



Universiteit
Leiden

The Netherlands

Spectral imaging and tomographic reconstruction methods for industrial applications

Zeegers, M.T.

Citation

Zeegers, M. T. (2023, May 31). *Spectral imaging and tomographic reconstruction methods for industrial applications*. Retrieved from <https://hdl.handle.net/1887/3619550>

Version: Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/3619550>

Note: To cite this publication please use the final published version (if applicable).

4

Multi-channel discrete tomography

In this chapter, we will look at discrete tomography where the number of materials in an object is limited. This can be exploited to make accurate reconstructions with a limited number of projections. The DART algorithm has proven to be a successful algorithm in this regard, but its usage is limited to objects with a few different materials. To push the number of materials that DART can handle, multi-channel data recorded with advanced tomographic imaging techniques – such as spectral X-ray tomography – can be used.

To effectively deal with multi-channel data, we present Multi-Channel DART (MC-DART). This class of algorithms is a generalization of DART to multiple channels and combines the information for each separate channel-reconstruction in a multi-channel segmentation step. We demonstrate in a range of simulation experiments with spectral X-ray imaging that MC-DART is capable of producing more accurate reconstructions compared to single-channel DART.

This chapter is based on:

M. T. Zeegers, F. Lucka, and K. J. Batenburg. “A Multi-Channel DART algorithm”. In: *International Workshop on Combinatorial Image Analysis*. (Porto, Portugal). Ed. by R. P. Barneva, V. E. Brimkov, and J. M. R. S. Tavares. Springer, 2018, pp. 164–178.

4.1 Introduction

Tomography is a non-invasive technique for creating 2D or 3D images of the inner structure of an object. Projections of the object are acquired by sending photonic or particle beams (e.g. X-rays, electrons, neutrons) through the object in a particular direction and measuring the signal resulting from interaction of the beam and the object at a detector. By acquiring this data from multiple positions and under various angles, a collection of projections is obtained. An image of the interior of the object is then reconstructed by applying a reconstruction algorithm to this projection data. Tomography is successfully used in many fields, including medical imaging [127] and electron tomography in materials science [93, 196]. If a large number of accurate projection images are available, solving the reconstruction problem is straightforward by a closed-form inversion formula [51]. Practical constraints on the dose, acquisition time or available space can impose limitations on the number of projections that can be taken, the angular range, or the noise level of the data, resulting in artefacts in the reconstructed images if standard reconstruction methods are used [127].

Discrete tomography is a powerful technique for dealing with such limited tomographic data. It can be applied if the object consists of only a limited number of materials with homogeneous densities. The Discrete Algebraic Reconstruction Technique (DART) [30, 31] is an algebraic reconstruction method for discrete tomography that alternates between continuous reconstruction steps and discretization of the image intensities by segmentation. The DART algorithm has demonstrated to obtain higher image quality reconstructions with limited projections and angles compared to standard reconstruction methods. Numerous successive studies have improved the DART algorithm, which include automatic parameter estimation (PDM-DART [3] and TVR-DART [328]), multi-resolution reconstruction (MDART [65]), relaxing voxel constraints (SDART [42]) and adaptive boundary reconstructions (ADART [181]). Nevertheless, a key limitation of DART is that it can only improve reconstruction quality if the number of different materials in the object is relatively small. The main reason is that for a larger number of materials, the segmentation step is no longer effective [30, 31].

In some cases it is possible to obtain tomographic information in *multiple measurement channels*. For instance, in X-ray imaging the beams are typically polychromatic, i.e. X-ray photon energies are distributed over a spectrum. Each material in the object has different attenuation properties for different X-ray energy levels. Whenever a single X-ray energy value is desired the range of energies within the beam can be narrowed by applying filters at the X-ray source [51]. Some detectors are capable of separating the incoming photons into energy bins while counting (e.g. HEXITEC [295]). In these cases spectral multi-channel projection data are acquired, providing additional information about the object at different energies. Compared to the single-channel setting, where each material has a single

attenuation value in the reconstructed image, in the multi-channel setting the attenuation value for each material varies along the channels. In this way, a tomographic dataset of the object is acquired for each channel, where the attenuation values of the materials change throughout these datasets. This multi-channel imaging can potentially yield extra information about the materials. With more materials in the object, especially with similar attenuation features at a fixed energy, having data from multiple channels enables a better separation during segmentation. A conceptual example of this is shown in Figure 4.1. It is hard to separate points based on their attenuation values in a one-dimensional energy space. For instance, the right side of the blue area might as well be assigned to the green or yellow material during segmentation. With two energy dimensions the points are easily separable, since each voxel value lies close to its attenuation cloud center.

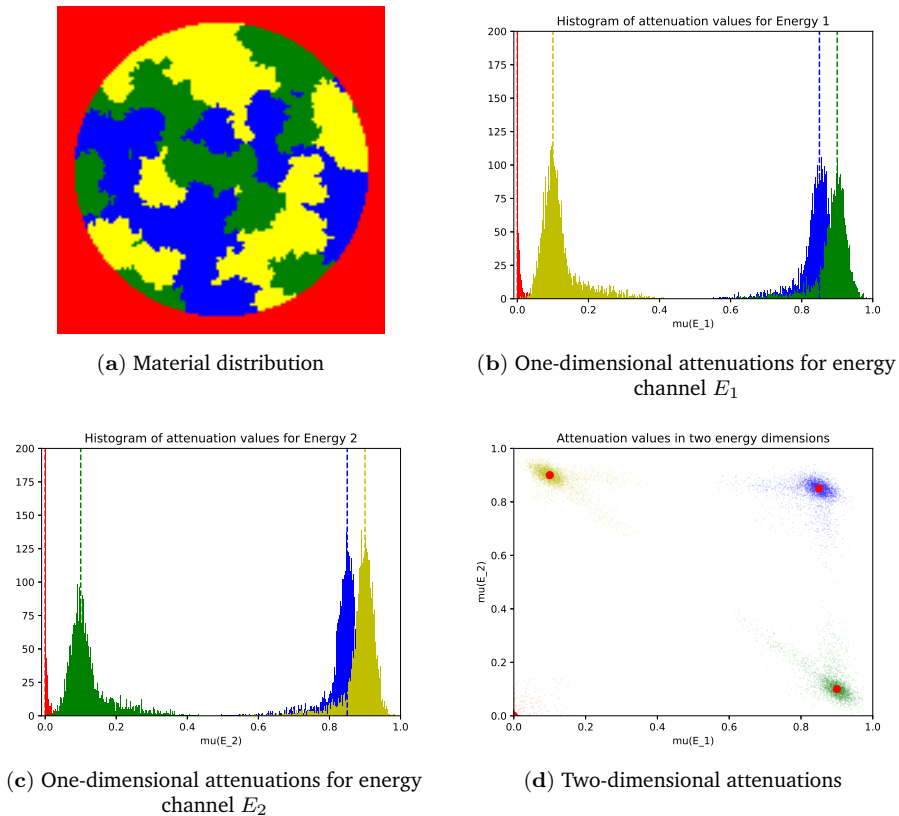


Figure 4.1: Elementary example of separation difficulties during segmentation. (a) Distribution of the three materials (blue, yellow, green) in the object. The background is indicated in red. (b) Histogram of attenuation values of pixels at energy E_1 (above) and E_2 (below). Vertical lines show true material attenuations. (c) Attenuations of the materials (red dots) and computed attenuations by a reconstruction algorithm for each voxel (colors indicate the materials these belong to).

that these spectra are artificial and not likely to occur in real-world examples.

In this chapter we present a new class of algorithms that combines DART with multi-channel imaging for solving discrete multi-channel reconstruction problems. Our method can combine the information from multiple channels to produce a final segmentation that is superior to that of the (single-channel) DART algorithm. Note that since this new method is designed by means of modules or subroutines that are interchangeable (as with DART), the method is essentially a class of algorithms providing a framework for dealing with multi-channel data. For simplicity, however, we will frequently call this framework an algorithm.

This chapter is structured as follows. Section 2 introduces the multi-channel discrete tomography problem. In Section 3 the DART algorithm is restated and the *Multi-Channel DART* (MC-DART) algorithm is introduced. Results of experiments with this algorithm are reported in Section 4. Finally, Section 5 presents the conclusions of this study.

4.2 Problem formulation

The standard (single-channel) tomography problem can be modelled as a system of linear equations. The image is characterized by a vector of voxel attenuation values $\mathbf{x} \in \mathbb{R}^n$, where n is the number of voxels. We will work with 2D images, but the problem formulation and methods in this work can easily be extended to the 3D setting. We will refer to the image pixels as *voxels* to distinguish these from detector pixels. We will often interchangeably speak of voxels and their corresponding indices. The projection values (also called data) are given as the vector $\mathbf{p} \in \mathbb{R}^l$, where l is the number of projection angles times the number of detector pixels. The reconstruction problem can then be described by solving the following set of linear equations for \mathbf{x} :

$$\mathbf{W}\mathbf{x} = \mathbf{p}. \quad (4.1)$$

Here \mathbf{W} is the projection matrix, also called the forward operator [140]. This matrix incorporates the contribution of each voxel to each projection, where element w_{ij} indicates the contribution of voxel j to projection i . Applying the operator \mathbf{W} on a vector \mathbf{x} results in the forward projection (the sinogram). Since inverting the matrix \mathbf{W} is computationally too expensive (or not even possible, for example when the problem is ill-posed) the reconstruction problem is to find a solution \mathbf{x}^* whose forward projection $\mathbf{W}\mathbf{x}^*$ matches the projection data best with respect to some norm $\|\cdot\|$.

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{W}\mathbf{x} - \mathbf{p}\| \quad (4.2)$$

Since this is a least squares problem over \mathbb{R}^n , a solution always exists. For simplicity of notation we also assume that it is unique. A vector that encapsulates noise

from real-world examples can also be modelled with (4.2). In our experiments with phantom examples in Section 4.4 there is no noise.

In the *discrete tomography problem*, the image to be reconstructed consists of a limited number of materials with homogeneous densities, each having an attenuation which is known beforehand by means of the set $\mathcal{R} = \{\rho_1, \dots, \rho_m\}$, where m is the number of different materials in the object. Therefore the problem to be solved becomes finding a vector $\mathbf{x} \in \mathcal{R}^n$ that matches the data best:

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathcal{R}^n} \|\mathbf{W}\mathbf{x} - \mathbf{p}\|. \quad (4.3)$$

Note that this is a minimization problem over a non-empty finite set. Hence, a minimum always exists. Again, the minimizer does not need to be unique but we use this notation throughout the chapter for simplicity.

In the *multi-channel* setting different properties of the target can be individually interrogated and measured. The information of each property is obtained through a separate channel. An example of a channel is an energy level, as in the example in Section 4.1. In Figures 4.1b and 4.1c, the channels are the two energy levels revealing attenuations of the object at different energies. In a more abstract way the object is described as set of voxels with labels instead of attenuation values, since each material has different attenuation values in different channels. The material labels are values in the set $\mathcal{M} = \{1, 2, \dots, m\}$. The channel indices are given by $\mathcal{E} = \{E_1, E_2, \dots, E_C\}$ where the number of channels is given by C . Again, the attenuations are known beforehand in the sets $\mathcal{R}_{E_1} = \{\rho_{1,1}, \dots, \rho_{1,m}\}$, $\mathcal{R}_{E_2} = \{\rho_{2,1}, \dots, \rho_{2,m}\}, \dots, \mathcal{R}_{E_C} = \{\rho_{C,1}, \dots, \rho_{C,m}\}$. In this setting, let $\mathcal{R} = \cup \mathcal{R}_{E_c}$. The function $\mu : \mathcal{M} \times \mathcal{E} \mapsto \mathcal{R}$ maps the label-channel combinations to their attenuation value, so the attenuation of a material with label s at channel E_c is given by $\mu(s, E_c)$. Note that there is not necessarily a one-to-one correspondence between the attenuation values and the material-channel combinations, because some combinations can have the same attenuation value. In this multi-channel case the projection data are given by a vector of projection data vectors at various channels:

$$\mathbf{P} = (\mathbf{p}_{E_1}, \dots, \mathbf{p}_{E_C}) \in \mathbb{R}^{n \times C}. \quad (4.4)$$

For each channel E_c the reconstruction problem for \mathbf{x}_{E_c} is given by the following set of linear equations:

$$\mathbf{W}\mathbf{x}_{E_c} = \mathbf{p}_{E_c}, \quad E_c \in \{E_1, \dots, E_C\}. \quad (4.5)$$

For $\mathbf{y} \in \mathcal{M}^n$, define $\mu(\mathbf{y}, E_c) = (\mu(\mathbf{y}_1, E_c), \dots, \mu(\mathbf{y}_n, E_c))^\top$ as the vector of voxel attenuation values at channel E_c . The multi-channel problem is now defined as follows. Given data vector \mathbf{P} and projection matrix \mathbf{W} , find a labelling vector $\mathbf{y}^* \in \mathcal{M}^n$ such that for each channel E_c the difference between forward projection $\mathbf{W}\mu(\mathbf{y}^*, E_c)$ and data is minimal with respect to some norm $\|\cdot\|$:

$$\mathbf{y}^* = \arg \min_{\mathbf{y} \in \mathcal{M}^n} \sum_{h=1}^C \|\mathbf{W} \mu(\mathbf{y}, E_c) - \mathbf{p}_{E_c}\|. \quad (4.6)$$

Note that for one channel the minimization problem is equivalent to (4.3) where the labelling is given by the attenuation values \mathbf{x} , by setting $\mu(\mathbf{y}, E_1) = \mathbf{x}$ and $\mathcal{M}^n = \mathcal{R}^n$ and $\mathbf{p}_{E_1} = \mathbf{p}$:

$$\mathbf{y}^* = \arg \min_{\mathbf{y} \in \mathcal{M}^n} \|\mathbf{W} \mu(\mathbf{y}, E_1) - \mathbf{p}_{E_1}\| \quad (4.7)$$

$$= \arg \min_{\mathbf{x} \in \mathcal{R}^n} \|\mathbf{W} \mathbf{x} - \mathbf{p}\|. \quad (4.8)$$

4.3 Algorithms

In this section, the Multi-Channel DART (MC-DART) framework for solving the minimization problem of Eq. (4.6) is introduced. We first explain the DART algorithm as given in [30] by discussing the overall structure and its building blocks. We then describe each building block of the MC-DART algorithm separately in more detail. Note that ASTRA [1, 215] provides an implementation for numerically computing all projection matrices in these algorithms, either by storing the full matrix or doing all necessary computations in a matrix-free way.

4.3.1 DART

The DART algorithm attempts to solve the optimization problem of Eq. (4.3) by iteratively alternating between continuous reconstruction steps and discrete segmentation steps. The number of materials in the object to be reconstructed and their attenuation values should be known beforehand, given by the function μ . The algorithm consists of several phases, which are indicated in the flow-chart in Figure 4.2. The pseudocode of DART is given in Algorithm 1.

Algorithm 1 DART

Input: $\mathbf{W}, \mathbf{p}, \mathcal{R}$
1: $\mathbf{x}^0 \leftarrow \text{Mask-ARM}(\mathbf{W}, \mathbf{p}, \mathbf{1}_n, \mathbf{0}_n)$
2: **for** $k = 1$ **to** K **do**
3: $\mathbf{y}^k \leftarrow \text{Seg}(\mathbf{x}^k, \mathcal{R})$
4: $\mathbf{M}^k \leftarrow \text{Mask}(\mathbf{y}^k)$
5: $\mathbf{x}^k \leftarrow \text{Mask-ARM}(\mathbf{W}, \mathbf{p}, \mathbf{M}^k, \mathbf{x}^{k-1})$
6: **Output:** $\mathbf{x}^K, \text{Seg}(\mathbf{x}^K, \mathcal{R})$

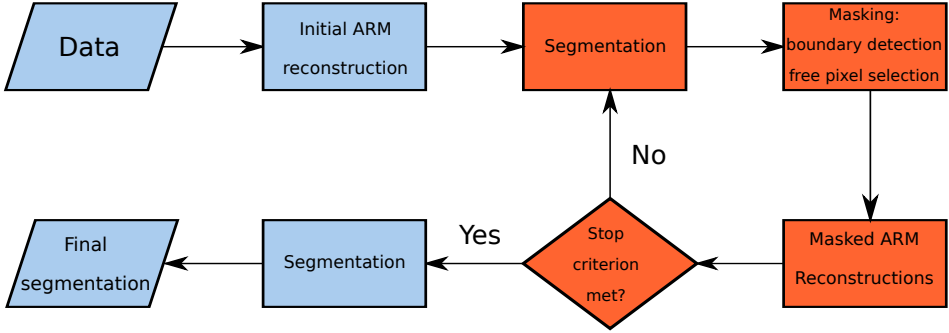


Figure 4.2: Flow chart of the DART algorithm. The DART iteration activities are indicated in red and the initialization and post-segmentation activities are indicated in blue.

Initialization

In the initialization phase, given the projection data \mathbf{p} and the projection properties by means of \mathbf{W} , an initial reconstruction \mathbf{x}^0 is calculated using an Algebraic Reconstruction Method of choice (hereafter referred to as the ARM), for example ART, SART or SIRT [140]. With the initial reconstruction \mathbf{x}^0 at hand, the main loop of the DART algorithm begins.

Segmentation

In this main loop, in iteration k the image \mathbf{x}^{k-1} is segmented using a simple thresholding scheme, forming the image $\mathbf{y}^k \in \mathcal{R}^n$, by computing for every voxel j the closest material attenuation value:

$$\mathbf{y}_j^k = \begin{cases} \rho_1, & \mathbf{x}_j^{k-1} < \frac{1}{2}(\rho_1 + \rho_2) \\ \rho_2, & \frac{1}{2}(\rho_1 + \rho_2) \leq \mathbf{x}_j^{k-1} < \frac{1}{2}(\rho_2 + \rho_3) \\ \vdots & \\ \rho_m, & \frac{1}{2}(\rho_{m-1} + \rho_m) \leq \mathbf{x}_j^{k-1} \end{cases} \quad (4.9)$$

$$= \arg \min_{\rho \in \mathcal{R}} \|\mathbf{x}_j^{k-1} - \rho\|_2. \quad (4.10)$$

The second expression is easier to generalize to a higher-dimensional setting, which will be done in Section 4.3.2.

Boundary detection and masking

A set of voxels in the segmentation is selected for a new reconstruction to refine the resulting image. First, the set $B^k \subset \{1, \dots, n\}$ of boundary voxel indices is determined based on the segmentation. Various schemes can be applied for boundary detection. Additionally, a set $U^k \subset \{1, \dots, n\}$ of free voxel indices is

determined, where each voxel is included with a certain probability $1 - \beta$, with $0 \leq \beta \leq 1$. The process of selecting the voxels $U^k \cup B^k$ to be reconstructed and the voxels to be left out is called masking. Note that in the initialization phase all voxels are included in the mask.

Masked ARM reconstructions

The set of free voxel indices $U^k \cup B^k$ are subjected to a new ARM reconstruction. This is done by computing the forward projection of the voxels (\mathbf{y}_j^k) with $j \notin U^k \cup B^k$, and subtracting this from the input data \mathbf{p} to obtain the residual sinogram $\overline{\mathbf{p}}^k$. The subproblem that has to be solved in this phase is:

$$\overline{\mathbf{W}}^k \overline{\mathbf{x}}^k = \overline{\mathbf{p}}^k. \quad (4.11)$$

In Eq. (4.11) matrix $\overline{\mathbf{W}}^k$ is defined by $\overline{\mathbf{W}}^k = (w_{ij})_{j \in U^k \cup B^k}$ and vector $\overline{\mathbf{x}}^k$ to be found has length $|U^k \cup B^k|$. Thus, the system of equations contains the same number of equations as Eq. (4.2) but has fewer unknowns. The system is solved using a fixed number of ARM iterations, taking the values of $(\mathbf{x}_j^{k-1})_{j \in U^k \cup B^k}$ as the starting condition. The complete reconstruction \mathbf{x}^k at the end of iteration k is then formed by merging $\overline{\mathbf{x}}^k$ with \mathbf{y}^k .

Some DART implementations also include a smoothing step at this point. The entire loop is repeated a predefined number of times. After the loop ends, the image is segmented one more time. An example of the DART algorithm on a two-material phantom is given in Figure 4.3. Note that the DART algorithm has many degrees of freedom. This includes the number of ARM iterations in the initialization phase, the number of DART iterations, the number of ARM iterations during these DART iterations, the fixing probability β , and possibly parameters in the smoothing operation. The quality of the reconstructions also depends on the tomographic setup, such as the number of projections and the number of projection angles, and on the complexity of the object, including the number of materials and different attenuation values. Despite the DART algorithm performing well in practice, it is a heuristic method for which no solution guarantees exist [28]. The DART algorithm is also highly modular. Approaches for segmentation, boundary detection, reconstruction (ARM) and possible smoothing can easily be changed without sacrificing the overall structure of the algorithm. For the multi-channel algorithm proposed in this section, the segmentation phase is adapted to using all multi-channel reconstructions as input.

The complexity of the framework depends on the algorithms that are used for reconstruction and segmentation. In this work, we use SIRT as the reconstruction algorithm and the thresholding segmentation as described above. Therefore, in this case, the DART algorithm has a time complexity of $O(Kn(m+l))$. The space complexity of our implementation is $O(ln)$.

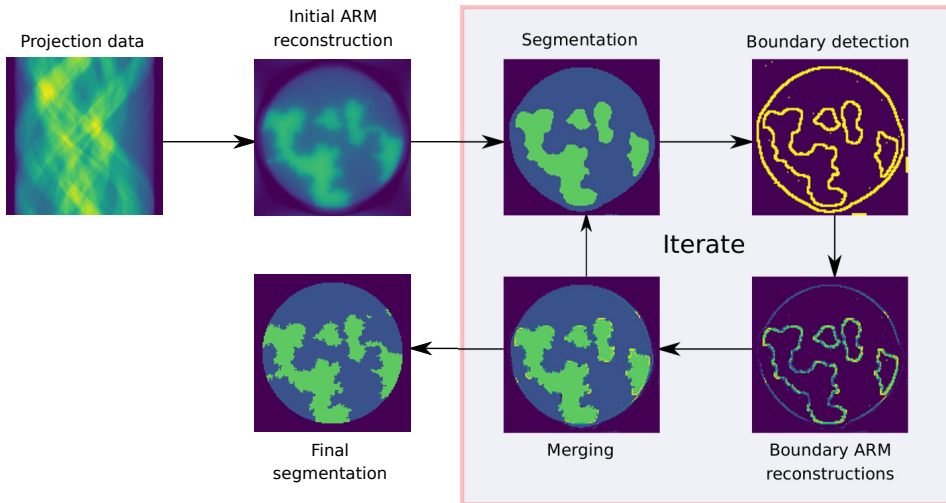


Figure 4.3: An example of DART applied to simulated X-ray data from a two-material phantom.

4.3.2 Multi-channel DART

We now present the *Multi-Channel DART* (MC-DART) algorithm and outline its separate building blocks. Most focus will be on the multi-channel segmentation. Note that labelling single-channel images separately by attenuation values does not work here, since across multiple channels different materials can have the same attenuation. Therefore, there are some slight changes in the other blocks as well due to a new labelling mechanism. The algorithm structure is shown in the flow-chart in Figure 4.4. The pseudocode of MC-DART is given in Algorithm 2.

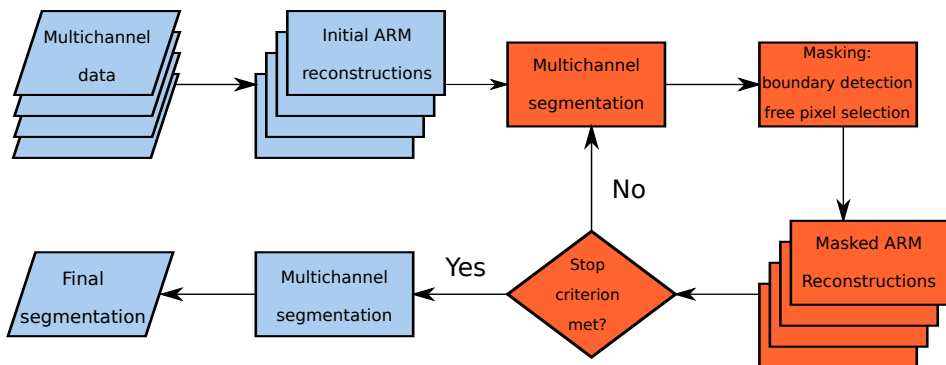


Figure 4.4: Flow chart of the MC-DART algorithm. A stacked number of activities indicate that these are applied at different channels simultaneously.

Algorithm 2 MC-DART

Input: $\mathbf{W}, \mathcal{E}, \mathbf{P}, \mathcal{R}, \mathcal{M}, \mu$

- 1: **for** $c = 1$ **to** C **do**
- 2: $\mathbf{x}_{E_c}^0 \leftarrow \text{Mask-ARM}(\mathbf{W}, \mathbf{p}_{E_c}, \mathbf{1}_n, \mathbf{0}_n)$
- 3: **for** $k = 1$ **to** K **do**
- 4: $\mathbf{y}^k \leftarrow \text{MCSEg}(\mathbf{X}^k, \mathcal{R}, \mathcal{M}, \mu)$
- 5: $\mathbf{M}^k \leftarrow \text{Mask}(\mathbf{y}^k)$
- 6: **for** $c = 1$ **to** C **do**
- 7: $\mathbf{x}_{E_c}^k \leftarrow \text{Mask-ARM}(\mathbf{W}, \mathbf{p}_{E_c}, \mathbf{M}^k, \mathbf{x}_{E_c}^{k-1})$
- 8: **Output:** $\mathbf{X}^K, \text{MCSEg}(\mathbf{X}^K, \mathcal{R}, \mathcal{M}, \mu)$

Initialization

In the multi-channel setting we start out with a vector of projection data \mathbf{P} at various channels and the matrix \mathbf{W} as before. For each channel E_c a reconstruction $\mathbf{x}_{E_c}^0$ is computed using the selected ARM. This results in C initial reconstructions for the MC-DART loop.

Multi-Channel segmentation

Given the reconstructions for all channels, similar to the DART segmentation, the multi-channel segmentation will determine a label image $\mathbf{y}^k \in \mathcal{M}^n$. Let $\boldsymbol{\mu}(s) = (\mu(s, E_1), \dots, \mu(s, E_C)) \in \mathcal{R}_{E_1} \times \mathcal{R}_{E_2} \times \dots \times \mathcal{R}_{E_C}$ be the vector of all attenuation values at each energy for material $s \in \mathcal{M}$, and let $\mathbf{X}^k(\cdot, j) = (\mathbf{x}_{j, E_1}^k, \dots, \mathbf{x}_{j, E_C}^k)$ be the vector of all attenuation values of voxel j at each channel. We compute the segmented image by computing for each voxel $j \in \{1, \dots, n\}$ the label using a basic thresholding scheme:

$$\mathbf{y}_j^k = \arg \min_{s \in \mathcal{M}} \|\mathbf{X}^k(\cdot, j) - \boldsymbol{\mu}(s)\|_2. \quad (4.12)$$

Essentially, this operation selects the material label for which the multi-dimensional difference between the material attenuation and voxel attenuations is smallest.

Masking and boundary detection

The masking works exactly the same as in the single-channel case. Given the segmentation \mathbf{y}^k the masking produces a set $U^k \cup B^k$ of voxel indices to be included in the multi-channel reconstructions.

Multi-Channel reconstructions

In the MC-DART algorithm the reconstructions are handled separately for each energy. Thus, in MC-DART iteration k the ARM is invoked C times to find $\bar{\mathbf{x}}_{E_c}^k$

for each channel c in

$$\overline{\mathbf{W}}^k \overline{\mathbf{x}}_{E_c}^k = \overline{\mathbf{p}}_{E_c}^k. \quad (4.13)$$

The resulting (merged) reconstructions are then given by $\mathbf{X}^k := (\mathbf{x}_{E_1}^k, \dots, \mathbf{x}_{E_C}^k) \in \mathbb{R}^{n \times C}$.

An example of the MC-DART algorithm on a two-material phantom with two channels is given in Figure 4.5. As with DART, the complexity of this framework depends on the reconstruction and segmentation methods that are chosen, as well as the extent of parallelization. If we use SIRT and the multi-channel segmentation method as described above and use a completely sequential implementation, the time complexity of MC-DART is $O(CKn(l+m))$. Because of the dependencies on the methods, we rather speak of a *relative complexity* of MC-DART to DART, which we define as the ratio of the sequential MC-DART complexity to that of DART, irrespective of the subroutines used. This relative time complexity is $O(C)$. The space complexity of this algorithm instance of MC-DART is $O(Cn)$, resulting in a relative space complexity of $O(C)$ as well.

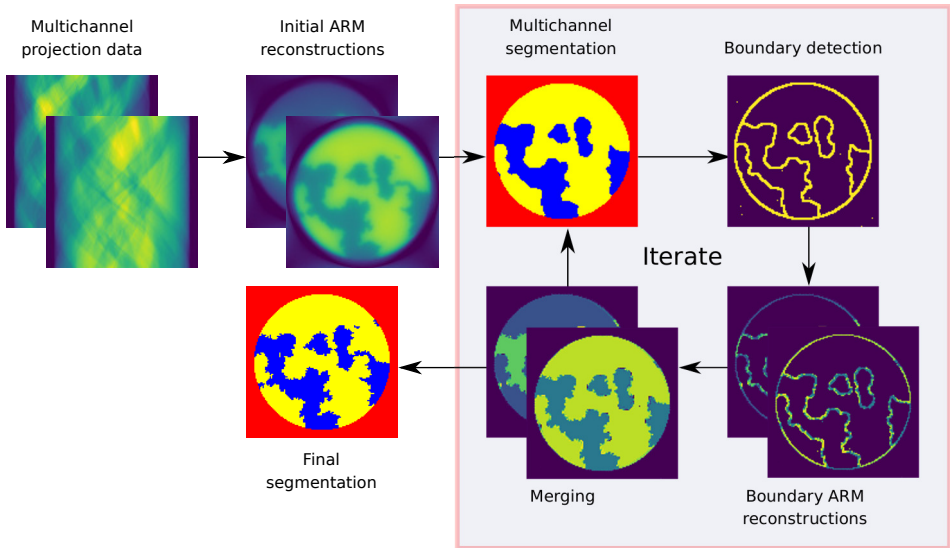


Figure 4.5: An example of MC-DART applied to simulated two-channel X-ray data from a two-material phantom.

4.4 Experiments and results

In this section the performance of the described MC-DART framework in terms of reconstruction and segmentation is presented. A series of experiments have been designed in which the number of channels C and different materials m are varied. For each experiment, multiple random phantoms are created. The size of these two-dimensional phantoms is 128×128 pixels, and each phantom consists of a circular disk containing a random parcellation among m materials in such a way that the total surface is approximately equal for each material. An example of this random phantom is given in Figure 4.1a, where $m = 3$. Given the number of materials and channels, random attenuation spectra are generated by assigning a random number $\mu(s, E_c) \sim \mathcal{U}(0, 1)$ for each channel-material combination, where $s \in \{1, \dots, m\}$ and $E_c \in \{E_1, \dots, E_C\}$. With this way of generating spectra no dependencies between channels are established. Note that in most practical applications such dependencies do exist, as materials all have their own attenuation spectrum. For each phantom, reconstructions are made. The reference values for the tomographic setup and the parameter values of the MC-DART reconstruction algorithm for these reconstructions are summarized in Table 4.1. For multi-channel segmentation the method as described in Section 4.3.2 is used.

Parameter	Reference value
Angles	32 (equidistant)
ARM	SIRT
Start iterations	10
MC-DART iterations K	10
ARM iterations	10
Fix probability β	0.99

Table 4.1: Reference values for the parameters of the tomographic setup and the reconstructions algorithm for all experiments.

We vary the number of channels $C \in \{1, \dots, 10\}$ and materials $m \in \{2, \dots, 10\}$ independently. For each combination, a random phantom \mathbf{y}_{init} is created, after which data \mathbf{P} is generated by applying the forward projection as described in Section 4.2 on the phantom by applying μ and \mathbf{W} on \mathbf{y}_{init} . In all experiments parallel-beam geometries are used and the detector size is 128 pixels. After this, the MC-DART algorithm as described in Section 4.3.2 is applied with $K = 10$ MC-DART iterations. The final segmentation is compared to the original phantom and the pixel error is computed, which is defined as the number of pixels in the final segmentation \mathbf{y}^K that are labelled differently compared to the corresponding pixels in the original phantom \mathbf{y}_{init} . Only the pixels in the inner disk of the phantoms are taken into account. All experiments are repeated for and averaged over 100 runs with different phantoms.

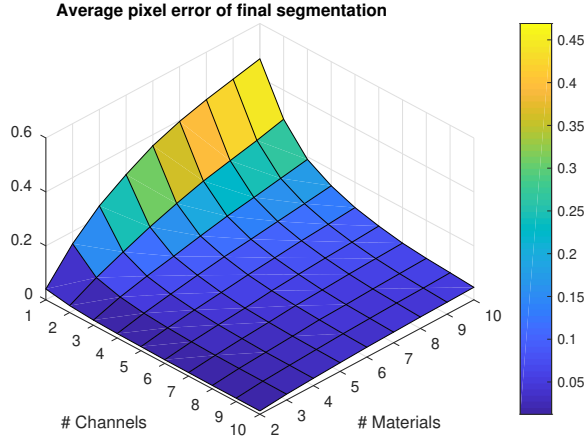


Figure 4.6: Pixel error percentage for different number of material-channel combinations.

The creation of random phantoms is implemented in MATLAB. The remainder of the experiment setup scripts are implemented in Python. The reconstruction algorithms, including the MC-DART algorithm, are implemented in Python, where the ASTRA Toolbox [1, 2] is used to take care of the ARM invocations and forward projections, including the masking in each MC-DART iteration and the creation of matrices \mathbf{W} and $\overline{\mathbf{W}}^k$ based on the geometric properties.

Figure 4.6 shows the percentage of misclassified pixels with respect to the number of pixels in the inner disk. The percentage is lowest when the number of materials is low and the number of channels is high, while the percentage is highest when the number of channels is low and the number of materials is high. Given a number of channels, the percentage seems to scale logarithmically with the number of materials. On the other hand, given a number of materials, the percentage seems to scale exponentially with the number of channels for larger number of materials. Therefore, in this setup, the addition of only a few channels improves the reconstruction quality considerably. Figure 4.7 shows examples of the reconstructions at the corners of the curved plane of Figure 4.6.

We have investigated the effect of changing the parameters that are shown in Table 4.1. The number of starting iterations has no effect on the pixel error percentage curve. For these parameters, we found that increasing the number of MC-DART iterations further than 4 had no significant effect on the reconstructions. This threshold depends on the number of ARM iterations in each MC-DART iteration. Also, the quality of the reconstructions increases only marginally when β is increased. However, the pixel error percentage drops considerably as the number of ARM iterations during an MC-DART iteration increases. Also, when scanning

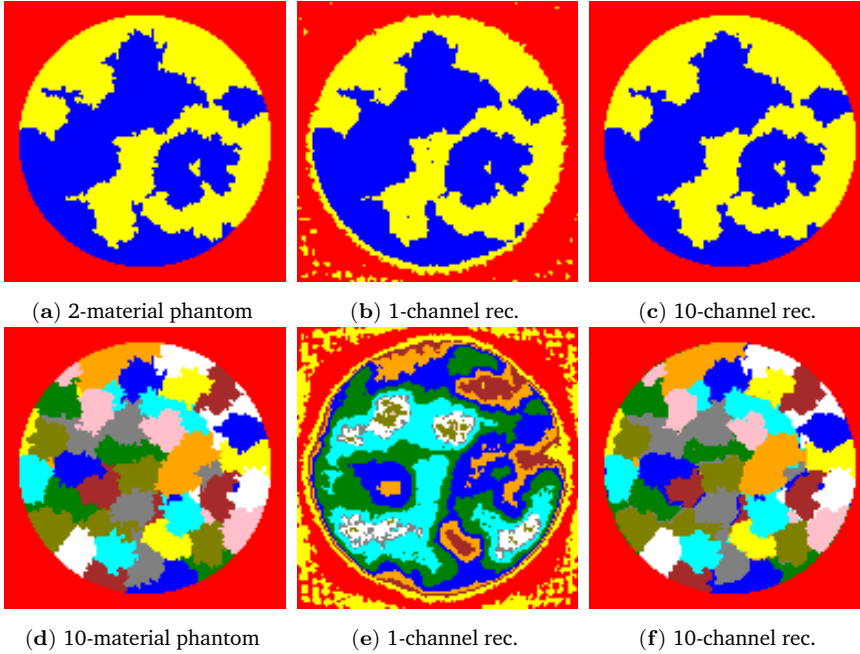


Figure 4.7: Reconstructions for various setups. (a, d) Phantoms used with two and ten materials respectively. (b, e) Reconstructions using one channel. The mislabelled yellow pixels are because the attenuation of the yellow material is very close to zero. (c, f) Reconstructions using ten channels.

data from many angles are available, the reconstruction quality improvements with multiple materials become much better. For only 2 angles, the reconstruction between $C = 1$ and $C = 10$ channels improves from pixel error percentage 27% to 23% for two materials and from 55% to 41% for ten materials. In comparison, for as much as 128 angles the reconstructions between $C = 1$ and $C = 10$ channels improve by from 3% to less than 1% for two materials and from 46% to 4% for ten materials. We conclude that in all these cases the MC-DART algorithm gives better results when more channels are available.

Additionally, apart from the pixel error, we investigate how the number of assigned pixels per material class behaves as the MC-DART reconstruction proceeds. The results are shown in Figure 4.8. A random phantom with four different materials and background is used. The number of channels is set to $C = 10$, and for each channel c and material m a random attenuation value $\mu(s, E_c) \sim \mathcal{U}(0, 1)$ is generated. Then the MC-DART algorithm is applied to this phantom in two different experimental setups. In the first experiment, the number of MC-DART iterations is set to 10 and the number of ARM iterations per MC-DART iteration is set to 10. After each MC-DART iteration, the number of pixels assigned are calculated for each class. During the first four MC-DART iterations the number

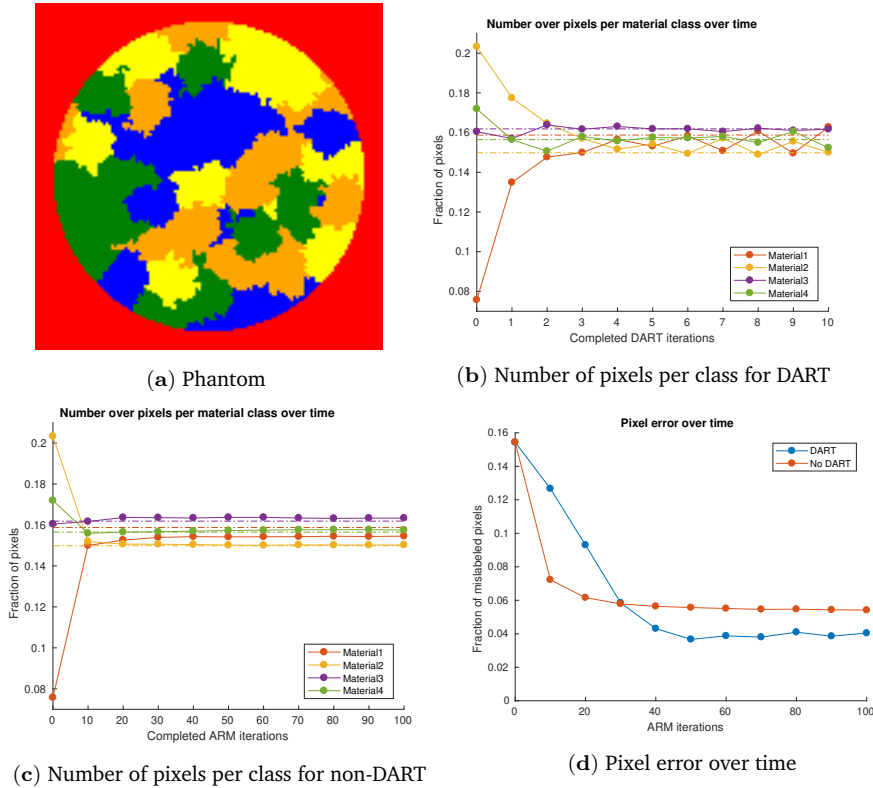


Figure 4.8: Convergence behavior for material classes with $C = 10$ channels. (a) Phantom that has been used with $m = 4$ materials (b) Graph showing the behavior for each material class in a DART routine for this phantom. The number of iterations for the initial reconstruction is set to 2, the number of DART iterations is 10, the number of ARM iterations is set to 10, the number of angles is 8, fixing probability is set to $\beta = 0.99$. The chosen ARM is SIRT. Shown are the number of pixels assigned per class during segmentation after each DART step, with the true value of these indicated by dashed lines. (c) Number of pixel assigned per class over number of ARM iterations. The number of DART iterations is 0, and instead we apply 100 ARM iterations with 2 initial iterations. The results are based on intermediate segmentations after each 10 ARM iterations, but these segmentations are not used in further iterations. Background pixels are excluded from the results (d) Pixel error over number of ARM iterations for both approaches.

of assigned pixels is converging towards their real values. After this, the graphs enter an oscillatory phase in which for each class the number of assigned pixels alternates between two values whose average is not necessarily the real number of pixels for that class. For comparison, in the second experiment the same setup is used, but without using MC-DART iterations and applying the same ARM for 100 iterations instead. In this way the ARM is effectively invoked equally often. After each 10 iterations a segmentation is made based on the current reconstruction and the pixels per class are measured, but no new forward projections are calculated from these segmentations and used in subsequent ARM iterations. In this case the number of pixels converges much more quickly for each class. Also, there is no oscillatory phase and the number of pixels are just as close to their true values as with the DART approach. However, plotting the total pixel error over time reveals that the pixel error in the non-DART case is higher. The pixel error for the MC-DART case needs more time to stabilize to its oscillatory phase, but the values are eventually lower than in the non-MC-DART case.

4.5 Discussion

This chapter presents the first steps to implement a multi-channel reconstruction technique using multi-channel segmentation. Currently, there are no standard approaches for the discrete multi-channel problem presented in Section 4.2. We propose a framework in which reconstruction and segmentation techniques can be exchanged. The modules in the framework can be adjusted to the problem to be solved. For instance, segmentation can be performed with neural network based methods. The proposed method is not aimed at optimizing reconstructions with state-of-the-art ARMs or segmentation techniques but at presenting a framework to work with multi-channel data. If more data from different channels are available, this implementation outperforms DART but it does not mean that the problem is optimally solved. To further develop this technique and transfer it to real-world settings, real-data properties should be taken into account. These properties include the correlation of attenuation values between channels and noise contained in the projection data. In our study we only make use of the multi-channel data during segmentation. Another approach could be to use the multi-channel data during reconstruction, modelling the reconstruction problem as a large inverse problem where the unknowns are the material concentrations in each pixel (e.g. see [143, 275]). However, solving this problem is much more involved and the MC-DART framework presented in this chapter provides a simple but effective alternative of separating materials using multi-channel data.

4.6 Conclusions

A new class of algorithms for solving discrete multi-channel reconstruction problems has been proposed. This framework uses the strength of DART regarding dealing with limited data in a multi-channel setting by using a multi-channel segmentation method. The experiments have shown that combining information from different channels by a multi-channel segmentation method increases the reconstruction quality compared to the single-channel DART algorithm. Therefore, we conclude that the MC-DART framework is a promising approach for dealing with multi-channel data.

Code and data availability

The source code to reproduce all results in this chapter are available on <https://github.com/mzeegers/DRCNN> [311]. These Python scripts make use of the ASTRA toolbox [1, 2]. The data configurations generated for this chapter are available in the MC-DART source code repository.

