for a computer is impossible to know if it is right or wrong, the RMSE is the metric to look at. For this matching tentative, the RMSE is 0.5243, which is five time the one obtained with correct match. However, it is not to exclude that with greater coordinate error, the RMSE could be smaller, because the point cloud "expands" to match with the plane size.

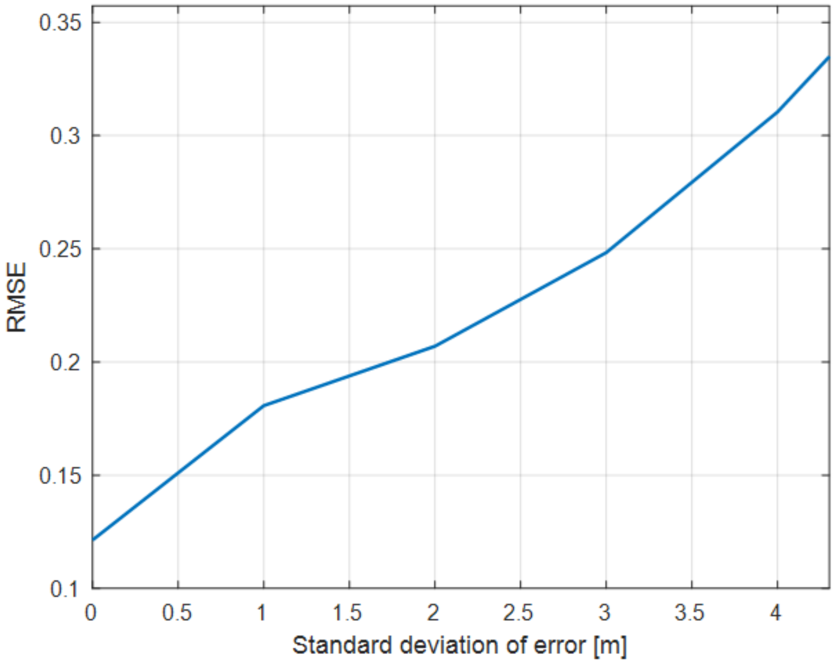


Figure 4.10: RMSE with increasing coordinate estimation error

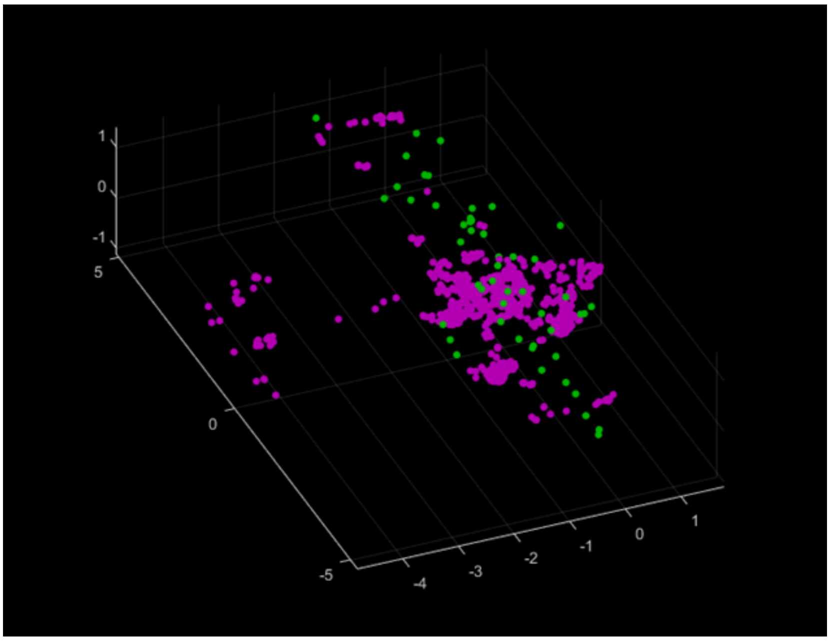


Figure 4.11: T-72 and P-92 planecloud point matching

4.1.4 RMSE assessment

The final steps consist in calculating the RMSE regarding the polarimetric response matching and how to combine the results from the geometrical and polarimetric matching. The complete algorithm will be better analyzed in the next reports, but here we show the principle of work of the process.

For the sake of simplicity, and due to the fact that a precise polarimetric model to be inserted into the CAD is needed, let's assign to the CAD points in the point cloud, the values of the amplitudes extracted from the slave dataset (assuming that we extract several more scatterers to complete the CAD point cloud which is, anyway, already downsampled), while keeping for comparison the values in the master image.

The rationale is that the two images differ only by a few degrees in the azimuthal plane. Particularly master and slave have been acquired with aspect angles $4°$ far apart.

The estimated $RMSE_{POL}$ is 3.93, which is an order of magnitude greater with respect to the $RMSE_{GEO}$. For this reason, is reasonable to use a weighted system, or to use in two different phases the two metrics.

In Fig. 4.12 the value of the $RMSE_{POL}$ is plotted against growing mismatch in polarimetric similarity, while in 4.13 it is possible to see how the $RMSE_{TOT}$ tends to grow with growing α. For this reason, if a single weighted metric is used, it is advisable to use α not larger that 0.5.

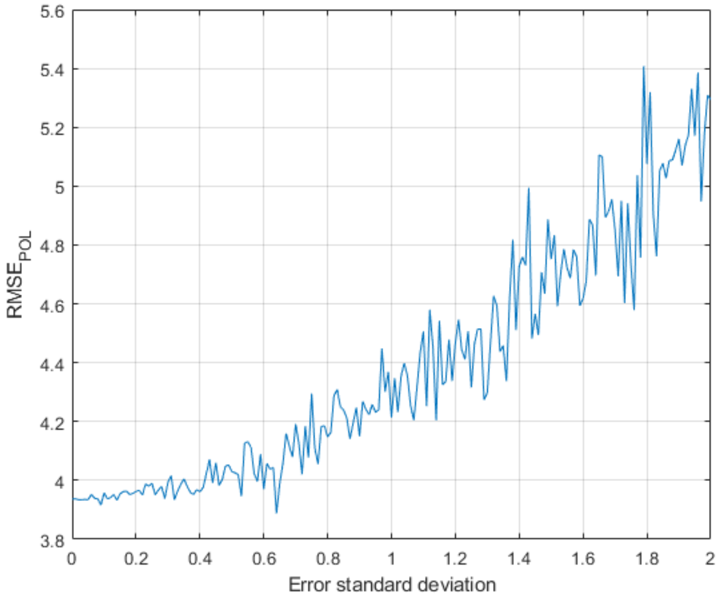


Figure 4.12: Polarimetric RMSE against mismatch error (standard deviation)

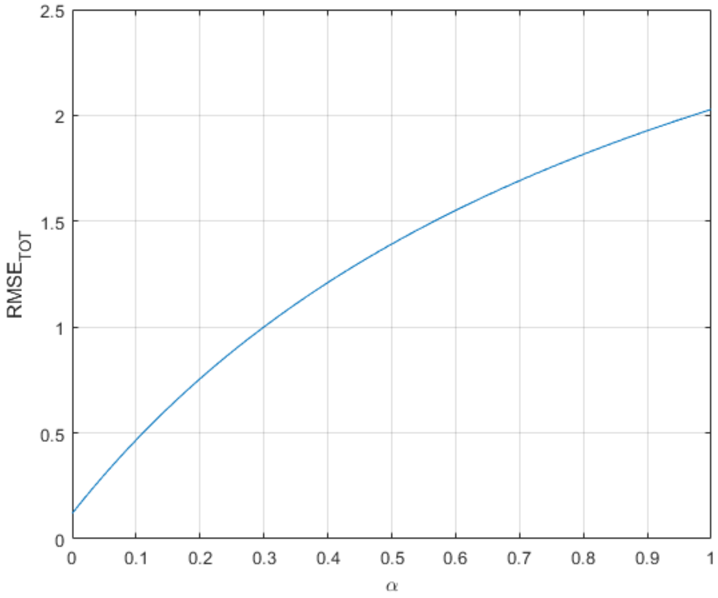


Figure 4.13: Variation of total RMSE with increasing α

4.2 DL-based ATR

The terms deep learning refer to a subset of machine learning methods based on artificial neural network that recently became of huge interest among the researchers. As stated in the previous report, the main features that led to such increase in popularity are their power and flexibility. Indeed, also thanks to the always increasing amount of available training data, these algorithms rapidly became the new state-of-the-art in different areas such as Natural Language Processing, Speech Recognition, Image Classification or, more recently, Point-Cloud Classification.

Employing deep learning, we are shifting the burden of designing the feature extractor from the researchers to the network. A feature extractor is a set of mathematical operations that modify the input data in order to reduce its dimensions while keeping all the relevant features. This approach is equal to a projection of the input data onto a smaller features subspace where targets of interest are more easily distinguishable.

In traditional classifiers, exploiting the most prominent features of the targets, the feature extractor is manually designed, or "handcrafted", in order to increase the aforementioned distinguishability. Despite being more reliable, this approach is less powerful then the deep learning-based whenever we have the possibility to train the classifier with a sufficiently large amount of data. Indeed, training a deep neural network can be seen as an estimation problem where we have to estimate the value of all the parameters, or weights, in order to approximate the underlying function that maps the input to the right output. But, since the number of weights can easily be in the millions, the amount of available data becomes a problem itself.

Summarizing, the deep learning approach offers higher performances than the traditional one at the cost of explainability and complexity of the training data.

4.2.1 State of the art

One of the first problem that needs to be addressed is the format of the input data. A 3D object can be represented in a multitude of formats, like voxels, meshes, graphs or point clouds, and, when training deep neural networks, the input data plays a huge role in achieving good performance. All of the aforementioned formats are 3D, which means that the intrinsic tridimensional nature of the object is maintained.

On the other hand we have 2D representations, in which the tridimensional object is projected onto a specific domain to form a bidimensional image. 2D representation benefits from the wide effort devoted to the task of image classification but looses important geometrical information during the

projection process.
In between, there are 2.5D representations which encode the third dimension inside the resulting 2D image in order to avoid the information loss due to the projection. The encoding can be done using the depth information alone (i.e. creating depth images) or, if present, using also the colour information, like in the RGB-D format. For each of this formats, various neural network architectures have been recently proposed and tested on the Modelnet dataset [24], available at [25]. Table 4.1 reports the analyzed architectures namely: SimpleView, Point Cloud Transformer (PCT) and Dynamic Graph Convolutional Neural Network (DGCNN).

Representation	**Network**	**Score**
2.5D	SimpleView [26]	93.9%
Point Cloud	PCT [27]	93.2%
Graph	DGCNN [28]	92.9%

Table 4.1: Classification accuracy for each neural network, as reported in the corresponding paper.

In [26] a simple but effective multiview architecture for 2.5D images is presented. SimpleView creates 6 depth images obtained from as many orthogonal directions and extracts meaningful information from each point of view. After a fusion layer, the last fully connected layer is used to obtain the classification scores. The feature extractor, which is a modification of the ResNet18 model but with one-fourth filters, was designed in order to have a comparable amount of parameters with respect to point-based classifier. In Figure 4.14 the SimpleView architecture is shown.

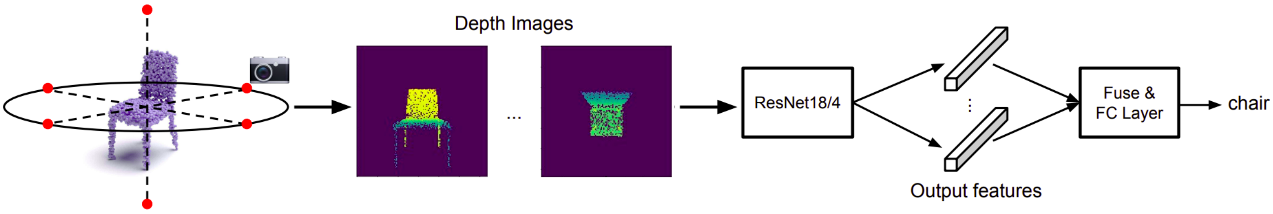


Figure 4.14: SimpleView, procedure and architecture. The depth images are colored just to enhance their meaning, this architecture only accept 1D images as input.

Multiview classifiers, which, like SimpleView, are based on 2D-CNN, have the best performance but lack the “spatial awareness” of other 3D classifiers. Especially with point cloud where there are no textures or clear edges, these kind of classifiers mainly rely on the object shape which, in itself, is heavily dependant on the number of points of the 3D object. A possible solution to this problem are 3D-CNN, however, the computational complexity of such architectures is still greater then their benefits, making their use unfeasible. Lastly, CNNs extract only local features because the convolutional filter only consider few adjacent pixels, missing therefore global features that could be used to improve classification accuracy.

Transformer networks are relatively new if compared to the more established multiview algorithm, but gained a lot of interest thanks to their success in natural language processing (NLP) [29]. As explained in [27], this kind of neural network are based solely on the attention mechanism which allows the network to estimate the importance of each input element with respect to all the others, effectively extracting contextual information. Experiments done for the NLP task showed the advantages of Transformer over traditional recursive or convolutional network, such as shorter training time and superior performance. Furthermore, this architecture already demonstrated an high generalization capability when applied to image classification, speech recognition or, as in our case, point cloud classification.

Neural network that works on graphs were recently proposed as a solution to the locality problem of CNNs. Constructing a graph from the point cloud allows the network to evaluate the relations even between distant points, improving the overall quality of the extracted information. In [28], the authors of DGCNN proposed a dynamically updated graph that is recomputed after each layer. This means that, after each layer and for each point, the k-nearest neighbours are not computed using the distances from the original point cloud but rather using the distances computed in the feature space obtained after the last layer. Indeed, two points that are close in the feature space are also semantically similar and this relation wouldn't be detected using only spatial distances. This set of operations is performed in the *EdgeConv* block, where each point is elaborated with respect to all the others. The output, for each combination of input elements, is an edge feature set, which is then aggregated using the $\sum$ or $\max$ functions to obtain the responses for corresponding points. The architecture of DGCNN is shown in figure 4.15

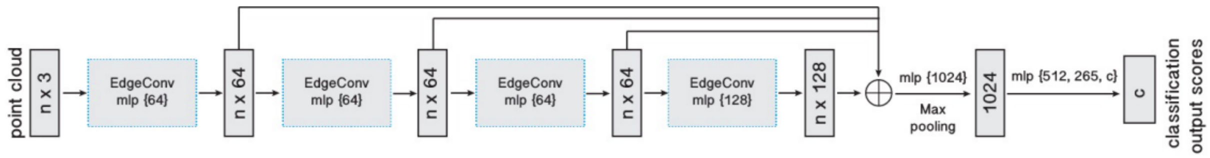


Figure 4.15: Architecture of the DGCNN classification branch.

Each method has its own advantages and disadvantages, but for our purpose we are more interested in some specific aspect:

- The network must accept a variable number of points. Given the non-cooperative nature of the targets of interest, it is reasonable to assume that the number of points returned by the radar system will not always be the same. In fact, this value can range from a few to several tens.
- The targets dimension should be taken into account during the classification process since it is one of the most discriminative feature.
- The chosen architecture should give us the possibility to easily enrich the input data with polarimetric information.

Given these new requirements, their recent success [27, 29] and the performances obtained, we decided to use Point Cloud Transformer as network architecture and point clouds as the input data.

4.2.2 Point clouds and Transformer

A point cloud is a list of coordinates describing a set of points, typically placed on the outer envelope of the represented object, in an euclidian 3-dimensional space. This data structure doesn't require projection or quantization but is permutation invariant, hence the list of points can be scrambled without affecting the represented object. Furthermore, since this is the raw output format of the 3D reconstruction algorithm, there is also no loss of spatial information.
PCT is based on Transformer, which were developed for NLP and are therefore inherently permutation invariant, making it well-suited for learning from point cloud.
A Transformer is an encoder-decoder structure based on self-attention, a concept introduced in [30] that immediately gained the interest of the NLP research community. It is composed of three main modules used for input embedding, positional encoding and self-attention evaluation. An *embedding* is an alternative representation for each element of the input data in an higher dimensional space where, if two elements are close, then they are probably also semantically similar. The positional encoding, in the original Transformer architecture, accounts for word ordering and is generated from the input sentence. Finally, the attention module is the core of the Tranformer and is in charge of generating the attention features later used for classification.

Self-attention takes as input the sum of the output of the other two modules (i.e. the input embedding and the positional encoding) and computes, through linear layers, three vectors for each input element: *key*, *query* and *value*. The attention scores are calculated with a dot-product between a *query* and all the other element's *key*, including itself. Then, the softmax function is applied to the attention scores vector and each element is multiplied by its corresponding *value*. The results are summed element wise to obtain the attention feature vector. The diagram of Figure 4.16 is taken from a simple case study [31] with only 3 input elements and shows this procedure for input 1. But, in practice, it is repeated for each *query*, or, equivalently, for each input element.

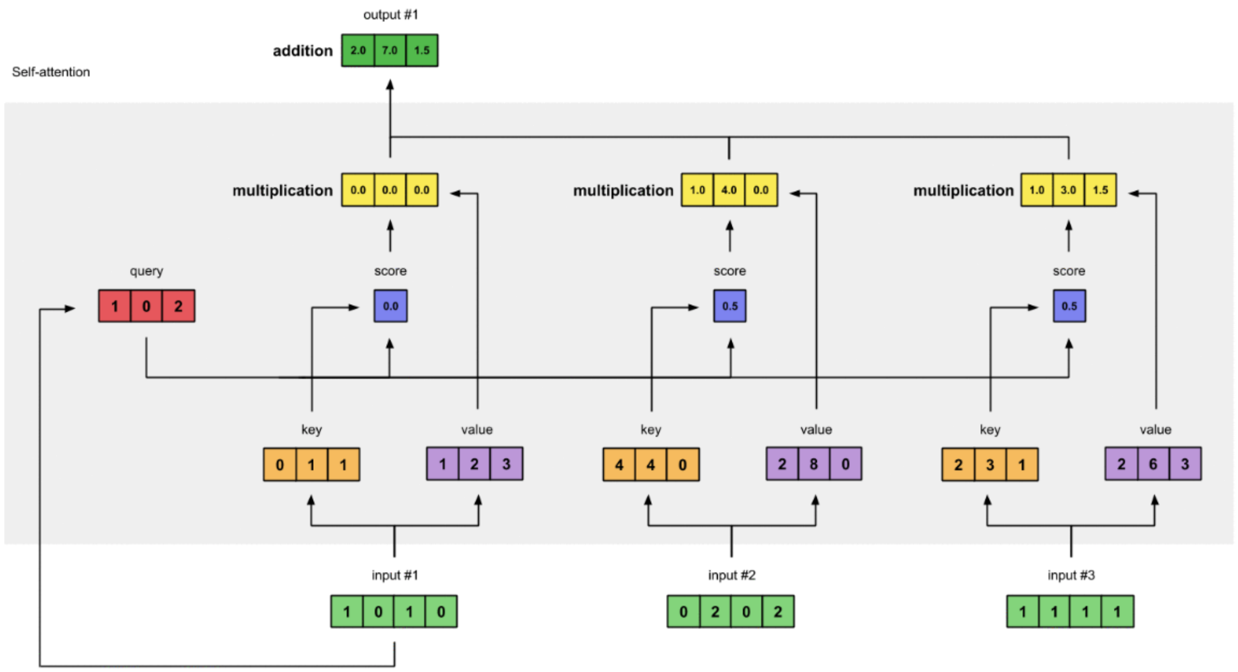


Figure 4.16: Diagram explaining the self-attention procedure for a simple case with 3 input elements.

In a Transformer-based architecture, self-attention module are stacked on top of each other in order to extract more complex and meaningful information. Depending on the task to be accomplished this attention features are either further refined or directly fed into a fully connected layer, which, typically, is trained to act as a classifier.

Despite the similarities, point clouds are more difficult to learn than languages and the authors of PCT had to slightly modify the original Transformer architecture. The main differences are here reported:

- Coordinate-based input embedding module: in a point cloud there is no meaning associated with the points ordering and the aforementioned positional encoding can't be applied. On the other hand, each point, represented with x,y,z coordinates in a tridimensional euclidean space, already contains the positional information required and therefore the first two encoding modules can be fused in a single embedding module based on the coordinates of each point.

- Neighbor embedding module: the output embedding of the previously described coordinate-based module are computed using individually each point and are therefore only weakly related to the semantic content of the whole point cloud. To address this issue, the authors proposed the neighbor aggregation strategy to further enrich the semantical meaning and the geometrical information of the embeddings. To do so, the k-nearest neighbor (k-NN) are computed for some sample points and then, a layer aggregates the features of those local neighbors. A similar mechanism is used in DGCNN architecture.
- Optimized offset-attention module: another problem that needs to be address when learning from point cloud comes from rigid transformations. Using the absolute coordinate system, the same object can be completely different in terms of coordinates when a rigid transformation is applied. For this reason, using relative coordinates can generally improve the robustness of the architecture. The offset-attention tries to solve this issue by replacing the attention features with the offset between the un-elaborated input and the output attention features.

The final PCT architecture is composed of the embeddings module, 4 optimized offset-attention module stacked on top of each other followed by a Linear-BatchNorm-ReLu (LBR) and two Linear-BatchNorm-ReLu-Dropout (LBRD) layers. Figure 4.17 shows the described architecture.

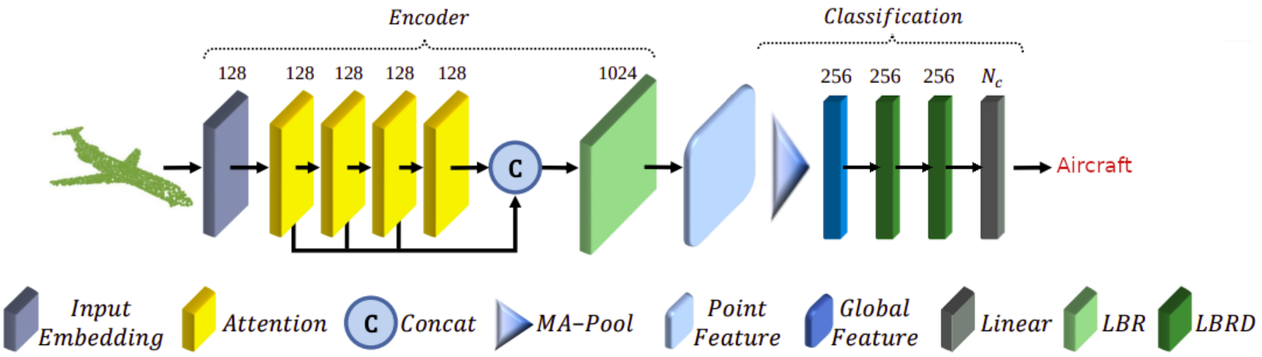


Figure 4.17: Architecture of the PCT's classification branch. LBR is used to identify the combined use of Linear, BatchNorm and ReLu. LBRD is analogous to LBR but with a Dropout layer after the ReLu.

The authors also proposed an input embedding module made of multiple linear layers but the slight performance gain isn't worth the increase in complexity and was not considered during our test.

4.2.3 Dataset and implementation

The original PCT implementation is written in Jittor but other unofficial implementations are available as well. We decided to use the one based on PyTorch library, downloadable from [32], since it is one of the most widely used and supported machine learning library. The first test was performed to validate the training set up and was therefore carried out using the original hyperparameters and the Modelnet40 dataset. The succeeding tests were performed on our custom dataset, specifically generated in order to be more similar to the environment that the radar system will face.

Point clouds of various objects, like vehicles, ships or airplanes, can be found online as CAD model [33]. These objects are designed using polygon meshes, usually triangles, and their representation often contains a list of the polygons vertexes. The list of vertexes is a point cloud representation of the outer envelope of the object and can therefore be used as a starting point in the generation of the new dataset. In table 4.2 the used models and their corresponding number of points are shown.

Model	N° of pts	Model	N° of pts	Model	N° of pts	Model	N° of pts
A3000	6478	C550	2165	TU160	8157	Fishing	39683
B52	30606	F16	2630	TU160C	8155	Fremm	20523
Boeing747	8569	learjet85	14055	Carrier	112828	Maersk	714903
Boeing727	5193	MIG29	1738	Corvette	427805	Oilgas	1022854
C-130H	30936	MIG35	18701	Ferry	1847	Patrol	391306

Table 4.2: The used model are drown from a possible set of ground and air targets.

A downsampled version of each model, made of nearly 50 points, is generated using a voxel-based subsampling function. The input point cloud is regularly subdivided in 3D grid cells, or voxels, each cell then returns the centroid of the points in itself by averaging the corresponding coordinates. Changing the dimensions of the voxels affects the downsampling factors, as well as the amount of points after the filtering. Each subsampled model is then manually enriched with the insertion of points in all the position that are representative of the target or that could be particularly prominent in an ISAR image. After this process, 64 points are left for each model. In order to generate a sufficient amount of training data, we combined the original point clouds with the corresponding subsampled versions by averaging close points, a detailed description of the procedure is reported in Algorithm 3. Currently only this data augmentation technique has been tested but further researches will be conducted on other techniques.

Algorithm 3: Dataset generation procedure

Data: Original point cloud of each model and corresponding subsampled version
Result: A variable number of data augmented point clouds.

```
for each original PC and its corresponding subsampled version do
    $avgDist \leftarrow$ CalcAvgDistance(subsampled PC) ;
    $auxPC \leftarrow$ Copy(subsampled PC) ;
    $listOfPts \leftarrow$ Shuffle(original PC) ;
    for each point $P$ in $listOfPts$ do
        $minDist \leftarrow$ Min( Norm(P - auxPC)) ;
        $minInd \leftarrow$ Argmin( Norm(P- auxPC) );
        if $minDist \leq avgDist/2$ then
            $auxPC[minInd] \leftarrow \frac{P + auxPC[minInd]}{2}$;
            $AugmentedPoints \leftarrow AugmentedPoints + 1$;
        end if
        if $AugmentedPoints \geq 0.75 \times$ Length(auxPC) then
            savePC() save to file the new point cloud
            $auxPC \leftarrow Subsampled\ PC$ ;
            $AugmentedPoints \leftarrow 0$ ;
        end if
    end for
end for
```

In Table 4.3 the number of newly generated point clouds for each model is reported. All the point clouds generated with the previously described algorithm have 64 points.

Model	N° of PC	Model	N° of PC	Model	N° of PC	Model	N° of PC
A3000	7	C550	2	TU160	23	Fishing	7
B52	63	F16	5	TU160C	17	Fremm	5
Boeing747	15	learjet85	24	Carrier	78	Maersk	719
Boeing727	9	MIG29	2	Corvette	89	Oilgas	450
C-130H	54	MIG35	62	Ferry	3	Patrol	12

Table 4.3: Each model with the corresponding number of subsampled point clouds (64 points) generated using the described procedure.

In order to verify the capabilities of the network to learn from point clouds with just 64 points, the generated dataset is divided in two simple classes: aircraft and ships. The models with more generated point clouds are used in the training phase, while the models less represented are used in the test phase. This means that the network is trained on the point clouds of, for example, the Boeing747 but is tested on the Cessna C550, a never before seen model of aircraft. This division helps us understand if the network is actually able to extract meaningful information from the input data or is just memorizing (i.e. it is overfitting). After 250 training epochs the achieved overall accuracy is 100%.

The second experiment with 64 points is performed with all the 20 models used as individual classes. The subsampled versions of each model are split in training set and test set with a 70/30 ratio. At the end of the training phase, the accuracy on the test set is equal to 99%. A performance analysis with respect to the size of the input point cloud for the network trained on 64 points is given in image 4.18.

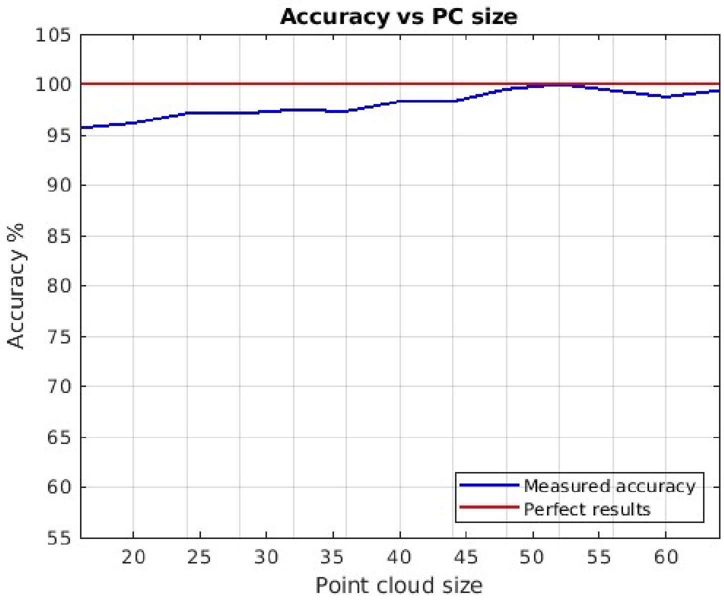


Figure 4.18: Classification accuracy of the PCT network vs the number of points of the input data. The network was trained with 64 points element but tested on a different number.

Chapter 5

Conclusions

The project is progressing according to the work plan. Two methods for the estimation of the target height using fully polarimetric data have been proposed and compared. A careful and deep analysis of the implemented methods has been conducted through the definition of suitable metrics. The results have shown the effectiveness of the use of polarimetric information when forming 3D InISAR images, when compared against single polarization algorithms. The algorithms have been tested both on simulated and real data. The simulated data allowed us to numerically quantify the accuracy of the proposed methods. The real data proves the effectiveness of the algorithms on real data, and proves the limits of the SPAN-based with respect to the coherence-based. Finally two ATR algorithms have been proposed and preliminary implemented with the aim to validate their feasibility. One falls in the branch of the template matching approaches the second falls in the branch of machine learning. The work will continue with the final implementation and refinement of the classification algorithms and their performance evaluation.

Bibliography

[1] M. Martorella, D. Stagliano, F. Salvetti, and N. Battisti, “3D interferometric ISAR imaging of noncooperative targets,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 4, pp. 3102–3114, 2014.

[2] B. Tian, Z. Lu, Y. Liu, and X. Li, “Review on interferometric ISAR 3D imaging: Concept, technology and experiment,” *Signal Processing*, vol. 153, pp. 164–187, 2018.

[3] E. Giusti, S. Ghio, and M. Martorella, “Drone-based 3d interferometric isar imaging,” in *2021 IEEE Radar Conference (RadarConf21)*, 2021, pp. 1–6.

[4] D. Li and Y. Zhang, “A fast normalized cross-correlation algorithm for insar image subpixel registration,” in *2011 3rd International Asia-Pacific Conference on Synthetic Aperture Radar (APSAR)*, 2011, pp. 1–4.

[5] M. Martorella, A. Cacciamano, E. Giusti, F. Berizzi, B. Haywood, and B. Bates, “CLEAN Technique for Polarimetric ISAR,” *International Journal of Navigation and Observation*, vol. 2008, pp. 1–13, Aug 2008.

[6] F. Mancuso, E. Giusti, and M. Martorella, “Polarimetric three-dimensional isar imaging,” in *2022 IEEE Radar Conference (RadarConf22)*, 2022, pp. 1–6.

[7] C.-A. Deledalle, L. Denis, and F. Tupin, “NL-InSAR: Nonlocal Interferogram Estimation,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 4, pp. 1441–1452, 2011.

[8] S. R. Cloude and K. P. Papathanassiou, “Polarimetric SAR interferometry,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 5, pp. 1551–1565, 1998.

[9] J. L. Gomez-Dans, “On the use of polarimetry and interferometry for SAR image analysis,” *Ph.D. thesis, Sheffield Centre Earth Obs. Sci., Univ. Sheffield, Sheffield, U.K.*, 2004.

[10] J. L. Gomez-Dans and S. Quegan, “Constraining coherence optimization in polarimetric interferometry of layered targets,” *in Proceedings of POLinSAR, Frascati, Italy: ESA*, pp. 1–6, Jan. 2005.

[11] E. Colin, C. Titin-Schnaider, and W. Tabbara, “An interferometric coherence optimization method in radar polarimetry for high-resolution imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 1, pp. 167–175, 2006.

[12] G. Wang, X.-G. Xia, and V. Chen, “Three-dimensional ISAR imaging of maneuvering targets using three receivers,” *IEEE Transactions on Image Processing*, vol. 10, no. 3, pp. 436–447, 2001.

[13] G. Margarit, J. J. Mallorqui, J. Fortuny-Guasch, and C. Lopez-Martinez, “Phenomenological vessel scattering study based on simulated inverse sar imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 4, pp. 1212–1223, 2009.

[14] Y. Li, S. Quan, D. Xiang, W. Wang, C. Hu, Y. Liu, and X. Wang, “Ship recognition from chaff clouds with sophisticated polarimetric decomposition,” *Remote Sensing*, vol. 12, no. 11, 2020.

[15] Y. Yamaguchi, T. Moriyama, M. Ishido, and H. Yamada, “Four-component scattering model for polarimetric SAR image decomposition,” *IEEE Transaction on Geoscience and Remote Sensing*, vol. 43, no. 8, pp. 1699–1706, Aug. 2005.

[16] “T-72A MBT Main Battle Tank - Russia,” https://www.armyrecognition.com, December 2021, [Accessed: 2022-05-29].

[17] U. A. Force, “Sensor Data Management System (SDMS),” Dataset online available: https://www.sdms.afrl.af.mil/, [Accessed: 2021-23-12].

[18] S. Cloude and E. Pottier, “A review of target decomposition theorems in radar polarimetry,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, no. 2, pp. 498–518, 1996.

[19] P. Rosen, S. Hensley, I. Joughin, F. Li, S. Madsen, E. Rodriguez, and R. Goldstein, “Synthetic aperture radar interferometry,” *Proceedings of the IEEE*, vol. 88, no. 3, pp. 333–382, 2000.

[20] J. Yang, Y.-N. Peng, and S.-M. Lin, “Similarity between two scattering matrices,” *Electronics Letters*, vol. 37, pp. 193–194(1), February 2001. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/el_20010104

[21] J. Cai, M. Martorella, Q. Liu, Z. Ding, E. Giusti, and T. Long, “Automatic target recognition based on alignments of three-dimensional interferometric isar images and cad models,” *IEEE Transactions on Aerospace and Electronic Systems, vol. 56, no. 6*, Dec. 2020.

[22] M. M. et al, “Target recognition by means of polarimetric isar images,” *IIEEE Trans. Aerosp. Electron. Syst., vol. 47*, Jan. 2011.

[23] B. D. F. Zhang and L. Zhang, “Saliency-guided unsupervised feature learning for scene classification,” *IEEE Transactions on Geoscience and Remote Sensing, vol. 53, no. 4*, April 2015.

[24] Z. Wu, S. Song, and A. e. a. Khosla, “3d shapenets: A deep representation for volumetric shapes,” 2014. [Online]. Available: https://arxiv.org/abs/1406.5670

[25] “Princeton modelnet,” accessed: 2022-05-30. [Online]. Available: https://modelnet.cs.princeton.edu/

[26] A. Goyal, H. Law, and B. e. a. Liu, “Revisiting point cloud shape classification with a simple and effective baseline,” 2021. [Online]. Available: https://arxiv.org/abs/2106.05304

[27] M.-H. Guo, J.-X. Cai, and Z.-N. L. et al., “PCT: Point cloud transformer,” *Computational Visual Media*, vol. 7, no. 2, pp. 187–199, apr 2021.

[28] Y. Wang, Y. Sun, and Z. e. a. Liu, “Dynamic graph cnn for learning on point clouds,” 2018. [Online]. Available: https://arxiv.org/abs/1801.07829

[29] J. Devlin, M.-W. Chang, and K. e. a. Lee, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2018. [Online]. Available: https://arxiv.org/abs/1810.04805

[30] A. Vaswani, N. Shazeer, and N. e. a. Parmar, “Attention is all you need,” 2017. [Online]. Available: https://arxiv.org/abs/1706.03762

[31] “Illustrated: Self-attention,” accessed: 2022-05-30. [Online]. Available: https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

[32] “Pct_pytorch,” accessed: 2022-05-30. [Online]. Available: https://github.com/Strawberry-Eat-Mango/PCT_Pytorch

[33] “Cadnav,” accessed: 2022-05-30. [Online]. Available: https://www.cadnav.com/3d-models/