Intersection-based slice motion estimation for fetal brain imaging

Chloé Mercier, Sylvain Faisan, Alexandre Pron, Nadine Girard, Guillaume Auzias, Thierry Chonavel, François Rousseau

Intersection-based slice motion estimation for fetal brain imaging

Chloe Mercier[a,*], Sylvain Faisan[b], Alexandre Pron[c], Nadine Girard[c], Guillaume Auzias[c], Thierry Chonavel[a], François Rousseau[d]

[a]*IMT Atlantique, Lab-STICC UMR CNRS 6285, Brest, France*
[b]*ICube Laboratory, University of Strasbourg, CNRS, Strasbourg, France*
[c]*Aix-Marseille Université, CNRS, Institut de Neurosciences de la Timone, Marseille, France*
[d]*IMT Atlantique, LaTIM U1101 INSERM, Brest, France*

Abstract

Fetal MRI offers a broad spectrum of applications, including the investigation of fetal brain development and facilitation of early diagnosis. However, image quality is often compromised by motion artifacts arising from both maternal and fetal movement. To mitigate these artifacts, fetal MRI typically employs ultrafast acquisition sequences. This results in the acquisition of three (or more) orthogonal stacks along different spatial axes. Nonetheless, inter-slice motion can still occur. If left uncorrected, such motion can introduce artifacts in the reconstructed 3D volume. Existing motion-correction approaches often rely on a two-step iterative process involving registration followed by reconstruction. They tend to detect and remove a large number of misaligned slices, resulting in poor reconstruction quality. This paper proposes a novel reconstruction-independent method for motion correction. Our approach benefits from the intersection of orthogonal slices and estimates motion for each slice by minimizing the difference between the intensity profiles along their intersections. To address potential misalignments, we present an innovative machine learning-based classifier for identifying misaligned slices. The

*Corresponding author

Email addresses: chloe.mercier@imt-atlantique.fr (Chloe Mercier), faisan@unistra.fr (Sylvain Faisan), alexandre.pron@gmail.com (Alexandre Pron), Nadine.GIRARD@ap-hm.fr (Nadine Girard), guillaume.auzias@univ-amu.fr (Guillaume Auzias), thierry.chonavel@imt-atlantique.fr (Thierry Chonavel), francois.rousseau@imt-atlantique.fr (François Rousseau)

parameters of theses slices are then corrected using a multistart optimization approach. Quantitative evaluation on simulated datasets demonstrates very low registration errors. Qualitative analysis on real data further highlights the effectiveness of our approach compared to state-of-the-art methods.

Keywords: Fetal Brain, Registration, Magnetic Resonance Imaging

1. Introduction

Magnetic Resonance Imaging (MRI) has been used to study fetal brain development since the 1980s (Huisman et al. (2002)). However, motion from both the mother and the fetus remains a significant challenge, limiting the quality and interpretability of the acquired images. In the context of antenatal imaging, the fetal brain is typically represented as three stacks of 2D slices, each acquired along one of the three orthogonal axes to provide radiologists with a 3D "vision" of the brain. Fig. 1 illustrates these stacks, acquired in the axial (first row), coronal (second row), and sagittal (third row) directions. Notably, resolution is highest in the acquisition direction and lower in the orthogonal directions.

The acquisition time of a slice is generally sufficiently short (less than 1 second) to "freeze" the motion. However, motion causes slice misalignments within the stack, resulting in geometric distortions. These distortions prevent the 2D slices from accurately representing the 3D geometry of the brain. Thus, it is necessary to estimate the motion retrospectively in order to reconstruct a 3D image of the fetal brain (Studholme and Rousseau (2014)). Applications of fetal MRI include segmenting the fetal brain (Uus et al. (2023)), building atlases (Gholipour et al. (2017)), biometric measurement (Ciceri et al. (2023)) and understanding normal development (Saleem (2013)).

Retrospective motion estimation is often associated with the reconstruction of a high-resolution 3D volume. In the Slice-to-Volume Reconstruction (SVR) method developed in Rousseau et al. (2005, 2006) and Jiang et al. (2007), the authors proposed to iteratively estimate the motion of individual slices by registering them to a high-resolution reference volume. During the first iteration, the reference volume corresponds to the reconstruction of one of the acquired stacks. In subsequent iterations, it corresponds to the volume reconstructed from the registered slices. The SVR reconstruction framework has been enhanced by employing a Maximum A Posteriori (MAP) Super-

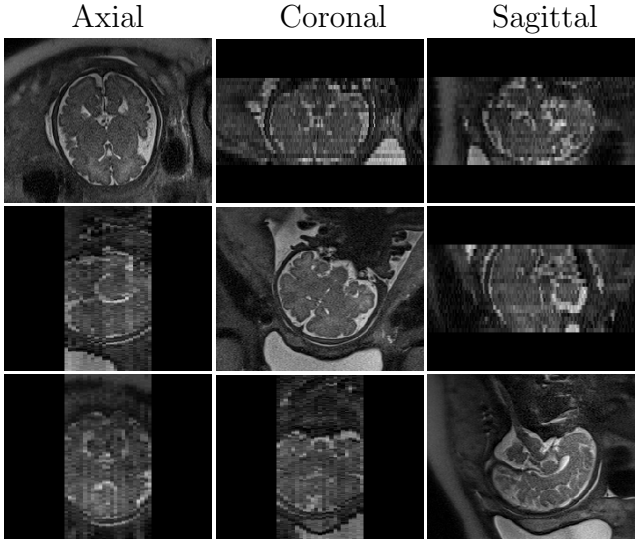


Figure 1: Triplanar visualization of three fetal brain stacks, each acquired along one of the three orthogonal directions (Axial, Coronal, and Sagittal), with each stack presented from top to bottom.

Resolution approach, and by replacing the L2 norm with the more robust norms (Huber and Total-Variation for instance) Gholipour et al. (2010); **?** . Kuklisova-Murgasova et al. (2012) and Ebner et al. (2020) proposed a complete framework for SVR in fetal MRI. Specifically, Kuklisova-Murgasova et al. (2012) incorporated bias field estimation and outlier management during the 3D reconstruction process, while Ebner et al. (2020) included brain masking, bias field correction, and outlier detection. However, a major limitation of the aforementioned methods, compared to more recent deep learning approaches, is their inability to effectively correct large motion distortions and their high computational cost. The implementation of an algorithm can, however, have a significant impact on execution time. For instance, Kainz et al. (2015) optimized the work of Kuklisova-Murgasova et al. (2012) by proposing a faster implementation using a GPU, which significantly reduced computation time.

Recently, the development of deep learning architectures has facilitated the emergence of new methods to improve the robustness of fetal motion estimation, specifically addressing the challenge of large motion. To tackle this issue, a mapping from 2D image slices to a 3D canonical atlas space can be learned using either Transformers Xu et al. (2022) or Convolutional

Neural Networks (CNNs) Hou et al. (2018). In Shi et al. (2022), the motion estimation step relies on an attention-based feature affinity fusion method that learns the correspondence between features extracted from 2D slices and those extracted from a reference volume. As in SVR, the motion correction and reconstruction steps are repeated iteratively in the deep learning-based methods Hou et al. (2018); Shi et al. (2022); Xu et al. (2023); Lv et al. (2024) (since all motion estimation algorithms require a 3D reference image).

In all the aforementioned methods, whether based on deep learning or not, motion estimation is closely tied to the reconstruction process. As a result, errors in the reconstructed volume can lead to inaccuracies in motion estimation. To address this limitation, Kim et al. (2009) proposed a method that corrects motion between slices without requiring a reference 3D volume. This approach is based on the principle that the intensity profiles along the intersection lines of slices from orthogonal stacks should be similar if they are perfectly aligned. In Ma et al. (2024), the motion correction network is based on a multi-scale feature fusion model, where features are extracted solely from 2D slices. Consequently, this approach also overcomes the limitation of requiring a reference 3D volume for motion correction.

In addition, fetal brain MRI reconstruction requires careful handling of outliers, which can include misaligned slices or low-quality slices since they can introduce artifacts into the reconstructed 3D volume. Current methods address this challenge in two main ways: outlier removal through iterative thresholding techniques (Ebner et al. (2020); Kim et al. (2009)) and minimizing the influence of outliers within the reconstruction process (Kuklisova-Murgasova et al. (2012); Gholipour et al. (2010); Xu et al. (2023)). In Ebner et al. (2020), a similarity metric is computed between each slice and the reconstructed volume. Slices with a similarity value below a predefined threshold are identified as outliers and excluded. A similar strategy is used by Ma et al. (2024). In Kim et al. (2009), registration is performed using a loss function that compares the intensity profiles along the intersection lines of slices. From this loss, a Mean Squared Error (MSE) is derived for each slice. If the MSE of a slice exceeds a threshold, the slice is considered an outlier. Xu et al. (2023) used Monte Carlo dropout during inference to provide a confidence score for each slice, which is used in outlier rejection. Gholipour et al. (2010) proposed replacing the L2 norm with the Huber norm in the cost function during reconstruction, improving robustness to outliers by reducing their weight in the final reconstruction. Robust statistics are also used in Kuklisova-Murgasova et al. (2012) during reconstruction.

Note that the proposed state-of-the-art approach may lose significant information by treating misaligned slices as corrupted and not considering their potential for inclusion in the reconstruction after better alignment.

1.1. Contribution

This paper introduces ROSI (Registration based on Orthogonal Slices Intersection), a novel method for fetal MRI motion correction that operates independently of the reconstruction process. Inspired by Kim et al. (2009), ROSI estimates slice motion by minimizing intensity differences along the intersection lines between slices from orthogonal stacks. This paper provides a detailed description of the intersection calculation and cost function, facilitating the reproducibility of the method. The code is available at: https://github.com/rousseau/pyrecon. Moreover, this work proposes training a machine learning classifier to identify misaligned slices with greater precision than traditional threshold-based methods. Our outlier detection strategy addresses a key challenge in fetal MRI motion correction: balancing outlier removal with information preservation. This work builds upon our previous research (Mercier et al. (2023)) and introduces a multistart strategy for realigning misaligned slices, enabling the correction and reintegration of useful information while removing truly corrupted slices prior to reconstruction.

To summarize, the contributions of this work are: 1) a motion correction method for fetal brain MRI based on the intersection of orthogonal slices, 2) a detailed description of the intersection calculation and cost function, 3) a machine learning classifier for detecting misaligned slices, and 4) a multistart strategy for the correction of misaligned slices.

The manuscript is structured as follows: Sec. 2 introduces the proposed method, with a general overview provided in Sec.2.1. Sec. 2.2 describes the modeling of the transformation matrix to be optimized for each slice, while Sec. 2.3 presents the loss function, and Sec. 2.4 details its optimization. Sec. 2.5 outlines the supervised learning approach for detecting misaligned slices, and Sec. 2.6 introduces the multistart approach for correcting misalignments. Sec. 3 covers the data and evaluation procedures. Finally, Sec. 4 compares the results obtained with state-of-the-art methods, using both simulated and real data.

2. Method

2.1. Description of the ROSI algorithm

The proposed method is based on the principle that slices originating from orthogonal stacks (e.g., axial, coronal, and sagittal views) should have consistent intensity values at their intersection points when perfectly aligned. This principle is illustrated in Fig. 2: misaligned slices (a) produce differing intensity profiles along their intersection (b), whereas perfectly aligned slices (c) result in consistent profiles.

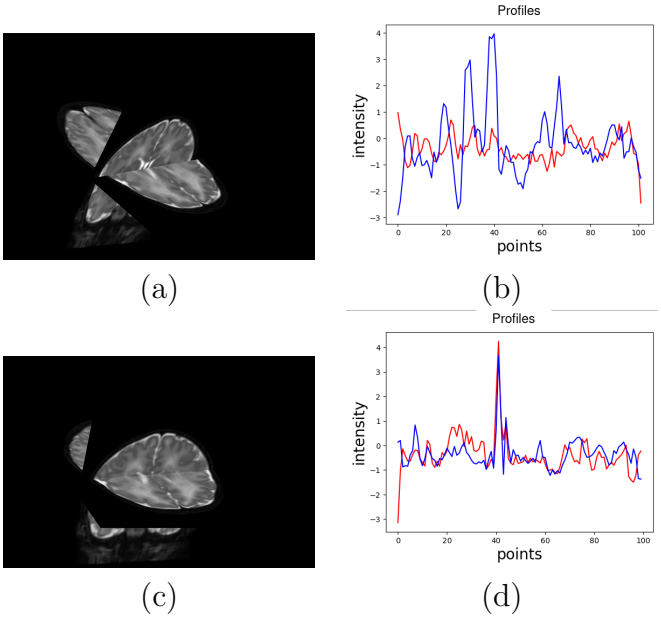


Figure 2: Visualization of the intersection between two orthogonal slices and their corresponding profiles. Two misaligned slices (a) have distinct profiles along their intersection (b) whereas two well-registered slices (c) display similar profiles (d).

A summary of the proposed algorithm is illustrated in Fig. 3. The first step (Fig. 3-1) involves aligning slices from orthogonal stacks by minimizing the loss described in Sec. 2.3 according to the parameters described in Sec. 2.2. However, the optimization may get trapped in local minima, leading to suboptimal alignment. To address this problem, the second step (Fig. 3-2) involves using a classifier to detect potentially misaligned slices (Sec. 2.5), followed by a multi-start procedure (Sec. 2.6) to improve the estimation of their positions. In the final step, an additional optimization stage is

performed, focusing only on slices that the classifier has identified as well-registered. This ensures that misaligned slices do not impact the alignment of the correctly registered slices.

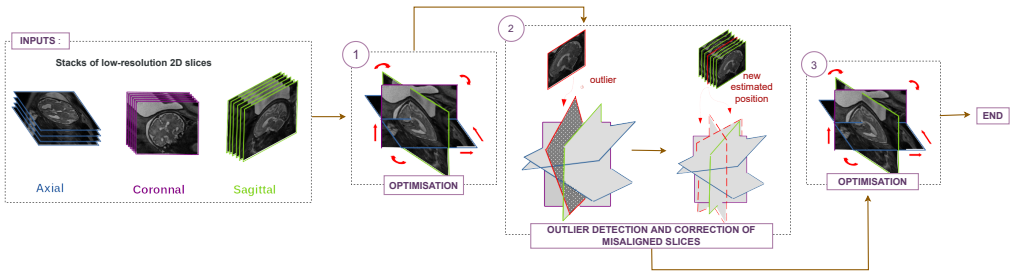


Figure 3: Overview of the proposed algorithm. The input consists of multiple orthogonal stacks, each containing slices. The algorithm proceeds in three stages: (1) alignment of slices by minimizing the loss function, (2) detection and correction of potentially misaligned slices using a classifier and multi-start optimization, and (3) final optimization applied only to well-registered slices to ensure precise alignment.

2.2. Motion Representation

To achieve optimal alignment, ROSI estimates a transformation matrix, $\mathbf{M}_k$, for each slice k using homogeneous coordinates, defined as:

$$\mathbf{M}_k = \mathbf{T}(\mathbf{c}_k)\mathbf{M}(\boldsymbol{\theta}_k, \mathbf{t}_k)\mathbf{T}(-\mathbf{c}_k)\mathbf{R}_{f(k)}\mathbf{R}_{k,2d\to 3d}. \tag{1}$$

$\mathbf{M}_k$ is the composition of five transformations: $\mathbf{R}_{k,2d\to 3d}$ is a rigid transformation that maps 2D coordinates of slice k to the corresponding 3D voxel coordinates within image $I_{f(k)}$ ($f(k)$ denotes the index of the stack to which slice k belongs, while $I_f(k)$ refers to the stack itself (comprising all its slices)). $\mathbf{R}_{f(k)}$ converts these voxel coordinates into the world reference system (millimeters). The objective is to estimate the matrix $\mathbf{M}(\boldsymbol{\theta}_k, \mathbf{t}_k)$, which encodes the rigid transformation for each slice k. This matrix incorporates six parameters: three for rotation angles ($\boldsymbol{\theta}_k$) using the Euler representation (Pio (1966)) and three for translation ($\mathbf{t}_k$) in millimeters: $\mathbf{M}(\boldsymbol{\theta}_k, \mathbf{t}_k) = \begin{bmatrix} \mathbf{R}_k & \mathbf{t}_k \\ \mathbf{0}^{[1\times 3]} & 1 \end{bmatrix}$. To achieve rotations around a specific point with coordinates $\mathbf{c}_k$, a translation of $-\mathbf{c}_k$ is applied (denoted as $\mathbf{T}(-\mathbf{c}_k)$). Next, the transformation $\mathbf{M}(\boldsymbol{\theta}_k, \mathbf{t}_k)$ is applied. Finally, the inverse translation $\mathbf{T}(\mathbf{c}_k)$ is performed. Note that $\mathbf{c}_k$ is equal to $\mathbf{R}_{f(k)}\mathbf{R}_{k,2d\to 3d}\mathbf{g}_k$, where $\mathbf{g}_k$

represents the 2-D coordinates of the centroid of the mask associated with slice k (brain masks have been automatically segmented in the images Ebner et al. (2018), associating each k with a corresponding mask denoted as m_k). This approach ensures that the rotation is estimated at the center of the region of interest.

2.3. Cost Function

The registration criterion for aligning images relies on the principle that well-registered slices should have similar intensity profiles along their intersection. To achieve this, the loss function computes the intensity profiles for all pairs of slices k and k' from orthogonal stacks along their intersections.

For a given k, k', this computation involves four main steps, detailed in Appendix A:

1. Deriving the plane equation of slice k in the world coordinate system (Appendix A.1)
2. Deriving the equation of the intersection line of the planes of slices k and k' in the world coordinate system (Appendix A.2)
3. Deriving the equations of the intersection line of the planes of slices k and k' in the 2D coordinate system of slice k and k' (Appendix A.3)
4. Generating corresponding points along the intersection line (Appendix A.4)

As a result of these steps, for two orthogonal slices k and k', we obtain a set of points in slice k, denoted as $\mathcal{V} = \{v_1, v_2, \ldots, v_{L_{k,k'}}\}$, along with their corresponding points in slice k': $\mathcal{V}' = \{v'_1, v'_2, \ldots, v'_{L_{k,k'}}\}$, such that:

$$\mathbf{M}_k(v_i) = \mathbf{M}_{k'}(v'_i).$$

Assuming that, in the case of accurate slice registration, the intensity of slice k at point v_i (denoted as $s_k(v_i)$) to match the intensity of slice k' at point v'_i (denoted $s_{k'}(v'_i)$), we propose a cost function based on the following quantities:

$$S^2(k, k') = \sum_{i=1}^{L_{k,k'}} (s_k(v_i) - s_{k'}(v'_i))^2 \mathbb{1}_{m_k(v_i)=1 \text{ or } m_{k'}(v'_i)=1}, \tag{2}$$

$$N(k, k') = \sum_{i=1}^{L_{k,k'}} \mathbb{1}_{m_k(v_i)=1 \text{ or } m_{k'}(v'_i)=1}, \tag{3}$$

where $\mathbb{1}_E$ is an indicator function that equals 1 if E is true and 0 otherwise, and m_k is the mask associated with slice k. It identifies the regions of interest within the slice, where $m_k(v_i) = 1$ indicates that the point v_i belongs to the regions of interest in slice k, and $m_k(v_i) = 0$ otherwise.

Moreover, as explained in Appendix A, we ensure that the set of points $\mathcal{V}$ (resp. $\mathcal{V}'$) sample all structures of interest that lie along the portion of the intersection line within slice k (resp. k'). Consequently, if the slices are misaligned, v_i (or v_i') may not fall within the support of slice k (or slice k'). To address this issue, we assume that the slices have a value of zero outside their respective supports. Finally, the intensity values $s_k(v_i)$ and $s_{k'}(v_i')$ are obtained by linear interpolation whereas $m_k(v_i)$ and $m_{k'}(v_i')$ are obtained by nearest neighbour interpolation.

The same steps are repeated for each possible pair of slices k, k'. The loss function to be minimized writes:

$$\frac{\sum_{k,k' \ ; \ k>k'} S^2(k,k')}{\sum_{k,k' \ ; \ k>k'} N(k,k')}. \tag{4}$$

2.4. Optimization

The criterion is optimized using an alternating block optimization method, where parameters associated with slice k (Θ_k and t_k) are updated while parameters associated with other slices remain unchanged. The computation of the criterion in Eq. (4) benefits from the fact that, for each update, many terms in the loss function do not need to be recalculated. Specifically, when optimizing slice k_{opt}, the values of $S^2(k,k')$ and $N(k,k')$ do not need to be recalculated if $k \neq k_{\text{opt}}$ and $k' \neq k_{\text{opt}}$, since the parameters for slices other than k_{opt} remain fixed: only the terms involving k_{opt} and its intersections with other slices need to be computed.

The parameter optimization for a slice is performed using the Nelder-Mead method, which is a direct-search method using a polytope, called a simplex, to find local solutions. The simplex is composed of $n+1$ vertexes, in n dimensions, and undergoes four different operations: reflection, expansion, contraction and shrinking to reach the local minima (Singer and Nelder (2009)). Two hyperparameters control the Nelder-Mead approach: the initial simplex size (denoted as ds) as well as the final simplex size (denoted as fs) that serves as the stopping criterion for the Nelder-Mead method.

The algorithm (Alg. 1 in Appendix B) proceeds iteratively. In each iteration, the algorithm updates the parameters of each slice sequentially.

If, during an iteration, the changes in the parameters of a slice fall below a defined threshold th, this indicates local convergence for that slice, and it is no longer optimized in subsequent iterations. The algorithm repeats this process until all slices have reached local convergence. If the slices achieve local convergence at different iterations, the optimization process restarts for all slices using the same hyperparameters. However, if all slices achieve local convergence within the same iteration (global convergence), the optimization also continues, but with reduced hyperparameters to allow for finer adjustments. The simplex sizes ds and fs, as well as th, are reduced three times throughout the optimization process. This gradual reduction allows for more precise refinement in the later stages of the optimization, ensuring that the algorithm progresses from broad exploration to fine-tuning. The algorithm terminates when global convergence is reached with the final, reduced hyperparameters. Initially, we choose $ds = 4$, $fs = 0.25$ and $th = 2$ at the first iteration. We conducted experiments using different values for ds, fs, and th, and observed that their choice does not significantly impact the final result across a wide range of values.

2.5. Supervised learning approach for misaligned slice detection

We propose a supervised learning strategy to detect misaligned slices. A random forest classifier is selected for its demonstrated effectiveness in outlier detection tasks, outperforming approaches like Support Vector Machines (SVMs) (Klapwijk et al. (2019)). To create the training and testing datasets, we simulate 12 acquisitions with varying levels of motion (medium, large, and very large motion), as described in Sec. 3.1. These simulated datasets are registered using the alternating block optimization procedure (Alg. 1). For each slice, it is necessary to determine whether the slice is properly registered. To achieve this, we compute a metric called the mean target registration error (TRE) for each slice k. This metric represents the average distance (in millimeters, within the world coordinate system) between the points of slice k and the corresponding points from other slices when perfectly aligned. Details of the TRE computation are provided in Sec. 3.2. Since poorly registered slices can skew the mean TRE of other slices, we implement an iterative approach to robustly identify misaligned slices. First, the mean TRE is calculated for all slices. The slice with the highest mean TRE is identified, and if its value exceeds 1.5 mm, it is classified as poorly registered and excluded from the next computation. The mean TRE is then recalculated for the remaining slices. This process is repeated

iteratively until all remaining slices have a mean TRE below 1.5 mm. These slices are considered well-registered. The dataset is randomly divided into two subsets, with 50% of the slices allocated for training and 50% for testing.

We propose to use three features to detect misaligned slices. The first feature leverages intensity information from the images, while the other two exploit the correspondence between structures of interest (e.g., brain mask segmentation).

For the first feature, we expect the term

$$\text{avg}_{k'} \left(\frac{S^2(k,k')}{N(k,k')} \right) \tag{5}$$

to be small when slice k is well-registered, and large otherwise (avg denotes the averaging operator). However, this formulation has two major limitations:

1. The presence of misaligned slices k' can significantly increase the term of Eq. 5, even when slice k itself is well-registered. Consequently, poorly registered slices k' may inflate the term, increasing the risk that k is incorrectly identified as misaligned. To address this issue, we replace the average operator with the median operator, which is less sensitive to outliers.
2. Noise can act as a confounding factor; even if slice k is well-registered, high noise levels may inflate the feature value. Consequently, we propose to normalize the term according to the noise that corrupts the data.

We propose finally the following feature, denoted $F_1(k)$:

$$F_1(k) = \text{med}_{k'} \left(\frac{1}{\sigma^2_{f(k)} + \sigma^2_{f(k')}} \frac{S^2(k,k')}{N(k,k')} \right),$$

where $\sigma_{f(k)}$ is the standard deviation of the noise corrupting the image $I_{f(k)}$. The noise standard deviation $\sigma_{f(k)}$ is estimated using the method proposed in Immerkaer (1996), applied within the region defined by the brain mask.

The second feature evaluates the alignment of the masks and is obtained by calculating a median Dice index for each slice k. In segmentation tasks, the Dice index between two sets X and Y is defined as $2|X \cap Y|/(|X| + |Y|)$, where $|X|$ represents the cardinality of X. Therefore, $F_2(k)$ is defined as :

$$F_2(k) = med_{k'} \left(\frac{2M(k,k')}{P(k,k') + Q(k,k')} \right) \tag{6}$$

where

$$M(k,k') = \sum_{i=1}^{L_{k,k'}} \mathbb{1}_{m_k(v_i)=1 \text{ and } m_{k'}(v'_i)=1}, \tag{7}$$

and where

$$P(k,k') = \sum_{i=1}^{L_{k,k'}} \mathbb{1}_{m_k(v_i)=1}, Q(k,k') = \sum_{i=1}^{L_{k,k'}} \mathbb{1}_{m_{k'}(v'_i)=1}. \tag{8}$$

The Dice index can be influenced by the volume of the structures of interest, making it less reliable for slices near the edge of the brain, where the masked volume is smaller. To address this limitation, a third feature $F_3(k)$ is introduced and computed as follows:

$$F_3(k) = med_{k'} \left(2M(k,k') - P(k,k') - Q(k,k') \right). \tag{9}$$

2.6. Multistart approach for misaligned slices

The registration process relies on an optimization algorithm that may only converge to local optima. We propose a multistart strategy to escape these local minima. As before, the parameters associated with a slice k are updated using the Nelder-Mead algorithm, while keeping the parameters of other slices fixed. However, there are four major differences.

First, to save time, we update only the parameters of potentially misaligned slices. The classifier is used to estimate the probability p_k of each slice k being misaligned. We then focus on updating the slices in $\mathcal{M}$, the set of indices corresponding to slices identified as misaligned (i.e., where $p_k > 0.2$).

Next, during the update of slice k, the optimization algorithm is run multiple times with different initializations, and the best transformation obtained is retained. The different initializations are obtained by taking advantage of well-aligned slices that are nearby k (Appendix C). To achieve this, we construct the set $\mathcal{A}$, which contains the indices of slices identified as likely well-aligned (i.e., where $p_k < 0.5$).

Next, when updating the parameters of a slice, the misalignment of other slices impacts the derived cost function, which in turn affects the quality of the registration. Therefore, we aim to reduce the influence of misaligned

slices on the loss. This is done as follows: when updating the parameters of a slice k of $\mathcal{M}$, the loss to be optimized considers only the slices in $\mathcal{A}$ along with k. This ensures that potentially misaligned slices do not influence the estimation related to slice k.

Finally, we observed that adding a term to the loss function that encourages mask overlap can sometimes help escape local minima more effectively. Thus, we attempt to update the parameters of misaligned slices either by using the original loss or by using the following modified loss (note that the original loss corresponds to the case $w = 0$):

$$\frac{\sum_{k,k' \ ; \ k>k'} S^2(k,k')}{\sum_{k,k' \ ; \ k>k'} N(k,k')} + \omega \frac{\sum_{k,k';k>k'} 2M(k,k')}{V}, \tag{10}$$

where V is the sum of the mask volumes ($\sum_{k,k';k>k'} \sum_v \mathbb{1}_{m_k}(v) = 1$), and $M(k,k')$ is defined in Eq. 7.

The multistart procedure is detailed in Algorithm 2 (Appendix D).

3. Data and Evaluation Procedure

3.1. Data

To address the lack of ground truth, we generated synthetic data from 3D high-resolution images available in the dHCP database (Edwards et al. (2022)). We simulated realistic fetal MRI scenarios, including three orthogonal stacks (axial, sagittal, and coronal) with clinically relevant parameters: a slice thickness of 3 mm and an in-plane resolution of 0.5 x 0.5 mm. To thoroughly evaluate the robustness of our algorithm, we generated data with four distinct levels of motion (Fig. 4). For each slice, motion parameters were randomly assigned, with the rotation (in degrees) and translation (in mm) drawn independently from a uniform distribution. The four levels of motion were: low ($\boldsymbol{\theta}_k \in [-1,1]$, $\mathbf{t}_k \in [-1,1]$), medium ($\boldsymbol{\theta}_k \in [-3,3]$, $\mathbf{t}_k \in [-3,3]$), large ($\boldsymbol{\theta}_k \in [-5,5]$, $\mathbf{t}_k \in [-5,5]$), and extra-large ($\boldsymbol{\theta}_k \in [-8,8]$, $\mathbf{t}_k \in [-8,8]$).

In addition to the simulated data, acquisition from 49 fetuses aged between 24th and 37th weeks were included in this study. The retrospective use of MRI data acquired during clinical routine at the Timone Hospital was approved by the Aix-Marseille University ethics committee (ref 2022-04-14-003). All data were acquired in ultra-fast spin echo (HASTE) on three different Siemens scanners (Skyra and Spectra (3T) n=20 and SymphonyTim

(1.5T) n=29). The data were pre-processed to correct for the bias caused by the inhomogeneity of the magnetic field.

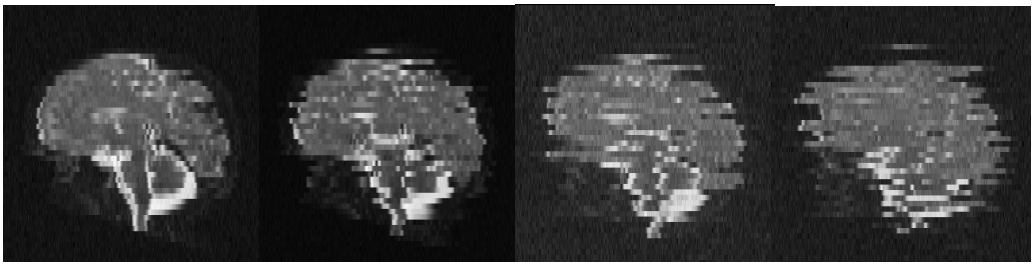


Figure 4: Example of simulated data for the axial orientation. The data is generated with varying levels of motion and visualized in the sagittal view.

3.2. Metrics for assessing registration performance

A first approach to assess the quality of image registration is to use the registered slices to reconstruct a 3D image. In order to reconstruct the data registered with ROSI, two different approaches have been employed: NiftyMIC (Ebner et al. (2020)) and NeSVoR (Xu et al. (2023)).

For real data, quantitative evaluation is not possible due to the absence of ground truth. Instead, following the methodology proposed in Sanchez et al. (2023), the visual quality of the reconstructed images was assessed by two independent raters using the approach described in Sec. 3.4.

For synthetic data, the reconstructed image can be directly compared to the ground truth reference image. The closer the two images, the better the registration quality. To quantify the similarity between the two images, we use the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM). In addition, a more direct approach to assess the quality of the image registration process involves using the mean Target Registration Error (TRE). This metric represents the average distance, in millimeters, between corresponding points in the world coordinate space. To compute the mean TRE for a given slice k, we consider all intersecting slices of k, similar to the procedure used for the loss function. Let slice k' intersect with slice k. The corresponding points in slices k and k' are obtained using the same method as in the cost function computation. This method generates a set of points in slice k, denoted as $\mathcal{V} = \{v_1, v_2, \ldots, v_{L_{k,k'}}\}$, and their corresponding points in slice k', denoted as $\mathcal{V}' = \{v'_1, v'_2, \ldots, v'_{L_{k,k'}}\}$, such that $\mathbf{M}_k(v_i) = \mathbf{M}_{k'}(v'_i)$. The

ground truth transformations, $\mathbf{M}_k^{GT}$ and $\mathbf{M}_{k'}^{GT}$, are used so that $\mathbf{M}_k^{GT}(v_i) = \mathbf{M}_{k'}^{GT}(v_i')$. The mean TRE for slice k can then be computed as follows:

$$TRE_k = \frac{1}{N_k} \sum_{k'} \sum_{i=1}^{L_{k,k'}} ||\mathbf{M}_k(v_i) - \mathbf{M}_{k'}(v_i')|| \mathbb{1}_{m_k(v_i)=1 \text{ or } m_{k'}(v_i')=1}, \text{ with} \quad (11)$$

$$N_k = \sum_{k'} \sum_{i=1}^{L_{k,k'}} \mathbb{1}_{m_k(v_i)=1 \text{ or } m_{k'}(v_i')=1}, \quad (12)$$

Note that the summation $\sum_{i=1}^{L_{k,k'}}$ corresponds to summing over the indices of the elements in the sets $\mathcal{V}$ and $\mathcal{V}'$, which are obtained from slices k and k'.

The presence of a strongly misaligned slice can result in significant distances, which may subsequently lead to an increase in the mean TRE of well-registered slices. To address this, it is possible to compute the median TRE for each slice k, considering the mean TRE of all pairwise intersections involving slice k.

3.3. Metrics for assessing the classifier performance

We assess slice classification performance by comparing the true labels with those predicted by the classifier using standard metrics. The True Positive Rate (TPR), also known as recall or sensitivity, is defined as the number of misaligned slices correctly labeled by the model, divided by the total number of misaligned slices. The False Positive Rate (FPR), or probability of false alarms, is the number of well-registered slices incorrectly classified as misaligned, divided by the total number of well-registered slices. Precision is the number of misaligned slices correctly labeled by the model, divided by the total number of slices labeled as misaligned by the classifier. Finally, the F1-score is the harmonic mean of precision and TPR.

All these metrics require setting the ground truth for whether a slice is well-registered or misaligned. For synthetic data, this is achieved by considering slices with a median TRE greater than 1.5 as misaligned. For real data, slice registration quality is assessed by calculating a residual image, defined as the absolute difference between the high-resolution reconstructed image and each slice. Based on this residual image, slices are manually classified.

3.4. Qualitative Evaluation of 3D reconstructed images

Due to the absence of ground truth data for real clinical data, a quantitative evaluation of the 3D reconstruction quality was not feasible. Consequently, a qualitative assessment of the reconstructed volumes was conducted, involving two raters. The methodology employed was inspired by the work of Sanchez et al. (2024). The reconstruction are classified into four categories:

1. Good: The reconstructed image is of high quality, with no visible artifacts and all brain structures clearly defined.
2. Acceptable: The reconstructed image is of high quality, with minimal artifacts and most brain structures identifiable.
3. Poor: The reconstructed image has significant artefacts that prevent detailed examination.
4. Failed: The reconstructed image is severely compromised or non-existent, with no recognizable brain structures.

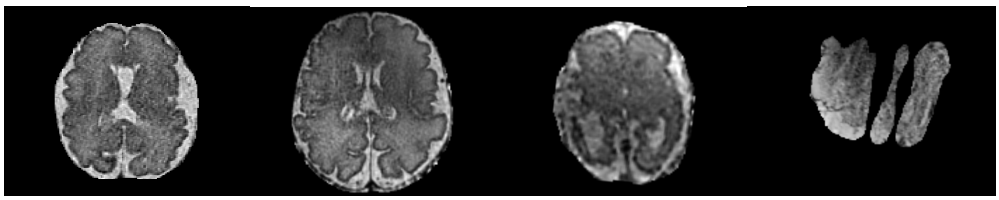


Figure 5: Examples of reconstructed volumes: good, acceptable, poor or failed.

Reconstructions categorized as good or acceptable were suitable for subsequent processing steps, such as segmentation, while those classified as poor or failed were of insufficient quality. Fig. 5 illustrates an example of three-dimensional reconstruction for each of the aforementioned categories.

4. Results

The evaluation consists of three parts: optimization performance, classification algorithm effectiveness, and the impact of implementing a multistart procedure.

4.1. Results on simulated data

4.1.1. Evaluation of the optimization step

We focus here on the alternating block optimization procedure (Alg. 1), omitting both the outlier detection and multistart strategies in ROSI. This approach is referred to as ROSI-noODM (no outlier detection and multistart). The performance of ROSI-noODM is benchmarked against the widely used NiftyMIC algorithm (Ebner et al. (2020)) and the more recent NeSVoR algorithm (Xu et al. (2023)). For a fair comparison, the outlier detection step was also excluded from NiftyMIC.

Fig. 6 shows the median TRE (Sec. 3.2) before and after registration for all three algorithms, evaluated on simulated datasets with varying motion levels (small, medium, large, and extra-large). The results demonstrate that ROSI-noODM outperforms the other methods overall, maintaining consistently low TRE values across all motion levels after registration, even though a small subset of slices remains poorly registered for large and extra-large motion. NeSVoR performs well for small to large initial motion but struggles with extra-large motion, where a significant number of slices exhibit high errors. In contrast, NiftyMIC is effective for small motion but fails to handle medium to extra-large inter-slice motion adequately.

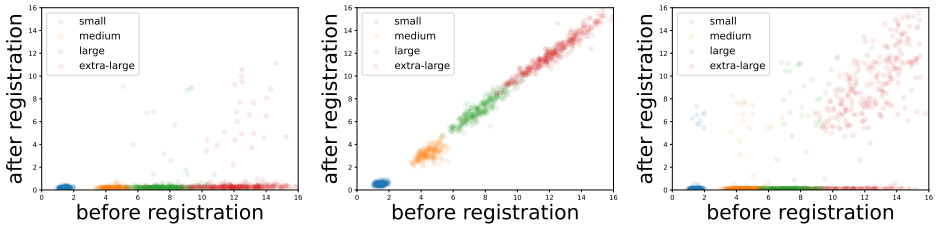


Figure 6: Median TRE (in millimeters) before registration (x-axis) versus after registration (y-axis) for simulated data. Each point represents a slice. Results are shown for (a) ROSI-noODM, (b) NiftyMIC (without outlier detection), and (c) NeSVoR.

A second approach to assess registration performance involves comparing the 3D reference image with the reconstruction obtained from the registered slices (Sec. 3.2). Using ROSI-noODM as the motion correction method, we reconstructed a 3D image with either the NeSVoR or NiftyMIC reconstruction algorithm. These methods are referred to as ROSI-noODM+NeSVoR and ROSI-noODM+NiftyMIC, respectively.

In Tab. 1 and Tab. 2 (first two columns with numerical values), we compare the NiftyMIC method (motion correction and reconstruction) to ROSI-noODM+NiftyMIC. Using ROSI-noODM for motion correction consistently resulted in higher average PSNR and SSIM values across all simulated motion levels compared to NiftyMIC.

In the last two columns of Tables 1 and 2, we compare NeSVoR (motion correction and reconstruction) to ROSI-noODM+NeSVoR. While the differences in PSNR and SSIM were less pronounced compared to NiftyMIC, higher values were still achieved using ROSI-noODM+NeSVoR in most cases (with one exception).

Table 1: Mean PSNR (dB) ↑ obtained on 3-D reconstruction from simulated data. Standard deviation is given in parenthesis.

PSNR ↑				
Reconstruction method	NiftyMIC		NeSVoR	
Registration method	NiftyMIC	ROSI-noODM	NeSVoR	ROSI-noODM
Low	30.79 (0.52)	**32.65** (0.55)	33.34 (0.94)	**34.75** (1.01)
Medium	25.40 (0.49)	**31.54** (0.63)	**32.54** (0.39)	31.74 (0.33)
Large	23.90 (0.88)	**30.30** (0.41)	27.97 (1.88)	**30.14** (0.19)
Extra-large	22.32 (0.32)	**29.95** (0.53)	27.99 (1.86)	**29.25** (0.58)

Table 2: Mean SSIM ↑ obtained on 3-D reconstruction from simulated data. Standard deviation is given in parenthesis.

SSIM ↑				
Reconstruction method	NiftyMIC		NeSVoR	
Registration method	NiftyMIC	ROSI-noODM	NeSVoR	ROSI-noODM
Low	0.94 (0.007)	**0.95** (0.006)	0.98 (0.001)	**0.98** (0.003)
Medium	0.82 (0.037)	**0.94** (0.011)	0.94 (0039)	**0.97** (0.001)
Large	0.79 (0.027)	**0.92** (0.011)	0.94 (0.034)	**0.97** (0.003)
Extra-large	0.75 (0.032)	**0.92** (0.012)	0.94 (0.039)	**0.97** (0.002)

Fig. 7 presents examples of 3D reconstructions from simulated data with varying motion levels (low, medium, large, and extra-large). NiftyMIC fails to accurately reconstruct data with medium and large motion artifacts, while ROSI-noODM (regardless of the reconstruction method used) maintains good reconstruction quality across all motion levels. NeSVoR generally produces high-quality 3D reconstructions but its performance may be limited when processing data with extra-large motion, as shown in Fig. 7.

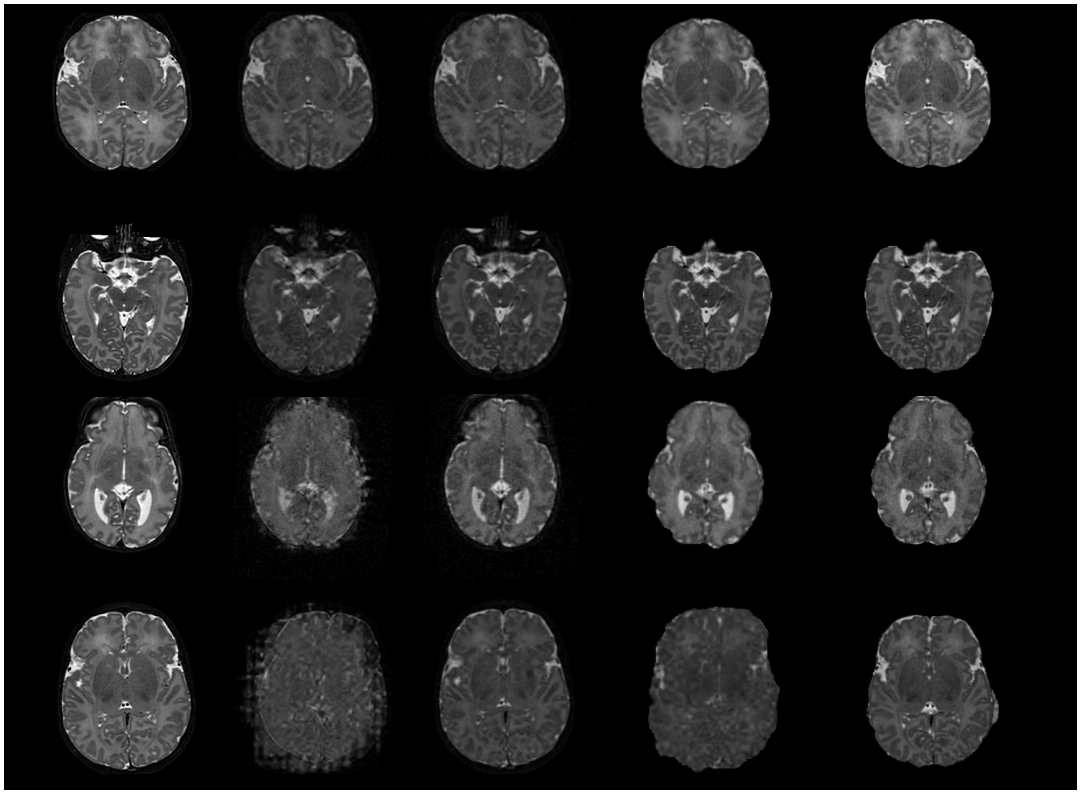


Figure 7: Examples of 3D reconstructions from simulated data with varying motion levels: low (row 1), medium (row 2), large (row 3), and extra-large (row 4). Left column: a slice from the high-resolution volume. Reconstructions are obtained with NiftyMIC for motion correction and reconstruction (second column), ROSI-noODM+ NiftyMIC (third column), NeSVoR for motion correction and reconstruction (fourth column), and ROSI-noODM+NeSVoR (last column).

4.1.2. Evaluation of the classification algorithm

This section evaluates the performance of the proposed outlier detection algorithm. As outlined in Sec. 2.6, the classifier is used to identify mis-

aligned and well-aligned slices after the alternating block optimization procedure (ROSI-noODM): the probability p_k that slice k is misaligned is used to determine the set $\mathcal{A}$ of slices that are confidently well-aligned (threshold of 0.5) and the set $\mathcal{M}$ of slices that are confidently misaligned (threshold of 0.2). Specifically, a slice k belongs to $\mathcal{M}$ if $p_k > 0.2$ and to $\mathcal{A}$ if $p_k < 0.5$.

NeSVoR is excluded from this comparison as it does not directly detect outlier slices. In contrast, NiftyMIC employs an iterative approach, calculating the normalized cross-correlation (NCC) between each registered slice and the reconstructed volume after each iteration. Slices with NCC values below a predefined threshold are classified as outliers and excluded. The threshold is progressively increased across iterations: 0.5 for the first, 0.65 for the second, and 0.8 for the final iteration.

Table 3 compares the outlier detection performance of ROSI (with a threshold of 0.5) and NiftyMIC using the evaluation metrics defined in Sec. 3.3. We also report the percentage of slices that should be considered as misaligned (defined as a median TRE above 1.5mm).

Table 3: Comparison of the outlier detection performances of ROSI and NiftyMIC on simulated data using the metrics defined in Sec. 3.3. We also give the percentage of slices that should be considered as misaligned (median TRE above 1.5 mm)

ROSI					
	TRE > 1.5	TPR/Recall $\uparrow$	FPR $\downarrow$	Precision $\uparrow$	F1-score $\uparrow$
Small	0%	$\times$	0%	$\times$	$\times$
Medium	2.7%	70%	0%	1	0.82
Large	0.5%	100%	5%	0.69	0.78
Extra-large	9.8%	92%	2.3%	0.77	0.91
NiftyMIC					
	TRE > 1.5	TPR/Recall $\uparrow$	FPR $\downarrow$	Precision $\uparrow$	F1-score $\uparrow$
Small	0%	$\times$	0.5%	$\times$	$\times$
Medium	100%	92%	$\times$	$\times$	$\times$
Large	100%	92%	$\times$	$\times$	$\times$
Extra-large	100%	92%	$\times$	$\times$	$\times$

Both classifiers perform well but operate on distinct data distributions. NiftyMIC always encounters scenarios where either all slices are well-aligned or all are misaligned. For ROSI, a threshold of $p_k = 0.5$ strikes an effective balance, retaining most well-aligned slices (according to the experiments, the FPR ranges from 0% to 5%) while filtering out a significant proportion of

misaligned ones (the TPR ranges from 70% to 100%). Note that lowering the threshold to $p_k = 0.2$ increases the false positive rate but enhances the detection of misaligned slices, making it an appropriate choice for constructing the set $\mathcal{M}$ of misaligned slices.

4.1.3. Evaluation of the multistart approach for misaligned slices

This section examines the impact of the multistart procedure implemented in ROSI. We present the results obtained immediately after applying this procedure, referred to as ROSI-M (it includes the optimization, the outlier detection and correction of misaligned slices). This method differs from ROSI in two key ways: after the multi-start procedure, ROSI considers only the slices in the set $\mathcal{A}$, excluding all others, and performs a final optimization of the loss function using only these slices (Fig. 3), whereas ROSI-M retains all slices and does not perform a final optimization. Fig. 8 presents the results obtained with ROSI-M. The median TRE is very low for nearly all slices, indicating significant improvement compared to ROSI-noODM (Fig. 6). This clearly highlights the interest of the multistart approach.

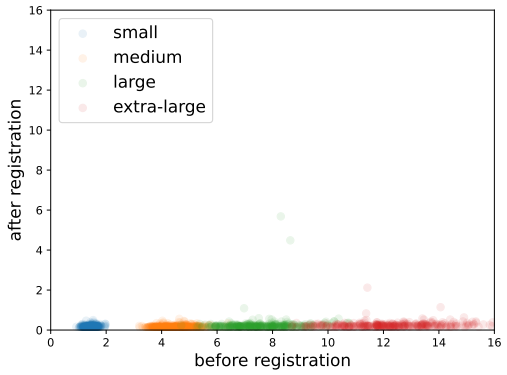


Figure 8: Median TRE (in millimeters) before registration (x-axis) versus after registration (y-axis) for simulated data registered with ROSI-M. Each point represents a slice.

These positive results are confirmed by Tab. 4, which shows a very low percentage of slices with a median TRE exceeding 1.5 mm. However, it is worth noting that the classifier fails to detect misaligned slices (with a threshold of 0.5), but the false positive rate remains very low.

Table 4: Percentage of outliers (slices with median TRE above 1.5 mm) for simulated data registered with ROSI-M and classifier performance on the registered slices.

	TRE>1.5	TPR ↑	FPR ↓
Small	0%	×	0%
Medium	0%	×	0%
Large	0.5%	0%	0.2%
Extra-Large	0.8%	0%	0.2%

4.2. Results on real data

As for the simulated data, we first evaluate the optimization step. Secondly, we assess the interest of outlier detection and evaluate the benefit of multistart. Finally, ROSI is compared with the methods Ebner et al. (2020), Kuklisova-Murgasova et al. (2012) and Xu et al. (2023) from the state-of-the-art.

4.2.1. Evaluation of the optimization step

As explained in Sec. 3.2, registration performance for real data is assessed by visually inspecting the quality of the 3D-reconstructions obtained from the registered slices. Results of ROSI-noODM+NiftyMIC (see Sec. 4.1.1) are compared in Tab. 5 with those of NiftyMIC (for motion correction and reconstruction). For a fair comparison, the outlier detection step was also excluded from NiftyMIC. Motion correction with ROSI-noODM resulted in reconstructions with better anatomical details and fewer artifacts compared to NiftyMIC. Although there was a discrepancy in the evaluation between the two raters, we observed that ROSI-noODM + NiftyMIC produced a higher number of good reconstructions and fewer poor or failed reconstructions compared to NiftyMIC in both cases.

Table 5: Visual assessment by two raters of 49 fetal data reconstructions obtained with NiftyMIC (no outlier removal) and ROSI-noODM + NiftyMIC.

Quality	Rater 1		Rater 2	
	NiftyMIC	ROSI-noODM + NiftyMIC	NiftyMIC	ROSI-noODM + NiftyMIC
Good	15	17	12	14
Acceptable	11	10	7	8
Poor or Failed	23	22	30	27

4.2.2. Evaluation of the classification algorithm

This section evaluates the outlier detection algorithm applied to the slices registered with ROSI-noODM. It is worth noting that the classifier was trained on synthetic data generated from images of the dHCP database (Edwards et al. (2022)). Here, we test it on real data acquired during clinical routines at the Timone Hospital. Note that reconstructions classified as failed were excluded from this analysis due to the difficulty of manually rating slices when a large number are misaligned. In the end, the dataset contained 9.6% of misaligned slices. Tab. 6 presents the different metrics for various thresholds.

The classifier performed well in detecting misaligned slices in the clinical dataset. With a threshold of 0.2, it successfully identifies over 80% of misaligned slices while maintaining a low false alarm rate (below 10%). This confirms that 0.2 is an effective threshold for selecting slices in the multi-start procedure (i.e., for defining the set $\mathcal{M}$). At a threshold of 0.5, fewer misaligned slices are detected (more than 60%), but the false positive rate drops to under 3%, ensuring that only a small number of well-aligned slices are excluded from the set $\mathcal{A}$.

Table 6: Classification performance on real data for different thresholds (0.1 to 0.5): a slice k is labeled as misaligned if p_k exceeds the threshold.

	0.1	0.2	0.3	0.4	0.5
TPR/Recall ↑	0.92	0.81	0.74	0.71	0.66
FPR ↓	0.19	0.09	0.05	0.04	0.03
Precision ↑	0.31	0.46	0.58	0.63	0.67
F1-score ↑	0.46	0.58	0.65	0.66	0.66

We also assess the impact of incorporating the outlier detection algorithm on the final 3D reconstruction. The key difference from Sec. 4.2.1 is that, when registration is performed with ROSI-noODM, only the slices from set $\mathcal{A}$ ($p_k < 0.5$) are used for reconstruction with NiftyMIC. The method is refered to ROSI-noM+NiftyMIC. Additionally, we compare our approach to the full NiftyMIC method, which includes the outlier removal step (not applied in Sec. 4.2.1). Table 7 summarizes the evaluation of the reconstructions obtained with ROSI-noM+NiftyMIC and NiftyMIC, along with the percentage of rejected slices. With ROSI-noM, the benefit of removing outliers is clear, as the number of reconstructed volumes rated as "Good" increased for both

raters, while the number of reconstructions rated as "Poor" or "Failed" decreased. The advantage of outlier rejection is less apparent in the case of NiftyMIC. Finally, it is worth noting that the percentage of rejected slices was generally smaller with ROSI-noM than with NiftyMIC.

Table 7: Visual assessment by two raters of 49 fetal data reconstructions obtained with NiftyMIC and ROSI-noM + NiftyMIC. It also gives the percentage of slices removed by each algorithm during outlier detection.

Quality	Rater 1		Rater 2	
	NiftyMIC		NiftyMIC	
	reconstructed volumes	% of rejected slices	reconstructed volumes	% of rejected slices
Good	17	8%	12	4 %
Acceptable	7	14%	8	17 %
Poor or Failed	25	39%	29	47%
	ROSI-noM+NiftyMIC		ROSI-noM+NiftyMIC	
	reconstructed volumes	% of rejected slices	reconstructed volumes	% of rejected slices
Good	18	6%	17	7%
Acceptable	11	7%	6	7%
Poor or Failed	20	32%	26	26%

4.2.3. Evaluation of the multistart approach for misaligned slice

This section examines the contribution of the multistart procedure. Reconstructions classified as poor or failed were excluded from the analysis, as the multistart procedure is ineffective when too many slices are misaligned. Table 8 gives the number of slices identified as misaligned in the real dataset, using two thresholds (0.2 and 0.5), before and after the multistart procedure, as well as the total number of slices (in parentheses). The multistart procedure was applied to a total of 302 slices. Without the multistart step, 146 slices would have been excluded from the reconstruction. After its application, only 106 slices were ultimately rejected for the final reconstruction. This highlights the effectiveness of the multistart procedure in correcting the parameters of misaligned slices: it recovered 40 (146 - 106) slices. Moreover, an analysis of the remaining uncorrected slices reveals that they are often located at the extremities of the brain or exhibit artifacts such as spin history effects or blurring.

Table 9 presents a quantitative assessment of the impact of the proposed multistart approach on reconstruction quality. This evaluation includes the full registration process of ROSI, followed by reconstruction. Results can be compared to those in Tab.7, which were obtained with the same configuration but without the multistart procedure. The results show that using

the multistart procedure reduces the number of rejected slices, though it has minimal effect on the quality of the reconstructed volumes.

Table 8: Number of slices identified as misaligned by the classifier in the real dataset, using two thresholds (0.2 and 0.5), before and after the multistart procedure. The total number of slices is shown in parentheses.

before correction ($p_k > 0.2$)	302 (1784)
after correction ($p_k > 0.2$)	242 (1784)
before correction ($p_k > 0.5$)	146 (1784)
after correction ($p_k > 0.5$)	106 (1784)

Table 9: Visual assessment by two raters of fetal data reconstructions obtained using ROSI + NiftyMIC (including the multistart procedure) (failed and poor cases were excluded). It also gives the percentage of slices removed by the algorithm during outlier detection.

Quality	Rater 1		Rater 2	
	reconstructed volumes	% of rejected slices	reconstructed volumes	% of rejected slices
Good	18	5%	17	5 %
Acceptable	11	3%	6	4%

4.3. Comparison with the state of the art

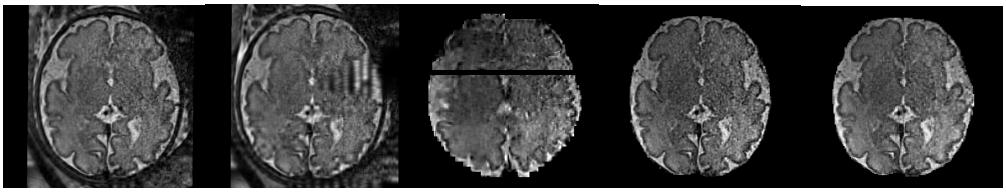
This section compares reconstructions obtained from slices registered with ROSI to those produced by three state-of-the-art methods: NiftyMIC (Ebner et al. (2018)), NeSVoR (Xu et al. (2023)), and SVRTK (Kuklisova-Murgasova et al. (2012)). To reconstruct a 3D volume from motion-corrected data using ROSI, we employ either the reconstruction algorithm from NiftyMIC or NeSVoR. Additionally, NeSVoR is often paired with Svort, which provides an initial motion correction estimate for the slices. In real data, sudden large motions (e.g., a 90° rotation or more) can occur, which, although rare, cannot be corrected by approaches like NeSVoR alone, NiftyMIC, SVRTK, or ROSI. To address this, Svort was also used to initialize the motion correction before applying ROSI when the reconstruction was carried out using the NeSVoR method.

In Table 10, a higher number of reconstructed volumes were rated "Good" when using the proposed method for motion correction, compared to NiftyMIC and SVRTK. Additionally, both raters observed a lower number of poor and failed reconstructions. With NeSVoR, the difference was less significant,

Table 10: Visual assessment by two raters of 49 fetal data reconstructions obtained with ROSI+NiftyMIC, NiftyMIC, SVRTK, ROSI+NeSVoR, NeSVoR.

	Rater 1					Rater 2				
Reconstruction method	NiftyMIC		SVRTK	NeSVoR		NiftyMIC		SVRTK	NeSVoR	
Registration method	ROSI	NiftyMIC	SVRTK	ROSI	NeSVoR	ROSI	NiftyMIC	SVRTK	ROSI	NeSVoR
Good	18	17	15	23	24	17	12	9	17	18
Acceptable	11	7	15	15	14	6	8	9	13	13
Poor or Failed	20	25	19	11	11	26	29	31	19	18

with very similar reconstructions. Raters 1 and 2 found, however, that one reconstructed volume was of slightly better quality with registration from NeSVoR.



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

Figure 9: 3D reconstructions obtained with (a) ROSI+NiftyMIC (b) NiftyMIC (c) SVRTK (d) ROSI+NeSVoR (e) NeSVoR.

Fig. 9 presents 3D reconstructions in axial orientation, generated using the methods outlined above. Artifacts are visible in the reconstructions obtained with NiftyMIC and SVRTK (Fig. 9(b) and (c)). In contrast, reconstructions using NeSVoR (Fig. 9(e)), as well as those combining ROSI with NiftyMIC (Fig. 9(a)) or ROSI with NeSVoR (Fig. 9(d)), exhibit comparable high quality.

5. Conclusion

In this paper, we introduce ROSI (Registration based on Orthogonal Slices Intersection), a novel reconstruction-independent method for motion correction in fetal MRI. Inspired by Kim et al. (2009), ROSI leverages intersections between slices from orthogonal stacks for robust motion estimation. Additionally, a machine learning classifier is employed to identify outlier slices, incorporating both image intensity and brain masking information. Finally, a multistart strategy is implemented to correct the positioning of

outlier slices detected by the classifier. This approach shows very good performance on simulated data. For real data, ROSI outperforms NiftyMIC and SVRTK, achieving similar results to NeSVoR .

Concerning potential improvement of the method, the current classifier algorithm is designed to detect misaligned slices, encompassing both those exhibiting local minima in motion estimation and those corrupted by artifacts. A potential enhancement would involve differentiating between these two types of slices. This would allow for a more targeted application of the subsequent multi-start algorithm, optimizing computational efficiency. Furthermore, integrating ROSI into a comprehensive framework, (Ebner et al. (2020); Kuklisova-Murgasova et al. (2012)), would significantly enhance its accessibility and potential impact within the field

6. Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used Gemini, Google to rewrite and clarify sentences using grammatically correct English. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Acknowledgements: This work was supported by the French National Research Agency (projects ANR- 19-CHIA-0015, ANR-19-CE45-0014, ANR-21-NEU2-0005).

Appendix A. Calculation of slice Intersections for registration

The registration criterion is based on the principle that well-registered slices should exhibit similar intensity profiles along their intersection points. Calculating these intersections enables us to compare these profiles and quantify the alignment between slices, adjusting the transformation parameters accordingly. This appendix describes the process of deriving the necessary equations to identify corresponding points along the intersection of each pair of slices k and k'. This calculation involves four main steps:

1. Deriving the plane equation of slice k in the world coordinate system (Appendix A.1)
2. Deriving the equation of the intersection line of the planes of slices k and k' in the world coordinate system (Appendix A.2)
3. Deriving the equations of the intersection line of the planes of slices k and k' in the 2D coordinate system of slice k and k' (Appendix A.3)
4. Generating corresponding points along the intersection line (Appendix A.4)

Appendix A.1. Deriving the plane equation of slice k in the world coordinate system

The equation of the plane of slice k is expressed as:

$$\mathbf{n}_k(x - \mathbf{p}_k) = 0 \tag{A.1}$$

where $\mathbf{n}_k$ is a normal vector to the plane, and $\mathbf{p}_k$ a point on the plane.

The position of the slice k in 3D space is defined by its transformation matrix $\mathbf{M}_k$ (Eq.1). First, $\mathbf{p}_k$ can be set to $\mathbf{M}_k.(0,0,0,1)^T$. Next, in the general case where $R_{f(k)}$ (Eq.1) represents an affine transformation, $\mathbf{n}_k$ can be computed by applying $\mathbf{M}_k$ to two non-collinear vectors of the k-th slice (in the 2D coordinate system of slice k) and taking their cross product:

$$\mathbf{n}_k = [\mathbf{M}_k.(1,0,0,0)^T] \wedge [\mathbf{M}_k.(0,1,0,0)^T]. \tag{A.2}$$

In the following, we assume that $\mathbf{n}_k$ is normalized to have a unit magnitude.

Appendix A.2. Deriving the equation of the intersection line of the planes of slices k and k' in the world coordinate system

Consider two orthogonal slices, k and k'. The equation of the intersection line can be derived from the condition that a point lies on both the planes of slices k and k'. Using Equation A.1, the equation of the line in the 3D space writes:

$$(g\mathbf{n}_k + h\mathbf{n}_{k'}) + \lambda \mathbf{n}_k \wedge \mathbf{n}_{k'}. \tag{A.3}$$

λ is a scalar parameter that defines points along the line by scaling the direction vector $\mathbf{n}_k \wedge \mathbf{n}_{k'}$, starting from the point $g\mathbf{n}_k + h\mathbf{n}_{k'}$. g and h are constants determined by solving the system of equations obtained by substituting Eq. A.3 into Eq. A.1.

$$\begin{cases} g + h\mathbf{n}_k\mathbf{n}_{k'} = \mathbf{n}_k\mathbf{p}_k \\ g\mathbf{n}_k\mathbf{n}_{k'} + h = \mathbf{n}_{k'}\mathbf{p}_{k'}. \end{cases} \tag{A.4}$$

Solving the system of equations yields:

$$g = \tfrac{1}{1-\alpha^2}(\beta - \gamma\alpha), \quad h = \tfrac{1}{1-\alpha^2}(\gamma - \alpha\beta), \tag{A.5}$$

where $\alpha = \mathbf{n}_k.\mathbf{n}_{k'}$ $\beta = \mathbf{n}_k.\mathbf{p}_k$, and $\gamma = \mathbf{n}_{k'}.\mathbf{p}_{k'}$.

Appendix A.3. Deriving the equations of the intersection line of the planes of slices k and k' in the 2D coordinate system of slice k and k'

The equation of the intersection line in the 2D coordinate system of slice k can be easily obtained by applying the inverse transformation $\mathbf{M}_k^{-1}$ to the 3D intersection line equation (Eq. A.3):

$$\mathbf{M}_k^{-1}.(g\mathbf{n}_k + h\mathbf{n}_{k'}) + \lambda \mathbf{M}_k^{-1}.(\mathbf{n}_k \wedge \mathbf{n}_{k'}). \tag{A.6}$$

Similarly, in the 2D coordinate system of slice k', the equation becomes:

$$\mathbf{M}_{k'}^{-1}.(g\mathbf{n}_k + h\mathbf{n}_{k'}) + \lambda \mathbf{M}_{k'}^{-1}.(\mathbf{n}_k \wedge \mathbf{n}_{k'}). \tag{A.7}$$

Appendix A.4. Generating corresponding points on the intersection line

By construction, using the same scalar λ in both Eq. A.6 and A.7 gives us two corresponding points, v_i (in slice k) and v'_i (in slice k'), such that $\mathbf{M}_k(v_i) = \mathbf{M}_{k'}(v'_i)$. This means that they correspond to the same physical location, according to $\mathbf{M}_k$ and $\mathbf{M}_{k'}$.

However, the support of the slices is bounded (it forms a rectangle), so we cannot choose just any value for λ. Let $g_k(.)$ be the function that associates a scalar λ with a 2D point through equation A.6. Then, we can easily compute an interval $J_k = [\lambda_k^{min}, \lambda_k^{max}]$ such that:

- The points $g(\lambda_k^{min})$ and $g(\lambda_k^{max})$ lie on the boundary of the support of slice k.
- $\forall \lambda \in [\lambda_k^{min}, \lambda_k^{max}], g(\lambda)$ is inside support of slice k.

The same process can be applied to slice k', yielding another interval, $J_{k'}$.

It is worth noting that J_k (or $J_{k'}$) may, in theory, be empty or reduced to a single point. However, this scenario should not occur in practice, as the slices are derived from orthogonal stacks. Nevertheless, if the transformations are poorly estimated, such cases may arise.

Finally, the intersection line is sampled at regular intervals (every millimeter) as follows: a list of uniformly spaced λ values is computed to span precisely the union of J_k and $J_{k'}$ (which may consist of a single interval or two disjoint intervals). By doing so, we ensure that the points v (resp. v') sample all structures of interest that are on the segment of the slice k (resp. k'). Once this list of λ values is obtained, Eqs. A.6 and A.7 are applied to compute the corresponding coordinates in slice k and slice k'.

In summary, for two orthogonal slices k and k', we obtain a set of points in slice k, denoted as $\mathcal{V} = \{v_1, v_2, \ldots, v_{L_{k,k'}}\}$, along with their corresponding points in slice k': $\mathcal{V}' = \{v'_1, v'_2, \ldots, v'_{L_{k,k'}}\}$, such that:

$$\mathbf{M}_k(v_i) = \mathbf{M}_{k'}(v'_i).$$

Appendix B. Alternating block optimization procedure

The algorithm in given in Alg. 1.

Algorithm 1 Alternating block optimization procedure

Input : $A = \{1, ..., n\}$, n the number of slices.
Output: $\boldsymbol{\theta}_k$ and $\mathbf{t}_k$ for each slice k.

for v in 1,2,4,8 **do** :
  $ds = 4/v$, $fs = 0.25/v$, $th = 2/v$
  while global convergence has not been achieved **do**
    Set B to A
    while B is not empty **do**
      for each slice k in B **do**
        Update: $(\boldsymbol{\theta}_k^{old}, \mathbf{t}_k^{old}) \rightarrow (\boldsymbol{\theta}_k^{new}, \mathbf{t}_k^{new})$
        if $||(\boldsymbol{\theta}_k^{old}, \mathbf{t}_k^{old}) - (\boldsymbol{\theta}_k^{new}, \mathbf{t}_k^{new})||^2 < th$ **then**
          ▷ local convergence is achieved for slice k
          remove k from B
        else
          global convergence is not achieved
        end if
      end for
    end while
  end while
end for

Appendix C. Multistart Initialization

Appendix C.1. Initialization Strategy for Multi-Start Optimization

During the update of slice k in the multi-start procedure (Sec. 2.6), the optimization algorithm is executed multiple times with different initializations. We detail here how these initializations are generated by leveraging well-aligned slices nearby k. Appendix C.2 and Appendix C.3 explain how initial positions for slice k can be generated using either two correctly aligned surrounding the slice k (Appendix C.2) or a single nearby well-aligned slice (Appendix C.3). In practice, we identify, if possible, three well-registered slices before and after slice k. We can then run the approach of Sec. Appendix C.2 with $3 \times 3 = 9$ different configurations and the approach of Sec. Appendix C.3 with $3 + 3 = 6$ configurations, which results in a maximum of 15 initial positions. Note that some positions may be redundant, so to ensure a diverse set of starting points, redundant positions are removed.

These initial positions are not optimal (since they are not computed from

the images), so we use the following approach, inspired by Jenkinson and Smith (2001), to determine 5 possible initializations for each of the 15 positions, leading to a maximum of 75 initializations.

For a given initial position $(\boldsymbol{\theta}_k, \mathbf{t}_k)$, we create a grid of 125 rotation points ranging from $(\boldsymbol{\theta}_k - 6)$ to $(\boldsymbol{\theta}_k + 6)$. At each grid point, the rotation matrix is fixed, and we estimate the translation $\mathbf{t}_k$ by minimizing the Dice score:

$$\frac{\sum_{k,k';k>k'} 2M(k,k')}{\sum_{k,k';k>k'} P(k,k') + Q(k,k')}, \tag{C.1}$$

where M, P and Q are defined in Eqs. 7 and 8. We prefer optimizing the Dice score instead of the loss function because convergence is much faster. Finally, we estimate the loss at each point on the rotation parameter grid and select the five smallest local minima from the grid search as potential initializations for the optimization algorithm.

Appendix C.2. Initialization : first case

We consider the scenario where a slice k, potentially misaligned, is bounded by two correctly aligned slices k_1 and k_2 ($k_1 < k < k_2$), which originate from the same stack of slices ($f(k_1) = f(k_2) = f(k)$). The goal is to initialize $\mathbf{M}(\boldsymbol{\theta}_k, \mathbf{t}_k)$ from $\mathbf{M}(\boldsymbol{\theta}_{k_1}, \mathbf{t}_{k_1})$ and $\mathbf{M}(\boldsymbol{\theta}_{k_2}, \mathbf{t}_{k_2})$.

In the absence of any relative motion between slices in the same stack, $\mathbf{T}(\mathbf{c}_k)\mathbf{M}(\boldsymbol{\theta}_k, \mathbf{t}_k)\mathbf{T}(-\mathbf{c}_k)$ would remain the same for all slices k in the stack. This can be understood from Eq. 1, where $\mathbf{R}_{f(k)}\mathbf{R}_{k,2d\to 3d}$ converts the 2D coordinates of slice k into the world reference system. This motivates the following approach to initialize $\mathbf{M}(\boldsymbol{\theta}_k, \mathbf{t}_k)$:

1 Compute $\mathbf{M}_{avg}$ as the weighted average of $\mathbf{T}(\mathbf{c}_{k_1})\mathbf{M}(\boldsymbol{\theta}_{k_1}, \mathbf{t}_{k_1})\mathbf{T}(-\mathbf{c}_{k_1})$ and $\mathbf{T}(\mathbf{c}_{k_2})\mathbf{M}(\boldsymbol{\theta}_{k_2}, \mathbf{t}_{k_2})\mathbf{T}(-\mathbf{c}_{k_2})$.

2 Compute the parameters of $\mathbf{M}(\boldsymbol{\theta}_k, \mathbf{t}_k)$ such that $\mathbf{T}(\mathbf{c}_k)\mathbf{M}(\boldsymbol{\theta}_k, \mathbf{t}_k)\mathbf{T}(-\mathbf{c}_k) = \mathbf{M}_{avg}$, namely, $\mathbf{M}(\boldsymbol{\theta}_k, \mathbf{t}_k) = \mathbf{T}(-\mathbf{c}_k)\mathbf{M}_{avg}\mathbf{T}(\mathbf{c}_k)$.

The first step uses interpolation of rigid transformations. The geodesic path between two rigid transformation $\mathbf{T}_1$ and $\mathbf{T}_2$ is: $e^{t\cdot\log(\mathbf{T}_2\mathbf{T}_1^{-1})}\mathbf{T}_1$, with $t \in [0,1]$. To compute $\mathbf{M}_{avg}$, we use the previous equation with $\mathbf{T}_i = \mathbf{T}(\mathbf{c}_{k_i})\mathbf{M}(\boldsymbol{\theta}_{k_i}, \mathbf{t}_{k_i})\mathbf{T}(-\mathbf{c}_{k_i})(i = 1,2)$, and $t = \frac{k-k_1}{k_2-k_1}$. The second step is straightforward.

Appendix C.3. Initialization : second case

We consider the case where a slice k, potentially misaligned, is close to a correctly aligned slice k_1 from the same stack ($f(k_1) = f(k)$). The goal is to initialize $\mathbf{M}(\boldsymbol{\theta}_k, \mathbf{t}_k)$ from $\mathbf{M}(\boldsymbol{\theta}_{k_1}, \mathbf{t}_{k_1})$. Using the same reasoning as in Appendix C.2, we can compute $\mathbf{M}(\boldsymbol{\theta}_k, \mathbf{t}_k)$ such that:

$$\mathbf{T}(\mathbf{c}_k)\mathbf{M}(\boldsymbol{\theta}_k, \mathbf{t}_k)\mathbf{T}(-\mathbf{c}_k) = \mathbf{T}(\mathbf{c}_{k_1})\mathbf{M}(\boldsymbol{\theta}_{k_1}, \mathbf{t}_{k_1})\mathbf{T}(-\mathbf{c}_{k_1}), \tag{C.2}$$

or equivalently,

$$\mathbf{M}(\boldsymbol{\theta}_k, \mathbf{t}_k) = \mathbf{T}(-\mathbf{c}_k + \mathbf{c}_{k_1})\mathbf{M}(\boldsymbol{\theta}_{k_1}, \mathbf{t}_{k_1})\mathbf{T}(-\mathbf{c}_{k_1} + \mathbf{c}_k). \tag{C.3}$$

Appendix D. Multistart procedure

The multistart procedure is detailed in Algorithm 2. It involves iterating through the following two steps: updating all parameters associated with the slices in $\mathcal{M}$ (the set of indices corresponding to slices identified as misaligned, i.e., where $p_k > 0.2$) and then recalculating $\mathcal{M}$ and $\mathcal{A}$ (the set of indices corresponding to slices identified as well-aligned, i.e., where $p_k < 0.5$). During the first iteration, we use the loss from Eq. 10 with $w = 0$, and in subsequent iterations, we use $w = 1$. The process stops when $\mathcal{M}$ does not change during an iteration with $w = 1$. Although we do not have a formal proof of convergence, we observed that the algorithm consistently converged in all studied cases.

Algorithm 2 Multistart procedure for slice alignment

Initialize $w \leftarrow 0$ (for the loss function in Eq. 10)
Compute $\mathcal{M}$ (misaligned slices) and $\mathcal{A}$ (well-aligned slices)
while convergence is not achieved **do**
  for each slice $k \in \mathcal{M}$ **do**
    Generate m initializations from $\mathcal{A}$ (Appendix C)
    Optimize Eq. (10) for slice k, using slices in $\mathcal{A}$ and k for the loss calculation, starting from the m initializations.
    Set slice k's parameters to those minimizing the cost
  end for
  Compute $\mathcal{M}$ (misaligned slices) and $\mathcal{A}$ (well-aligned slices)
  if $\mathcal{M}$ has not changed and $w = 1$ **then**
    Convergence is achieved
  end if
  Update $w \leftarrow 1$ (switch to alternative loss function in Eq. 10)
end while

References

Ciceri, T., Squarcina, L., Pigoni, A., Ferro, A., Montano, F., Bertoldo, A., Persico, N., Boito, S., Triulzi, F.M., Conte, G., et al., 2023. Geometric Reliability of Super-Resolution Reconstructed Images from Clinical Fetal MRI in the Second Trimester. Neuroinformatics 21, 549–563.

Ebner, M., Wang, G., Li, W., Aertsen, M., Patel, P.A., Aughwane, R., Melbourne, A., Doel, T., David, A.L., Deprest, J., et al., 2018. An automated localization, segmentation and reconstruction framework for fetal brain MRI, in: Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I, Springer. pp. 313–320.

Ebner, M., Wang, G., Li, W., Aertsen, M., Patel, P.A., Aughwane, R., Melbourne, A., Doel, T., Dymarkowski, S., De Coppi, P., et al., 2020. An automated framework for localization, segmentation and super-resolution reconstruction of fetal brain MRI. NeuroImage 206, 116324.

Edwards, A.D., Rueckert, D., Smith, S.M., Abo Seada, S., Alansary, A., Almalbis, J., Allsop, J., Andersson, J., Arichi, T., Arulkumaran, S., et al.,

2022. The developing human connectome project neonatal data release. Frontiers in neuroscience 16, 886772.

Gholipour, A., Estroff, J.A., Warfield, S.K., 2010. Robust super-resolution volume reconstruction from slice acquisitions: application to fetal brain MRI. IEEE Transactions on Medical Imaging 29, 1739–1758.

Gholipour, A., Rollins, C.K., Velasco-Annis, C., Ouaalam, A., Akhondi-Asl, A., Afacan, O., Ortinau, C.M., Clancy, S., Limperopoulos, C., Yang, E., et al., 2017. A normative spatiotemporal MRI atlas of the fetal brain for automatic segmentation and analysis of early brain growth. Scientific reports 7, 476.

Hou, B., Khanal, B., Alansary, A., McDonagh, S., Davidson, A., Rutherford, M., Hajnal, J.V., Rueckert, D., Glocker, B., Kainz, B., 2018. 3-D reconstruction in canonical co-ordinate space from arbitrarily oriented 2-D images. IEEE Transactions on Medical Imaging 37, 1737–1750.

Huisman, T.A., Martin, E., Kubik-Huch, R., Marincek, B., 2002. Fetal magnetic resonance imaging of the brain: technical considerations and normal brain development. European radiology 12, 1941–1951.

Immerkaer, J., 1996. Fast noise variance estimation. Computer vision and image understanding 64, 300–302.

Jenkinson, M., Smith, S., 2001. A global optimisation method for robust affine registration of brain images. Medical Image Analysis 5, 143–156.

Jiang, S., Xue, H., Glover, A., Rutherford, M., Rueckert, D., Hajnal, J.V., 2007. MRI of moving subjects using multislice snapshot images with volume reconstruction (SVR): application to fetal, neonatal, and adult brain studies. IEEE Transactions on Medical Imaging 26, 967–980.

Kainz, B., Steinberger, M., Wein, W., Kuklisova-Murgasova, M., Malamateniou, C., Keraudren, K., Torsney-Weir, T., Rutherford, M., Aljabar, P., Hajnal, J.V., et al., 2015. Fast volume reconstruction from motion corrupted stacks of 2D slices. IEEE Transactions on Medical Imaging 34, 1901–1913.

Kim, K., Habas, P.A., Rousseau, F., Glenn, O.A., Barkovich, A.J., Studholme, C., 2009. Intersection based motion correction of multislice

MRI for 3-D in utero fetal brain image formation. IEEE Transactions on Medical Imaging 29, 146–158.

Klapwijk, E.T., Van De Kamp, F., Van Der Meulen, M., Peters, S., Wierenga, L.M., 2019. Qoala-T: A supervised-learning tool for quality control of FreeSurfer segmented MRI data. Neuroimage 189, 116–129.

Kuklisova-Murgasova, M., Quaghebeur, G., Rutherford, M.A., Hajnal, J.V., Schnabel, J.A., 2012. Reconstruction of fetal brain MRI with intensity matching and complete outlier removal. Medical Image Analysis 16, 1550–1564.

Lv, Y., Tan, J., Chen, S., Zhang, H., Cheng, W., Yang, C., Xu, X., Zhang, X., 2024. Two-stage implicit representation reconstruction with iterative registration and double-channel encoder for fetal mri, in: 2024 IEEE International Symposium on Biomedical Imaging (ISBI), IEEE. pp. 1–5.

Ma, L., Lin, W., Zhang, H., Li, G., 2024. Automated fetal brain volume reconstruction from motion-corrupted stacks with deep learning, in: 2024 IEEE International Symposium on Biomedical Imaging (ISBI), IEEE. pp. 1–5.

Mercier, C., Faisan, S., Pron, A., Girard, N., Auzias, G., Chonavel, T., Rousseau, F., 2023. Retrospective motion estimation for fetal brain MRI, in: 2023 Twelfth International Conference on Image Processing Theory, Tools and Applications (IPTA), IEEE. pp. 1–6.

Pio, R., 1966. Euler angle transformations. IEEE Transactions on automatic control 11, 707–715.

Rousseau, F., Glenn, O., Iordanova, B., Rodriguez-Carranza, C., Vigneron, D., Barkovich, J., Studholme, C., 2005. A novel approach to high resolution fetal brain mr imaging, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2005: 8th International Conference, Palm Springs, CA, USA, October 26-29, 2005, Proceedings, Part I 8, Springer. pp. 548–555.

Rousseau, F., Glenn, O.A., Iordanova, B., Rodriguez-Carranza, C., Vigneron, D.B., Barkovich, J.A., Studholme, C., 2006. Registration-based approach for reconstruction of high-resolution in utero fetal MR brain images. Academic Radiology 13, 1072–1081.

Saleem, S.N., 2013. Fetal Magnetic Resonance Imaging (MRI) A Tool for a Better Understanding of Normal and Abnormal Brain Development. Journal of child neurology 28, 890–908.

Sanchez, T., Esteban, O., Gomez, Y., Pron, A., Koob, M., Dunet, V., Girard, N., Jakab, A., Eixarch, E., Auzias, G., et al., 2023. FetMRQC: an open-source machine learning framework for multi-centric fetal brain MRI quality control. arXiv preprint arXiv:2311.04780 .

Sanchez, T., Mihailov, A., Gomez, Y., Juan, G.M., Eixarch, E., Jakab, A., Dunet, V., Koob, M., Auzias, G., Cuadra, M.B., 2024. Assessing data quality on fetal brain mri reconstruction: a multi-site and multi-rater study, in: International Workshop on Preterm, Perinatal and Paediatric Image Analysis, Springer. pp. 46–56.

Shi, W., Xu, H., Sun, C., Sun, J., Li, Y., Xu, X., Zheng, T., Zhang, Y., Wang, G., Wu, D., 2022. Affirm: Affinity fusion-based framework for iteratively random motion correction of multi-slice fetal brain MRI. IEEE Transactions on Medical Imaging 42, 209–219.

Singer, S., Nelder, J., 2009. Nelder-mead algorithm. Scholarpedia 4, 2928.

Studholme, C., Rousseau, F., 2014. Quantifying and modelling tissue maturation in the living human fetal brain. International Journal of Developmental Neuroscience 32, 3–10.

Uus, A.U., Kyriakopoulou, V., Makropoulos, A., Fukami-Gartner, A., Cromb, D., Davidson, A., Cordero-Grande, L., Price, A.N., Grigorescu, I., Williams, L.Z., et al., 2023. BOUNTI: Brain vOlumetry and aUtomated parcellatioN for 3D feTal MRI. bioRxiv .

Xu, J., Moyer, D., Gagoski, B., Iglesias, J.E., Grant, P.E., Golland, P., Adalsteinsson, E., 2023. NeSVoR: Implicit Neural Representation for Slice-to-Volume Reconstruction in MRI. IEEE Transactions on Medical Imaging 42, 1707–1719.

Xu, J., Moyer, D., Grant, P.E., Golland, P., Iglesias, J.E., Adalsteinsson, E., 2022. SVoRT: iterative transformer for slice-to-volume registration in fetal brain MRI, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 3–13.