Integrated Registration, Segmentation, and Interpolation for 3D/4D Sparse Data

Adeline Paiement

University of BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements for the degree of Doctor of Philosophy in the Faculty of Engineering, Visual Information Laboratory.

December 2013

34132 words

Abstract

We address the problem of object modelling from 3D and 4D sparse data acquired as different sequences which are misaligned with respect to each other. Such data may result from various imaging modalities and can therefore present very diverse spatial configurations and appearances. We focus on medical tomographic data, made up of sets of 2D slices having arbitrary positions and orientations, and which may have different gains and contrasts even within the same dataset. The analysis of such tomographic data is essential for establishing a diagnosis or planning surgery.

Modelling from sparse and misaligned data requires solving the three inherently related problems of registration, segmentation, and interpolation. We propose a new method to integrate these stages in a level set framework. Registration is particularly challenging by the limited number of intersections present in a sparse dataset, and interpolation has to handle images that may have very different appearances. Hence, registration and interpolation exploit segmentation information, rather than pixel intensities, for increased robustness and accuracy. We achieve this by first introducing a new level set scheme based on the interpolation of the level set function by radial basis functions. This new scheme can inherently handle sparse data, and is more numerically stable and robust to noise than the classical level set. We also present a new registration algorithm based on the level set method, which is robust to local minima and can handle sparse data that have only a limited number of intersections. Then, we integrate these two methods into the same level set framework.

The proposed method is validated quantitatively and subjectively on artificial data and MRI and CT scans. It is compared against a state-of-the-art, sequential method comprising traditional mutual information based registration, image interpolation, and 3D or 4D segmentation of the registered and interpolated volume. In our experiments, the proposed framework yields similar segmentation results to the sequential approach, but provides a more robust and accurate registration and interpolation. In particular, the registration is more robust to limited intersections in the data and to local minima. The interpolation is more satisfactory in cases of large gaps, due to the method taking into account the global shape of the object, and it recovers better topologies at the extremities of the shapes where the objects disappear from the image slices. As a result, the complete integrated framework provides more satisfactory shape reconstructions than the sequential approach.

Declaration

I declare that the work in this dissertation was carried out in accordance with the Regulations of the University of Bristol. The work is original, except where indicated by special reference in the text, and no part of the dissertation has been submitted for any other academic award.

Any views expressed in the dissertation are those of the author and in no way represent those of the University of Bristol.

The dissertation has not been presented to any other University for examination either in the United Kingdom or overseas.

SIGNED: DATE:

Acknowledgements

I would like to begin by thanking my supervisor Majid Mirmehdi, for all his guidance and sage advice, and making sure I got through the PhD. I would also like to thank Nishan Canagarajah and Alin Achim, for very useful and essential comments, helping to keep me on the right track. My special thanks go to Xianghua Xie, who has given me so much advice and help throughout the PhD, and to Mark Hamilton for his high availability and fruitful discussions. Thanks also to my former supervisors at Strasbourg and Honolulu, Christian Boily, Isabelle Scholl, and Shadia Habbal, for setting me on the research path in the first place.

I owe a large amount of gratitude to my family and to Damien, for their support and encouragement throughout the PhD. I particularly appreciate their understanding and their not minding all the missed birthdays and events.

I am enormously grateful to my fantastic team of proofreaders, whose insightful comments and invaluable suggestions made this thesis into what it is. They are, in no particular order: Osian Haines, Toby Perrett, David Hanwell, Austin Gregg-Smith, Jack Greenhalgh, Teesid Leelasawassuk, and Damien Bonneau.

I would like to thank all of my friends in the lab, and in Bristol generally, who have made the past four years so enjoyable. The PhD would have been a very different experience without their friendship and support. This includes, but is not limited to: Nicola, Osian, José, Sion, David, Toby, Jack, Austin, Oliver, Teesid, Elena, Dima, John, Jorge, Josh and Liz, Raffaele and Eleonora, Alberto and Daniela, James, Karl, and Tom.

Finally, I would like to thank the authors of [14] who allowed me to compare to their method by kindly both running their software on my data and making their code available for further experiments. The CT-scan datasets of bones used in the experiments were from the Laboratory of Human Anatomy and Embryology, University of Brussels (ULB), Belgium.

Publications

The work described in this thesis has been presented in the following publications:

1. Adeline Paiement, Majid Mirmehdi, Xianghua Xie, Mark Hamilton. Simultaneous level set interpolation and segmentation of short-and long-axis MRI. *Proceedings of Medical Image Understanding and Analysis (MIUA)*, pp. 267–272, July 2010.

2. Adeline Paiement, Majid Mirmehdi, Xianghua Xie, Mark Hamilton, Integrated Segmentation and Interpolation of Sparse Data. *IEEE Transactions on Image Processing (IEEE-TIP)*, Vol. 23, Issue 1, pp. 110–125, January 2014.

3. Adeline Paiement, Majid Mirmehdi, Xianghua Xie, Mark Hamilton, Registration and Modeling from Sparse and Misaligned Image Volumes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE-PAMI)*, Submitted.

Contents

List of Figures

List of Tables

List of Acronyms

AAM	Active Appearance Model
ASM	Active Shape Model
CT	Computed Tomography
ECG	ElectroCardioGram
FFT	Fast Fourier Transform
FMM	Fast Multipole Method
ICM	Iterated Conditional Modes
IReSISD	Integrated Registration, Segmentation, and Interpolation for Sparse Data
ISISD	Integrated Segmentation and Interpolation for Sparse Data
LA	Long-Axis
LV	Left Ventricle
MCF	Mean Curvature Flow
MI	Mutual Information
MLE	Maximum Likelihood Estimator
MPLE	Maximum Pseudolikelihood Estimator
MQ	Multiquadric
MRF	Markov Random Field
MRI	Magnetic Resonance Imaging/Image
NMI	Normalised Mutual Information
ODE	Ordinary Differential Equation

PC	Piecewise Constant
PDE	Partial Differential Equation
RBF	Radial Basis Function
RGB-D	Red Green Blue + Depth
ROI	Region of Interest
RV	Right Ventricle
SA	Short-Axis
SDF	Surface Diffusion Flow
SM+[14]	Sequential Method implemented with [14]
SPD	Strictly Positive Definite
SR+ISI	Sequential Registration with Integrated Segmentation and Interpolation
SR[33]	Extended use of Sequential Registration [33]

List of Symbols

IReSISD

a_1 Angle of rotation of 1°.

a_2 Angle of rotation of 0.1°.

α_n Angle of rotation of sequence n around the x-axis.

β_n Angle of rotation of sequence n around the y-axis.

$\widetilde{\frac{\partial g_i^n}{\partial t}}_{min}$ Minimum displacement amongst all sequences for the i^{th} component of g in absolute values.

$\mathbb{D}$ Number of dimensions.

g^n Registration function of sequence n.

γ_n Angle of rotation of sequence n around the z-axis.

λ_n Estimation of misalignment of sequence n with all other sequences.

λ_n^{global} Estimation of misalignment of sequence n with all other sequences.

offset_i Offset removed from the i^{th} component of g for all sequences.

$\mathit{offset}_{\mathbf{T}}$ Offset removed from $\mathbf{T}$ for all sequences.

$\mathit{offset}_{\boldsymbol{\theta}}$ Offset removed from $\boldsymbol{\theta}$ for all sequences.

$\Omega_{\neq}^n$ Domain where ϕ and the speed computed from the images of sequence n have opposite signs.

$\Omega_{(S^n \circ g^n) \times (S^m \circ g^m) < 0}$ Domain where images of sequence n and images of all other sequences m produce speeds of opposite signs for the contour C.

Ω^n Domain of the images of sequence n.

$\Omega_{\phi > 0}$ Domain where ϕ is positive.

P^n Number of points used in the computation of the update of the registration of sequence n.

R^n Rotation matrix of sequence n.

$\mathit{rotation}^n$ Amount of rotation for aligning sequence n at the current iteration.

S_n Speed imposed on the contour by the data of sequence n.

shift^n Amount of shift for aligning sequence n at the current iteration.

$\widehat{T_i^n}$ Initialisation for T_i^n.

$\widetilde{\frac{\partial \mathbf{T}^n}{\partial t}}$ Scaled $\mathbf{T}^n$ according to λ_n.

$\mathbf{T}^n$ Translation vector of sequence n.

$\boldsymbol{\theta}^n$ Vector of the rotation angles of sequence n.

$\widehat{\theta_i^n}$ Initialisation for θ_i^n.

$\widetilde{\frac{\partial \boldsymbol{\theta}^n}{\partial t}}$ Scaled $\boldsymbol{\theta}^n$ according to λ_n.

ISISD

$\hat{\boldsymbol{\alpha}}$ Fourier transform of $\boldsymbol{\alpha}$.

α_i^l Other notation for α_i that highlights that its associated RBF is of flatness level l.

β Slope parameter of the inverse multiquadric RBF that is set to the number of dimensions in our implementation.

D Maximum distance in pixels that the contour C is allowed to travel during one iteration.

δ_ϵ Approximation of the Dirac delta function.

$\frac{\partial \phi}{\partial t}_{conv.}$ Speed of the level set function ϕ as computed by a conventional variational level set method.

ϵ Width of δ_ϵ.

γ Flatness parameter of the inverse multiquadric RBF.

H_l Modified Heaviside function that is equal to 1 inside Ω_l and 0 outside.

$\boldsymbol{J}$ Jaccard similarity coefficient.

l_i Flatness level of the i^{th} RBF.

M_{01} Number of voxels that are outside the first region and inside the second one.

M_{10} Number of voxels that are inside the first region and outside the second one.

M_{11} Number of voxels that are inside both regions.

ω Frequency coordinate in the Fourier domain.

Ω_l Domain of the control points associated with RBFs of flatness level l.

$\hat{\phi}$ Fourier transform of ϕ.

$\hat{\psi}$ Fourier transform of ψ.

ψ^l RBF of flatness level l.

S Simpler notation for $\frac{\partial \phi}{\partial t}_{conv.}$.

S_{data} Speed for the contour C computed from image data.

S_{geom} Speed for the contour C computed from geometric considerations.

Level Set Based Registration

$\hat{C}$ Contour in registered image.

$\hat{f}$ f in the registered image.

$\hat{f}_{in}$ f_{in} in the registered image.

$\hat{f}_{out}$ f_{out} in the registered image.

g Registration function that defines a mapping between C and $\hat{C}$.

g_i i^{th} component of g.

g' Jacobian of g.

M Scaling matrix.

m Determinant of M.

R Rotation matrix.

R_θ Matrix of rotation angle θ.

$\mathbf{T}$ Translation vector.

T_i i^{th} component of the translation vector.

$\mathbf{u}_i$ Unit vector oriented in the i^{th} direction.

Level Set Segmentation

C Active contour.

c_{in} Mean intensity of the image inside C.

c_{out} Mean intensity of the image outside C.

δ Dirac delta function.

E Segmentation energy for the level set.

E_{PC} Segmentation energy formulation for a piecewise constant level set model.

E_{reg} General region-based segmentation energy formulation for a level set.

$E_{reg,C}$ General region-based segmentation energy formulation for a level set with a constraint on the properties of the contour.

$\mathcal{F}$ Functional that defines energy E.

F Speed of contour C in the direction of its normal.

f Difference $f_{in} - f_{out}$ that may be seen as a speed for the contour.

f_c Function that describes the contour C.

f_{in} Function that describes the image inside C.

f_{out} Function that describes the image outside C.

F_{PC} Speed for the contour C as derived from energy E_{PC}.

$F_{reg,C}$ Speed for the contour C as derived from energy $E_{reg,C}$.

H Heaviside function.

I Image intensity.

κ Curvature of C.

$\mathbf{N}$ Normal to contour C pointing outwards.

ν_c Weight of the contour term in $E_{reg,C}$.

ν_{in} Weight of the inside region term in E_{reg} or $E_{reg,C}$.

ν_{out} Weight of the ouside region term in E_{reg} or $E_{reg,C}$.

Ω Domain of the image or volume being segmented.

Ω_{in} Region inside the contour C.

Ω_{out} Region outside the contour C.

$P_{in}(I)$ Probability of intensity I to belong to the object.

$P_{out}(I)$ Probability of intensity I to belong to the background.

ϕ Level set embedding function.

t Time.

Modelling from RGB-D Data

$\mathfrak{C}$ Confidence in the choice of speed.

$\overrightarrow{CP}$ Unit vector from the camera to the data point.

$\mathfrak{d}$ Unsigned distance to the data point in voxels.

$S_{\mathfrak{d}}$ Speed for the contour C based on $\mathfrak{d}$.

τ Parametric coordinate along a line-of-sight.

τ_{ls} First intersection of the level set surface with the line-of-sight.

MRF Based Image Interpolation

C_i Clique of order i.

Δ_k Unit quantised transformation.

E_{MRF} Energy of the MRF.

E_{OL} Observation likelihood energy of the MRF.

E_P Prior energy of the MRF.

F_i either I_i or $|\nabla_M I_i|$.

G Gaussian with standard deviation σ_G.

I_i Image of index i.

J Jacobian of u.

l Resolution level.

λ Weight of the third potiential term in energy E.

$\nabla_M I_i$ Gradient of the intensity of image I_i after convolution with a Gaussian kernel M of standard deviation σ_M.

P Probability.

$\mathbf{r}$ Coordinate in an image.

$\mathbf{r}_{I_0,b}$ **and** $\mathbf{r}_{I_1,b}$ Corresponding points in I_0 and I_1 respectively according to the backward transformation between the two images.

$\mathbf{r}_{I_0,f}$ **and** $\mathbf{r}_{I_1,f}$ Corresponding points in I_0 and I_1 respectively according to the forward transformation between the two images.

$\mathbf{r}_\tau$ Position of a gap point located at a fraction τ of the distance between the two images.

ρ Radius of a region around a site.

$r_{i,s}$ i^{th} component of $\mathbf{r}_s$.

$\mathbf{r}_{k_s}$ Coordinate at grid site s in image I_1.

$\mathbf{r}_n$ A spatial shift of n pixels.

$\mathbf{r}_s$ Coordinate at grid site s in image I_0.

s Grid site.

σ_0 Weight of the first potiential term in energy E.

σ_1 Weight of the second potiential term in energy E.

σ_G Standard deviation of the Gaussian G that is equal to $\rho - \frac{1}{2}$.

u Deformation fonction.

V_i i^{th} potential term of the MRF energy E.

W_i Confidence in the selection of two corresponding points in the two images.

$\boldsymbol{X}$ Field of quantised deformations X_s.

$\boldsymbol{x}$ Value of $\boldsymbol{X}$.

ξ Weight of the fourth potiential term in energy E.

X_s Quantised deformation at grid site s.

x_s Value of deformation at site s.

NMI Based Registration

$\mathcal{H}$ The Shannon-Wiener entropy measure.

p_i Probability of a variable to take the value indexed by i.

Prior Knowledge Based Modelling of the heart

d Nominal distance between the endo- and epicardium.

ν Weight of $S_{coupling}$ in the computation of S.

ϕ_1 Level set function that segments the left ventricle cavity.

$S_{coupling}$ Speed for the contour C that inforces a coupling between the endocardium and the epicardium.

S_{image} Speed for the contour C computed from the image data.

w Transition width between prior knowledge and image driven speeds.

RBF

$\mathbf{A}$ Matrix of values of the RBFs at the control points.

$\boldsymbol{\alpha}$ Vector of the weight coefficients.

α_i Weight of the i^{th} RBF in the interpolation.

$\mathbf{B}$ Vector of speeds of the level set function at the controls points and expressed in terms of α and $\mathbf{\Psi}$.

β^n Uniform symmetric B-spline of degree n.

$\mathbf{f}$ Matrix of values of the interpolated function at the control points.

f_i Value of the interpolated function at the i^{th} control point.

γ_i Flatness coefficient of the i^{th} MQ spline RBF.

$\mathbf{H}$ Matrix defining the linear system of $N + 4$ equations of the RBF interpolation.

h Spacing of the control point grid in [1].

N Number of RBFs and control points used in the interpolation.

$\mathbf{P}$ Matrix of coordinates of the control points.

p First-degree polynomial which accounts for the linear and constant portions of ϕ in its RBF interpolation.

p_i Coefficients of p.

$\mathbf{\Psi}(\mathbf{x})$ Vector of values of the RBFs at a given point $\mathbf{x}$.

$\psi(\mathbf{x})$ n dimensional RBF centered on 0 defined as equal to $\tilde{\psi}(\|\mathbf{x}\|)$.

ψ_i RBF centered at control point $\mathbf{x}_i$.

$\tilde{\psi}$ Radial Basis Function: $\mathbb{R}^+ \to \mathbb{R}$ centered on 0.

ς Parameters of 2D anisotropic Gaussian RBFs.

ς_{ij} j^{th} parameter of the i^{th} 2D anisotropic Gaussian RBF.

$\mathbf{x}_i$ Control point of an RBF.

Chapter 1

Introduction

Recovering an object's shape from 3D or 4D measurements requires *segmenting* the imaged volume in order to delineate the object's boundaries. Its large range of applications includes, but is not limited to, characterising objects' shapes and dynamics, visualising, archiving, and distributing them. A well known example of an archiving and distribution application is the Digital Michelangelo Project [27], which aimed at preserving Michelangelo's sculptures by recording digital copies. This project also enabled a large community of experts worldwide to work based on the models, without the need to obtain physical access to the sculptures. Similarly, in medical imaging, it may be desirable to archive the shape and dynamics of organs or tumour in order to monitor their temporal evolution in the long term. Medical imaging also uses object modelling extensively to help establish a diagnosis or plan surgery, by taking advantage of the easier visualisation and characterisation provided by this technique. Thus, the shape and dynamics of an organ or a tumour, for example, may be visualised in 3D or 4D and from arbitrary view points. As will be illustrated in Chapter 2 and Appendix C, such visualisation would not be possible by using the raw image data directly. The shape and dynamics of an organ or tumour may also be easily characterised after modelling them, *e.g.* by computing their volume and how it changes over time.

Due to the great variety of applications and imaging modalities, object shape measurements may have various forms, from 2D slices which span the imaged volume in tomography, to 3D point clouds acquired by range sensors. This work focuses on the modelling of 3D and 4D objects from *sparse*[1] and *misaligned* medical tomographic data, and aims to be general enough to handle data from any medical tomographic modality, regardless of its type and spatial configuration. Sparse and misaligned data are commonly found in these modalities, and raise a number of issues which make the segmentation of such data more challenging than with ideally dense and well-aligned measurements.

The first issue of such datasets is that misaligned data provide incoherent information to the segmentation stage, since they locate the object at different positions. Misalignments in 3D tomographic medical images may be caused by movements of the imaged object. They are particularly frequent due to the difficulty for the patient to remain perfectly still during the full time of the acquisition, which takes typically 30 min for a Magnetic Resonance Imaging (MRI) scan for example. The images acquired (*e.g.* Fig. 1.1) must be *registered* prior to any other processing. However, a second issue arises when the data are sparse and do not span the whole imaged volume, since this registration is further complicated by the limited number of intersection points available to match the different acquisitions together.

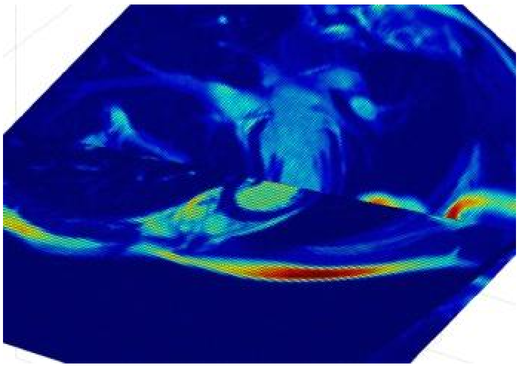


Figure 1.1: *Example of misaligned dataset – two orthogonal horizontal and vertical medical images of the chest acquired separately. The round shape in the horizontal image, and the elongated shape in the vertical one, are cross sections of the left ventricle of the heart. Their misalignment is due to the patient's movements between the acquisitions.*

A third issue of our tomographic datasets is that many acquisition protocols do not measure the full imaged volume, and the resulting 3D and 4D datasets contain gaps of various extents, which do not provide the data support needed by segmentation. This occurs often in medical imaging, especially with MRI scans, where volumes are commonly made up of thick and widely spaced slices, as illustrated in Fig. 1.2. In particular, medical imaging protocols often require integrating the signal over a thick slice of the volume

[1] in the sense that the data are incomplete and the images do not cover the full volume

in order to improve its quality by increasing the signal to noise ratio. This sets an undesirable limit on the possible slice spacings. Thus, in 4D cardiac MRIs produced by a 1.5 Tesla scanner, a typical slice thickness is 7 mm, hence slice spacing is often 7 mm or larger. In addition, clinicians commonly choose to acquire largely spaced slices in order to decrease the acquisition time and reduce discomfort for patients who are asked to remain perfectly still until the end of the acquisition, and sometimes to hold their breath repeatedly for around 10 s and occasionally up to 30 s at a time. Such data acquisition protocols result in very sparse volumes containing large gaps between the 2D slices, with widths of around 8 mm to 16 mm for in-plane pixel sizes of ∼1.8 mm for cardiac cine MRIs. Therefore, *interpolation*, either of the images or of the object's shape, is needed in order to recover the closed 3D shape of the object.

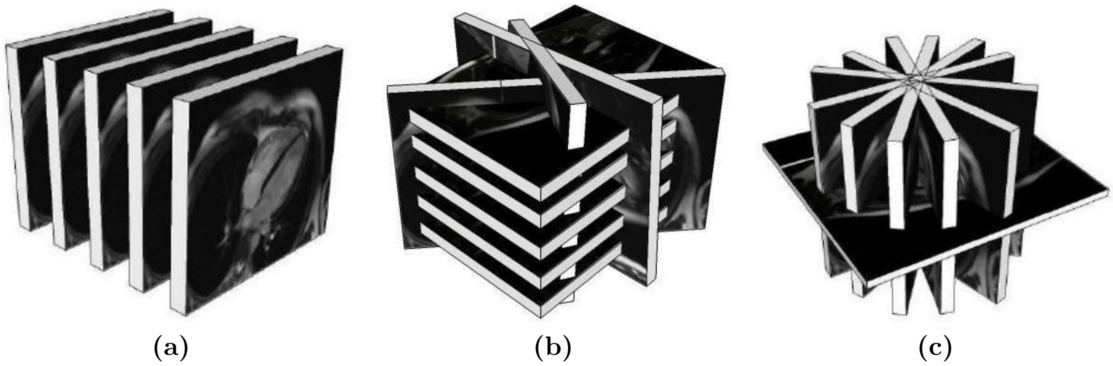


Figure 1.2: *Examples of sparse datasets – slice thickness is imposed by the scanner's limitations, while spacing is adjusted as a compromise between accuracy and patient comfort. (a) simple stack of parallel slices, (b) standard spatial configuration of a cardiac MRI: a stack of parallel short-axis slices plus a few long-axis ones, and (c) radial dataset.*

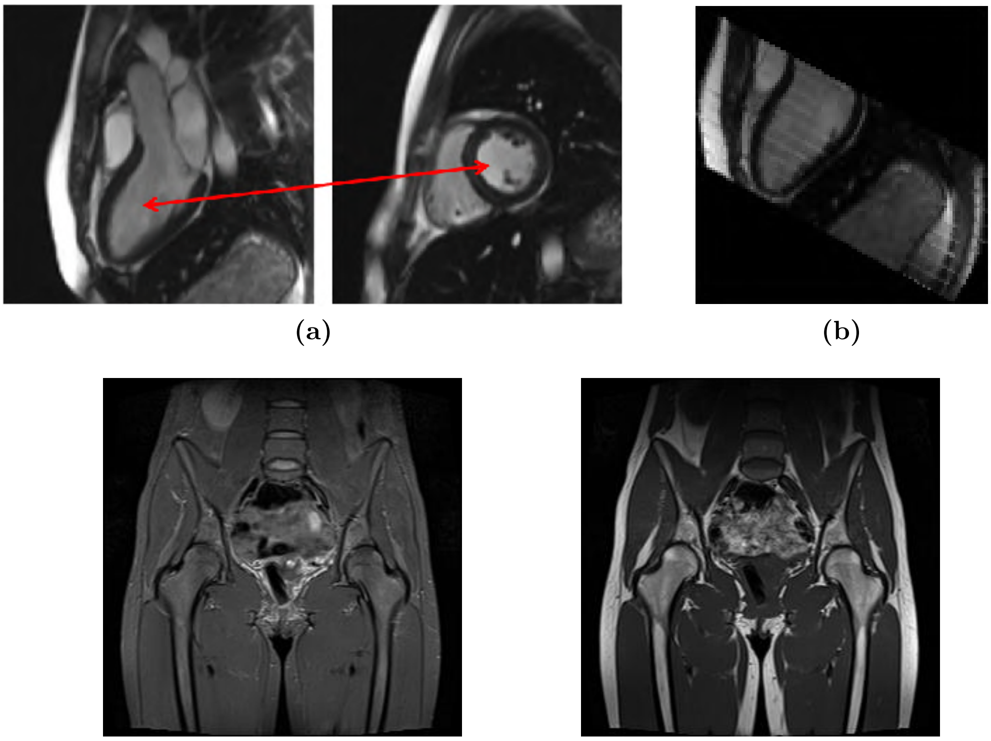
It is important to note that the three processes required to model objects from sparse and misaligned data – namely registration, segmentation, and interpolation – are closely intertwined. Several measurements may drive the segmentation of a volume concurrently, but only on the condition that these measurements are aligned with each other, since otherwise they would provide incoherent information to the segmentation process. Similarly, sparse volumes do not contain enough information to drive the segmentation everywhere. As such, segmentation of sparse and misaligned data cannot be performed without registration and interpolation. Similarly, registration is made difficult by the sparseness of the data, and shape information yielded by the segmentation would drive it more efficiently and reliably than raw data values. Likewise, interpolation can also gain robustness by exploiting shape information and segmentation results. It naturally follows that these three processes would benefit from being integrated in a single framework. Such integration has been attempted in the past, as will be discussed in Chapter 3, and will be a focus of this work.

Further, this work aims to be general enough to be used on a variety of data, regardless of their spatial configuration. Depending on the modality, the slices which make up a tomographic medical volume may have greatly varying spatial configurations, since a large variety of imaging protocols exist, producing from simple stacks of parallel and equally spaced images (Fig. 1.2a), to more complicated spatial configurations (Figs. 1.2b,c). In addition, the positions and orientations of medical slices must be adapted to the morphology of the patient, and therefore vary in number, location, orientation, and spacing, between both patients and imaged organs. Theoretically, it is reasonable to exploit all available information in order to obtain the most robust modelling possible. In practice, this approach raises multiple issues, especially when interpolating between sets of slices having different orientations (Figs. 1.2b,c for example), as will be discussed in more detail in Chapter 3. As a result, exploiting all available information is a difficult task, and several existing methods fail to achieve this, as in the case of cardiac cine MRI, where slices of unsupported orientations and the valuable information they contain are often discarded, *e.g.* in [13, 14]. The method proposed in this work is able to process slices having arbitrary positions and orientations, and is therefore general enough to process data of any spatial configuration.

Another problem addressed by this work is the different gains and contrasts which can occasionally be found in the same dataset. In particular, cine MRI acquisition protocols tend to produce images having different gains and contrasts, especially between different image orientations. A common example is cardiac MRI, as illustrated at the top row of Fig. 1.3. Images from different modalities may also be found in the same dataset, as shown in the bottom row of Fig. 1.3. The different appearances of such images makes their simultaneous processing even more challenging, especially when trying to interpolate pixel intensities in the gaps in the data.

In summary, the challenges addressed by this work are:

1. the misalignments in the data, which provide conflicting information to the segmentation and interpolation stages,
2. the difficult registration due to the sparsity of the data,
3. the gaps in the datasets, which make interpolation necessary in order to recover a closed shape of the object,



(a) (b)

(c) (d)

Figure 1.3: *Images having different gains and contrasts within the same dataset. Top row: short- and long-axis slices of a cardiac MRI dataset. (a) long-axis (left) and short-axis (right) views. The red arrows point to corresponding areas in the two images with different respective intensities. (b) long-axis view with intersecting short-axis slices superimposed, which have different gains and contrasts. Bottom row: pelvis MRIs acquired at the same location with different scanner parameters: (c) STIR, and (d) T1 protocols.*

4. the different gains and contrasts in the images of the same dataset,

5. the great variety of possible spatial configurations, and

6. the inter-dependency of the three processes of registration, segmentation, and interpolation.

1.1 Overview of the Proposed Method

This work addresses the challenges listed previously by adopting a new strategy, which integrates the three stages of registration, segmentation, and interpolation into a single framework based on a level set method. Level set methods are known to be flexible since they can segment arbitrary shapes and topologies, without the need for a training phase, and they can use any segmentation criteria. This flexibility allows us to model a great variety of shapes from data from various modalities and that have different spatial configurations.

In Chapter 4, we first propose a new *integrated segmentation and interpolation* level set framework which inherently handles sparse data of arbitrary spatial configurations and appearances. This new framework is general purpose, and it performs interpolation at the same time as segmentation, thanks to it's level set formulation. More specifically, it interpolates the shape of the object in the gaps in the data through the level set interface, while this interface also segments both the data and the volume. The evolution of this level set is driven by all available data points, with no consideration of their location. It can also make use of any type of data, and of multiple types of data simultaneously, through the choice of an appropriate segmentation algorithm. The interpolation properties of this framework are evaluated on a variety of medical datasets.

Then in Chapter 5, we add *registration* to our integrated level set framework. We first present a level set based registration method, which is suited for aligning non-overlapping sparse data that may suffer from a limited number of intersections. We then extend this new registration method to a global formulation in order to increase its robustness to local minima. We also evaluate the registration and the complete reconstruction of our fully integrated framework on multiple data from different modalities.

Finally, in Chapter 6, we propose an application of our general purpose integrated framework to 4D modelling of the heart from cardiac cine MRIs. We exploit the cyclical nature of the cardiac motion to improve consistency of the modelling. Simple segmentation criteria based on intensity are used to segment both cavities of the heart, and we demonstrate the use of prior knowledge in our framework to constrain the more difficult segmentation of the myocardium. To the best of our knowledge, this is the first time that the heart is modelled using a fully integrated level set framework. This has the advantages of producing robust registration and interpolation, as well as very detailed shapes. This also allows the segmentation and modelling to be improved when novel segmentation algorithms are developed, since they can be easily integrated in the framework.

1.2 Contributions

The original contributions of this work are the following:

1. We introduce and implement the concept of interpolating a level set segmenting surface rather than the data directly.

2. We introduce a new level set scheme, based on the interpolation of its implicit function by Radial Basis Functions (RBFs). Such RBF based interpolation of a level set function was proposed in the past for segmenting images, but this method does not support sparse datasets, as will be explained in Chapter 3. Our proposed framework combines, for the first time as far as we know, the two uses of RBFs, namely interpolation and segmentation, and it can therefore handle sparse data inherently. This scheme is also more numerically stable, and its segmentation is more robust to noise, than with classical level set methods.

3. We propose a new level set based registration method for sparse and non-overlapping data, which exploits segmentation information. Local and global formulations are proposed and combined, which provide better accuracy and robustness to local minima respectively.

4. We integrate the three processes of registration, segmentation, and interpolation in a single framework based on level sets. To the best of our knowledge, this is the first time that these three stages are combined in a level set framework.

5. The proposed method is very general and can model objects

   - of any shape, since it is not limited to a given class of objects like model based methods,
   - from data having any spatial configuration, in particular any number of 2D slices having any position and orientation,
   - from data from any modality, including multimodality data, given an appropriate choice of segmentation method, *e.g.* using edges, intensity or prior knowledge
   - using any strictly positive definite (SPD) RBF.

1.3 Thesis Structure

The thesis is laid out as follows:

- In Chapter 2, we present the sparse and misaligned medical tomographic data which are used in this work.
- In Chapter 3, we review existing methods for object modelling from sparse and misaligned sets of slices. We also describe in more detail related methods which inspired the presented work, and methods which we will compare against in our evaluations of our proposed framework.
- In Chapter 4, we introduce our new level set framework for simultaneous segmentation of sparse image volumes and interpolation of the shape of the object, and we provide both quantitative and qualitative evaluations of the quality and robustness of interpolation on medical tomographic data.
- In Chapter 5, we introduce our new simultaneous registration and segmentation method, and we integrate it to the level set framework of Chapter 4 in order to add registration to it. We evaluate both the registration method and the complete framework on various medical tomographic data.
- In Chapter 6, we propose an application of our framework to modelling the heart from cardiac cine MRI.
- Chapter 7 concludes this thesis and discusses possible extensions of this work.
- In Appendix A, we demonstrate the possibility to apply our framework to modelling from 3D point clouds and RGB-D data.
- In Appendix B, we propose another use of the proposed framework for image inpainting and the restoration of videos.
- Appendix C presents some examples of raw and segmented cardiac cine MRI datasets.

Chapter 2

Data

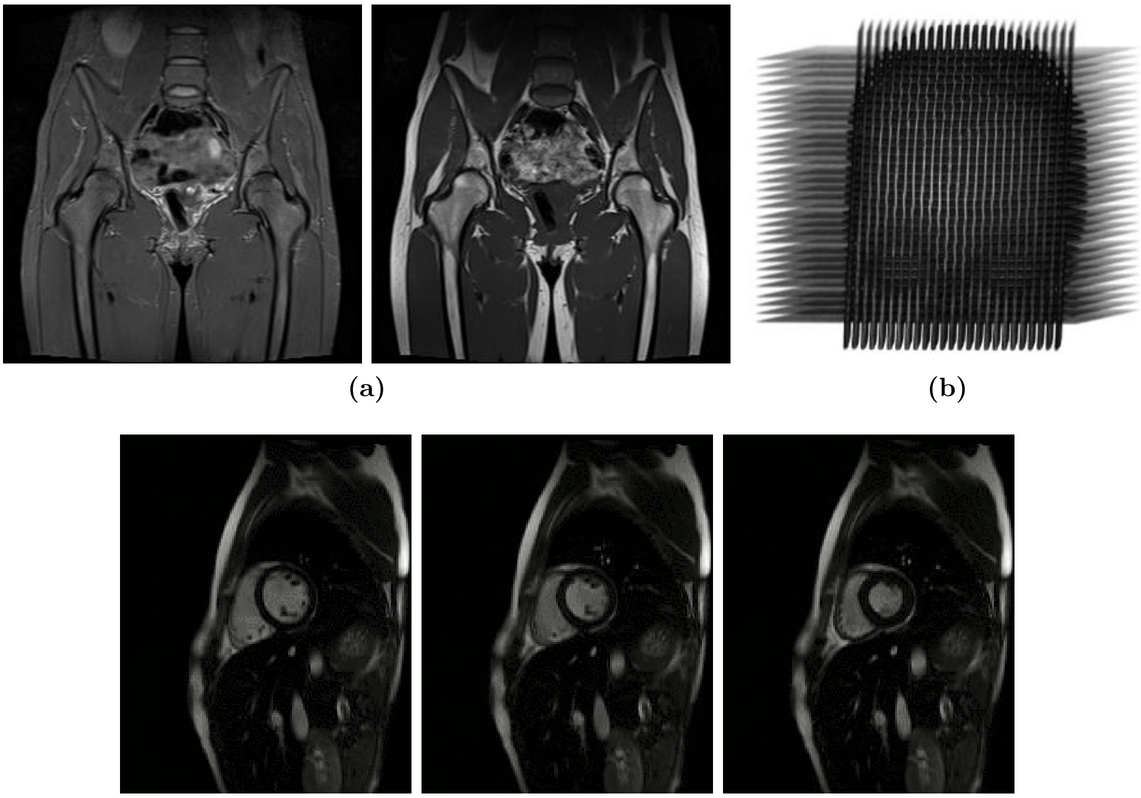
In this chapter, the sparse and misaligned medical tomographic data that is used to illustrate and evaluate the proposed framework is presented. Typical examples of such data are MRI and Computed Tomography (CT) data, which will be described in Section 2.1, with an emphasis on cardiac cine MRI. The characteristics of the data used in our experiments in Chapters 4, 5, and 6 will be provided in Section 2.2.

2.1 Sets of 2D Medical Tomographic Images

3D and 4D medical tomographic data produced by MRI and CT scanners are in the form of several 2D images which span the imaged volume. One dataset is generally made up of several independent sets of images acquired during a single run of the acquisition device. We refer to such independent sets of images as *sequences*. The data may be acquired in 3D or 4D and using different techniques, thus leading to a range of different types of sequences. They may be broadly classified between "spatial sequences", when their images are slices[1] spanning the whole volume and considered to be acquired at the same time, and "temporal sequences", when their images are several time-frames of a single slice. We identify three types of sequences resulting from different image acquisition scenarios:

[1] where "slice" denotes a spatial position and orientation of an image in the 3D or 4D volume

- Stacks of slices acquired with different scanner parameters, as illustrated in Fig. 2.1a. Such sequences qualify as spatial sequences.
- Stacks of slices acquired with different slice orientations, as in Fig. 2.1b. These sequences are also spatial sequences.
- Temporal stacks of time-frames of a single slice, as in Fig. 2.1c. These are temporal sequences.



(a)

(b)

(c)

Figure 2.1: *Examples of datasets made up of independent sequences. (a) and (b) spatial sequences: (a) one slice position of two MRI sequences of pelvis with two different scanner parameters – left: Short Time Inversion Recovery MRI, and right: T1 MRI, (b) MRI scan of a head with two different slice orientations (axial and sagittal), and (c) three time-frames of a temporal sequence of a cine cardiac MRI at one slice position.*

The sequences of a dataset, regardless of their type, are acquired independently and successively, during consecutive and short runs of the acquisition device. During the acquisition of each sequence, the patient is required to be perfectly still, and occasionally to hold their breath. This is achieved easily since the acquisition time of one sequence is very short (of the order of a few seconds). Therefore, all the images of a sequence, spatial or temporal, can reasonably be assumed to be spatially aligned with each other and can be treated as a rigid dataset, and are moved in unison during registration.

However, spatial alignment is not assured between different sequences. A detailed scan requires typically 30 minutes to cover the whole imaged volume, during which it is difficult to remain perfectly still. In addition, if the subject is required to repeatedly hold their breath, as it is usual in cardiac MRI, the breath holds may not be the same. These two effects – patient movements and inconsistent breath-holds – commonly lead to different spatial positions of the region of interest, resulting in sequences being often shifted and rotated with respect to others, as shown earlier in Fig. 1.1.

Frequently, 3D and 4D medical data are acquired with large gaps between their slices in order to reduce the acquisition time. In the case of spatial sequences, this strategy reduces the duration of each acquisition, and the patient needs to stay still during shorter periods of time. In the case of temporal sequences, the total number of sequences is reduced. In both cases, the resulting gaps between the slices are generally of the order of a few pixels.

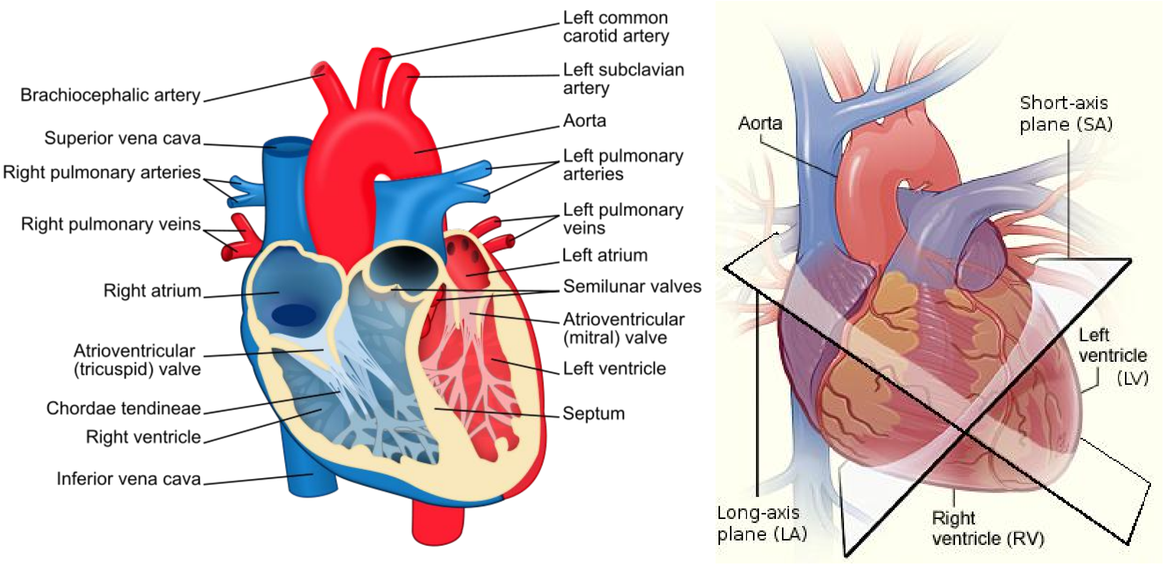
Cine MRI of the heart is an imaging protocol which is particularly subject to these high misalignments and large gaps. Therefore, it will be given special attention in the experiments in Chapter 6, and it is described in more detail next.

2.1.1 Cardiac Cine MRI

The modelling of the heart throughout a cardiac cycle provides essential information for establishing a diagnosis, for example measuring the volumes of the hearts chambers and their rates of change which allow assessing the ventricular and atrial function (see Fig. 2.2a for an outline of the heart's morphology). Cine MRI has placed itself as the most suited imaging modality for this purpose, due to it being harmless to the patient, its ability to image soft tissue, and its high in-plane spatial and temporal resolution.

Cardiac cine MRIs are 4D datasets made up of temporal sequences. The time-frames, as well as the slice positions and orientations, are chosen by the radiologist to match the dynamics and morphology of the patient's heart.

As illustrated in Fig. 2.2b, the orientation of cardiac slices follow the two principal orientations of the heart. Short-axis (SA) slices (Fig. 2.3a) cut the heart perpendicular to its main axis, and span the heart from the bottom (apex) to the valves or the atria, while long-axis (LA) slices cut the heart along its main axis, either through one ventricle (and possibly atrium) as in Fig. 2.3b, or through both ventricles (and atria) as in Fig. 2.3c.



(a) (b)

Figure 2.2: *Morphology based position and orientation of slices in cardiac cine MRI. (a) anatomy of the heart, seen as in cut in a long-axis plane – blue components indicate de-oxygenated blood pathways and red components indicate oxygenated blood pathways, (b) position and orientation of slices in cardiac cine MRI: examples of short- and long-axis slices. Note that other short-axis slices are acquired parallel to the currently displayed slice, while other long-axis slices may be acquired at any position perpendicular to the short-axis plane. The figure (a) is reproduced from [69], and the figure (b) is reproduced from [4] with modifications.*

In addition to being adapted to the morphology of the patient, the spatial configuration of cardiac cine MRIs depends also on the interest of the physician, who may want to examine the dynamics of a chamber, or to look more closely at a valve for example. Thus, different imaging protocols may be used which produce the various configurations that were presented in Fig. 1.2. Traditionally, cardiac cine MRIs are made up of a stack of parallel SA slices, that allow observing the left ventricle (LV), which is easily recognised on such slices due to its round cross-section. In particular, the myocardium (black circular muscle indicated by a red arrow in Fig. 2.3a) can be conveniently examined for possible defects. The right ventricle (RV), indicated by the green arrow in Fig. 2.3a, may also be segmented from these slices, although its boundaries look more fuzzy than those of the LV and are more challenging to delineate. LA slices are becoming increasingly popular for examining the apex of the heart and the valves (respectively the red arrow

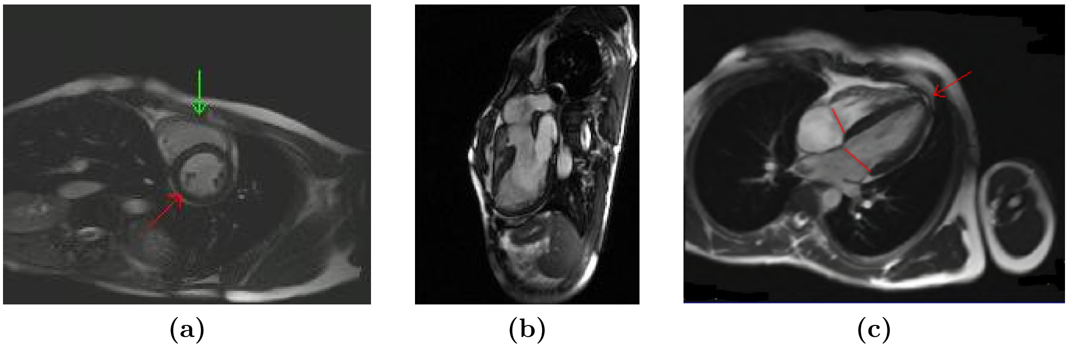


Figure 2.3: *Short- and long-axis cardiac cine MRI. (a) SA slice – the red arrow indicates the position of the myocardium and LV, and the green arrow points to the RV, (b) 2-chambers LA slice, (c) 4-chambers LA slice – the red arrow and lines highlight the apex and valves which are better visualised in LA slices.*

and lines in Fig. 2.3c), and it is increasingly common for about two to four of them to be acquired in addition to the traditional SA slices, as in Fig. 1.2b. Datasets made up of LA slices, such as the radial dataset of Fig. 1.2c, are also encountered more and more often.

All the time-frames of a single slice position are acquired during a single run of the MRI scanner, with the patient holding their breath in order to avoid image blurring due to respiration movements. Generally, 25 time-frames are acquired and cover the whole cardiac cycle. Their acquisitions are triggered by the electrocardiogram (ECG), so that the same time-frame in all the different temporal sequences always corresponds to the same moment in the cardiac cycle. In other words, the images are temporally aligned and do not require registration in the time dimension. Note that alignment is not necessarily (and is usually not) achieved in the spatial domain.

Indeed, remember that all the time-frames of a single slice position constitute a temporal sequence which is acquired separately from the others. The protocol for cardiac MRI acquisition is generally for the patient to breathe freely between the acquisition of two sequences, then to prepare for the acquisition of the next sequence by either filling or emptying their lungs, ideally in the exact same way for all acquisitions, and then to hold their breath during the acquisition of the sequence. Such protocol makes it very likely for the sequences to be misaligned with each other, due to patient movement during the free-breathing periods and inconsistent breath-holds. The latter is particularly frequent and happens when the remaining amounts of air in the lungs vary from one sequence to another, resulting in displacements of the heart inside the chest [36]. In the case of MRI acquired at end-expiration, the movements of the heart are commonly approximated by shifts in its SA plane, with very little shift along its long axis and rotation. For end-inspiration acquisitions, the translation of the heart in the LA direction and its rotation are more significant. Consequently, and in both cases, the different slices of the same dataset are very often misaligned with regards to others, as illustrated in Fig. 1.1, and they require registration.

A second characteristic of cardiac cine MRI is its incomplete coverage of the volume. The first cause of this lies in the physical limitation of the scanner, which requires integrating the images over a thick slice of the volume in order to improve the signal quality by increasing the signal to noise ratio. This sets an undesirable limit on the possible slice spacings. Thus, for a 1.5 T scanner, a typical slice thickness is 7 mm, hence slice spacing is usually 7 mm or larger.

The second cause of incomplete coverage of the volume is the large gaps between cardiac cine MRI slices, in order to reduce the acquisition time and number of breath-holds, and thus limit the patient's discomfort. Indeed, a full cardiac cine MRI scan takes usually $\sim$30 minutes, and a breath-hold typically lasts for around 10 seconds and occasionally up to 30 seconds. This prolonged period of immobility and the repeated breath-holds may be difficult for young and elderly patients, and for patients suffering from a cardiac disease. Therefore, radiologists routinely seek a compromise between a sufficient coverage of the imaged volume and a limited number of sequences together with a reduced acquisition time. Thus, a standard cardiac cine MRI contains $\sim$10 SA slices (this number varies with the size of the heart and the slice spacing) spaced by around 8 mm to 16 mm, for in-plane pixel sizes of $\sim$1.8 mm. Three or four additional LA slices may also be acquired. Therefore, cardiac cine MRI are often very sparse and require interpolation in order to make sense of the data and, for example, compute chamber volumes.

2.2 Datasets for Experiments

Our proposed framework will be assessed on a variety of artificial and real data in Chapters 4, 5, and 6. The artificial data will be used to evaluate accurately the interpolation and registration stages of our framework, while real data will provide a greater variety of shapes and more realistic test conditions. We now provide the detailed characteristics of both of these datasets, which are summarised in Table 2.1 at the end of this chapter.

2.2.1 Artificial Data

As highlighted in Chapter 1, the interdependency of segmentation with interpolation and registration is a key aspect of modelling from medical tomographic data and is a focus of this work. This interdependency makes it difficult to evaluate the interpolation and registration stages alone, and especially to assess their quality separately from the accuracy of the segmentation on real data. However, we note that as long as the segmentation is correct, the result of the interpolation and registration stages depends only on the spatial configuration of the images and on the shape of the reconstructed object. For this reason, we will first evaluate the interpolation and registration stages, in Sections 4.5 and 5.5 respectively, on artificial data that is very simply and reliably segmented, *e.g.* images made up of piecewise constant regions. To this end, we produce 3D datasets in which images are made up of two regions of constant intensity, inside and outside of the object to model, respectively, and we will process them using a segmentation model based on intensity.

We produce the piecewise constant datasets by extracting spaced slices from volumes containing several models. Our first model, which is used to test the interpolation quality, is a geometrical shape made up of a cylinder and a hemisphere (Fig. 2.4a). Both radial and axial slices are extracted from it. Two other models, used to test both interpolation and registration, are the shapes of two LV cavities of the heart, which are generated by segmenting two real cardiac MRI datasets, visually inspected for no visible misalignment and containing almost no gaps, hence requiring only a minimal amount of inter-slice interpolation (Figs. 2.4b and 2.4c). From each LV model, we make 16 datasets whose slice positions and orientations are the ones of 16 human volunteer heart MRIs. The resulting artificial cardiac MRI datasets are made up of a stack of 4 to 19 SA slices having pixel size ranging from 1.771 to 2.083 mm and slice spacings ranging from 5 to 20 mm, plus no to 12 LA slices with similar pixel sizes and various positions and orientations. Therefore, even if the slices of these two times 16 datasets are black and white images, they contain the shape of real hearts, and their positions and orientations are also realistic since they are taken from real cardiac MRIs.

Similarly, registration will also be evaluated on an artificial brain's ventricles model (Fig. 2.4d), also generated from a real MRI with no or negligible gaps and misalignments, and which is used to produced 9 artificial brain datasets, using the positions and orientations of slices of 9 real brain MRIs. They are made up of two spatial sequences of 11 to 18 axial slices, with pixel size of 0.781 to 0.9375 mm and spacings of 3.3 to 6.6 mm, and one or two spatial sequences of 13 to 20 sagittal slices with pixel size of 0.898 to 0.371 mm and spacings of 3.3 to 4.8 mm.

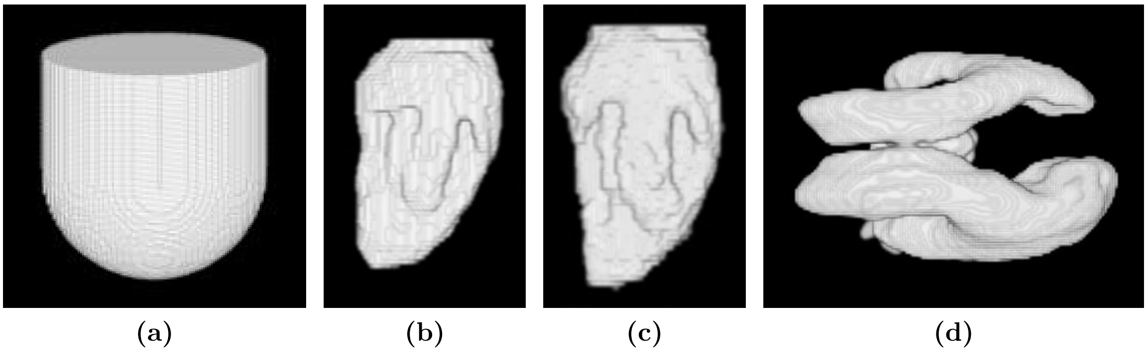


Figure 2.4: *3D models used to produce artificial datasets. (a) geometric model, (b) LV cavity model A, (c) LV cavity model B, (d) brain ventricles model.*

2.2.2 Real Data

We will also evaluate our proposed method on a collection of quasi-isotropic data from which a number of slices are removed in order to create gaps, and which are artificially misaligned. This will allow assessing quantitatively the interpolation and registration properties of our method using the segmentation of the original full-volume data as a ground-truth. These sets are one CT-scan of the acetabulum bone made up of a single spatial sequence with parallel slices spaced at 0.5 mm with pixel size of 0.49 mm, a CT-scan of the whole body also made up of a single spatial sequence with pixel size of 0.78 mm and gaps of 0.8 mm between the slices, from which we segment the LV cavity and kidney, and two MRI scans of a spherical phantom, each made up of one spatial sequence, with pixel sizes of 0.87 mm and 1.44 mm, and slice spacings of 0.9 mm and 1.5 mm respectively. For the interpolation tests, we make 8 datasets from each original set, 4 being made up of parallel slices spaced by 5, 10, 15 and 20 pixels respectively, and 4 with the same horizontal slices plus one additional vertical slice in the centre of the volume. For the registration tests, we create 4 datasets from each original set, each one made up of two stacks of axial and sagittal slices, with slice spacings of 5, 10, 15 and 20 pixels.

In our experiments, 17 cardiac cine MRIs will also be used. Sixteen datasets have the standard SA+LA configuration (Fig. 1.2b) and are made up of temporal sequences, with a stack of SA images and 0 to 12 LA images. The pixels range in size from 1.77 mm to 2.08 mm, while there are 5 mm to 20 mm gaps between the parallel SA images, and 25 time-frames. These datasets were acquired during clinical examinations of real patients, using a standard acquisition protocol. The last dataset has an axial configuration, similar to the dataset shown in Fig. 1.2c, and is made up of 12 LA slices plus one SA slice, with pixel size of 1.771 mm, and also 25 time-frames. All datasets have been acquired at end-expiration. The spatial configuration of one example of standard dataset is shown in Fig. 2.5. The images that make up this dataset are displayed in Figs. 8 to 14 of Appendix C, and for convenience the images of the first time-frame are presented in Fig. 2.6.

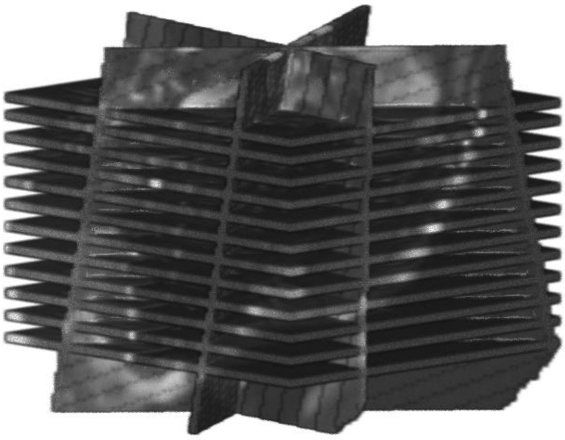


Figure 2.5: *Example of cardiac cine MRI made up of a stack of SA slices plus 3 LA slices – 3D view of the dataset at one time-frame. Note that the spatial configuration does not change between time-frames.*

Finally, in order to demonstrate the versatility of the proposed framework, we will also test our method on 9 sparse MRIs of neonatal brains made up of two spatial sequences that are stacks of T1- and T2-weighted images, where each stack is composed of parallel slices spaced at 3.3 mm with a pixel size of 0.78 mm. We will also use two sparse CT-scans of bones made up of a single spatial sequence each, whose parallel slices, spaced at 0.5 mm, have pixels of size 0.35 mm, and one MRI of the hip joints, made up of one T2-Weighted Fast Spin Echo axial (spatial) sequence, one Short Time Inversion Recovery coronal sequence, one T1-Weighted Spin Echo coronal sequence, and two T2-Weighted 2-D Multi Echo diagonal sequences focusing on each hip joint. All hip sequences are composed of 15 to 19 images, with pixel sizes ranging from 0.742 to 1.641 mm and spacings ranging from 3.3 to 6 mm. Examples of these images are provided in Figs. 2.7 to 2.9.

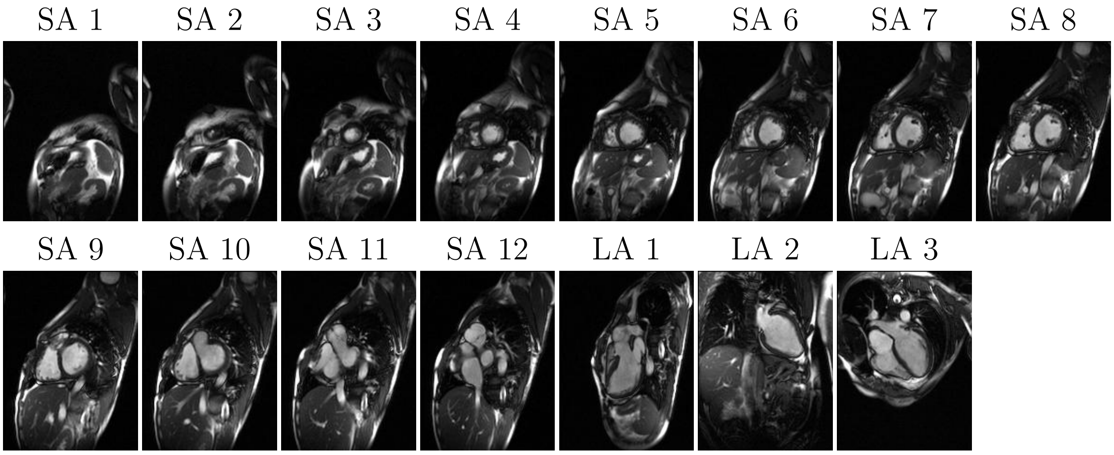


Figure 2.6: *Example of cardiac cine MRI made up of a stack of SA slices plus 3 LA slices – images of the first time-frame. Please refer to Figs. 8 to 14 of Appendix C for the complete dataset.*

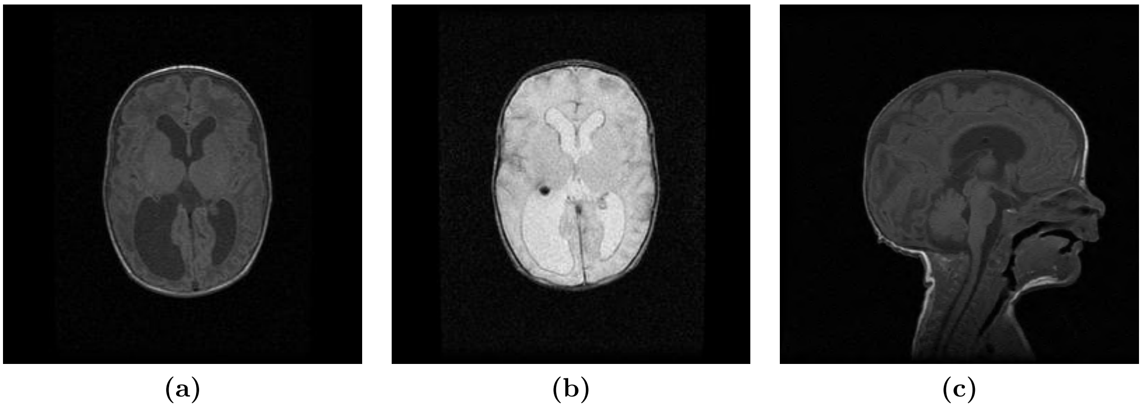


Figure 2.7: *Example of T1- and T2-weighted MRI of brain. Slices from the dataset of Fig. 2.1b. Central slices of the (a) T1-weighted, and (b) T2-weighted axial sequences, and of (c) a T1-weighted sagittal sequence. Note that not all datasets contain a sagittal sequence.*

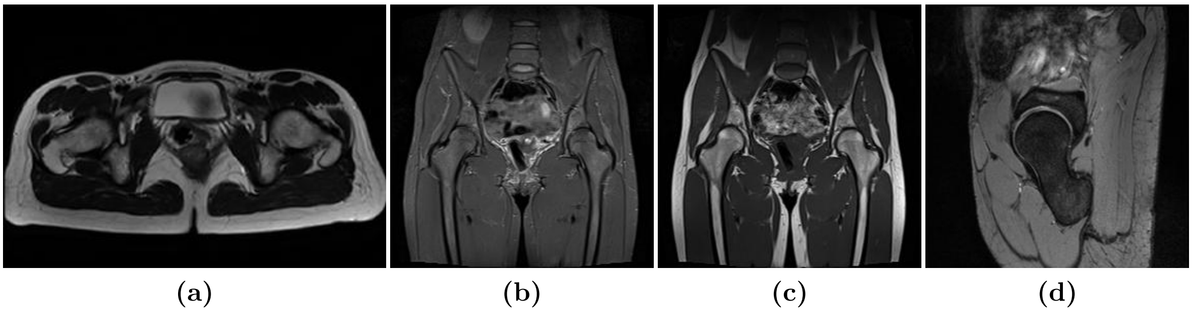


Figure 2.8: *Example of MRI of the hip joints: central slice of (a) a T2-Weighted Fast Spin Echo axial sequence, (b) a Short Time Inversion Recovery coronal sequence, (c) a T1-Weighted Spin Echo coronal sequence, (d) a T2-Weighted 2-D Multi Echo diagonal sequence.*

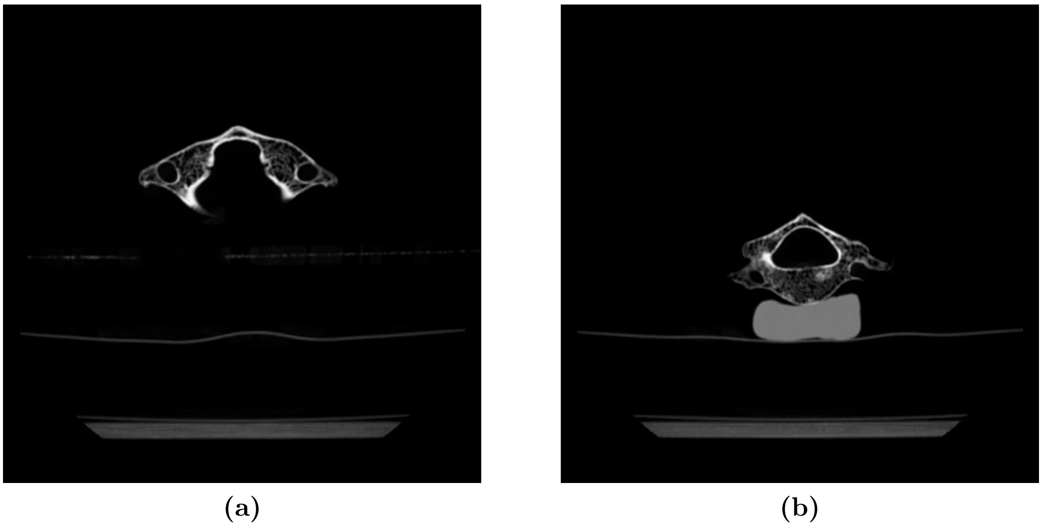


Figure 2.9: *Examples of CT-scans of bone: central slice of c1 (a) and c7 (b) vertebra datasets.*

Table 2.1: *Body of data used in this work*

Modality	**Object**	**Spatial configuration**				**Time-frames**	**Datasets**
		Orientation	Nb. slices	Slice spacing (mm)	Pixel size (mm)		
Artificial	Geometric model	Axial	12	5.73	1	1	1
Artificial	Geometric model	Radial	13	–	1	1	1
Artificial	LV	SA	4 to 19	5 to 20	1.77 to 2.08	1	32
		LA	0 to 12	–	1.77 to 2.08		
Artificial	Brain	Axial (2 sets)	11 to 18	3.3 to 6.6	0.78 to 0.94	1	9
		Sagittal (1 or 2 sets)	13 to 20	3.3 to 4.8	0.90 to 0.37		
CT	Acetabulum	Axial	132	0.5	0.49	1	1
CT	Kidney	Axial	142	0.8	0.78	1	1
CT	LV cavity	Axial	99	0.8	0.78	1	1
MRI	Phantom	Axial	192	0.9	0.87	1	1
MRI	Phantom	Axial	80	1.5	1.44	1	1
Cine MRI	Heart	SA	4 to 19	5 to 20	1.77 to 2.08	25	16
		LA	0 to 12	–	1.77 to 2.08		
Cine MRI	Heart	Radial	13	–	1.77	25	1
MRI	Brain	Axial (2 sets)	11 to 18	3.3 to 6.6	0.78 to 0.94	1	9
		Sagittal (1 or 2 sets)	13 to 20	3.3 to 4.8	0.90 to 0.37		
CT	Vertebra	Axial	47 or 62	0.5	0.35	1	2
MRI	Hip	Axial	19	6	0.74	1	1
		Coronal (2 sets)	19	4.8	1.64		
		Diagonal	15	3.3	0.86		

Chapter 3

Background

In this chapter, we present existing methods for object modelling from sparse and misaligned tomographic data, and we highlight shortcomings and problems that we address in this work. We also provide more details and formulations for the most popular of these works, which we will compare against in Sections 4.5 and 5.5 and Chapter 6. In the last section, we present the works which inspired and justified the proposed method.

3.1 Overview of Object Modelling Methods for Sets of Medical Slices

The approaches to modelling objects from sparse and misaligned sets of slices can be broadly classified into two main categories, namely sequential and integrated methods. The former performs registration, segmentation, and interpolation independently and in turn, while the latter performs two or all of these stages simultaneously. Tables 3.1 and 3.2 present an overview of the existing methods for registration and object modelling from 3D sparse sets of slices respectively. An overlap exists between these tables where the three stages are integrated (last row of Table 3.1).

Table 3.1: *Overview of existing approaches for registration of 3D sparse medical data*

Approach	**Type**	**Main drawbacks**	
Independent registration	Based on similarity measure	Unreliable similarity measure due to limited slice intersections	Failure is cascaded to subsequent modelling stage
	Based on segmentation results	Lack of robustness of the independent 2D segmentations	
Model based	Integrated with segmentation and interpolation	Training phase, Lack of flexibility	

Table 3.2: *Overview of existing approaches for 3D modelling from aligned sparse medical data*

Approach		**Type of segmentation**	**Type of interpolation**	**Main drawbacks**	
Sequential	Segmentation followed by interpolation	2D	Surface interpolation	Lack of robustness of the segmentation	First step is critical
	Interpolation followed by segmentation	3D	Image interpolation	Ill-posed interpolation problem	
Model based		3D, Registration and deformation of a model	Rely on model continuity	Training phase, Lack of flexibility	

3.1.1 Sequential Approaches

Traditionally, object modelling from sparse and misaligned sets of slices involves sequential registration, segmentation, and interpolation of the images. This sequential approach requires aligning the images at first, in order to ensure that they provide coherent information to the subsequent segmentation and interpolation processes. Interpolation usually follows registration, in order to fill the gaps in the data volume, before 3D segmentation is applied to the interpolated volume. However, in some cases, segmentation is performed first in 2D, then a 3D surface is interpolated from the resulting 2D contours. In the rest of this section, we list the most common registration methods first, then we describe existing interpolation solutions.

Independent Registration – Image registration methods can be classified into two categories: intensity-based and feature-based. Intensity-based methods align two images by comparing patterns in their intensity, while feature-based methods exploit information on the content of the images and correspondences between matching points between the two images' contents.

Note that most registration methods found in the literature apply to overlapping images, and little attention has been devoted to the registration of intersecting 2D images slices. Some authors, such as Chandler et al. [8], Shen et al. [43], Smolikova-Wachowiak et al. [48], proposed to align the 2D slices to a reference high-resolution volume. Such methods will not be considered in this study, since a full-volume data is usually not available in common medical tomographic applications.

Amongst intensity-based methods, algorithms using Mutual Information (MI) [11, 54] and Normalised Mutual Information (NMI) [50] are very popular for aligning multi-modality medical images or volumes. MI and NMI are similarity measures which estimate the quantity of information shared by two images, and are maximal when the two images are perfectly aligned. Indeed, when two images overlap perfectly, knowing the content of one of the image allows knowing the content of the other one completely, which means that they contain the same, shared, information. The NMI measure will be presented in more detail in Section 3.2.1. These two similarity measures are particularly popular in medical imaging, because they do not require the images to have the same intensities, and they are therefore suited to align images from different modalities.

Although most MI and NMI registration algorithms have been proposed for overlapping images and dense volumes, a few works have focused on sparse data. In particular, Lotjonen et al. [33] proposed to use NMI to register, by translations, cardiac MRIs made up of two perpendicular stacks of parallel 2D slices separated by large gaps. This method is very popular for cine cardiac MRI registration, *e.g.* [3, 16, 24, 32, 44], and will be described in more detail in the formulation Section 3.2.1. A drawback of similarity measure-based registration methods is that they require enough intersection points between the images in order to compute significant and reliable similarity measures. In the case of Lotjonen et al. [33], the method was evaluated on relatively dense datasets, made up of two orthogonal stacks of SA and LA slices, so each image had several intersecting images and a reasonable number of intersection points with them. In our experiments in Section 5.5, we will show that this method performs less satisfactorily on sparser datasets.

Other intensity based methods have been proposed *e.g.* by Slomka et al. [46] and Elen et al. [17] who registered cardiac cine MRIs, made up of one two-chamber and one four-chamber LA slices plus a stack of SA slices, by optimizing the match of intensity profiles along the intersections of the various slices. In [46], the images were registered by in-plane translations and out-of-plane rotations in the case of the SA images. In [17], all images were registered by 3D translations.

Li and Denney [29] presented a feature-based method to register internally intersecting and largely spaced cardiac MRI slices. Contrary to the previous methods, they used the segmentation contours obtained from preliminary independent 2D segmentations of the images, rather than measures of intensity similarity. The drawback of this approach is that registration is highly dependent on the quality of segmentation, while the independent 2D segmentations may not be consistent with each other.

Independent Interpolation – Following registration, modelling an object from sparse sets of 2D slices requires interpolating in the gaps in the data. Li et al. [28] and Pihuit et al. [41], amongst others, proposed to first segment the slices independently, then to interpolate the yielded 2D contours into a 3D surface. Again, the independent 2D segmentations may not be consistent with each other, and incorrect segmentations would not provide a reliable ground for the shape interpolation. Liu et al. [31] proposed a method to interpolate surfaces from segmentation contours obtained from image slices having arbitrary positions and orientations. In their work, the segmentations are performed manually to ensure they are correct.

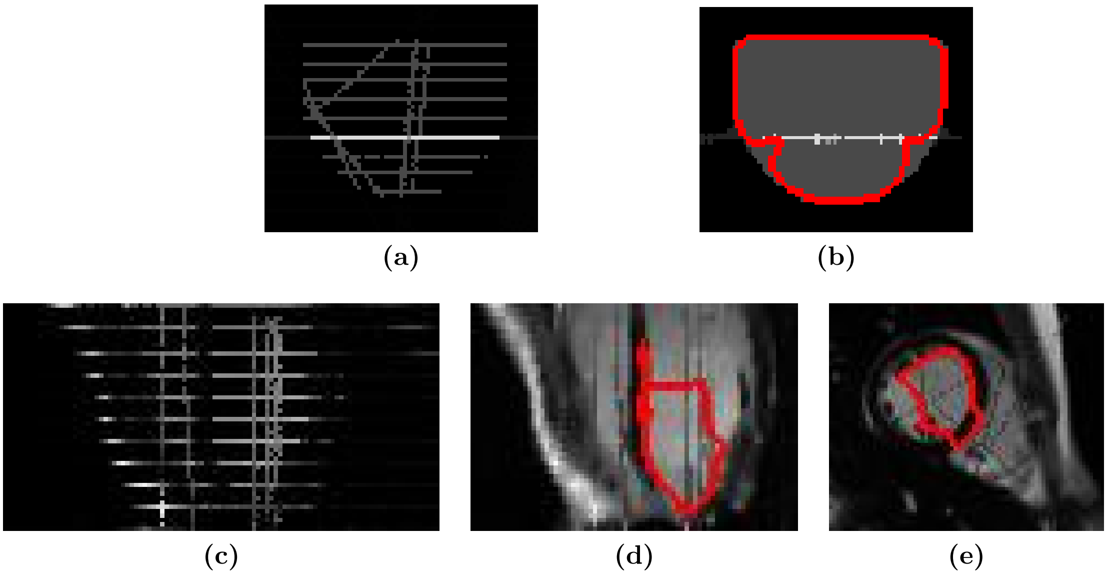
More commonly, a 3D image is first interpolated from the 2D slices, *e.g.* as in [14, 40, 42, 63], and then it is segmented in a second step. Image interpolation methods may be separated into two categories: intensity-based and shape-based methods. Amongst intensity-based methods, Rahman and Wesarg [42] and Yuan and Yuan [63] proposed super-resolution approaches to fuse two orthogonal stacks of parallel slices. Rahman and Wesarg [42] used an observation model to estimate the initial image from the observed ones, and Yuan and Yuan [63] used a wavelet fusion technique which selects the direction of interpolation that better preserves edges. Shape-based methods attempt to preserve the shape of objects while interpolating the images, and rely on these shapes to drive the interpolation. They therefore tend to be more robust than intensity-based methods and preserve edges better. Grevera and Udupa [21] have interpolated image intensity by first converting an N-D grey-level image into an (N+1)-D binary image, using the grey-level values to derive the additional dimension. This effectively converts the image into a surface embedded in an (N+1)-D volume. A distance function-based surface interpolation algorithm was then used to close the holes in the surface, before converting it back into an N-D image. Morigi and Sgallari [37] noted that this shape-based interpolation is the same as interpolating every grey-level line (*i.e.* iso-surface intensity contour lines) using distance transform interpolation. Since level sets are distance functions, they proposed to use them to interpolate between parallel slices by level set morphings. Pan et al. [40] and Cordero-Grande et al. [14] performed interpolation between matching regions of parallel images. These matching regions were found using preliminary segmentations of the images in [40], and non-rigid, topology-preserving registration in [14]. The method of Cordero-Grande et al. [14] will be compared against in Section 4.5 and Chapter 6, and is therefore described in more detail in Section 3.2.2.

The problem of interpolating sets of 2D images is ill-posed, since the slices may have various positions and orientations, as seen in Fig. 1.2, and most of the available interpolation methods can only handle stacks of parallel and equally spaced images, *e.g.* [14, 37]. This limitation may lead to the loss of crucial information, such as in cardiac MRI, where only SA images are traditionally used, while LA images, when available, are often discarded. Yet these LA slices contain unique and important information on the position of the heart valves and the shape of the apex of the ventricles, as highlighted in Section 2.1.1 and illustrated in Fig. 2.3b, and they are therefore valuable for the segmentation.

Another issue for image intensity interpolation arises from the different gains and contrasts that images of the same dataset may have, as seen in Fig. 1.3. This commonly leads to the creation of interpolation artefacts which may bias the subsequent segmentation, as in Fig. 3.1. Woo et al. [59] proposed in their work on super-resolution by volume fusion to equalise the intensities of corresponding regions in three non-isotropic volumes, prior to reconstructing a high resolution fused volume. Their intensity equalisation stage used an intensity matching method based on spline regression. However, modifying the intensities of medical images may be controversial because of the risk in erasing valuable information and creating erroneous features.

For similar reasons, the image interpolation approach is also restricted to images of the same kind, while in some applications it may be beneficial to segment data from different modalities simultaneously. Such data may highlight different properties of the object, as seen in Fig. 2.1a, and the combination of different images may allow the signal to noise ratio to be improved. An example will be given in Chapter 4 where T1 and T2 weighted MRI of brain are processed simultaneously.

Finally, these sequential approaches may lack robustness, since the success of one stage



(a) (b)

(c) (d) (e)

Figure 3.1: *Image interpolation followed by segmentation of SA and LA slices having different gains and contrasts. The interpolation is performed by the image intensity interpolation of [21]. Segmentation is done using an edge based segmentation algorithm [61] and its result is displayed in red. Top row: artificial geometrical shape dataset, bottom row: real cardiac MRI. (a) and (c): central vertical slice of the dataset viewed from the side, (b), (d) and (e): interpolation and segmentation results viewed in a central vertical slice ((b) and (d)) and in a central SA (horizontal) slice (e).*

depends on the success of previous stages. In addition, performing registration, segmentation, and interpolation in turn may not be appropriate since they are inherently related, as explained in the introduction chapter.

3.1.2 Integrated Approaches

A few attempts have been made to integrate two or three of these stages in a single framework rather than process them sequentially. Notably, registration and interpolation have been combined with segmentation, respectively, and all together, in order to exploit the shape information contained in segmentation results, and thus gain extra robustness.

Integrated Registration and Segmentation – Registration and segmentation of sets of 2D slices have been combined to update the two stages simultaneously, mostly using statistical frameworks, *e.g.* [60] and [19] amongst others, or iteratively in turn, *e.g.* [51]. Le Guyader and Vese [25] proposed to align an image and a template while segmenting the image in a level set framework, by deforming a displacement vector field under the guidance of an energy functional which measures the quality of the segmentation of the image. These methods are designed to register overlapping images and can not handle several intersecting 2D images in a 3D volume.

Yezzi et al. [62] presented a level set based method to register and segment two overlapping images or volumes simultaneously while minimising an energy containing a region-based segmentation term and an alignment term. This method does not apply to sparse datasets made up of more than two images, but its integration strategy inspired our proposed integrated registration method for multiple sparse data, which will be presented in Chapter 5. We will detail this method in Section 3.3.3.

Integrated Segmentation and Interpolation – Methods which register and deform a model on the images rely on the continuity of the model to segment the gaps in the data. This can be viewed as simultaneous interpolation and segmentation. Amongst others, van Assen et al. [52] used Active Shape Models (ASMs), Zambal et al. [65] used combined 2D Active Appearance Models (AAMs) and a 3D shape model, Lotjonen et al. [34] used a triangulated surface model, and Cordero-Grande et al. [13] applied a parametric model to process sparse cardiac MRIs. The main drawback of these model-based methods is that they require a training phase and are restricted to the class of objects they have been trained with. They may also lack flexibility to finely model shapes presenting a high degree of variation.

When dealing with misaligned data, these integrated segmentation and interpolation methods, hereafter also referred to as semi-sequential, require an independent registration to be performed beforehand. In Section 5.5 and Chapter 6, such semi-sequential approaches with a separate registration stage will be shown to lack robustness.

Integrated Registration, Segmentation, and Interpolation – Zambal et al. [64] also registered, segmented, and interpolated sparse 3D cardiac MRI data made up of parallel SA images using a two component deformable model which provided both interpolation and registration. The model components were updated iteratively: 2D AAMs segmented the individual slices, and a global shape model linked the 2D AAMs together. Registration was performed at every iteration by aligning the AAMs on the global 3D shape model. Interpolation was provided by the continuity of the model. This method also suffers from the drawbacks of the model-based methods, notably the lack of flexibility and the requirement of a training phase.

The method of Zambal et al. [64] is designed for modelling the heart from cardiac MRI, therefore it would have been desirable to compare our proposed fully integrated framework against it in our experiments on cardiac cine MRIs in Chapter 6. However, the authors informed us that their code and prior (AAM and 3D shape) models were no longer available. Since this method is the only work we know of that integrates the three stages of registration, segmentation, and interpolation, we can only compare our framework against sequential and semi-sequential methods in Chapters 4, 5, and 6. We now describe these methods in more detail.

3.2 Focus on State-of-the-Art Methods used for Comparison

We now provide more details on key state-of-the-art registration and image interpolation methods that were mentioned in Section 3.1 and that will be used for comparative evaluation in Chapters 4, 5, and 6.

3.2.1 NMI Based Registration and Movement Correction in Cardiac MRIs

Normalised Mutual Information – NMI is a similarity measure proposed by Studholme et al. [50] to register 3D multi-modality medical images with varying degrees of overlap. It is based on information theory, and in particular on the Shannon-Wiener entropy measure $\mathcal{H}$, which estimates the uncertainty in a random variable, and is equal to

$$\mathcal{H}\left(p_1, p_2, ..., p_N\right) = -\sum_{i=1}^{N} p_i \log p_i \ , \tag{3.1}$$

where i are the indexes of the possible values for the variable, with associated probabilities p_i.

This entropy can be used to evaluate the amount of information contained in an image, with the random variable being the pixel intensity, and its probability distribution being given by the normalised histogram of the image. A way of interpreting this measure is that, when the random variable has a high probability of having a given value, then it is easy to guess which value will be yielded by a random draw on the variable, *e.g.* the random pick of a pixel in the case of an image. The uncertainty on the variable is therefore low, and so is the associated entropy $\mathcal{H}$. In other words, the information content is low, since very few bits are needed to encode it. On the contrary, if the variable has equal chances to have several values, then the guess on the outcome of a random draw is more difficult and the uncertainty on the variable and its entropy are high. An image with a high entropy requires therefore more bits to code its information content, and it is said to contain a lot of information.

The joint entropy of two images is an interesting tool to estimate the amount of information they share and therefore register them. This joint entropy may be computed by considering the random variable $V(i_1, i_2)$, with i_1 and i_2 being the intensities of a pixel in the two images respectively. The distribution of this variable is provided by the joint histogram of the two images. When the images are well aligned, the most probable intensity of a given object in one image should always be associated with the corresponding most probable intensity of the same object in the second image. Thus, the associated value of the joint random variable should have a frequent occurrence, and the joint entropy should be low. On the contrary, when the two image are badly aligned, knowing the intensity of one pixel in an image does not allow guessing the intensity of the same pixel in the second image with a high certainty, and therefore the joint entropy is high. Using this idea, authors such as Collignon et al. [12] and Studholme et al. [49] investigated the use of joint entropy to register images.

However, this approach makes the assumption that if there are large regions in the two images, a good registration of the images will optimise their overlap in order to maximise the co-occurrence of their most probable values. This may not always be true, notably for images with only partial overlap of a large homogeneous background. In such cases, the optimal alignment and overlap of the background, which contains less uncertainty on the joint intensities, may not be the optimal registration of the foreground object.

Viola and Wells [54] and Collignon et al. [11] addressed this issue by proposing a new measure of similarity, which prevents the artificial minimisation of the marginal entropies of both images in the region of overlap, by including them in the similarity measure. Their new measure was the relative entropy, or Mutual Information, of the two images A and B, which should be maximum for well aligned images:

$$MI(A, B) = \mathcal{H}(A) + \mathcal{H}(B) - \mathcal{H}(A, B)\ . \tag{3.2}$$

Nevertheless, Studholme et al. [50] showed that MI still does not completely remove the influence of the overlap on the images. Indeed, in some cases, its maximisation may seek to optimise the marginal entropies by varying the size of the overlap region, without modifying the joint entropy. Therefore, they proposed a new measure, the Normalized Mutual Information, which is independent of changes in the marginal entropies in their region of overlap, and whose maximisation seeks to minimise the joint entropy with respect to the marginal ones:

$$NMI\left(A,B\right)=\frac{\mathcal{H}(A)+\mathcal{H}(B)}{\mathcal{H}(A,B)}\,. \tag{3.3}$$

NMI Based Internal Registration of SA and LA Cardiac Cine MRI Slices – Lotjonen et al. [33] proposed to apply NMI to the registration, by translation, of sparse cardiac cine MRI datasets made up of two stacks of SA and LA slices. They maximised the NMI between one slice and all other intersecting slices, iteratively. At each iteration, a slice was chosen randomly, and was moved in the direction which better increases its NMI with all its intersecting slices.

Because each temporal sequence of a dataset is acquired during one breath-hold, all of its time-frames can be considered to be spatially registered with each other, and so they were moved together when aligning the different sequences. The authors found that using all the time-frames to compute the NMI did not increase the accuracy of the method, therefore they recommended to use six time-frames only. The displacement step at each iteration was initially set to the size of an image pixel, *i.e.* 1.4 mm in their experiments. Then, it was reduced by a factor of two to refine the registration.

Note that the datasets of Lotjonen et al. [33] were relatively dense when compared with ours (see Chapter 2), since they were made up of two stacks of parallel SA and LA slices, containing respectively 4 to 6 SA and 4 to 8 LA slices, and thus a lot of intersections. Our experiments in Chapter 5 will show that, for sparser datasets, the NMI method struggles to compute accurate similarity measures in the very limited overlap region available.

3.2.2 Object-Based Tomographic Image Interpolation

Cordero-Grande et al. [14] proposed a method to interpolate pixel intensities between parallel and equally spaced images. It falls in the object-based category, since it uses non-rigid registration to find matching points between two images to interpolate the intensity. It is currently the state-of-the-art image interpolation for cardiac cine MRI, and we will compare the interpolation results of our proposed method against it in Chapters 4 and 6. Thus, we present it in more detail now.

Non-rigid registration between two adjacent slices – The registration seeks a deformation function $u(\mathbf{r})$ which transforms a coordinate $\mathbf{r}$ in an image I_0 into the coordinate of a matching point in a second image I_1. This function is computed in a multi-resolution manner, with the result u^l at a resolution level l as the initialisation of the next finer level u^{l-1}. At each level, the update of the transformation function is allowed to take values in a discrete set of transformations only, such that $u^l(\mathbf{r}) = \mathbf{r} + u^{l+1}(\mathbf{r}) + k\Delta_k^l$, with Δ_k^l being the unit quantised transformation at resolution l, and $k \in K$ an integer in the space of authorised deformations K.

The transformation function u is obtained as the maximum *a posteriori* configuration of a Markov Random Field (MRF), defined as the field $\boldsymbol{X}$ of the quantised deformations X_s at grid sites $s \in S$ (S being the ensemble of sites at a given resolution), which take the values $\boldsymbol{x} = \{x_s = k_s, s \in S\}$. The MRF is built by defining dependencies between neighbouring sites, with the neighbouring relation between two sites s and s' being defined as $\|\mathbf{r}_s - \mathbf{r}_{s'}\|^2 \leq 2$. In a Bayesian formulation, and according to the Hammersley-Clifford theorem [22], it's posterior probability $P(\boldsymbol{x}|I_0, I_1)$ is expressed in terms of an energy $E_{MRF}(\boldsymbol{x}, I_0, I_1)$ which is the sum of some prior and observation likelihood energies E_P and E_{OL} respectively.

The MRF integrates intensity and gradient magnitude matching constraints between two corresponding areas of the registered images in the observation likelihood energy function E_{OL}, expressed as two potential terms $V_1(x_s, I_0, I_1)$ and $V_1(x_s, \|\nabla_M I_0\|, \|\nabla_M I_1\|)$.

$$V_1(x_s, F_0, F_1) = \sum_{|n| \leq \rho} G(n) \frac{|F_0(\mathbf{r}_s + \mathbf{r}_n) - F_1(\mathbf{r}_{k_s} + \mathbf{r}_n)|}{|F_0(\mathbf{r}_s + \mathbf{r}_n)| + |F_1(\mathbf{r}_{k_s} + \mathbf{r}_n)|}, \tag{3.4}$$

where F_0 and F_1 are either the intensity of the two images or the magnitude of their gradients after convolution with a Gaussian kernel M of standard deviation σ_M. $\mathbf{r}_s$ is the position of site s in the first image, $\mathbf{r}_{k_s}$ is the position of the corresponding point in the other image, $\mathbf{r}_n$ is a spatial shift of n pixels, ρ is the size of the region around site s which is considered for computing the potiential, and G is a Gaussian with standard deviation $\sigma_G = \rho - \frac{1}{2}$, which ensures that pixels located at distance n from site s, with $|n| > \rho$, are negligible.

A third constraint on the MRF, defined as the following term in the prior energy E_p,

$$V_2\left(x_{s_1}, x_{s_2}\right) = \frac{\left\|\left(\mathbf{r}_{k_{s_2}} - \mathbf{r}_{s_2}\right) - \left(\mathbf{r}_{k_{s_1}} - \mathbf{r}_{s_1}\right)\right\|}{\left\|\mathbf{r}_{s_2} - \mathbf{r}_{s_1}\right\|}\ , \tag{3.5}$$

penalises important variations of u between two neighbouring sites s_1 and s_2. Indeed, $\mathbf{r}_{k_s} - \mathbf{r}_s$ expresses the deformation at site s between the two images, and its variation between the two sites is minimised.

A second term in the prior energy function E_p seeks to preserve the topology, *i.e.* the connectivity and neighbourhood relationships of the image's objects, by restricting the Jacobian of the transformation function u to positive values. Indeed, functions that have a positive Jacobian preserve the orientation. This restriction of the Jacobian to positive values is the main contribution of [14]. Its formulation is based on the approximation of the Jacobian J inside a square defined by four summit positions $\mathbf{r_1}$, $\mathbf{r_2}$, $\mathbf{r_3}$, and $\mathbf{r_4}$, by the linear interpolation of its values $J\left(\mathbf{r_i}\right)$ at the four summit points. A necessary and sufficient condition for J to be positive inside the square is that all four $J\left(\mathbf{r_i}\right)$ are positive. Thus, the topology preservation term attempts to restrict the finite difference approximation of the Jacobian at every site to positive values. Given a site s_2, the finite difference is calculated at s_2 using the three summits s_1, s_2 and s_3, where s_1 and s_3 are defined as being aligned with s_2 along the $\mathbf{e_1}$ and $\mathbf{e_2}$ axes respectively. Thus, the penalisation of the approximation of the Jacobian at s_2 that is not positive is given by

$$V_3\left(x_{s_1}, x_{s_2}, x_{s_3}\right) = U\left(\frac{\left(r_{1,k_{s_2}} - r_{1,k_{s_1}}\right)\left(r_{2,k_{s_2}} - r_{2,k_{s_3}}\right)}{\left(r_{1,s_2} - r_{1,s_1}\right)\left(r_{2,s_2} - r_{2,s_3}\right)} - \frac{\left(r_{2,k_{s_2}} - r_{2,k_{s_1}}\right)\left(r_{1,k_{s_2}} - r_{1,k_{s_3}}\right)}{\left(r_{2,s_2} - r_{2,s_1}\right)\left(r_{1,s_2} - r_{1,s_3}\right)}\right)\ , \tag{3.6}$$

with

$$U\left(x\right) = \begin{cases} 0 & \text{if } x > 0 \\ 1 & \text{otherwise} \end{cases}\ . \tag{3.7}$$

r_{i,s_j} denotes the i^{th} component of $\mathbf{r}_{s_j}$. Remember that $\mathbf{r}_{k_{s_j}}$ is the position of the point in image I_1 corresponding to the site s_j of image I_0.

The final energy function for the posterior field is then

$$
\begin{aligned}
E_{MRF}\left(\mathbf{x}, I_0, I_1\right) = \quad & \sigma_0 \sum_{s \in C_1} V_1\left(x_s, I_0, I_1\right) + \\
& \sigma_1 \sum_{s \in C_1} V_1\left(x_s, \left\|\nabla_M I_0\right\|, \left\|\nabla_M I_1\right\|\right) + \\
& \lambda \sum_{s_1, s_2 \in C_2} V_2\left(x_{s_1}, x_{s_2}\right) + \\
& \xi \sum_{s_1, s_2, s_3 \in C_3} V_3\left(x_{s_1}, x_{s_2}, x_{s_3}\right) \ ,
\end{aligned}
\tag{3.8}
$$

with σ_0, σ_1, λ and ξ weighting the influence of their associated term. C_i are cliques of order i, which contain i elements chosen amongst the grid sites, such that, for $i > 1$, any pair of their elements are neighbours.

For each resolution level l, the field is optimised and its parameters are jointly estimated. Parameter λ is estimated using the Maximum Pseudolikelihood Estimator (MPLE) in order to maximise the prior associated with the third energy term in (3.8). The two σ_i are estimated using the Maximum Likelihood Estimator (MLE) in order to maximise the observation likelihood. ξ is not estimated, but instead it is set at a high value in order to ensure that the topology is preserved. The Iterated Conditional Modes (ICM) algorithm is used to optimise the MRF and obtain the transformation function u at grid sites. After u is obtained at the last, finer resolution level, it is linearly interpolated between the grid sites.

Interpolation between two adjacent slices – The previous registration method is used by Cordero-Grande et al. [14] to find corresponding points in two images, between which to interpolate the images' intensities. Both forward and backward registrations are computed, in order to ensure that the whole procedure is unbiased, *i.e.* the roles of I_0 and I_1 are interchangeable. The two results are used jointly to compute the interpolation.

Let's denote by $\mathbf{r}_\tau$ a gap point located at a fraction τ of the distance between the two images. Then, two sets of two corresponding points $\mathbf{r}_{I_0,f}$ and $\mathbf{r}_{I_1,f}$, and $\mathbf{r}_{I_0,b}$ and $\mathbf{r}_{I_1,b}$ are selected, one for each forward and backward transformation. Their choice is such that the transformation path which links two corresponding points passes through $\mathbf{r}_\tau$. Note that the topology preservation constraint of the registration algorithm ensures that only one such path passes through the gap point for each registration direction. The intensity at the gap point is then obtained as the sum of the linear interpolations along the two forward and backward paths, weighted by terms W_i which reflect the confidence in the selection of the paths and image points,

$$I\left(\mathbf{r}_{\tau}\right)=\frac{W_{f}\left(\mathbf{r}_{I_{0},f}\right)\left[\left(1-\tau\right)I_{0}\left(\mathbf{r}_{I_{0},f}\right)+\tau I_{1}\left(\mathbf{r}_{I_{1},f}\right)\right]+W_{b}\left(\mathbf{r}_{I_{1},b}\right)\left[\left(1-\tau\right)I_{0}\left(\mathbf{r}_{I_{0},b}\right)+\tau I_{1}\left(\mathbf{r}_{I_{1},b}\right)\right]}{W_{f}\left(\mathbf{r}_{I_{0},f}\right)+W_{b}\left(\mathbf{r}_{I_{1},b}\right)}\,, \tag{3.9}$$

$$W_{i}\left(\mathbf{r}_{I_{j},i}\right)=\left[-\log\frac{P\left(I_{0}\left(\mathbf{r}_{I_{0},i}\right),I_{1}\left(\mathbf{r}_{I_{1},i}\right)|x_{s};\sigma_{0}\right)P\left(\left\|\nabla_{M}I_{0}\left(\mathbf{r}_{I_{0},i}\right)\right\|,\left\|\nabla_{M}I_{1}\left(\mathbf{r}_{I_{1},i}\right)\right\||x_{s};\sigma_{1}\right)}{\sum\limits_{k_{s}\in K}P\left(I_{0}\left(\mathbf{r}_{I_{0},i}\right),I_{1}\left(\mathbf{r}_{I_{1},i}\right)|k_{s};\sigma_{0}\right)P\left(\left\|\nabla_{M}I_{0}\left(\mathbf{r}_{I_{0},i}\right)\right\|,\left\|\nabla_{M}I_{1}\left(\mathbf{r}_{I_{1},i}\right)\right\||k_{s};\sigma_{1}\right)}\right]^{-1}\,, \tag{3.10}$$

with i being f or b, and j being 0 and 1, for the forward and backward registrations respectively.

3.3 Focus on Key Related Works

This section details the formulation of methods which are the basis or inspired the new framework which we will present in Chapters 4 and 5. We first provide a general presentation of the level set method for image segmentation, which we will extend in Chapter 4 to handle sparse data by combining the segmentation and interpolation abilities of Radial Basis Functions. Then, we introduce an integrated registration and segmentation method for dense data, based on level sets, which we will generalise and extend in Chapter 5 to deal with sparse data.

3.3.1 Level Set Method

The level set method was proposed by Osher and Sethian [39] for simulating and tracking moving interfaces. It has become a widely used tool for image segmentation, where it falls into the *active contour* category, which evolves a contour towards the boundaries of the object to be segmented.

The key idea to the level set method is to define the active contour C as the zero-level iso-contour of an embedding function ϕ,

$$C = \{x : \phi(x) = 0\} \ . \tag{3.11}$$

We adopt the common convention of ϕ being positive inside the contour and negative outside. This concept is illustrated in Fig. 3.2. In this case, the normal $\mathbf{N}$ to C can be computed as $\mathbf{N} = -\frac{\nabla\phi}{\|\nabla\phi\|}$.

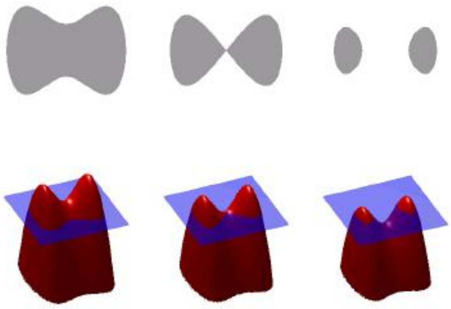


Figure 3.2: *Level set method – illustration of a 2D contour embedded in a 3D function. The figure is reproduced from [38].*

This embedding allows evolving the contour through the update of ϕ, by solving the following Partial Differential Equation (PDE),

$$\frac{\partial\phi}{\partial t} - F\,\|\nabla\phi\| = 0 \ , \tag{3.12}$$

where F is the speed of the contour C in the direction of its normal, which can be computed using any method of choice, based on the image data or on some internal geometrical constraint for the contour.

Equation (3.12) is more easily solved if the implicit function ϕ remains smooth and if the norm of its gradient stays close to 1 as the contour evolves, notably due to the term $\|\nabla\phi\|$. Therefore, it is usually required to re-normalise ϕ regularly during the iterative segmentation process. This is commonly achieved by solving the following equation, at regular intervals,

$$\frac{\partial\phi}{\partial t} = sign\,(\phi)\,(1 - \|\nabla\phi\|) \ . \tag{3.13}$$

The computation of (3.12) also requires a careful choice of a numerical scheme in order to maintain numerical stability [39]. Thus, upwind schemes must be used in the compu-

Table 3.3: *Summary of the main uses of RBFs*

Use of RBF	Segmentation	Interpolation
Scattered data, *e.g.* [18]		✓
Object modelling from point clouds, *e.g.* [5]		✓
Level set implicit function, *e.g.* [1, 45, 56]	✓	
Proposed modelling from sparse data (Chapter 4)	✓	✓

tation of the finite differences, in order to take into account the direction of propagation of the information. For example, when using a first order upwind scheme, the choice of forward or backward finite difference is performed dynamically depending on the speed of the interface, in order to only use points which have already been swept by the moving interface. These upwind schemes may sometimes be difficult to implement. This motivated the introduction of RBF interpolated level set methods, based on Ordinary Differential Equations (ODEs), which we will present in Section 3.3.2.

3.3.2 RBF Methods

RBFs are radially symmetric functions, centred at a control point $\mathbf{x}_i$, which can be expressed as

$$\psi_i(\mathbf{x}) = \tilde{\psi}(\|\mathbf{x} - \mathbf{x}_i\|) \ , \tag{3.14}$$

where $\tilde{\psi} : \mathbb{R}^+ \to \mathbb{R}$, and $\|.\|$ usually denotes the Euclidean norm. Thus the value of an RBF at a point $\mathbf{x}$ depends only on the distance of this point to the control point $\mathbf{x}_i$ of the RBF. Some functions that qualify as an RBF are thin-plate splines, cubic splines, polyharmonic splines, Gaussians, multiquadrics (MQ) or Wendland RBFs [18].

RBFs are commonly used for interpolating scattered data, and this led to the development of two other major uses for RBFs, namely reconstruction of 3D surfaces from point clouds, and image segmentation in a level set framework. These three main uses of RBFs are summarised in Table 3.3 and will be described next. We will propose a new use for RBFs in Chapter 4 (last row of Table 3.3).

RBF Interpolation of Scattered Data – A function $\phi(\mathbf{x})$ may be interpolated by a set of N RBFs centred on N control points $\mathbf{x}_i$ as

$$\phi\left(\mathbf{x}\right)=\sum_{i=1}^{N}\alpha_i\psi_i\left(\mathbf{x}\right)+p\left(\mathbf{x}\right)\ , \tag{3.15}$$

where α_i is a weight associated to the ith RBF and $p\left(\mathbf{x}\right)=p_0+p_1x+p_2y+p_3z$ is a first-degree polynomial which accounts for the linear and constant portions of $\phi\left(\mathbf{x}\right)$. In order to ensure a unique solution, additional constraints must be respected when $p\left(\mathbf{x}\right)$ is not null:

$$\textstyle\sum_{i=1}^{N}\alpha_i=0\ ;\quad \sum_{i=1}^{N}\alpha_ix_i=0\ ;\quad \sum_{i=1}^{N}\alpha_iy_i=0\ ;\quad \sum_{i=1}^{N}\alpha_iz_i=0\ . \tag{3.16}$$

If the values f_i of the interpolated function are known at the control points, *i.e.*

$$\phi\left(\mathbf{x}_i\right)=f_i,\quad i=1\ldots N\ , \tag{3.17}$$

then, the coefficients α_i and p_j can be obtained by solving the linear system of $N+4$ equations defined by (3.15) to (3.17), which can be rewritten in matrix form as

$$\mathbf{H}\boldsymbol{\alpha}=\mathbf{f}\ , \tag{3.18}$$

with

$$\mathbf{H}=\begin{bmatrix}\mathbf{A} & \mathbf{P}\\ \mathbf{P}^T & \mathbf{0}\end{bmatrix}\in\mathbb{R}^{(N+4)\times(N+4)}\ , \tag{3.19}$$

$$\mathbf{A}=\begin{bmatrix}\psi_1\left(\mathbf{x}_1\right) & \cdots & \psi_N\left(\mathbf{x}_1\right)\\ \vdots & \ddots & \vdots\\ \psi_1\left(\mathbf{x}_N\right) & \cdots & \psi_N\left(\mathbf{x}_N\right)\end{bmatrix}\in\mathbb{R}^{N\times N}\ , \tag{3.20}$$

$$\mathbf{P}=\begin{bmatrix}1 & x_1 & y_1 & z_1\\ \vdots & \vdots & \vdots & \vdots\\ 1 & x_N & y_N & z_N\end{bmatrix}\in\mathbb{R}^{N\times 4}\ , \tag{3.21}$$

$$\boldsymbol{\alpha}=\begin{bmatrix}\alpha_1 & \cdots & \alpha_N & p_0 & p_1 & p_2 & p_3\end{bmatrix}^T\in\mathbb{R}^{N+4}\ , \tag{3.22}$$

$$\mathbf{f} = \left[\begin{array}{cccccccc} f_1 & \cdots & f_N & 0 & 0 & 0 & 0 \end{array} \right]^T \in \mathbb{R}^{N+4} . \tag{3.23}$$

Object Reconstruction from Point Clouds by Implicit Surface Fitting – Carr et al. [5] exploited the interpolation property of RBFs to reconstruct closed 3D surfaces from sparse point clouds. Their method is based on the representation of the surface as the zero-level iso-contour of an embedding distance function, similar to the embedding of the segmenting contour of a level set method (see Section 3.3.1).

The embedding function is defined as the distance to the closest point of the surface, and its values can be computed initially along the normals of the surface for each data point. If the normals are not available, they must be estimated from the point cloud as a preliminary step, *e.g.* as in [2]. Then, RBF interpolation is performed on the embedding function in order to recover its values everywhere in the volume. As a result, the surface is implicitly defined by the embedding function fitted to the data points, and can be extracted as the zero-level iso-contour of the embedding function.

Noise and outliers may greatly reduce the performance of this method by biasing the computation of the initial values of the embedding function. Therefore, a smoothness term has been proposed by Carr et al. [5] to reduce the influence of noisy points.

For large point clouds, the evaluation of (3.18) can be extremely time and memory consuming, and Carr et al. [5] recommend the use of the Fast Multipole Method (FMM) [20] in order to reduce the computation costs. This method approximates the computations which involve RBF centres located far from the point for which a value is estimated. Carr et al. [5] also proposed a greedy algorithm that reduces the number of data points used in the computation while maintaining a given desired accuracy.

RBF Interpolated Level Set – Wang and Wang [55] proposed a new level set framework based on the interpolation of the level set function by RBFs. This new framework has the advantage of replacing the usual PDE presented in Section 3.3.1 by a simpler system of coupled ODE, and it does not require reinitialisations.

In their implementation, the authors chose to use MQ splines, which are known for providing the best interpolations of scattered data [18] and have the form

$$\psi_i\left(\mathbf{x}\right) = \sqrt{\left(\mathbf{x} - \mathbf{x}_i\right)^2 + \gamma_i^2} \ , \tag{3.24}$$

where γ_i is a constant which sets the flatness of the RBF. The control points were chosen to be the grid points of the discretised space.

Since an approximation of the level set function ϕ can be defined by the coefficients $\boldsymbol{\alpha}$ according to (3.15), Wang and Wang [55] proposed to evolve ϕ by updating $\boldsymbol{\alpha}$. Under the assumption that space and time are separable variables, (3.15) is rewritten as

$$\phi\left(\mathbf{x},t\right)=\mathbf{\Psi}^{T}\left(\mathbf{x}\right)\boldsymbol{\alpha}\left(t\right)\ , \tag{3.25}$$

with

$$\mathbf{\Psi}\left(\mathbf{x}\right)=\left[\begin{array}{ccccccc}\psi_1\left(\mathbf{x}\right) & \cdots & \psi_N\left(\mathbf{x}\right) & 1 & x & y & z\end{array}\right]^{T}\in\mathbb{R}^{(N+4)\times 1}\ . \tag{3.26}$$

Inserting (3.25) into (3.12) and (3.16) yields

$$\mathbf{\Psi}^{T}\left(\mathbf{x}\right)\frac{d\boldsymbol{\alpha}}{dt}-F\left(\mathbf{x},\boldsymbol{\alpha}\right)\left|\left(\nabla\mathbf{\Psi}\left(\mathbf{x}\right)\right)^{T}\boldsymbol{\alpha}\right|=0 \tag{3.27}$$

and

$$\sum_{i=1}^{N}\alpha_i\left(t\right)=0\ ;\ \ \sum_{i=1}^{N}\alpha_i\left(t\right)x_i=0\ ;\ \ \sum_{i=1}^{N}\alpha_i\left(t\right)y_i=0\ ;\ \ \sum_{i=1}^{N}\alpha_i\left(t\right)z_i=0\quad , \tag{3.28}$$

or, in matrix form:

$$\mathbf{H}\frac{d\boldsymbol{\alpha}}{dt}-\mathbf{B}\left(\boldsymbol{\alpha}\right)=0\ , \tag{3.29}$$

with

$$\mathbf{B}\left(\boldsymbol{\alpha}\right)=\left[\begin{array}{cccccccc}F\left(\mathbf{x}_1,\boldsymbol{\alpha}\right)\left|\left(\nabla\mathbf{\Psi}\left(\mathbf{x}_1\right)\right)^{T}\boldsymbol{\alpha}\right| & \cdots & F\left(\mathbf{x}_N,\boldsymbol{\alpha}\right)\left|\left(\nabla\mathbf{\Psi}\left(\mathbf{x}_N\right)\right)^{T}\boldsymbol{\alpha}\right| & 0 & 0 & 0 & 0\end{array}\right]^{T}\in\mathbb{R}^{(N+4)\times 1}\ . \tag{3.30}$$

(3.29) is a system of N+4 ODEs which, when solved, allows updating $\boldsymbol{\alpha}$ using *e.g.* an Euler forward scheme:

$$\boldsymbol{\alpha}\left(t+1\right)=\boldsymbol{\alpha}\left(t\right)+\triangle t\mathbf{H}^{-1}\mathbf{B}\left(\boldsymbol{\alpha}\left(t\right)\right)\ . \tag{3.31}$$

Note that $\mathbf{H}$ is a $(N+4)\times(N+4)$ matrix, thus its inversion requires a considerable amount of computing time and/or storage for large N. Wang and Wang [55] recommended to use a fast evaluation method based on the FMM, such as in [5]. However, this may be tedious to implement and may not improve the overall performance of the algorithm sufficiently when dealing with very large data sets such as 4D data. Indeed, Carr et al. [5] had to combine the FMM with a reduction of the number of control points in order to ensure reasonable performance for very large data sets.

Variational Approaches – Some later works addressed the inversion issue of [55] by deriving new evolution equations for $\boldsymbol{\alpha}$ using a variational approach. These new ODEs are based on a gradient descent and do not require matrix inversion any more. They are therefore more efficient to solve than (3.29).

Slabaugh et al. [45] used a small number of 2D anisotropic Gaussian RBFs, whose standard deviations, orientations, and positions of control points, as well as the number of RBFs, are updated at the same time as the interpolation coefficients $\boldsymbol{\alpha}$. They first obtained a new ODE by deriving the variation of the Chan-Vese energy functional [7] with regards to the interpolation coefficients and the parameters ς of the RBFs:

$$\frac{d\alpha_i}{dt} = \int_C F_{PC} \psi_i dC \ , \tag{3.32}$$

and

$$\frac{d\varsigma_{ij}}{dt} = \int_C F_{PC} \frac{\partial \phi}{\partial \varsigma_{ij}} dC \ , \tag{3.33}$$

with

$$F_{PC} = \left(I\left(\mathbf{x}\right) - c_{in}\right)^2 - \left(I\left(\mathbf{x}\right) - c_{out}\right)^2 + \gamma \nabla \left(\frac{\nabla \phi}{\|\nabla \phi\|} \right) \ . \tag{3.34}$$

i denotes the i^{th} RBF, with $i \in \{1..N\}$ and N the total number of RBFs. j is the j^{th} parameter of the RBF. I is the intensity of the image, and c_{in} and c_{out} are the mean intensities inside and outside of the contour C respectively, as defined in [7]. Then, after noting that F_{PC} is the classic level set speed of [7], they extended the obtained equations to edge-based segmentation algorithms, by replacing F_{PC} by the classic geodesic flow.

As the contour and the RBFs' positions evolve, some RBFs may need to be merged or added. Two RBFs are merged when they are too close to each other, by replacing them with an RBF having the sum of their respective properties and their average position. On the contrary, one new RBF is added where the gradient of ϕ is too high, if no RBF is already nearby.

Bernard et al. [1] used B-spline RBFs with control points placed on a regular grid and spaced by an integer h in all directions. They proposed to minimise a region-based energy functional of the form

$$E_{reg,C}(\phi) = \nu_{in} \int f_{in}(\mathbf{x}, \phi(\mathbf{x})) H(\phi(\mathbf{x})) + \nu_{out} \int f_{out}(\mathbf{x}, \phi(\mathbf{x})) (1 - H(\phi(\mathbf{x}))) + \nu_c \int f_c(\mathbf{x}, \phi(\mathbf{x})) \delta(\phi(\mathbf{x})) \|\nabla(\phi(\mathbf{x}))\| \ , \tag{3.35}$$

with f_{in}, f_{out} and f_c describing the object and background regions and the contour respectively, and ν_{in}, ν_{out} and ν_c the corresponding weighting coefficients, and H being equal to 1 inside the contour and 0 outside. The resulting update scheme for the interpolation coefficients was:

$$\frac{d\alpha_i}{dt} = \sum_{\mathbf{x}_j \in \mathbb{Z}^d} F_{reg,C}(\mathbf{x}_j) \beta^n \left(\frac{\mathbf{x}_j}{h} - \mathbf{x}_i \right) \ . \tag{3.36}$$

β^n is the uniform symmetric B-spline of degree n. $F_{reg,C}$ is named feature function by Bernard et al. [1] because it reflects the features of the object, and it is equal to:

$$F_{reg,C}(\mathbf{x}) = \left(\nu_{in} f_{in}(\mathbf{x}) - \nu_{out} f_{out}(\mathbf{x}) - \nu_c div \left(f_c(\mathbf{x}) \frac{\nabla\phi(\mathbf{x})}{\|\nabla\phi(\mathbf{x})\|} \right) \right) \delta(\phi(\mathbf{x})) \ . \tag{3.37}$$

Note that both the original method of Wang and Wang [55] and these variational methods use RBFs to perform segmentation, thus yielding a more numerically stable scheme, but they do not provide interpolation. They are therefore not appropriate for segmenting sparse data. The new RBF-interpolated level-set framework which will be proposed in Chapter 4 will perform both segmentation and interpolation, and therefore will be able to process sparse data.

3.3.3 Integrated Registration and Segmentation of Dense and Overlapping 2D/3D Data

Yezzi et al. [62] proposed an integrated registration and segmentation method for two overlapping and dense images or volumes, which motivated the new registration method which will be presented in Chapter 5 for sparse and partially intersecting data.

The algorithm of Yezzi et al. [62] simultaneously updates a level set contour, denoted as C and $\hat{C}$ in the two overlapping images or volumes to be registered respectively, and g a mapping between these contours such that $\hat{C} = g(C)$.

After noting that a region-based energy typically exhibits the form

$$E_{reg}(C) = \int_{\Omega_{in}} f_{in}(\mathbf{x})\, d\mathbf{x} + \int_{\Omega_{out}} f_{out}(\mathbf{x})\, d\mathbf{x}\ , \tag{3.38}$$

where f_{in} and f_{out} depend on the data, respectively inside and outside of the contour, the authors derived the evolution equations (3.40)for the contour and g by minimising an energy $E_{reg}(C, g)$:

$$E_{reg}(C, g) = \int_{\Omega_{in}} \left(f_{in} + |g'| \left(\hat{f}_{in} \circ g\right)\right)(\mathbf{x})\, d\mathbf{x} + \int_{\Omega_{out}} \left(f_{out} + |g'| \left(\hat{f}_{out} \circ g\right)\right)(\mathbf{x})\, d\mathbf{x}\ , \tag{3.39}$$

$$\begin{cases} \dfrac{\partial C}{\partial t} = & f\mathbf{N} + |g'| \left(\hat{f} \circ g\right) \mathbf{N} - \kappa \mathbf{N} & (3.40a) \\ \dfrac{\partial g_i}{\partial t} = & \int_C \left(\hat{f} \circ g\right) \left\langle \frac{\partial g}{\partial g_i}, \left(\left(g'\right)^{-1} |g'|\right)^T \mathbf{N} \right\rangle \quad , & (3.40b) \end{cases}$$

where $\langle\rangle$ is the scalar product and $\circ$ is the function composition operator. g_i is the i^{th} component of g, $|g'|$ is the determinant of g' the Jacobian matrix of g, $\mathbf{N}$ is the normal to the contour and κ its curvature, and $f = f_{in} - f_{out}$ may be seen as speeds for the contour.

If no constraint is imposed on the form of g, then g can be arbitrary between the two contours C and $\hat{C}$. In the extreme case where g is the mapping between the two contours prior to registration, the method would be equivalent to segmenting the two images independently. Thus, this method requires some structure to be imposed on g, and is well suited for rigid registration, where g has the imposed form of:

$$g(\mathbf{x}) = RM\mathbf{x} + \mathbf{T}\ , \tag{3.41}$$

with R being a rotation matrix, M a scaling matrix and $\mathbf{T}$ a translation vector. Equation

3.40 then becomes:

$$\begin{cases} \dfrac{\partial C}{\partial t} = & f\mathbf{N} + m\left(\hat{f} \circ g\right)\mathbf{N} - \kappa\mathbf{N} & (3.42a) \\ \dfrac{\partial g_i}{\partial t} = & \int_C \left(\hat{f} \circ g\right)\left\langle \frac{\partial g}{\partial g_i}, mRM^{-1}\mathbf{N}\right\rangle \quad , & (3.42b) \end{cases}$$

with m the determinant of M.

Registration relies on segmentation results through the use of the level set contour C and the segmentation speed $\hat{f}$ in (3.40b). This allows the method to demonstrate very good results on overlapping 2D and 3D data, especially for rigid registration. However, the method is limited to region-based segmentation algorithms and to two overlapping images or full volumes, and can not process sparse data. Therefore, we will propose a generalisation of this method to multiple and sparse, non-overlapping data, in Chapter 5.

3.4 Conclusion

In this chapter, we presented the current state-of-the-art of sparse and misaligned tomographic data analysis, with an emphasis on selected methods which we will compare against in our experiments in Chapters 4, 5, and 6. We highlighted that sequential methods are not best suited to solve the inter-dependent issues of registration, segmentation, and interpolation. However, the integration of these three stages has not so far been achieved in a satisfactory way. Most of the existing works only attempt to combine two of these stages, and they can not apply directly to sparse and misaligned data. The only method known to date that integrates all stages is limited to one specific application, *i.e.* heart modelling from cardiac cine MRI, and lacks flexibility for modelling both a variety of objects and shapes that may have important variations. Thus, there is a need for a method that integrates all three stages of registration, segmentation, and interpolation, and that is flexible enough to model any shape, and from any modality. Chapters 4 and 5 will be devoted to the design of such a method, following the inspirations of the previous and related works that we detailed in Section 3.3 of this chapter.

Chapter 4

ISISD: Integrated Segmentation and Interpolation for Sparse Data

In this chapter, we introduce a new general-purpose segmentation and interpolation framework for sparse data, which we refer to as ISISD. It extends the classical level set method, described in Section 3.3.1, in order to handle sparse data by providing interpolation of the object's shape in the gaps in the data in addition to the usual level set segmentation. Since segmentation and interpolation of such sparse data are two closely intertwined problems (see the discussion on their interdependency in Chapter 1), the strategy of ISISD is to integrate them into the same framework, for mutual benefit.

ISISD addresses the problems of data sparsity and varying spatial configurations, which are common issues in object modelling, as highlighted in Chapter 1 and detailed in Chapter 2. Specifically, in the case of medical images, it addresses the issue of thick and widely spaced slices whose positions and orientations may vary greatly depending on the imaging modality, patient, imaged organ, the pathology being investigated and the preferences of the operator. This will be demonstrated in Section 4.5 and Chapter 6. As will be illustrated in Appendix A, it also addresses the challenges of sparse and noisy data produced by depth cameras and range sensors, notably low resolution and irregularly spaced 3D point clouds, and holes such as those due to occlusions, low reflectance, constraints on sensor placement, or missing views.

Another problem addressed by ISISD is the variation in gain and contrast which can sometimes be found in the same dataset. A typical example is SA and LA slices in some cardiac MRIs, as was illustrated in Figs. 1.3a,b. It is also interesting to process images from different modalities, like in Figs. 1.3c,d, in order to exploit simultaneously the different information that they contain. In such cases, the traditional approach, which consists of interpolating image intensities (see Section 3.1.1), cannot be directly used.

4.1 Overview of ISISD

ISISD performs segmentation and interpolation simultaneously in order to let these two processes benefit from each other. Notably, interpolation exploits segmentation results, while 3D or 4D segmentation could not be performed without interpolation and would be restricted to separate and independent 2D segmentations of the slices, that may be mutually inconsistent. Contrary to the deformable model presented in Section 3.1.2, ISISD integrates them into a level set framework in order to benefit from the high flexibility and better computational efficiency of level set methods.

A key aspect of ISISD is that interpolation is performed on the level set's implicit surface which is segmenting the data and volume, rather than on image intensity, as is traditionally done in medical image analysis. Note that this approach is consistent with our strategy of integrating segmentation and interpolation. To begin with, speeds are computed for points of the implicit surface which have data support. Then, the remaining parts of the surface, *i.e.* the parts which are in the gaps, are updated in order to yield an appropriate interpolation of the shape of the object in the gaps. This is iterated until convergence of the surface on the object's boundary (see overview schematic in Fig. 4.1).

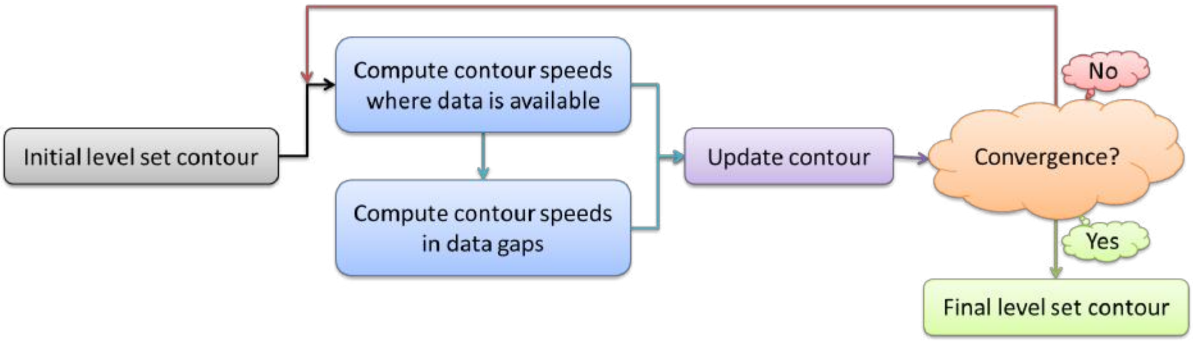

Initial level set contour
Compute contour speeds where data is available
Compute contour speeds in data gaps
Update contour
Convergence?
No
Yes
Final level set contour

Figure 4.1: *Overview of ISISD.*

Interpolation is provided by a novel level set scheme based on the RBF interpolation of the level set function. This new level set framework combines the two uses of RBFs which were presented in Section 3.3.2, namely interpolation and segmentation. It therefore benefits from the two advantages of these methods. First, similarly to [1, 45, 55], it yields a level set segmentation which is more robust to noise and more numerically stable than a classical level set one. Second, it can handle sparse data inherently and it automatically provides shape interpolation in the holes. Formulation of this novel level set framework and illustration of its interpolation ability will be provided later in this chapter.

Contrary to most image intensity interpolation methods which are restricted to stacks of parallel images, our shape interpolation scheme does not require a specific spatial configuration of the data. Hence, both sets of slices having arbitrary locations and orientations, and even sets of unorganised 3D points, are inherently supported.

In addition, speeds for surface points which have data support may be computed using any choice of segmentation algorithm. Both edge and region-based methods may be used as well as prior knowledge. Therefore, ISISD is very general and may segment various types of datasets, *e.g.* sets of medical image slices having various appearances by choosing an appropriate segmentation algorithm which exploits all available information. It is even possible to segment datasets from different modalities simultaneously by using different speed computation algorithms for each modality, as will be demonstrated later in this chapter.

In Section 4.2 we present the segmenting surface interpolation approach in more detail, together with its motivations and benefits. We also propose an initial and simple framework that implements this approach in an iterative way, and which is used to validate it. Section 4.3 describes the formulation of the ISISD framework that implements our approach in a more efficient and robust manner. Section 4.4 covers the main implementation aspects of ISISD, and evaluations of its interpolation properties are provided in Section 4.5.

4.2 Interpolation of a Segmenting Surface and Initial Implementation

In this section, we propose a new approach for the simultaneous segmentation and interpolation of sparse data. It extends the classical level set method to enable it to partition volumes that do not have a data support everywhere.

We also present a first framework that implements this approach by updating iteratively and concurrently the segmentation and interpolation processes. This framework has the advantage of being simply implemented and it will be used to validate our proposed concept of performing segmentation and interpolation simultaneously by interpolating the level set surface. A second framework that implements this concept in a more robust way will be presented in the next section.

4.2.1 Formulation

We propose to make sense of sparse data by *interpolating a level set implicit surface* where data is not available and *during the segmentation process*. Thus, we integrate segmentation and interpolation in a level set framework and in a novel way. This approach is a new contribution of this work.

Note that our proposed approach is similar to the method of Morigi and Sgallari [37], who interpolate a level set implicit surface on sets of slices using level set morphing to interpolate the surface between 2D contours which are grey-level lines of parallel images (see more detailed description of their method in Section 3.1). However, this method does not attempt to segment the data and is restricted to stacks of parallel slices. On the other hand, our proposed approach integrates this interpolation into a segmentation process. Therefore, for sets of slices, instead of interpolating every grey-level line, we interpolate only the shape being segmented, which corresponds to a grey-level line in the ideal segmentation case.

We first implement this segmenting level set surface interpolation approach by updating iteratively and concurrently the two stages of data segmentation and surface interpolation until convergence of both processes. We denote this framework as “concurrent framework”. Its formulation is the following. As for a conventional level set method, we compute speeds for the implicit surface at each iteration at the locations of points that have data support, according to the chosen segmentation algorithm. Points located in gaps between the image slices cannot be assigned a speed by the segmentation algorithm, due to the lack of data. This is a limitation of the classical level set framework. However, in our proposed framework, we suggest to generate speeds for these points using an interpolation algorithm (possible interpolation algorithms will be described later in this section). Then, these two types of speeds are used jointly to update the level set implicit surface, until convergence. The algorithm of this concurrent framework is summarised in Algorithm 4.1. In the next section, we will propose a second framework that provides a more robust implementation of our level set segmenting surface interpolation approach.

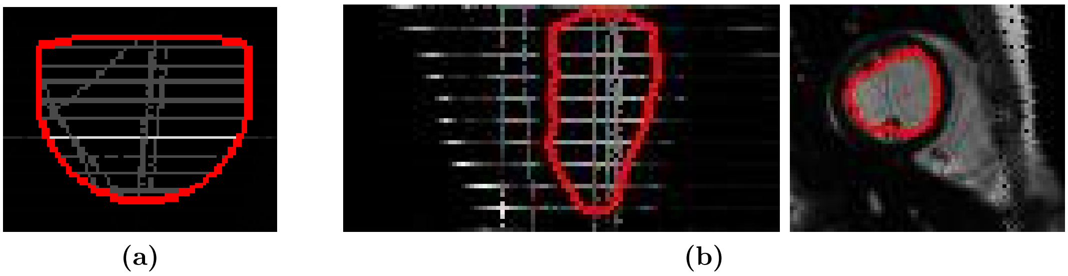
```
Initialise a surface through a level set embedding function
repeat
  for all points with data support do
    Compute speed according to the chosen segmentation method
  end for
  for all points without data support do
    Compute speed according to the chosen shape interpolation method
  end for
  Update the level set function
until convergence
```

Algorithm 4.1: *Integrated level set based segmentation and interpolation for sparse data by concurrent segmentation and interpolation of the segmenting surface.*

Since we want to handle datasets of arbitrary spatial configurations, we should choose an interpolation scheme for the level set implicit surface which is not restricted to parallel slices, but can handle both sets of slices having arbitrary positions and orientations and 3D point clouds. Possible and simple schemes are Mean Curvature Flow (MCF) [9] and Surface Diffusion Flow (SDF) [10], that yield minimal surfaces and surfaces having minimal curvature variations respectively. They ensure that the surface remains continuous and smooth between fixed parts of the surface, *i.e.* parts which are in the images' planes or at the location of a 3D data point, and whose position is decided by the segmentation stage.

4.2.2 Validation of our Level Set Segmenting Surface Interpolation Approach

In order to prove our proposed concept of interpolation of a segmenting level set surface, we tested our approach using our concurrent framework and the simple MCF and SDF for interpolation, implemented using the semi-implicit schemes proposed in [47]. We applied our method to both artificial images and cardiac MRIs, and in particular to the datasets of Fig. 3.1 that the traditional sequential image interpolation followed by segmentation failed to handle. The results for these datasets are shown in Fig. 4.2.



(a) (b)

Figure 4.2: *Integrated interpolation and segmentation, by MCF interpolation of the segmenting surface, of (a) an artificial geometrical shape dataset and (b) a real cardiac MRI, both made up of SA and LA slices having different gains and contrasts. (b) left: central vertical slice of the volume viewed from the side, right: central SA slice. Red: segmenting surface. Segmentation was done using an edge-based segmentation algorithm [61].*

A first benefit displayed by our segmenting surface interpolation approach is that the issue of different gains and contrasts, reported in Section 3.1, does not arise when interpolating the segmenting surface rather than image intensity, since no interpolation artefacts are produced which could bias the segmentation. Thus, in Fig. 4.2, the same edge-based segmentation algorithm as in Fig. 3.1 achieved satisfactory segmentations of the volumes. 2D images having different gains and contrasts due to different modalities may also be processed simultaneously, as in the two bottom rows of Fig. 4.4, where a region-based segmentation algorithm was used with different intensity models for each image in order to take into account their different appearances.

In addition, the problem of arbitrary spatial configuration is solved if the interpolation scheme used to update the implicit surface in the gaps in the data does not make any assumptions on the spatial configuration of the dataset. This is the case of the MCF and SDF schemes used in our preliminary tests. Using these schemes, slices having arbitrary positions and orientations could be processed, such as the SA and LA slices in the cardiac MRI dataset of Fig. 4.2b.

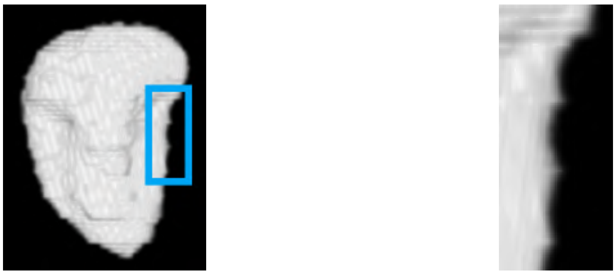
A third advantage of our integrated approach is that segmentation and interpolation are performed simultaneously and can therefore benefit from each other. To the best of our knowledge, this is the first time that segmentation and interpolation are combined in a level set framework. This integration is an important feature of our proposed method since these two processes are closely intertwined, as was highlighted in Chapter 1. Thus, in the tests of Fig. 4.2, interpolation succeeded through the use of segmentation results rather than image intensity. At the same time, the segmenting surface could delineate the whole objects directly in 3D, even in the gaps between the images, thus obtaining more coherent segmentations of the individual images. This benefit will be also illustrated by more results in Section 4.5, notably in cases were the integrated approach performs more robust interpolation than sequential image intensity interpolation followed by segmentation, due to the interpolation stage using shape information from the segmentation results.

4.2.3 Limitations of our Initial Framework

Our tests with our concurrent framework implemented with MCF and SDF validate our proposition to integrate segmentation and interpolation in the same framework and our approach to interpolate a segmenting surface rather than the images. However, the MCF and SDF schemes used to update the surface in the gaps show some limitations. First, the MCF method produces minimal surfaces between the intersections of the 3D implicit surface with the image planes. Depending on the applications, such surfaces may not be appropriate, as illustrated in Fig. 4.3. Indeed, Chopp and Sethian [9] showed that a cylinder evolved under the MCF between two fixed rings becomes a catenoid if the rings are close enough, and disappears if the distance between the rings is larger than a maximum distance. This caused the numerous catenoid-like shapes in Fig. 4.3.

Therefore, it may be more suitable to take into account the global geometry of the object in order to produce a better interpolation. Surfaces yielded by the SDF method tend to be more visually satisfactory, but like MCF, SDF is also not easily scalable. In particular, for large datasets and/or large gaps between the slices, many SDF iterations are required at each segmentation iteration in order to successfully drive the implicit surface from one side of the gap to the other. In addition, for bigger volumes or gaps, the method often fails to propagate the implicit surface satisfactorily, as in Figs. 4.4c,d where it is not interpolated smoothly in the gaps of a vertebra dataset, and in Figs. 4.4a,b,g where it fails to propagate respectively between radial slices, and to the upper and lower parts of brain ventricles which extend further up and down inside the skull, as shown by the raw data of Figs. 4.4h,i.

To sum up, the main limitation of our concurrent framework with diffusion flows is that it can only provide a local interpolation of the level set surface between two image planes. This motivates the development of a new implementation for our segmenting level set surface interpolation approach, that would provide a more global interpolation of the level set surface.



(a) **(b)**

Figure 4.3: *Limitations of interpolation by surface diffusion in gaps using an MCF – the modelled object is the left ventricle cavity of a heart reconstructed from an MRI dataset. (a) global view, (b) zoom of the highlighted part of (a). Note that the MCF scheme produces catenoid-like shapes in (b).*

4.3 RBF Interpolated Level Set

In this section, we address the limitations of our previous concurrent framework by introducing a new implementation of our segmenting level set surface interpolation approach that yields a more robust interpolation for the segmenting level set surface. This framework is based on the RBF interpolation of the level set function, and it provides interpolation at the same time as segmentation by combining the two uses of RBFs that were presented in Section 3.3.2, namely interpolation of sparse data and image segmentation respectively. It produces an intrinsic interpolation of a level set segmenting surface without the drawbacks of the MCF or SDF methods that were highlighted in Section 4.2.3. We denote as ISISD this segmenting level set surface interpolation framework implemented with our new RBF interpolated level set scheme.

4.3.1 Formulation

Let $\phi(\mathbf{x})$ be a level set function and $\psi_i(\mathbf{x})$ an SPD RBF centred on a control point $\mathbf{x}_i$. ϕ may be interpolated as

$$\phi(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \psi_i(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \psi(\mathbf{x} - \mathbf{x}_i), \tag{4.1}$$

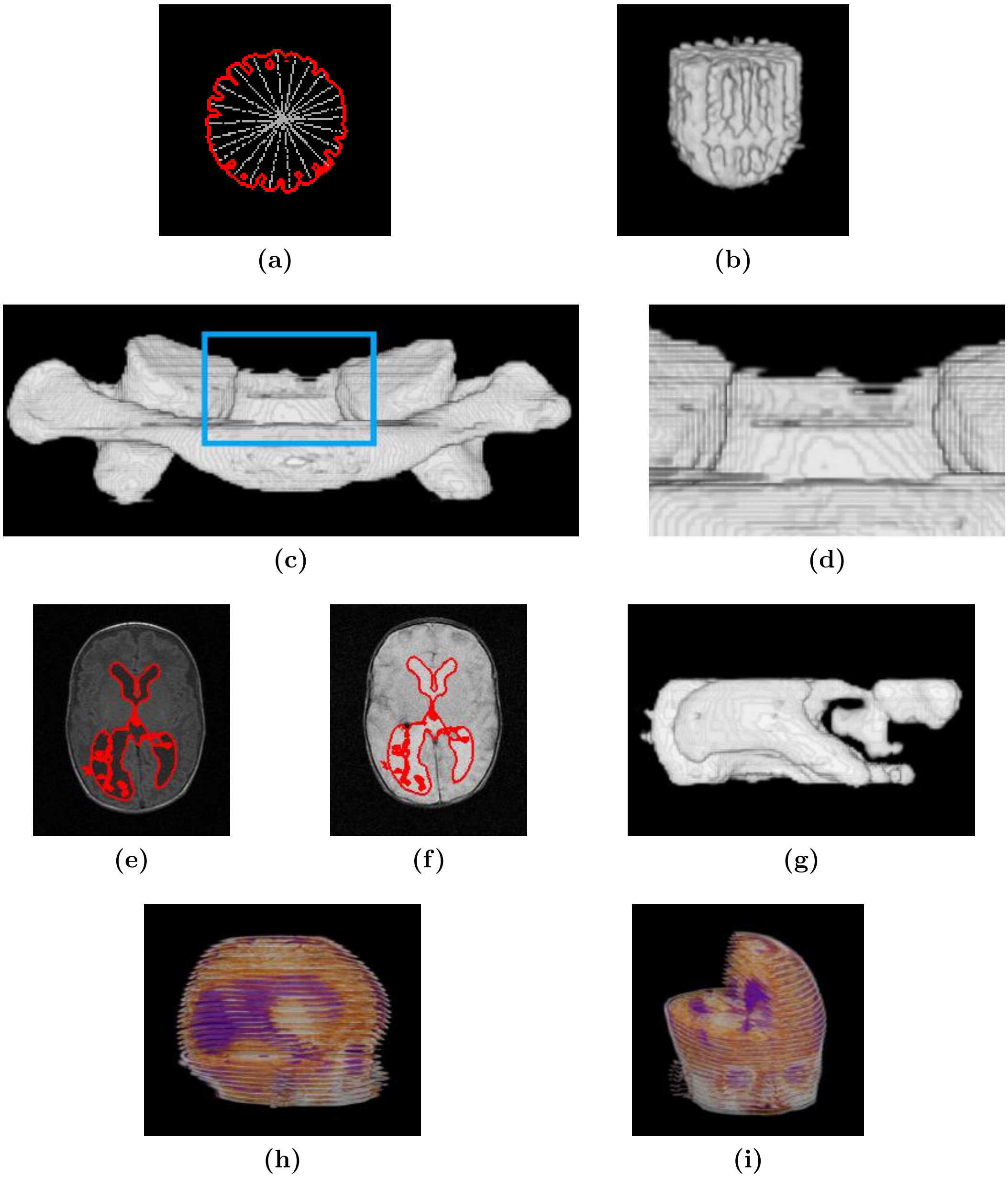


Figure 4.4: *Limitations of interpolation by surface diffusion in gaps using an SDF. First row: artificial geometric object modelled from a radial dataset: (a) central horizontal slice of the volume – red: segmenting surface, (b) 3D view of the reconstruction. Second row: vertebra modelled from a CT-scan made up of parallel slices: (c) reconstruction viewed from the front, and (d) zoom of the highlighted part of (c). Last two rows: brain ventricles modelled from (e) T1 and (f) T2 parallel slice MRIs, with (g) reconstruction viewed from the side. The brain dataset is shown in (h) from the side and (i) from the front, with parts removed in order to allow a better visualisation of the ventricles, in purple. Note that the SDF scheme fails to propagate the contour in particularly large gaps.*

where α_i are weighting coefficients. The method proposed by [56] to evolve ϕ through $\boldsymbol{\alpha} = [\alpha_1 \dots \alpha_N]^T$ involves inverting and storing an $N \times N$ matrix, which may be very time consuming and memory hungry when dealing with the large number N of points which compose a 3D or 4D dataset. Instead, following the idea of [1, 45] who addressed this inversion issue by deriving new ODEs using a variational method, we derive a new evolution for $\boldsymbol{\alpha}$ by minimising an energy functional $E[\phi]$ governing the segmentation of the 3D/4D volume Ω, *i.e.*

$$E[\phi] = \int_\Omega \mathcal{F}[\phi(\mathbf{x})]\, d\mathbf{x} = \int_\Omega \mathcal{F}\left[\sum_{i=1}^{N} \alpha_i \psi_i(\mathbf{x})\right] d\mathbf{x}. \tag{4.2}$$

$\mathcal{F}[\phi]$ may be any functional and is defined by the chosen segmentation method. Conventional variational level set methods derive an evolution equation for ϕ from (4.2) through a gradient descent method as

$$\frac{\partial \phi}{\partial t}_{conv.}(\mathbf{x}) = -\frac{\partial E}{\partial \phi}(\mathbf{x}) = -\frac{\partial \mathcal{F}}{\partial \phi}(\mathbf{x}). \tag{4.3}$$

Using (4.1), (4.3), and the chain rule, a gradient descent method on (4.2) yields the evolution equation for α_i such that

$$\frac{d\alpha_i}{dt} = -\frac{\partial E}{\partial \alpha_i} = -\int_\Omega \frac{\partial \mathcal{F}}{\partial \alpha_i}(\mathbf{x})\, d\mathbf{x} = -\int_\Omega \frac{\partial \mathcal{F}}{\partial \phi}(\mathbf{x}) \frac{\partial \phi}{\partial \alpha_i}(\mathbf{x})\, d\mathbf{x} = \int_\Omega \frac{\partial \phi}{\partial t}_{conv.}(\mathbf{x})\, \psi_i(\mathbf{x})\, d\mathbf{x}. \tag{4.4}$$

Let us rename $\frac{\partial \phi}{\partial t}_{conv.}(\mathbf{x})$ in (4.3) as $S(\mathbf{x})$. S is the speed of the moving level set contour C (or surface in 3D), and it commonly contains a term based on the Dirac function $\delta(\phi(\mathbf{x}))$ that restricts is to the contour C only. Therefore, we can simplify (4.4) into

$$\frac{d\alpha_i}{dt} = \int_C S(\mathbf{x})\, \psi_i(\mathbf{x})\, d\mathbf{x} \approx \int_\Omega \delta_\varepsilon(\phi(\mathbf{x}))\, S(\mathbf{x})\, \psi_i(\mathbf{x})\, d\mathbf{x}, \tag{4.5}$$

where δ_ε is an approximation of the Dirac function δ. The restriction of S to the contour C is necessary to ensure the stability of the method, *i.e.* if the object to be segmented is small with respect to the size of the image, the speeds of the background points would have an overwhelming effect in (4.4) – they would therefore occlude the influence of the more important speeds of the points located on the object edges. Our experiments showed that δ_ε should be larger with increasing flatness of the RBF in order to allow the contour to converge smoothly without oscillating around the edges of the segmented object. In practice, we used

$$\delta_\varepsilon(x) = \begin{cases} \frac{1+\cos\left(\frac{\pi x}{\varepsilon}\right)}{2\varepsilon} & |x| \leq \varepsilon \\ 0 & |x| > \varepsilon \end{cases}, \tag{4.6}$$

with $\varepsilon = 1$ for sharp RBFs and $\varepsilon = 3$ for flatter RBFs. Similarly to $\mathcal{F}$, S depends on the chosen segmentation method which may be any method of choice.

When segmenting sparse data, no data are available to compute a front's speed in the gaps, so S is set to zero there. Otherwise, S is computed for all images at their intersections with the contour, and the values of S are averaged when several images are present at the same location.

RBF based interpolation methods usually define one control point per data point. Instead, we choose to define one control point per voxel of a discrete space, allowing (4.5) to be re-written as a convolution which can then be computed efficiently using the Fast Fourier Transform (FFT)[1], as

$$\frac{d\alpha_i}{dt} = \sum_C S(\mathbf{x})\,\psi_i(\mathbf{x}) \approx \sum_\Omega \delta_\epsilon(\phi(\mathbf{x}))\,S(\mathbf{x})\,\psi(\mathbf{x}_i - \mathbf{x}) = ((\delta_\epsilon(\phi)\cdot S) \star \psi)(\mathbf{x}_i)\ , \tag{4.7}$$

where $\star$ is the convolution operator.

Note that the FFT assumes the signal to be periodic, and therefore, for non-periodic dimensions of the data (*i.e.* the spatial dimensions in the case of medical images, plus the temporal dimension in some cases), speeds S that are too close to an edge of the volume may have a significant influence on the other side of the volume. Such effect may bias the segmentation or create some artificial contours. In our experiments, this rarely happened since the objects and their associated contour speeds S were generally well centred in the volume and the edges were out of the influence area of the associated RBFs. However, if necessary, the problem can be easily solved by adding an empty border to the volume, that puts the volume's edges out of reach of the RBFs that are located at the positions of the contour speeds S.

[1] Indeed, a convolution of two functions in the spatial domain is equal to the multiplication of their Fourier transforms, that may be computed using the FFT.

$\frac{d\alpha_i}{dt}$ is then used to update $\boldsymbol{\alpha}$:

$$\boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}^{t} + dt\frac{d\boldsymbol{\alpha}}{dt}. \tag{4.8}$$

This step implicitly updates the level set function since ϕ is defined completely by $\boldsymbol{\alpha}$ according to (4.1). In addition, (4.1) implies that:

$$\frac{\partial\phi}{\partial t}(\mathbf{x}) = \sum_{i=1}^{N}\frac{d\alpha_i}{dt}\psi_i(\mathbf{x}) = \left(\frac{d\boldsymbol{\alpha}}{dt}\star\psi\right)(\mathbf{x})\,. \tag{4.9}$$

Note that this $\frac{\partial\phi}{\partial t}(\mathbf{x})$ is different from the one calculated by conventional level set methods in (4.3).

The initial $\boldsymbol{\alpha}$ may be easily computed in the Fourier domain:

$$\phi(\mathbf{x}) = \sum_{i=1}^{N}\alpha_i\psi(\mathbf{x}-\mathbf{x}_i) = (\boldsymbol{\alpha}\star\psi)(\mathbf{x})\,, \tag{4.10}$$

$$\hat{\boldsymbol{\alpha}}(\omega) = \frac{\hat{\phi}}{\hat{\psi}}(\omega)\,. \tag{4.11}$$

where $\hat{\boldsymbol{\alpha}}$, $\hat{\phi}$ and $\hat{\psi}$ are the Fourier transforms of $\boldsymbol{\alpha}$, ϕ and ψ respectively, and ω is the frequency.

When evolving the level set implicit surface under both data terms S_{data} and geometric terms S_{geom} which only depend on the geometric properties of ϕ, such as *e.g.* smoothing terms, then only the data terms should be used in the computation of S and its derived $\frac{\partial\phi}{\partial t}$ in (4.9), while the geometric terms should be used to update ϕ directly like in classical level set frameworks. Indeed, the geometric terms do not depend on data availability and can be computed everywhere in the volume, so they do not require any interpolation. In addition, their convolutions by an RBF are not suitable since it may modify their properties and degrade their performance. Therefore, when using such geometric terms, we recommend updating the level set function using the following evolution equations for ϕ, which combine (4.7), (4.9), and the geometric terms:

$$\frac{\partial\phi}{\partial t}(\mathbf{x}) = (((\delta_\epsilon(\phi)\cdot S_{data})\star\psi)\star\psi)(\mathbf{x}) + S_{geom}(\mathbf{x})\,, \tag{4.12}$$

$$\phi^{t+1} = \phi^{t} + dt \frac{\partial \phi}{\partial t} \ . \tag{4.13}$$

Note however that smoothing terms, such as the traditional curvature term, were found to be unnecessary in our experiments, thanks to the robustness to noise of the proposed framework, which we describe next in Section 4.3.2.

Unlike conventional level set methods, the proposed evolution scheme for the RBF interpolated level set does not use $\nabla\phi$ (see (4.7) and (4.8)). In addition, our experiments showed that ϕ tends to remain a smooth function due to the smoothing effect of the two convolutions by a decreasing RBF in (4.7) and (4.9). It is therefore unnecessary to regularly normalise ϕ to maintain $\|\nabla\phi\| = 1$. Finally, it is not necessary to compute an extended velocity for points which are not on the contour, as must be done with conventional level sets, because only contour points are used in (4.7).

```
Initialise a contour $C$ through a level set function $\phi(\mathbf{x})$
Compute the initial coefficients $\boldsymbol{\alpha}$ using (4.10)
repeat
    for all $\mathbf{x}$ on $C$ do
        Compute $S(\mathbf{x})$ according to the chosen segmentation method
    end for
    Compute $\frac{d\boldsymbol{\alpha}}{dt}$ using (4.7)
    Update $\boldsymbol{\alpha}^{t+1}$ using (4.8)
until convergence
```

Algorithm 4.2: *RBF interpolated level set for integrated segmentation and interpolation for sparse data*

This new level set scheme is summarised in Algorithm 4.2. It should be noted that it is different from [45] since it does not require adding, merging or removing RBFs, or updating the properties of the RBFs and their position. It is also more general than [1, 45] since any segmentation criteria, through the choice of F in (4.2), and any SPD RBF, may be used, in any number of dimensions. In addition, it can inherently handle sparse data and interpolate the shape of the object in gaps, while [1, 45] were designed to segment dense data only. It should also be stressed that, even if the proposed method looks similar to [5] which uses RBF interpolation to model objects from 3D point clouds, it differs from it by its ability to segment both the volume which contains the data, and the data itself.

We also note that, similar to our concurrent framework of Section 4.2 with the MCF and SDF schemes, this new framework can be used with any spatial configuration, since Algorithm 4.2 does not make any assumption on the type and position of the data. Notably, the data points do not need to be aligned on the grid of an image, and the method could handle unorganised 3D point clouds, as will be demonstrated in Appendix A. The difference in the processing of 3D point clouds and sets of 2D slices only arises in the computation of the front speed S, which depends on the chosen segmentation algorithm only. This choice is based on the type of data and the imaging modality, and will be discussed in Section 4.4.4.

4.3.2 Benefits of the RBF Level Set Scheme

Robustness to noise – The RBF interpolated level set framework has the interesting property of being very robust to noise in the speeds S, as illustrated in Fig. 4.5d (the influence of the RBF flatness on the level of robustness to noise will be discussed later, in Section 4.4.2). Classical level sets do not inherently handle noise, as shown in Figs. 4.5a,b, and usually require a curvature term in their evolution equations in order to achieve a result similar to Fig. 4.5d. This curvature term smooths the level set contour, and so it tends to prevent it from segmenting sharp corners when the smoothing is too high. It is unnecessary in our RBF interpolated level set scheme, since the robustness to noise is due to the two consecutive Gaussian-like smoothings that (4.7) and (4.9) apply to S in order to obtain $\frac{\partial \phi}{\partial t}$ (rather than to the contour directly). Indeed, as can be seen in Fig. 4.8g, when the speeds S produced by an object are in the minority with regards to surrounding speeds, their influence in the computation of $\frac{\partial \phi}{\partial t}$ becomes negligible, and the object is ignored by the contour. Note that these smoothings do not prevent the contour from segmenting sharp corners in Fig. 4.5d, as sometimes happens when using a curvature term.

Interpolation – The main advantage of the new RBF interpolated scheme when processing sparse data is that the segmenting contour is inherently interpolated in gaps in the images or volumes, since, even where S is set to zero because of a lack of data support, $\frac{\partial \phi}{\partial t}$ is not null. This can be seen as interpolating the speed of the contour in the gaps. Thus, this scheme implements our approach of interpolating a segmenting level set contour or surface, that was proposed in Section 4.2. Fig. 4.6 illustrates the concept with an example of an image with a gap being segmented. In Fig. 4.6b, one can assess using (4.7) and (4.9) that $\frac{\partial \phi}{\partial t}$ is respectively negative and positive at points A and C, and

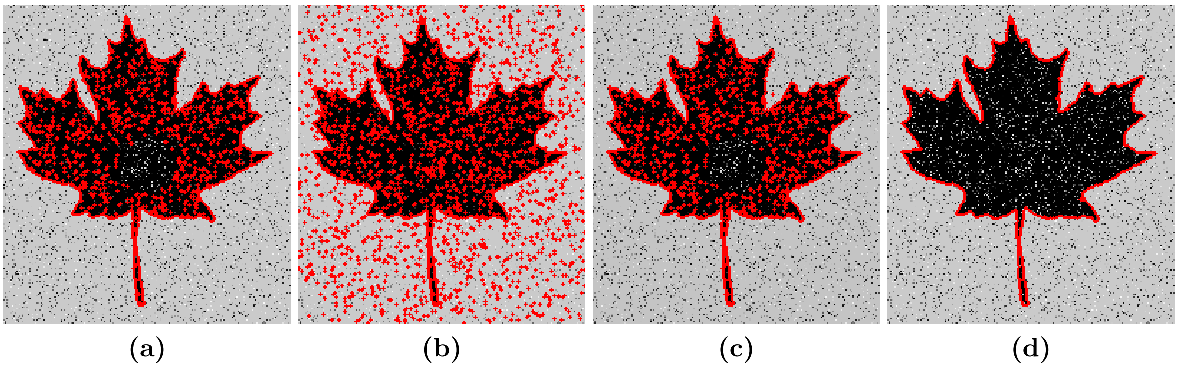


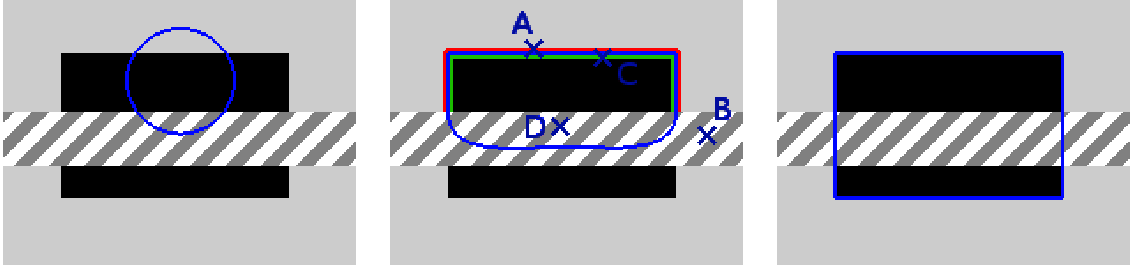
Figure 4.5: *Robustness to noise in the segmentation speeds S: noisy image segmented by the piecewise constant model used in [7] that produces noisy speeds S, using (a) conventional narrow-band level set, (b) Chan-Vese level set scheme [7], RBF interpolated level set with (c) a sharp RBF ($\gamma = 0.1$, with γ being an RBF flatness parameter which will be defined in Section 4.4.1), and (d) a flat RBF ($\gamma = 1$). Initialisation: small circle at the centre of the image. The piecewise constant model was used to compute $\frac{\partial \phi}{\partial t}_{conv.}$ for (a) and (b) and S for (c) and (d).*

thus pushes the contour towards the edges of the object. It is also positive at point D, pushing the contour towards the other side of the gap. At point B, the speed is negative and prevents the contour from leaking elsewhere in the gap.

This inherent interpolation of the contour in gaps, without the need for an additional interpolation stage such as the surface diffusion flows used in our concurrent framework in Section 4.2, is a new contribution of this work. Indeed, the RBF interpolated level set schemes of Wang and Wang [55], Slabaugh et al. [45], and Bernard et al. [1], do not provide such interpolation, since they were designed for segmentation only.

4.4 Implementation

The new RBF interpolated level set scheme presented previously is a general scheme which can use any SPD RBF and any segmentation algorithm. These choices should be made by the user depending on the data, in order to accommodate data of arbitrary types and appearances, and containing gaps of any sizes. In this section, we provide implementation recommendations for choosing both the RBF and the segmentation algorithm.



(a) (b) (c)

Figure 4.6: *Segmentation and interpolation on an image with a gap. (a) initial contour, (b) speeds $S(\mathbf{x})$ – green: positive; red: negative, (c) final segmentation. Blue: level set contour.*

4.4.1 Choice of RBF Type

An evolution scheme similar to that presented in Section 4.3.1 could be derived for conditionally positive definite RBF, with the level set function approximated as

$$\phi(\mathbf{x}) = p(\mathbf{x}) + \sum_{i=1}^{N} \alpha_i \psi_i(\mathbf{x}) \tag{4.14}$$

where p is a polynomial which accounts for the linear and constant portions of ϕ. However, in our experiments such functions yielded rather poor results, with over-smoothed $\frac{\partial \phi}{\partial t}$ and final ϕ. Thus, only globally defined SPD RBFs would be recommended for applications similar to ours.

We wish to emphasise the importance of the global support of the RBF. Even if the segmentation stage could work with both compactly supported and globally defined RBFs, the interpolation stage may fail where gaps are larger than their support. We tested Gaussian RBFs and inverse multiquadric RBFs of the form $\psi(\mathbf{x}) = \left(\|\mathbf{x}\|^2 + \gamma^2\right)^{-\frac{\beta}{2}}$, where γ defines the shape of the RBF, and found that the flatness of the RBF has much more influence on the segmentation and interpolation quality than its type. We observed that the fast decay of the Gaussian function tends to make the evolution of the contour very slow in gaps. Therefore, we chose to use inverse multiquadric RBFs, with β equal to the number of dimensions of the data to ensure the decrease is not too steep. The choice of γ allows a finer tuning of the decrease rate.

4.4.2 RBF Flatness: Influence on Segmentation and Interpolation Results

The flatness of the RBF influences the quality of the segmentation and of the interpolation, since, as illustrated in Fig. 4.7, different flatness rates will produce different weights $\psi_i(\mathbf{x})$ to the speeds $S(\mathbf{x})$ in (4.7), thus modulating the extent of the influence of the speeds S. The choice of RBF flatness, through the value of γ, depends on the nature of the data, *e.g.* its resolution (including in-plane resolution for images, and gap sizes) and the size of details on the object. As a result, a given value for γ may be particularly high for one dataset and low for an other one. However, the same value for γ can be used for processing similar datasets, such as in our experiments in Section 4.5 (see Table 4.1).

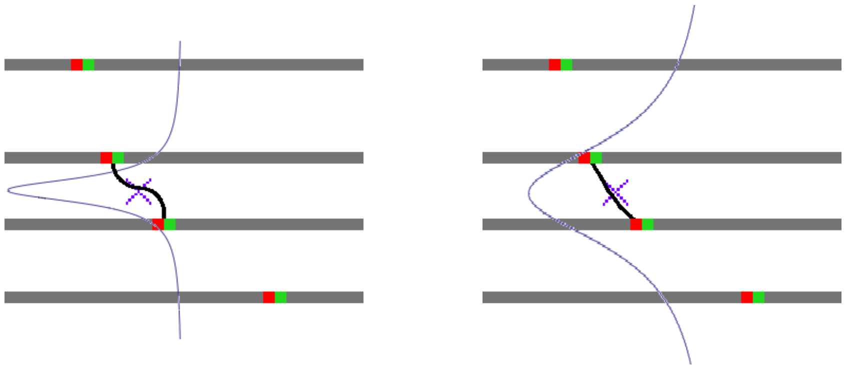


Figure 4.7: *RBF flatness and extent of the influence of speeds S – 2D image slices, seen from the side, with contour speeds in green (positive) and red (negative). A (a) sharp and (b) flat RBF, centred on a gap point (purple cross), is superimposed in blue, with the horizontal axis being its value and the vertical axis its spatial extent. Black: contour interpolated in the gap.*

Robustness to noise – The level of robustness to noise in the speeds S can be adjusted by setting the RBF flatness. Thus, in Figs. 4.5c and 4.11c, a sharp RBF provides very little robustness to noise, while in Figs. 4.5d and 4.11d, a flatter RBF prevents the segmentation from being impaired by the noise in S that is a consequence of the noise in the image. This effect of the RBF flatness on the robustness to noise can be understood by considering the example of Fig. 4.8, where a small hole is segmented when using a sharp RBF in the first row, but is missed in the second row where a flat RBF is used. Indeed, while the speeds S, shown in the second column, are negative inside the hole in both cases, they do not contribute equally to the computation of $\frac{\partial \phi}{\partial t}$ (third row) because of their different weights $\psi_i(\mathbf{x})$. In the case of a flat RBF, the influence of the few negative terms are overwhelmed by the numerous positive ones. As a result, the segmenting contour ignores the small hole, and all other holes or objects which are too small to produce significantly contributing speeds.

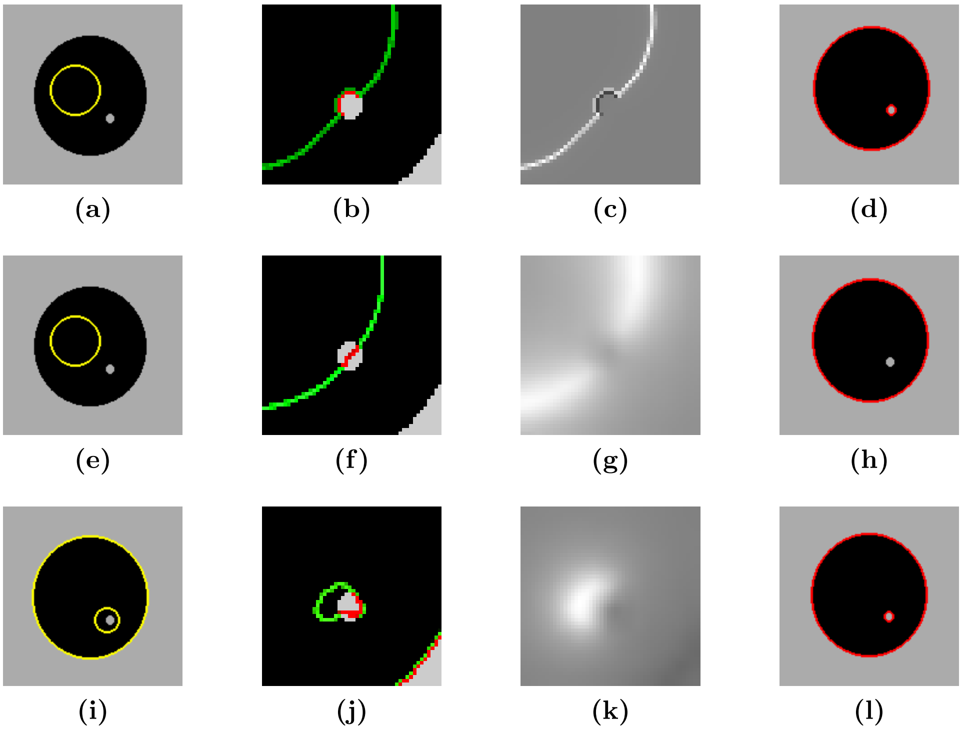


Figure 4.8: *Influence of the RBF flatness and of the initial position of the contour on the segmentation of a small hole. From left to right: initial contour, zoom on front speeds $S(\mathbf{x})$ at an intermediate state – green: positive; red: negative, zoom on $\frac{\partial\phi}{\partial t}(\mathbf{x})$ at an intermediate state – light grey: positive; dark grey: negative, final segmentation. Top row: a sharp RBF ($\gamma = 0.1$) segments the small hole in the middle of the object, while, in the middle row, a flatter RBF ($\gamma = 2$) treats the hole as noise and fails to delineate it. Bottom row: initialisation close enough to the edges of the small hole allows the speeds produced by the hole's edges to prevail against speeds produced by the object's points, resulting in the hole being segmented even with a flat RBF ($\gamma = 2$).*

A drawback of this robustness to noise is that an overly flat RBF, together with inadequate initial conditions, may prevent small holes and small objects from being segmented, as they would be treated as noise – see examples in Fig. 4.8h and Fig. 4.14e. In addition, if the surface of the modelled object has some fine details, such as bumps which are small in comparison to the gap sizes, then a particularly flat RBF results in a highly smoothing segmenting contour that misses the fine details, as illustrated in Fig. 4.9b.

Interpolation – On the other hand, a too-sharp RBF would provide only local interpolation, as illustrated in Fig. 4.7a, where only directly adjacent slices have an influence on a gap. This would result, for example, in a 'staircase-like' shape in the case of parallel

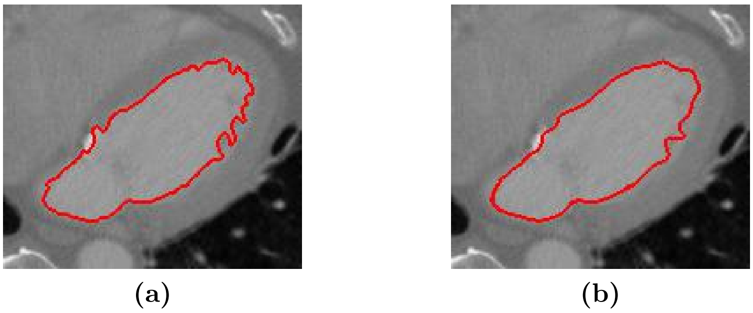


Figure 4.9: *Influence of the RBF flatness (via parameter γ) on the smoothing of surface details – the cavities of the left ventricle and atrium are segmented as one object. (a) a relatively sharp RBF ($\gamma = 2$) captures the irregularities of the edge of the modelled object, while (b) a flat RBF ($\gamma = 6$) smooths the segmentation and produces a less detailed contour. This image is a central long-axis slice of a CT-scan of the heart.*

slices (Fig. 4.14b). On the contrary, a flatter RBF allows the influence of speeds at data points to reach further into the gaps, as seen in Fig. 4.7b, and therefore provides a more global interpolation of the shape of the object (black curve in Fig. 4.7b).

Some situations require particularly flat RBFs in order to obtain a global enough interpolation. They arise for datasets containing large gaps, and/or slices that are at a small angle to the object's surface. This latter scenario is illustrated in Fig. 4.10 and will be evaluated in experiments on kidney and heart CT-scans reported in Section 4.5.5. In these two cases, a large γ had to be used in order to obtain a smooth and globally interpolated shape.

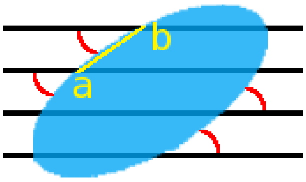


Figure 4.10: *Example of a dataset whose slices (black lines, seen from the side) have a small angle (red) with the surface of the modelled object (blue), resulting in a larger distance ab (yellow) between two neighbouring parts of the surface. Hence, such situations require a high degree of smoothing in the gaps.*

Segmentation energy – The robustness to noise and interpolation properties of our new RBF interpolated level set, and their sensitivity to the RBF flatness, are the main differences with a classical level set which would minimise the same segmentation energy

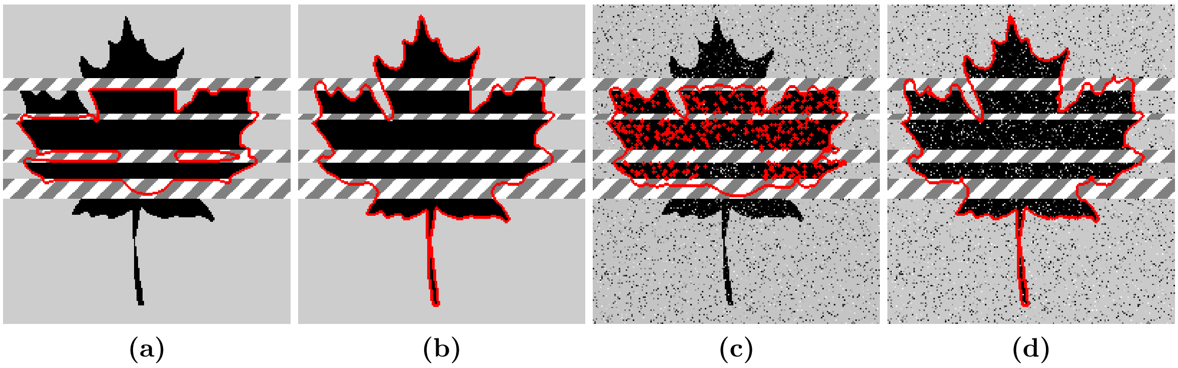


Figure 4.11: *Influence of the RBF flatness on interpolation and robustness to noise. Image with 'slices' of missing information (dashed areas) and noise being processed by ISISD using (a) and (c) a sharp RBF ($\gamma = 0.1$), and (b) and (d) a flat RBF ($\gamma = 1$). The piecewise constant model of [7] is used to compute S. Initialisation: small circle at the centre of the image. The conventional narrow-band level set and Chan-Vese level set schemes of Figs. 4.5a,b could not be used with these incomplete images because they can not handle missing information.*

(4.2). As explained in Section 4.3.2, they are due to the two consecutive smoothings which are applied to the front speeds S in order to compute the evolution of the level set function ϕ, and which impact the way the segmentation energy is minimised.

This modification of the minimisation of the segmentation energy is illustrated in Figs. 4.12 and 4.13, which present the influence of the RBF flatness on the minimisation of the segmentation energy in the case of noisy images and/or images with gaps. In these examples, the segmentation energy is computed in the case of a piecewise constant model [7]: $E_{PC} = \int_{\Omega_{in}} (I - c_{in})^2 + \int_{\Omega_{out}} (I - c_{out})^2$, where I denotes pixel intensity, Ω_{in} and Ω_{out} are respectively the interior and exterior of the contour, and c_{in} and c_{out} are the mean pixel values of the corresponding image areas.

When dealing with noisy images, sharp RBFs achieve the best segmentation energy minimisation (red curve in Fig. 4.12), and produce a segmentation (Fig. 4.5c) similar to the result of the conventional narrow-band level set (Fig. 4.5a) which is very sensitive to noise. A flatter RBF does not minimise the segmentation energy as efficiently (blue curve in Fig. 4.12), however this allows it to better handle the noise, since noisy pixels are now included into their correct regions (Fig. 4.5d).

When the image has gaps, a flat RBF also obtains a better interpolation (Fig. 4.11b) than a sharp RBF, which does not manage to propagate the contour in the gaps (Fig. 4.11a). The segmentation energy minimisation is therefore better (blue curve in Fig. 4.13a), since the segmenting contour can reach remote sides of gaps and segments the whole image. If the image with gaps is also noisy, the minimisation of the segmentation energy using a flat RBF is particularly non-optimal (blue curve in Fig. 4.13b), in order to both deal with the noise in the speeds S and perform the interpolation, in addition to segmenting the image. Note that in Fig. 4.13, the different steps in the red and blue curves are due to the contour propagating slowly in the gaps. At these stages, the segmentation of the image does not change significantly, and only the interpolation evolves. This evolution of interpolation is not visible on these graphs since they show the minimisation of the *segmentation* energy only.

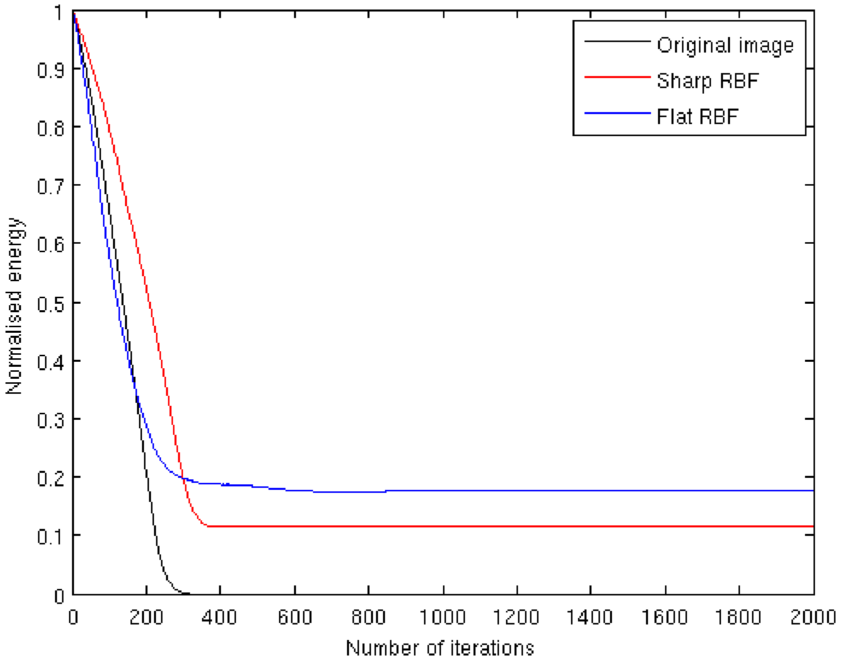
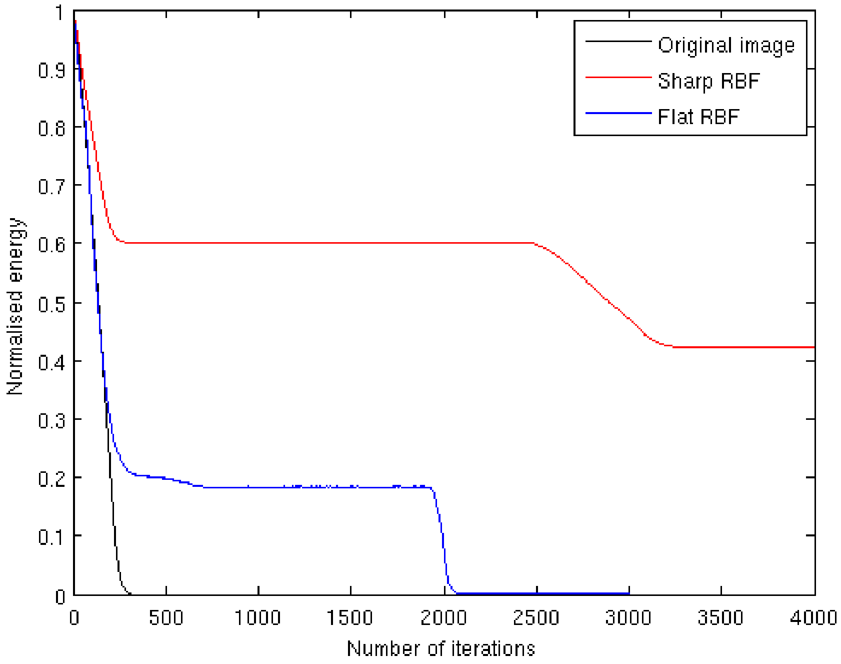


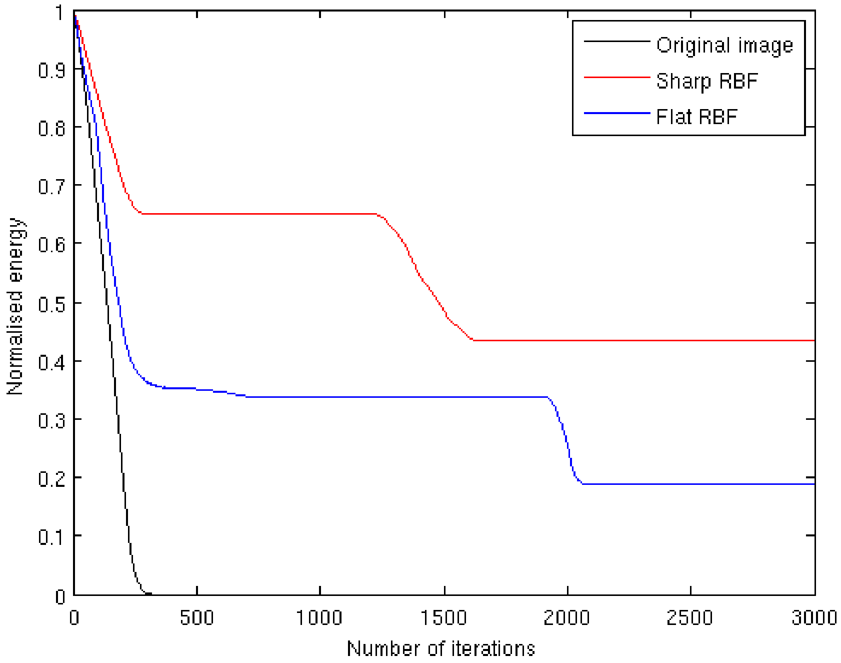
Figure 4.12: *Influence of the RBF flatness on the minimisation of the segmentation energy of a piecewise constant model on the noisy image of Figs. 4.5c,d. Black: original image (with no noise); red: sharp RBF ($\gamma = 0.1$); blue: flat RBF ($\gamma = 0.8$). The energies are normalised with their respective maximum values in order to allow comparison against the segmentation of the original image.*

4.4.3 Choice of γ and RBF Flatness

As discussed above, the flatness of the RBF ψ, through the value of γ, influences the quality of the segmentation and of the interpolation. Its choice is highly dependent on the nature of the data, *i.e.* on the level of robustness to noise required, the size of the object and of the gaps, and the level of detail on the surface of the object – these can



(a)



(b)

Figure 4.13: *Influence of the RBF flatness on the minimisation of the segmentation energy of a piecewise constant model on the images with gaps of (a) Figs. 4.11a,b without noise and (b) the noisy images of Figs. 4.11c,d. Black: original image (with no noise and no gaps); red: sharp RBF ($\gamma = 0.1$); blue: flat RBF ($\gamma = 0.8$). The energies are normalised with their respective maximum values in order to allow comparison against the segmentation of the original image.*

not be automatically evaluated before the object is modelled and, at this point, can only be assessed subjectively by the user from 3D visualisations of the data. Thus, its choice would be extremely difficult to automate, and currently needs to be determined by the user. However, as will be illustrated in our experiments, γ needs to be roughly tuned only, and the same value can generally be used for all datasets of the same type. In the simplest cases, an appropriate value for γ may be found easily which provides a trade-off between a suitable level of robustness to noise and a global enough interpolation. For example, $\gamma = 1.5$ was obtained empirically to model brain ventricles from five different datasets acquired by two different MRI scanners at 1T and 1.5T in Section 4.5.6. In other cases, we recommend two approaches, detailed below, which seek to provide a high degree of smoothing in the gaps while preserving the segmentation quality. The first approach preserves small objects and small holes which otherwise would be treated as noise when using a large value for γ. The second also preserves small details on the surface, *e.g.* small bumps, but at the expense of a slightly less satisfactory interpolation.

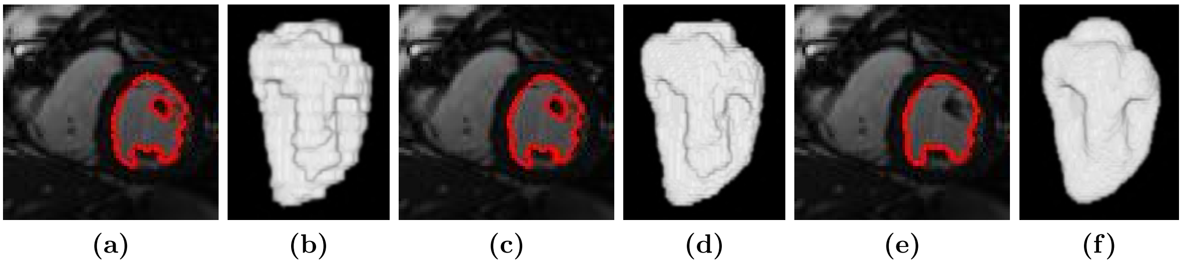


Figure 4.14: *Small object or hole preserving approach – the modelled object is the LV cavity of the heart of Fig. 4.3 with one central short-axis slice in the odd columns and 3D view in the even columns. Processing of a dataset in two stages: (a) and (b) initial segmented and locally interpolated surface using a sharp RBF ($\gamma = 0.7$), and (c) and (d) final segmented and globally interpolated surface using a flat RBF ($\gamma = 3$), to be compared with (e) and (f) processing of the dataset using a flat RBF ($\gamma = 3$) only. The small object or hole preserving approach ((a) to (d)) segments the papillary muscle inside the LV cavity, while the simplest approach with a single flat RBF ($\gamma = 3$) ((e) and (f)) treats it as noise.*

To obtain smooth and globally interpolated shapes while preserving small objects or holes – When a level set contour is initialised close to the edges of small objects or small holes, the speeds produced by these edges prevail against speeds produced by non-edge points, as illustrated in Figs. 4.8j and 4.8k, and the contour is therefore able to segment these small objects or holes, even when using a flat RBF (Fig. 4.8l). We generate such initialisations automatically by first processing the data with a low value for γ, which yields the required level of robustness to noise while preserving small objects and holes (Figs. 4.14a and 4.14b). This places the contour at the required initial position, *i.e.* on the edges of the modelled object and of the smaller objects and holes, *e.g.* as in Fig. 4.14a, with the LV cavity and the papillary muscle that is inside it. Then, starting from the contour or surface just obtained, we process the data with a high value of γ in order to refine the interpolation (Figs. 4.14c and 4.14d). Note that in Fig. 4.14c, the papillary muscle is still found, while it was not when the same value for γ was used, without preliminary segmentation, in Fig. 4.14e, and hence it was erroneously included in the LV cavity. This approach was used to process all our cardiac MRI datasets in Section 4.5.6 with γ being always set to 0.7 and 3 for the first and second steps respectively.

To preserve fine details on the surface at the expense of slightly less smooth and globally interpolated shapes – To prevent fine surface details being lost when a particularly high degree of smoothing in the gaps is required, we suggest the application of several RBFs, each having a different flattening strength, in the same interpolation, *i.e.*

$$\phi(\mathbf{x}) = \sum_{l=1}^{L} \sum_{\mathbf{x}_i \in \Omega_l} \alpha_i^l \psi^l(\mathbf{x} - \mathbf{x}_i), \tag{4.15}$$

where ψ^l is the RBF of flatness level l and Ω_l is the domain of the control points associated with RBFs of flatness level l, such that $\bigcup_l \Omega_l = \Omega$ and $\Omega_{l_a} \cap \Omega_{l_b} = \emptyset$ when $l_a \neq l_b$. These domains should be chosen such that sharp RBFs are used at data points, while, in the gaps, RBFs' flatness increases progressively away from data points. α_i^l is simply the previous coefficient α_i associated with the RBF ψ^l centred at point $\mathbf{x}_i$. We use this notation to establish that $\mathbf{x}_i$ and α_i are now associated with an RBF of a given flatness level l. Equations 4.7 and 4.9 then become:

$$\frac{d\alpha_i^{l_i}}{dt} = \left(\left(\delta_\epsilon \left(\phi \right) \cdot S \right) \star \psi^{l_i} \right) \left(\mathbf{x}_i \right), \tag{4.16}$$

with l_i such that $\mathbf{x}_i \in \Omega_{l_i}$, and

$$\begin{aligned} \frac{\partial \phi}{\partial t} \left(\mathbf{x} \right) &= \sum_{l=1}^{L} \sum_{\mathbf{x}_i \in \Omega_l} \frac{d\alpha_i^l}{dt} \psi^l(\mathbf{x} - \mathbf{x}_i) \\ &= \sum_{l=1}^{L} \sum_{\mathbf{x}_i \in \Omega} H_l \left(\mathbf{x}_i \right) \frac{d\alpha_i^l}{dt} \psi^l(\mathbf{x} - \mathbf{x}_i) \\ &= \sum_{l=1}^{L} \left(\left(H_l \cdot \frac{d\alpha^l}{dt} \right) \star \psi^{l_i} \right) \left(\mathbf{x} \right) \end{aligned} \tag{4.17}$$

with H_l being equal to 1 inside Ω_l, and 0 elsewhere. When using this surface-detail preserving approach, the initial α can not be computed as easily as in (4.10). However, it is easy to update ϕ directly using (4.16) and (4.17), without keeping track of α.

Although the resulting shapes will have a much higher degree of smoothing in the gaps than when using a sharp RBF only, they will be globally interpolated in the middle of the gaps only, where γ is highest, while the interpolation will be more local close to the data. Hence, the resulting interpolation is usually less satisfactory than with the small object or hole preserving approach.

Fig. 4.15 illustrates the ability of the two proposed approaches to process the extreme case of sets of parallel slices having particularly large gaps and a small angle with the modelled object's surface, hence requiring an exceptionally high degree of smoothing in the gaps. In Fig. 4.15a, the small object or hole preserving approach smooths out the fine details on the surface of the modelled object, while in Fig. 4.15b the surface-detail preserving approach is better at reconstructing them. On the other hand, in the dataset of Fig. 4.15c only four slices were available and thus did not contain much information on the fine details of the object's surface. Therefore, the small object or hole preserving approach could be used, with a very flat RBF ($\gamma = 12$) at the second step, without losing a significant amount of information. This approach obtained a better interpolation than the surface-detail preserving approach (Fig. 4.15d) which produced a more local interpolation close to the data.

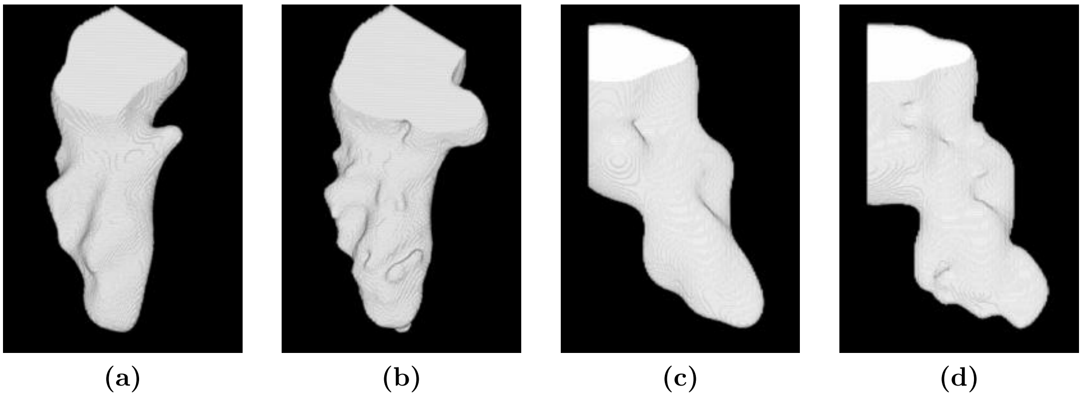


Figure 4.15: *Segmentation and interpolation on a dataset with large gaps and small angle between the images and the modelled object's surface – the modelled objects are the LV cavity of a heart from sets of parallel long-axis CT-scan slices. Gap size: (a) and (b) 15 pixels, (c) and (d) 20 pixels. Reconstruction by, (a) and (c): the small object or hole preserving approach ($\gamma = 3$ then (a) $\gamma = 9$, and (c) $\gamma = 12$), and (b) and (d): the surface-detail preserving approach (γ ranging from 3 to (b) 12, and (d) 15).*

4.4.4 Choice of Segmentation Algorithm

The segmentation algorithm determines the computation of the speed S of the level set implicit surface in Algorithm 4.2. Any segmentation algorithm may be used with ISISD, since no constraint is set on the computation of S. This choice of segmentation algorithm is dictated by the type of data and the imaging modality. Sets of 2D slices may be segmented using a number of image segmentation algorithms, such as edge based and region based methods, *e.g.* [6, 61] and [7, 23], as well as methods that use prior knowledge, *e.g.* [15, 26, 67].

As mentioned earlier, our framework may also be used with other spatial configurations such as 3D point clouds, since the interpolation of the level set segmenting surface does not make any assumption on the location of the data points. For 3D point clouds with surface orientation information (*e.g.* surface normals or line-of-sight), a region-based algorithm may be used, similar to [58], which would assign the speed $S(\mathbf{x})$ in (4.7) depending on whether the point $\mathbf{x}$ is behind or in front of the surface of the modelled object. Note that [58] is limited to dense point clouds while the intrinsic interpolation capability of the proposed framework allows it to apply the same segmentation algorithm to sparse point clouds. When there is no information on whether a point is inside or outside of the modelled object, an edge-based method may be used, which would attract the segmenting surface towards the 3D data points. A well known edge-based method for image segmentation, which was used by [68] to segment 3D point clouds, is Geodesic Active Contours [6]. The proposed framework could also use other, more robust, edge-based methods, such as [61]. Possible segmentation algorithms, in the case of medical image slices and RGB-D data, will be proposed in Section 4.5.1 and Appendix A respectively.

One benefit of our segmenting level set surface interpolation approach, stated in Section 4.2.2, is that when a dataset contains images with different gains and contrasts, or coming from different modalities, the speed S of the level set surface can be computed independently for each image, using different segmentation algorithms, as in Fig. 4.4. Then, the values in S for all images can be jointly used to update the level set surface, by computing $\frac{d\alpha_i}{dt}$ in (4.7) in the case of ISISD. Other examples of the use of this approach will be provided later, for example in Section 4.5 where T1 and T2 weighted MRIs of a brain are processed simultaneously (see Fig. 4.24).

In our approach, the choice of segmentation algorithm is as crucial to the success of interpolation as for segmentation, since interpolation relies on the position of the level set implicit surface in the volume, through the term S in (4.7) for ISISD. Thus, the outcome of the interpolation stage and of the integrated framework relies heavily on the choice of a robust segmentation algorithm by the user. It may be noted that, in the case of sets of 2D images, the integration of interpolation and segmentation allows using 3D or 4D segmentations rather than the less robust independent 2D segmentations. Nevertheless, segmentation remains a difficult task, and is central to the success of the integrated framework.

We note however that the failure of the segmentation in a part of the volume *has only an impact on the interpolation in the neighbourhood* of this part, while in the rest of the volume where the segmentation is satisfactory, the framework recovers quickly. This is illustrated in Fig. 4.16 where the LV cavity of a heart has been segmented and interpolated from a set of slices after replacing a central horizontal slice by a wrong one to simulate a failure in its segmentation.

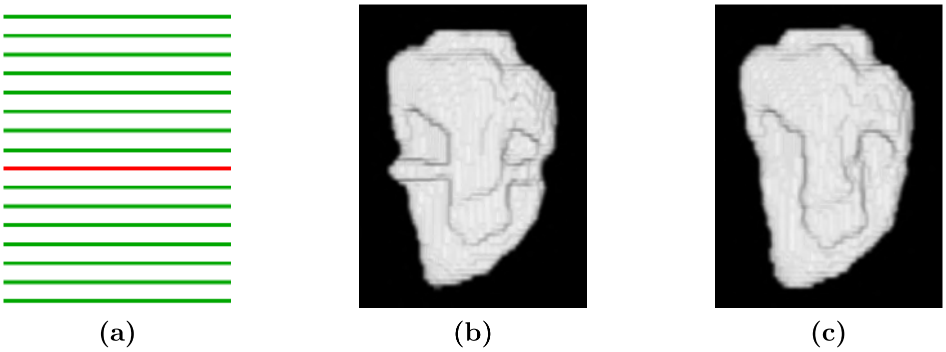


Figure 4.16: *Recovering after a local failure in the segmentation – the modelled object is the LV cavity of a heart of Figs. 4.3 and 4.14. (a) dataset viewed from the side – green: true slices; red: erroneous slice, (b) reconstructed object (same view), and (c) original object.*

4.4.5 Choice of Time Step

ISISD is very robust to numerical instabilities due to the use of an ODE rather than the traditional level set's PDE. In our experiments, no numerical instabilities arose, even for very large time steps, and without using any renormalisation. We can therefore choose the time step depending on the maximum number of voxels the implicit surface is allowed to travel through in one iteration. At a given position, a level set interface moves by distance D, defined as

$$D(\mathbf{x}) = \frac{dt \cdot \frac{\partial \phi}{\partial t}(\mathbf{x})}{\|\nabla \phi(\mathbf{x})\|}. \tag{4.18}$$

If D is too large, the implicit surface may pass through small objects without seeing them. In our experiments, we usually set $D = 1$ at the beginning of the process, which is then reduced automatically and progressively when oscillations of the implicit surface around the edges of the segmented object are detected. Lower D values allow the implicit surface to stabilise on object edges and to segment them more accurately.

4.5 Evaluation of Interpolation on Medical Tomographic Data

In this section, we demonstrate the application of ISISD to the modelling of objects from sparse sets of medical image slices which do not suffer from misalignment, and we evaluate its performance at interpolating in gaps between the slices. We also show the flexibility of ISISD by modelling objects having a large range of sizes and shapes, and from various modalities.

We first propose, in Section 4.5.1, two examples of segmentation algorithms, which are adapted to sets of medical slices while being simple and general enough to be applied to the various medical imaging modalities which we use in our experiments. Then, we present in Section 4.5.2 the evaluation protocol that will be used to assess the interpolation quality of ISISD.

These evaluations are performed in Sections 4.5.3 to 4.5.6, with a focus on the handling of arbitrary spatial configurations in Section 4.5.3, and on the accuracy and robustness of the interpolation in Sections 4.5.4 to 4.5.6, on artificial and real data respectively. Finally, some considerations on running time are discussed briefly in Section 4.5.8.

4.5.1 Segmentation Algorithm

We experimented with several segmentation algorithms, including the edge-based methods of [6, 61] and the region-based method of [7]. We selected two generic methods for our tests, based on their versatility, *i.e.* their ability to handle the variety of our data, and their simplicity and ease of use. Note that, although they yield acceptable segmentations in most of our tests, better segmentation results may be obtained by adapting the segmentation method to each type of dataset.

The first method is the piecewise constant (PC) model used in [7], which is intensity based and drives the segmentation according to:

$$S(\mathbf{x}) = (I(\mathbf{x}) - c_{out})^2 - (I(\mathbf{x}) - c_{in})^2 \ , \tag{4.19}$$

where $I(\mathbf{x})$ is the pixel intensity at point $\mathbf{x}$, and c_{in} and c_{out} are the mean intensities inside and outside of the object respectively. This method was selected as the most suitable segmentation approach for validation of our framework on artificial data made up of piecewise constant regions (see a description of our artificial data in Section 2.2.1).

For validation on real data, a method similar to [30] is used, which is based on a piecewise model and on a Parzen window method to estimate the distributions of image intensity inside and outside of the object:

$$S(\mathbf{x}) = \log P_{in}(I(\mathbf{x})) - \log P_{out}(I(\mathbf{x})) \ . \tag{4.20}$$

However, we did not use the contour smoothing and implicit function regularisation of [30] because our proposed method tends to produce smooth contours and does not require regularisation, as mentioned in Section 4.4.5. Also, we optimised the width of the Parzen window using log-likelihood maximisation, while [30] sets it equal to the standard deviations in the two regions. This algorithm was simple and generic enough to allow processing various types of images, *e.g.* both MRIs and CT-scans, with various amounts of noise.

4.5.2 Evaluation Protocol

This section is devoted to the evaluation of the interpolation quality of ISISD. We do not evaluate the quality of the segmentation, since, as explained earlier, this quality largely depends on the chosen segmentation algorithm, and this choice is highly dependent on the nature of the data. Since ISISD is general enough to be used on data from a variety of different modalities (*e.g.* MRIs and CT-scans as shown in our experiments), and can accept any level set segmentation algorithm for the computation of the speed S of the level set implicit surface in (4.7), as outlined in Section 4.4.4, we do not recommend any specific segmentation algorithm and let the user choose the most appropriate method depending on their data. Therefore, the choice of an optimum segmentation algorithm is beyond the scope of this work, and we only evaluate the quality of the interpolation.

In order to evaluate the quality of interpolation provided by ISISD, we compare with a state-of-the-art interpolation method on both artificial and real medical data, and on a variety of imaging modalities and imaged objects. Note that methods which integrate segmentation and interpolation in a model based framework are not general enough to allow processing shapes that have high variability, and we know of no other existing work that *integrates* the whole process of segmentation and interpolation for 2D slices. Therefore, we compare against the traditional *sequential* approach which performs interpolation and segmentation separately in turn.

As explained in Section 3.1, two sequential approaches exist: A) 2D segmentations of the image slices, followed by interpolation of the surface from the obtained binary masks, and B) interpolation of the images, followed by 3D segmentation of the obtained 3D volume. The type B sequential approach is more popular in automatic sequential processing of medical images, due to the more robust results produced by 3D segmentations over independent 2D segmentations of the slices. In addition, a fair comparison of the interpolation stage of the sequential and integrated approaches requires using the same segmentation stage in both methods. The ISISD framework can only use 3D and 4D segmentation by design, while sequential approach A is restricted to 2D independent segmentations. Due to these reasons, we compare against version B of the sequential approach, and we implement it with the state-of- the-art image interpolation method of Cordero-Grande et al. [14], which was described in Section 3.2.2, and the same level set segmentation method as in our ISISD framework. Hereafter, we refer to this sequential method as SM+[14] for "Sequential Method implemented with [14]").

For comparative evaluation against SM+[14], we measure accuracy on the 3D shape of the segmented and interpolated objects (again because the proposed method cannot separate segmentation from interpolation). For better parity, the segmentation algorithm of our framework is used to segment the interpolated volume produced by Cordero-Grande et al. [14]. We use the Jaccard similarity coefficient as an accuracy measure of the combined segmentation and interpolation. It measures the similarity between the recovered and the original volumes, and it is defined as

$$\boldsymbol{J} = \frac{M_{11}}{M_{11} + M_{01} + M_{10}} \, , \tag{4.21}$$

where M_{11} is the number of voxels that are inside both the evaluated and the ground-truth models, and M_{01} and M_{10} are the number of voxels that are inside one model and outside of the other. Note that other similarity measures may be used that would emphasise different aspects of the results. For example, while the Jaccard measure is well suited to compare the original and computed *volumes*, some other similarity measures such as the Hausdorff distance would be better suited to evaluate the smoothness of the reconstructed *surface* and its match with the original object's surface. In our experiments on artificial data of Section 4.5.4, the ground-truth is the model used to produce the artificial data. In our experiments of Section 4.5.5, in order to assess the accuracy of the interpolation on real data, we compare the reconstruction of the objects from the datasets with gaps against the reconstructions from the original full volume datasets. The same segmentation method is used in both cases to ensure that the segmentation results are similar and that the comparisons only evaluate the accuracy of the interpolation. In the experiments of Section 4.5.6, we perform subjective comparisons on the non-isotropic real datasets.

We evaluate especially the ability of the methods to interpolate from data having various spatial configurations, *i.e.* slices having arbitrary positions and orientations. This allows exploiting all available information without having to discard, for example, slices of unsupported orientations. Also, particular attention is paid to the quality of the interpolation in cases where a high amount of smoothing is required in the gaps. This is the case of gaps being comparatively too wide with respect to the size of the object and/or of slice orientations which do not match the principal orientation of the imaged object, as was illustrated in Fig. 4.10 and discussed in Section 4.4.2. Finally, we evaluate the capability of interpolating a variety of shapes, which can allow reconstructing either various objects or objects having a high degree of shape variability, *e.g.* some pathological organs such as brain ventricles of a patient suffering from hydrocephalus (see Fig. 4.24).

The values of γ used by ISISD are given in Table 4.1. For the SM+[14] approach, the image interpolation method of [14] was either ran and tuned by the authors of [14], or was ran using their implementation and the default parameters provided by the authors. Indeed, in this last case, it was found that modifying the method's parameters did not significantly change the results.

Table 4.1: *Values of parameter γ used in the interpolation experiments*

Experiment	Modelled object	Spatial configuration	γ
Artificial data	Geometric model	All	3
	LV models A and B	All	0.5 then 4
Real data (quantitative evaluation)	Phantom Low Res.	All	20
	Phantom High Res.	All	20
	Kidney	5 pixels spacing	3 to 7
		10 pixels spacing	3 to 9
		15 pixels spacing	3 then 9
		20 pixels spacing	3 then 12
	LV cavity	5 pixels spacing	3 to 7
		10 pixels spacing	3 to 9
		15 pixels spacing	3 to 12
		20 pixels spacing	3 then 12
	Acetabulum	All	9
Real data (qualitative evaluation)	Vertebrae	All	1.5
	Brain ventricles	All	1.5
	LV cavity	All	0.7 then 3

4.5.3 Robustness to Arbitrary Spatial Configurations

We first evaluate the ability of the methods to handle arbitrary spatial configurations, *i.e.* slices having diverse positions and orientations. The configurations tested in Figs. 4.17 and 4.18 are axial and radial datasets ((a) and (c) respectively), and (b) stacks of horizontal SA slices plus a few LA slices that have more complicated orientations and positions. This latter spatial configuration is typical of cardiac MRIs.

Both Cordero-Grande et al. [14] and ISISD could process the axial datasets. However, Cordero-Grande et al. [14] had to discard the LA slices of the SA+LA datasets, and could not process the radial datasets at all, because it is limited to stacks of parallel and equally spaced slices. On the contrary, ISISD could inherently handle all three spatial configurations, and exploited all available slices. This is due to the RBF based interpolation scheme which does not require any specific spatial arrangement of the data points. More examples of various spatial configurations will be presented in the next experiments.

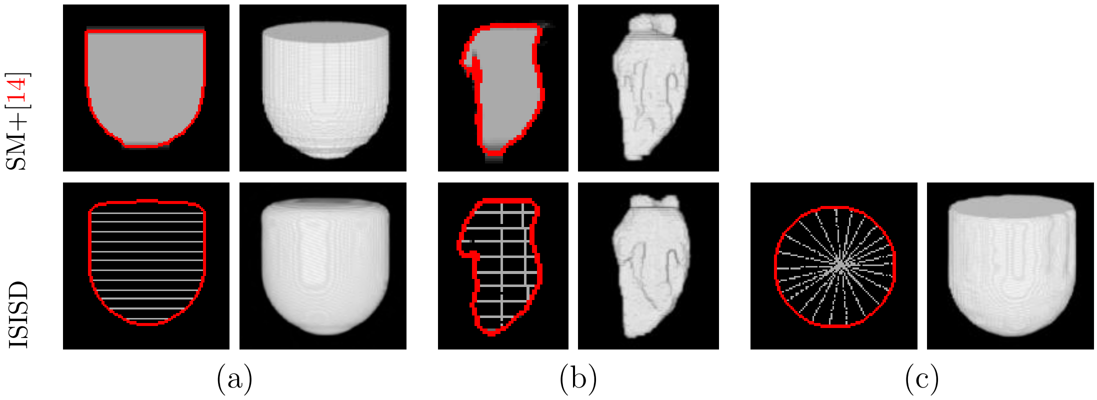


Figure 4.17: *Robustness to arbitrary spatial configurations of artificial slices. (a) Axial dataset – left: central vertical slice with segmenting contour in red, and right: 3D view of the reconstruction. (b) SA+LA dataset – left: central vertical slice, and right: 3D view. (c) radial dataset – left: central horizontal slice, and right: 3D view. The datasets contain the models of Figs. 2.4a and 2.4c presented in Section 2.2.1. The radial dataset could not be processed by [14] because this method requires parallel and equally spaced slices.*

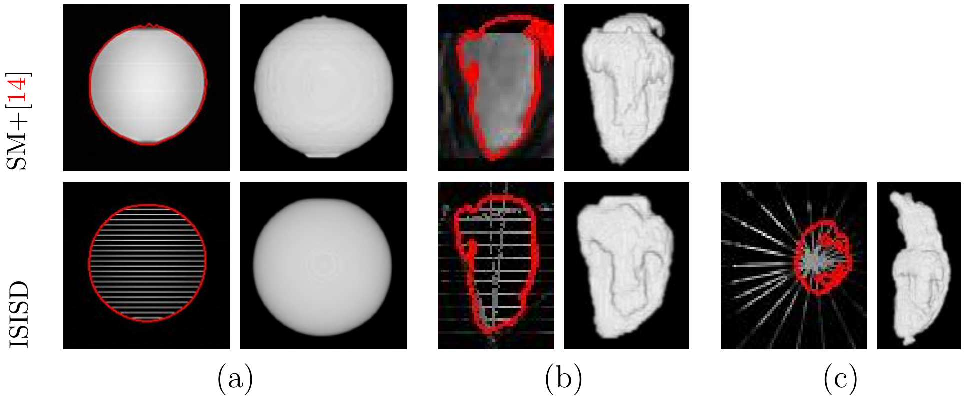


Figure 4.18: *Robustness to arbitrary spatial configurations of real slices. (a) Axial dataset – left: central vertical slice with segmenting contour in red, and right: 3D view of the reconstruction. (b) SA+LA dataset – left: central vertical slice, and right: 3D view. (c) radial dataset – left: central horizontal slice, and right: 3D view. The datasets are the MRI scan of a phantom and two cardiac MRIs, which were presented in Section 2.2.2. The radial datasets could not be processed by Cordero-Grande et al. [14] because this method requires parallel and equally spaced slices.*

4.5.4 Quantitative Evaluation of Accuracy on Artificial Data

The purpose of this experiment on artificial data is to assess the quality of interpolation separately from that of the segmentation, since segmentation should be perfect on the piecewise-constant images. It also establishes the limits of performance of the interpolation in the case of a perfect segmentation.

The geometrical object and LV models were modelled by SM+[14] and by ISISD from the artificial datasets, starting from a small spherical surface located at the centre of the volume, using the PC semgentation model described in Section 4.5.1. The geometrical object datasets were processed by ISISD using a single value of 3 for γ, and the LV model datasets were processed using the small object or hole preserving approach presented in Section 4.4.3, with γ being equal to 0.5 and 4 in the first and second steps respectively, for all LV datasets. The image interpolation method of Cordero-Grande et al. [14] was ran and tuned by the authors of [14]. They set parameter W_i of their method to 0.5, to reflect the very little amount of "randomness" in the piecewise constant images. Cordero-Grande et al. [14] could not process the LV datasets number 16 because their slice spacing is not constant through all the volume.

Fig. 4.17 presents the segmented and interpolated surfaces of the radial and axial datasets and of an example of a modified cardiac MRI, and all the results for the modified cardiac MRIs are shown in Figs. 4.19 and 4.20. Table 4.2 provides the Jaccard coefficients indicating the proportions of voxels correctly assigned to the objects.

Even when no attempt was made to tune γ to match the individual spatial configurations of the datasets, the ISISD framework yielded better results than SM+[14], with average accuracy measures of 0.943 against 0.924. It obtained a more global interpolation of the shape of the objects than Cordero-Grande et al. [14] on both the geometrical and LV datasets using the same slices, *i.e.* SA slices only, especially at the apex of the LV, with an average accuracy rate for the LV datasets with SA slices only of 0.935 against 0.923 for Cordero-Grande et al. [14]. The results were further improved by using all SA and LA slices, yielding an average accuracy rate of 0.941. It can be seen from Figs. 4.19 and 4.20 that Cordero-Grande et al. [14] sometimes produces "stair-case" like shapes at the apex of the two LV cavities, while ISISD reconstructs smoother and more global shapes.

For the proposed interpolation method, we identified two difficult types of surface areas whose reconstructions most contributed in reducing the Jaccard accuracy measure. The first type is the centres of gaps where the object's surface has a sharp angle with the image planes, as was illustrated in Fig. 4.10, and which requires a flat RBF in order to compensate for the increased distance between the data points and obtain a global and smooth interpolation. For example, this is the case for gaps near the apex of the LV models and at the bottom of the hemisphere of the axial dataset. Note that very large gaps would have the same effect, as will be illustrated in the next experiment. As

Table 4.2: *Jaccard Coefficients on Artificial 3D Data*

Dataset	SM+[14]	**ISISD**		**Dataset**	SM+[14]	**ISISD**	
Axial	0.959	0.967		Radial	-	0.984	
LV model A	SA	SA	SA+LA	LV model B	SA	SA	SA+LA
1	0.947	0.952	0.963	1	0.937	0.940	0.946
2	0.947	0.957	0.961	2	0.936	0.933	0.943
3	0.935	0.951	0.951	3	0.939	0.936	0.937
4	0.947	0.952	0.958	4	0.937	0.940	0.942
5	0.950	0.951	0.960	5	0.917	0.919	0.939
6	0.949	0.956	0.959	6	0.934	0.928	0.941
7	0.932	0.943	0.950	7	0.903	0.926	0.931
8	0.932	0.942	0.948	8	0.902	0.927	0.926
9	0.932	0.937	0.937	9	0.903	0.913	0.913
10	0.880	0.946	0.946	10	0.860	0.906	0.906
11	0.789	0.865	0.892	11	0.869	0.872	0.884
12	0.947	0.952	0.955	12	0.937	0.941	0.941
13	0.930	0.944	0.942	13	0.927	0.925	0.915
14	0.969	0.972	0.979	14	0.974	0.968	0.973
15	0.928	0.948	0.952	15	0.911	0.925	0.923
16	-	-	0.956	16	-	-	0.938
Average	**0.928 ±0.043**	**0.944 ±0.023**	**0.951 ±0.018**	**Average**	**0.919 ±0.029**	**0.927 ±0.021**	**0.931 ±0.020**
Average on all LV models	**0.923 ±0.036**	**0.935 ±0.024**	**0.941 ±0.021**	**Average on all datasets**	**0.924 ±0.036**	**0.943±0.022**	

discussed in Sections 4.4.2 and 4.4.3, a tradeoff must be found in this case, between a flat RBF which takes into account the global geometry of the object, and a sharper RBF which better preserves the surface's details. In this experiment, the small object or hole preserving approach achieved this tradeoff satisfactorily for the LV models, thus minimising the decrease of the Jaccard measure, and a high value for γ could be used with the axial dataset.

The second type is the disappearance of a structure between two slices where the proposed method has no other choice than to guess the shape of the missing extremity. It

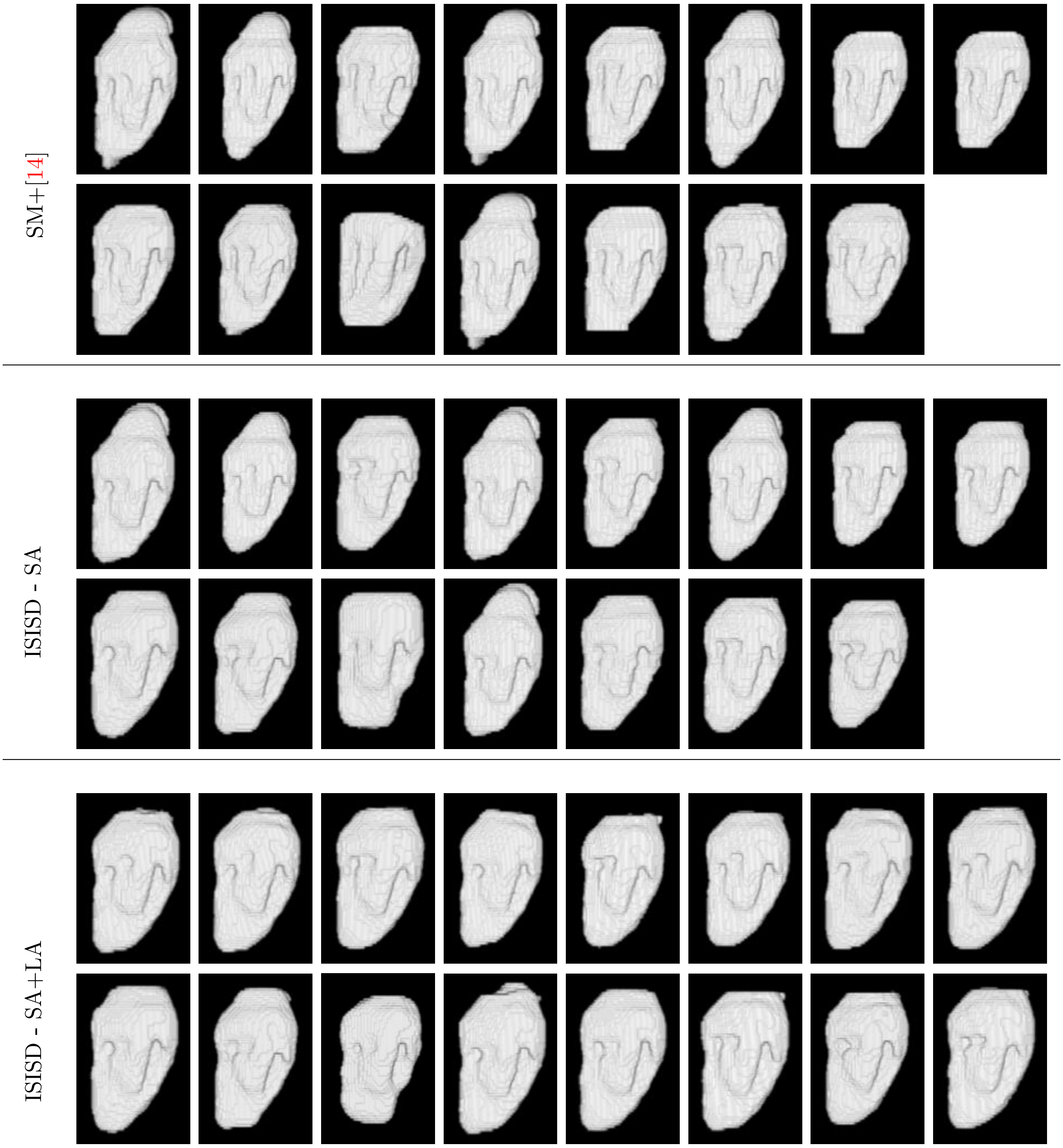


Figure 4.19: *Reconstruction of LV model A by SM+[14] and by ISISD, with and without LA slices. The last dataset could not be processed by [14] because its slices are not equally spaced.*

does so by attempting to preserve the smoothness of the global shape, and therefore produces plausible, yet possibly wrong, reconstructions. This is particularly true for large gaps where the amount of missing information is high. Overall, we found that the reduction of the Jaccard accuracy measures was due to the accumulation of several small deviations from the ground-truth surface, scattered over all the surface and with

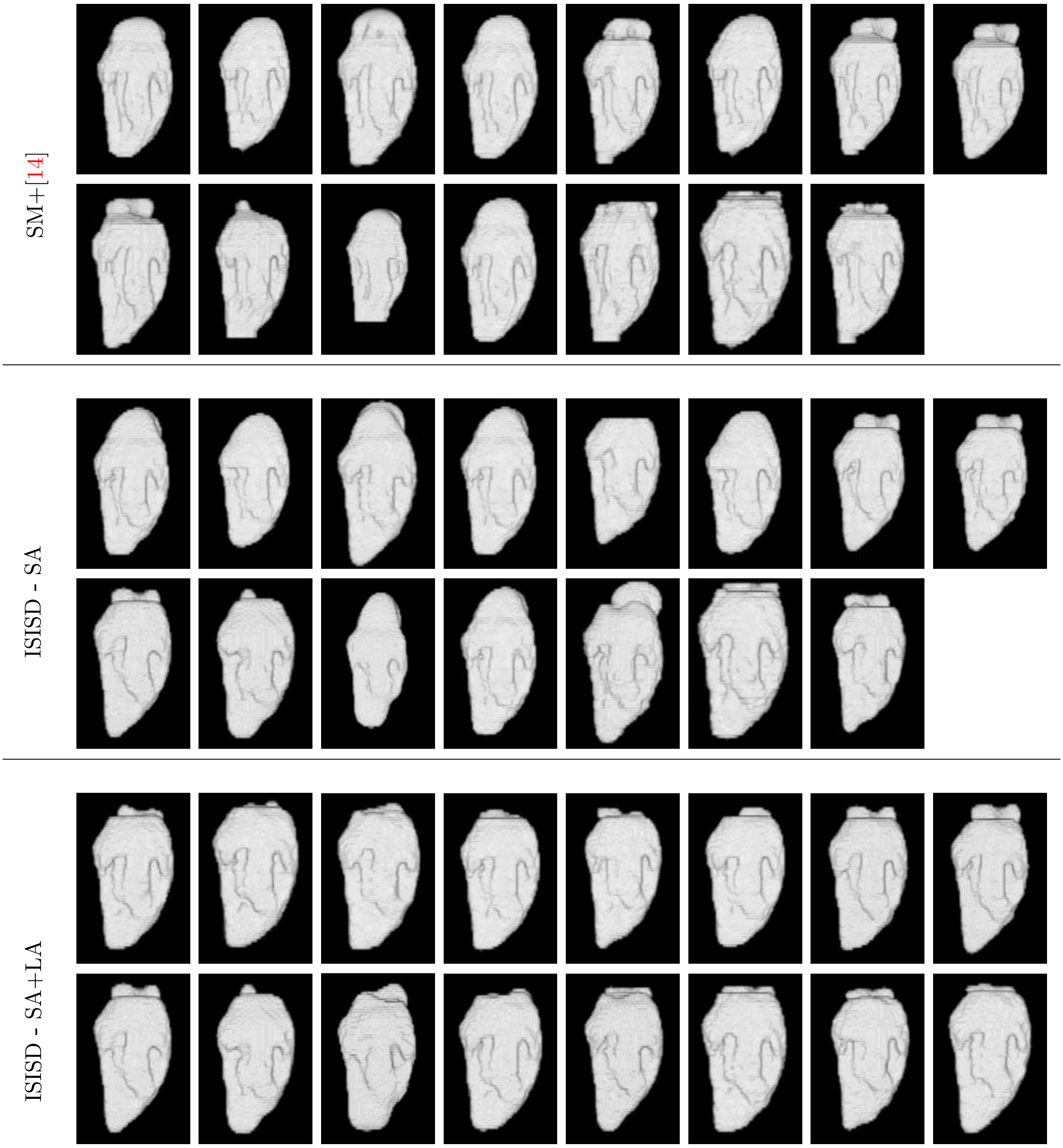


Figure 4.20: *Reconstruction of LV model B by SM+[14] and by ISISD, with and without LA slices. The last dataset could not be processed by [14] because its slices are not equally spaced.*

a slightly higher concentration in these two types of difficult areas. Note that the radial dataset does not suffer from any of these difficulties, and achieved the best Jaccard coefficient.

4.5.5 Quantitative Evaluation of Accuracy on Real Data

ISISD was also quantitatively evaluated on our real datasets made from the quasi-isotropic data from which a number of slices were removed, as described in Section 2.2.2. In order to demonstrate the versatility of the proposed method, we performed these tests on different objects having various shapes and sizes, and imaged using different modalities. These are the two phantom MRIs and the kidney, LV and acetabulum CT-scans. The segmentation algorithm based on pixel intensity classification using a Parzen window estimator, described in Section 4.5.1, was used for both compared methods.

The LV and Kidney datasets required a particularly large amount of smoothing in the gaps because of the combined effects of very large gaps and the unmatched orientations of (all) parallel slices against the orientation of the organs, hence causing those slices to have a small angle with the object surface. Therefore, they were reconstructed by ISISD using the surface-detail preserving approach for the smaller gap sizes (5-15 and 5-10 pixels for the LV and Kidney datasets respectively) and the small object or hole preserving approach for the larger gaps. γ was tuned depending on the gap size, ranging from 3 to 12, as shown in Table 4.1. The phantom and acetabulum datasets presented smoother surfaces, therefore a simpler scheme was used for them, with a single value of γ for all slice spacings. In these cases, γ was only roughly adapted to the size of the object and the level of detail of its shape, and was set to 9 for the acetabulum, and 20 for the two types of phantom. For the method of Cordero-Grande et al. [14], we used the three default parameters provided by the authors, as we found that modifying them did not significantly change the interpolation followed by segmentation results.

Table 4.3 provides the Jaccard coefficients for all of the datasets for SM+[14] and ISISD when the datasets were made up of parallel horizontal slices only (in the third and fourth columns of the table), and parallel horizontal slices plus one vertical slice (in the last column). Cordero-Grande et al. [14] could not be used in this last case, since it can not handle the vertical slice. Fig. 4.21 shows the modelled shapes for the 5 pixels and 20 pixels spacing datasets.

Even when γ was not finely tuned for all datasets, ISISD performed similarly or better than SM+[14]. The segmentations of the slices, in the planes of the images, were subjectively similarly good for the two compared methods for all slice spacings. However, the interpolation by ISISD produced better results in the gaps between the images, especially for larger gaps, with average accuracy measures of 0.937 and 0.946 across all

Table 4.3: *Jaccard Coefficients on Real 3D Data*

Dataset	**Slice spacing** (in pixels)	**SM+[14]**	**ISISD** Horz.	Horz.+Vert.
Phantom Low res.	5	0.970	0.989	0.990
	10	0.940	0.991	0.989
	15	0.923	0.978	0.987
	20	0.863	0.962	0.977
Phantom High res.	5	0.993	0.992	0.990
	10	0.980	0.991	0.993
	15	0.949	0.990	0.992
	20	0.917	0.985	0.990
Kidney	5	0.955	0.953	0.955
	10	0.895	0.887	0.900
	15	0.840	0.845	0.869
	20	0.771	0.802	0.846
LV	5	0.966	0.962	0.962
	10	0.919	0.916	0.922
	15	0.872	0.888	0.899
	20	0.750	0.835	0.864
Acetabulum	5	0.972	0.981	0.975
	10	0.948	0.968	0.968
	15	0.892	0.929	0.935
	20	0.881	0.903	0.915
Average of all datasets	**5**	**0.971±0.014**	**0.975±0.017**	**0.974±0.016**
	10	**0.936±0.032**	**0.951±0.047**	**0.954±0.041**
	15	**0.895±0.043**	**0.926±0.061**	**0.936±0.054**
	20	**0.836±0.072**	**0.897±0.079**	**0.918±0.065**
	all spacings	**0.910±0.066**	**0.937±0.059**	**0.946±0.049**

datasets and all spacings, for parallel horizontal slices and horizontal plus vertical slices respectively. SM+[14] only managed a rate of 0.910 for parallel horizontal slices and was inherently unable to use vertical slices due to the limitation of [14].

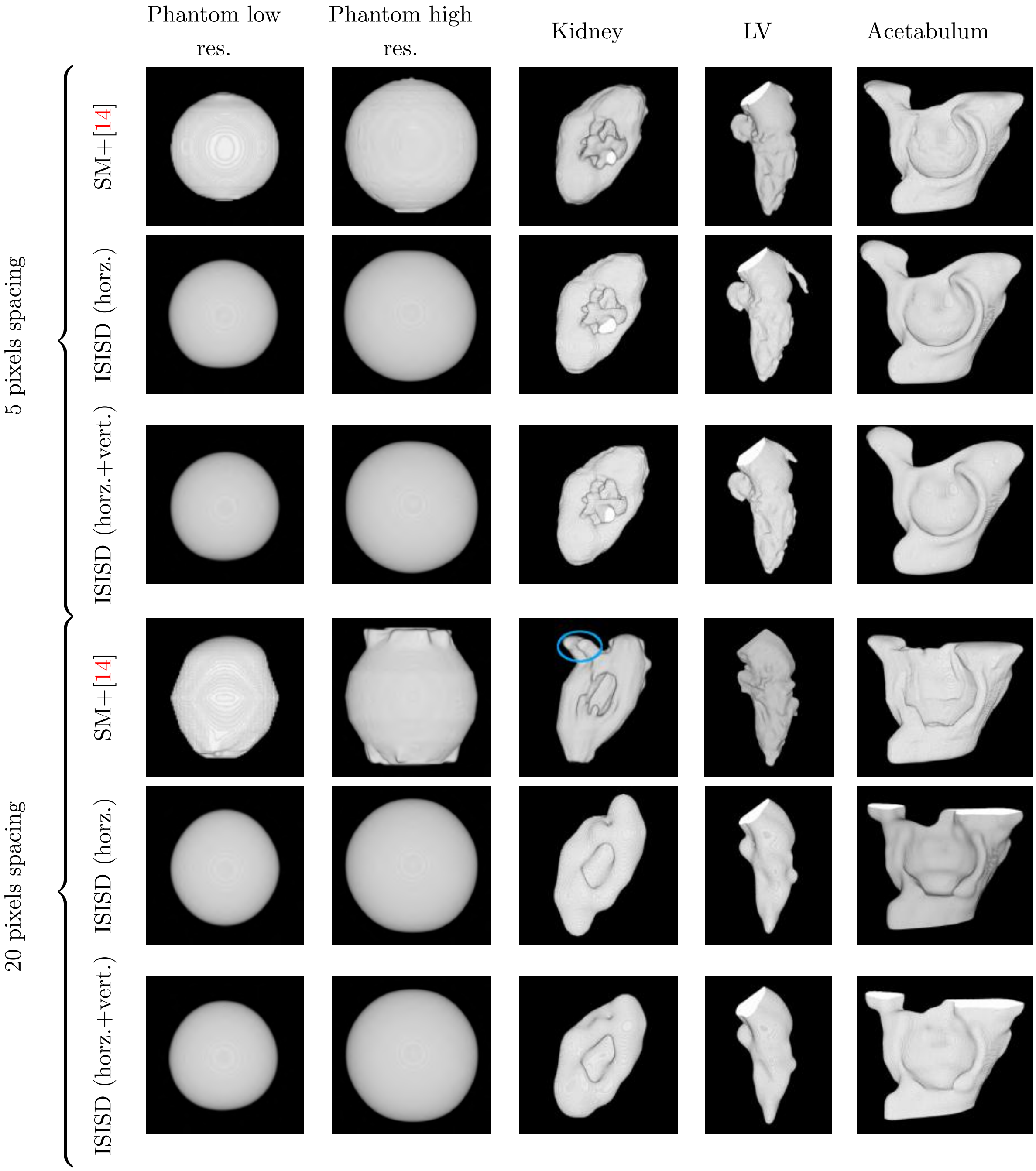


Figure 4.21: *Modelling of various shapes from real 3D data at two different slice spacings (5 and 20 pixels respectively), and from a variety of modalities. The blue circle highlights how Cordero-Grande et al. [14] struggles to reconstruct extremities of objects for large slice spacings, especially where a background of similar intensity can be confused with the object.*

Note that the addition of a single vertical slice increased the robustness of ISISD, especially in large gaps (right column of Table 4.3 and 3rd and 6th rows of Fig. 4.21), with average accuracy rates across all five datasets of 0.897 to 0.918 for 20 pixel spacings, even if the position and orientation of the additional slice were not chosen to match the position and orientation of the imaged object but instead were selected arbitrarily at the centre of the volume. Recall, Cordero-Grande et al. [14]'s approach is inherently unable to use LA images.

Cordero-Grande et al. [14] interpolates linearly between two neighbouring slices, thus producing a local interpolation of the shape, while our proposed method can take into account more than two slices, especially for large values of γ, and therefore it provides a more global interpolation. This is illustrated in the 4th row of Fig. 4.21 where the results of Cordero-Grande et al. [14] for 20 pixel spacing are made of straight segments because of the linear and local properties of the interpolation, and yield accuracy rates ranging from 0.750 to 0.917. On the contrary, in the 5th and last rows of Fig. 4.21, ISISD gave a smooth and rather round shape, and accuracy rates of 0.802 to 0.985 and 0.846 to 0.990 respectively.

Cordero-Grande et al. [14] also struggled where the object disappears between two slices, such as at the extremities of the phantom scans. This resulted in protrusions at these positions. For example, their method confused the object with a background of similar intensity in the 20 pixel spacing Kidney dataset (see blue circle in the 4th row of Fig. 4.21), which gave an accuracy rate of only 0.771.

Fig. 4.22 illustrates another example of shape detail extraction where, for the Kidney dataset with 5 pixels spacing, we can see that the topology of the vessels recovered by ISISD is much closer to the original volume than Cordero-Grande et al. [14]'s result, which fused some of the vessels together.

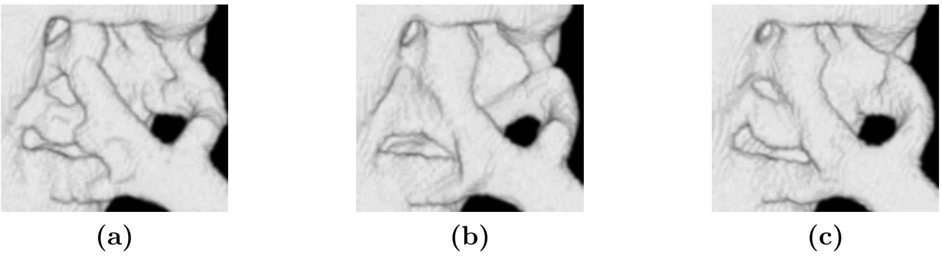


Figure 4.22: *Recovery of the topology of vessels from the Kidney dataset with 5 pixels spacing. (a) original dataset processed by (b) SM+[14] and (c) ISISD.*

4.5.6 Qualitative Analysis on Real CT-scans and MRIs

In this experiment, vertebrae were segmented from CT-scans, and the ventricles of the brain and the cavity of the LV were segmented from neonatal brain and cine MRIs respectively.

Two vertebrae (Fig. 4.23) and five brain ventricles (Fig. 4.24) were segmented using the Parzen estimator based segmentation algorithm. The datasets were visually inspected for any visible misalignment as a pre-processing step and also considered not to require much smoothing (unlike the data used in the previous experiment), because the gaps were smaller and the slices were acquired roughly perpendicular to the principal axes of the modelled objects. Therefore, the simplest scheme with a single value of $\gamma = 1.5$ was used for all the datasets of this experiment to provide a satisfactory trade-off between a suitable level of robustness to noise and a sufficiently global interpolation.

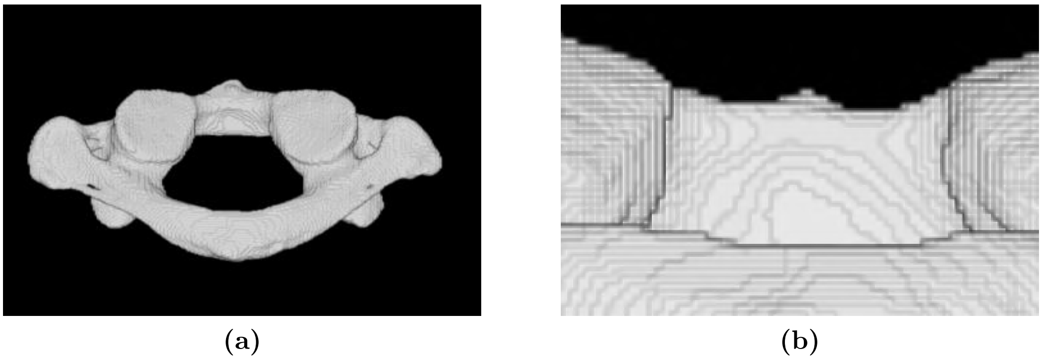


Figure 4.23: *3D segmentation and interpolation of the c1 vertebrae of Figs. 4.4c,d from a CT-scan. (a) global view, (b) zoomed view.*

We did not compare against SM+[14] in this experiment, because their method requires images spaced by at least twice their pixel size, which was not the case for the vertebrae CT-scans, and it can not handle images from different modalities simultaneously, such as the T1 and T2 images of the brain MRIs. On the contrary, the proposed method can jointly process the T1 and T2 images, by applying different segmentation algorithms to the computation of S for each modality, as suggested in Section 4.4.4.

Note that some methods perform multi-modal segmentation on brain MRIs with small gaps by “stacking” the image slices on top of each other, ignoring the gaps. The drawback of such methods is that they would discard any slice with a different orientation, such as sagittal and coronal slices, if available. By preserving the original spacing between the axial slices, our proposed framework can make use of these different slices. In Fig. 4.24,

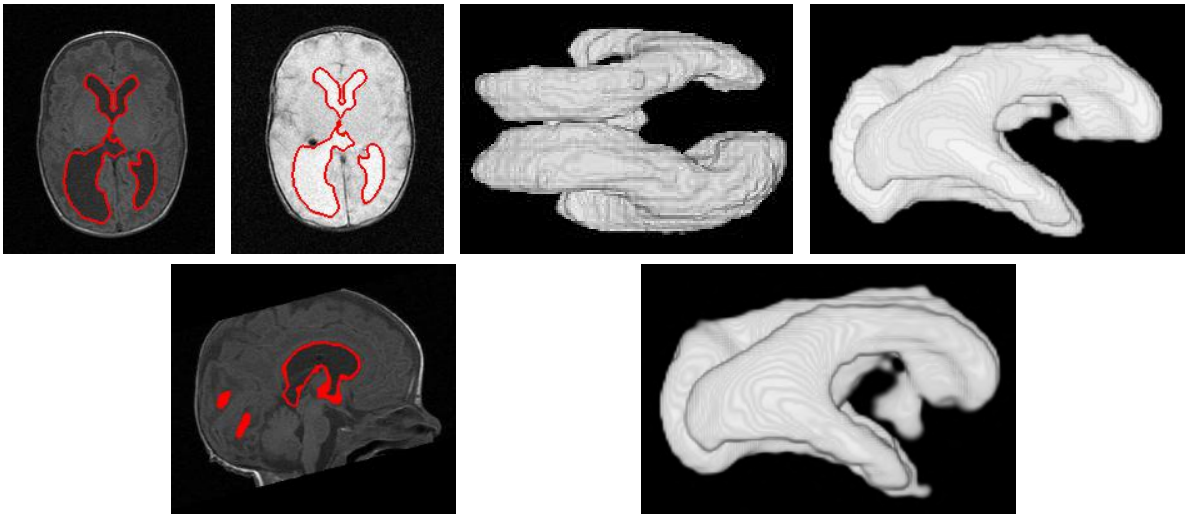


Figure 4.24: *3D segmentation and interpolation of the ventricles of the neonatal brain of the two bottom rows of Fig. 4.4 from combined T1 and T2 MRI. Top row: from left to right: T1 central slice, T2 central slice, axial 3D view, sagittal 3D view. Bottom row: with additional sagittal set of slices – left: central sagittal slice, right: sagittal 3D view.*

this allowed the use of an additional set of sagittal slices, which helped to segment a feature which was missed when using the axial slices only.

Contrary to Figs. 4.4c,d where the SDF method did not interpolate the bone's shape in the gaps satisfactorily, because of its local approach, ISISD produced a much more satisfactory global interpolation (Fig. 4.23). Similarly, in the two bottom rows of Fig. 4.4, the SDF method failed to extend the contour to all the parts of the brain ventricles due to the presence of large gaps in the data. In addition, the high degree of noise in the brain MRI datasets biased its segmentation. ISISD did not suffer from these issues and the segmenting contour reached all the parts of the brain's ventricles (Fig. 4.24). The segmentation was not impaired by the high level of noise, due to the inherent robustness of ISISD to noise in S, as described earlier in Section 4.3.2.

We modelled the cavity of the LV (Fig. 4.25) from cardiac MRIs, after visual inspection for no visible misalignment. The clear intensity separation between blood-pool and muscle allowed using the simple PC segmentation model. The datasets required about the same degree of smoothing as those of the vertebrae and neonatal brain, but the simplest scheme with a single value for γ failed to delineate the papillary muscles. Therefore, we used the small object or hole preserving approach presented in Section 4.4.3, with $\gamma = 0.7$ and $\gamma = 3$ for the first segmentation of small details stage and the second interpolation refinement stage respectively, for all the datasets. The RBF of the latter stage was sharp enough to preserve the details on the surface of the modelled objects. The image interpolation method of Cordero-Grande et al. [14] was ran and tuned by the authors of [14]. The cardiac MRIs are made up of 25 time-frames, but only one time-frame was used, as method [14] was designed for 3D segmentation and interpolation only. In addition, Cordero-Grande et al. [14] used only SA slices from these sets and discarded all LA slices.

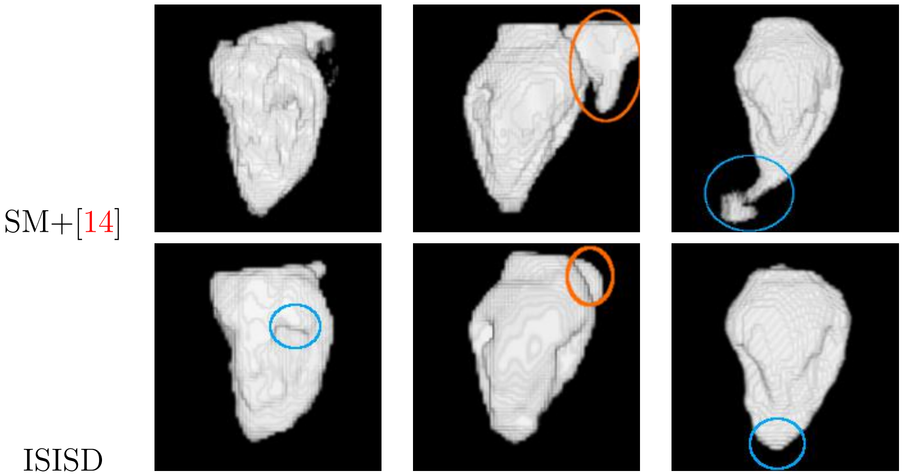


Figure 4.25: *3D segmentation and interpolation of the LV cavity from three different cardiac MRIs. The extraneous regions protruding from some LV cavities for all methods in some images,* e.g. *orange circles, are parts of the right ventricle which have been wrongly segmented by the chosen segmentation algorithm. The blue circles highlight situations where ISISD recovered a better topology than SM+[14].*

When subjectively compared, we observe that the segmentations of the 2D images of the cardiac datasets were similarly accurate for both methods, but the interpolation was better using ISISD, as illustrated by the blue highlighting circle in Fig. 4.25. Notably, the apex of the LV is better segmented and interpolated by ISISD than by SM+[14] thanks to the use of the LA slices.

4.5.7 Modelling of 4D Real Data

An example of a 4D segmentation and interpolation performed by ISISD is shown in Fig. 4.26. As in Section 4.5.6, we used the PC segmentation model to segment the LV cavity from cardiac cine MRIs. This cannot be compared with SM+[14] since [14] does not handle 4D data. Cordero-Grande et al. [14] can perform interpolation of the volume in each time-frame, but not across the time domain in an interframe fashion.

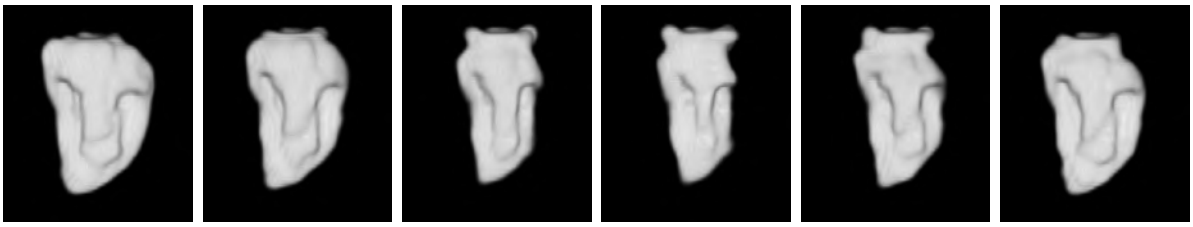


Figure 4.26: *4D segmentation and interpolation of the LV cavity from a cardiac cine MRI: 6 different time-frames.*

When processing 4D data, all time-frames are used simultaneously to update a 4D level set surface. The time dimension is treated in the exact same way as the other spatial dimensions, and in (4.7) the contour speeds S, weighted by 4D RBFs, are summed across both the spatial and the time domains. Therefore, neighbouring time-frames influence each other with weights (the RBF values) that decrease with distance in the time domain. This ensures temporal consistency in the resulting dynamical shape. Note that, in Section 4.3, FFTs were used to compute the convolutions of (4.7). As an additional bonus, FFT's inherent interpretation of periodic data allows us to exploit the cyclical nature of cardiac motion, since the first and last time-frames of the cycle are now neighbours as well[2].

4.5.8 Timing

The SM+[14] and ISISD frameworks were implemented in C++ and the experiments were run on a 1.6Ghz Linux machine. It is to be noted that timing comparisons against SM+[14] mainly highlight differences in the interpolation times, since our implementation of the level set framework was used for the segmentation stage of SM+[14]. Therefore, as will be detailed next, the main differences arise from the additional interpolation step of SM+[14], and varying numbers of iterations.

In the cases where the simple scheme with a single value for γ could be used, ISISD processed the data in about 5 to 15 minutes, depending on the size of the dataset. SM+[14] was moderately slower by approximately 5 to 20 minutes, since the running time of [14] had to be added to the segmentation time, and the segmentation itself (using

[2] Note that, as explained in Section 4.3.1, joint segmentation, interpolation, and tracking through time from non-periodic signals may be done by padding the data with a number of blank frames in order to limit the mutual influence of time-frames at the extremities of the signal. The extent of this influence decreases with the chosen flatness of the RBF.

the same segmentation framework as for ISISD) was slightly longer due to the higher number of data points.

Conversely, when two values for γ were used, due to the dataset requiring more interpolation, the proposed method needed more iterations and time (30 minutes to 2 hours) to process the data. Indeed, the second, interpolation refining step, required 4 to 10 times more iterations than the segmentation step in order to fully propagate the global interpolation in the gaps and replace the existing local one. Note that these computation times are still reasonable for the intended applications, which do not require real time processing. SM+[14] performed in the exact same way as in the first case, with the same number of data points and iterations required, and was therefore faster than the proposed approach for this configuration.

Table 4.4 summarises these timing comparisons.

Table 4.4: *Timing measures for ISISD and comparison against SM+[14]*

Processing scheme for ISISD	**SM+[14]**	**ISISD**
Single γ value	5 to 20 minutes	5 to 10 minutes
Small object or hole preserving approach with two γ values	5 to 20 minutes	30 minutes to 2 hours

4.6 Conclusion

In this chapter, an integrated segmentation and interpolation framework was introduced to reconstruct 3D and 4D objects from sparse datasets, based on the new concept of interpolating a segmenting level set contour, and on a novel RBF interpolated level set methodology. This new framework is general and can handle arbitrary spatial configurations, *i.e.* sets of any number of 2D slices having arbitrary positions and orientations.

Its use in modelling from sparse but aligned medical tomographic data was demonstrated, using two generic examples of segmentation algorithms. We evaluated the interpolation abilities of our ISISD framework and compared them against a state-of-the-art sequential method implemented with the image interpolation of Cordero-Grande et al. [14]. These tests were performed on a variety of data, both artificial and real, from different modalities, as well as on various shapes. Both 3D and periodic 4D shapes could be recovered, due to ISISD inherently handling periodicity.

ISISD proved to be very robust to noise and to large gaps, due to the interdependency of the segmentation and interpolation processes. In particular, the interpolation provided by ISISD was more robust to large gaps and to the object disappearing from consecutive slices than the image interpolation of Cordero-Grande et al. [14], and yielded smoother and more global shape interpolation.

Overall, ISISD was very flexible and could handle a great variety of shapes, spatial configurations, imaging modalities, and segmentation algorithms. Our proposed interpolation of the segmenting surface, rather than image intensities as in other works, allowed simultaneous handling of arbitrary spatial configurations and different gains and contrasts, such as images from different modalities, provided that they are aligned beforehand.

Note that correct registration of the data is of the utmost importance to provide coherent information to ISISD. Such registration would be needed to process sets of misaligned slices. The next chapter will address this registration issue.

Chapter 5

IReSISD: Integrated Registration, Segmentation, and Interpolation for Sparse Data

IReSISD improves upon ISISD by incorporating rigid registration into the framework described in the last chapter, in order to deal with any misalignments in the datasets. This integration allows it to model objects from several sequences[1] simultaneously by correcting any shifts or rotations caused by movements of the imaged object.

The inter-dependency of the three stages of registration, segmentation, and interpolation, highlighted in Chapter 1, is taken into account by integrating all three stages in the same framework. Thus, as in ISISD, interpolation exploits segmentation results while segmentation benefits from the extra support provided by interpolation. In addition, in this fully unified framework, registration can use segmentation and interpolation results for better robustness, while providing the aligned data that segmentation and interpolation require. This full integration of the three processes of registration, segmentation, and interpolation allows handling sparse and misaligned data with an increased robustness and accuracy for all three processes, as will be demonstrated in our experiments. To the best of our knowledge, this is the *first time that these three stages have been integrated in a level set framework*.

[1] Note that these sequences may equally be spatial or temporal sequences.

In IReSISD we propose a novel registration algorithm to address the problem of arbitrary spatial configuration of the data. Indeed, most image registration methods presented in Chapter 3 apply to overlapping images or full volumes only. The registration method of IReSISD can handle any spatial configurations, and especially sets of slices having arbitrary number, positions, and orientations, as will be illustrated by our experiments in Section 5.5. In Appendix A, we will demonstrate that it can also handle multiple 3D point clouds.

The registration method of IReSISD also addresses the issue of data sparsity and limited intersection between the sequences of a dataset, which was mentioned in Section 1 and Chapter 3. Indeed, traditional registration techniques estimate the best match between different sequences using raw data directly (see Section 3.1). However, in the case of sparse data where the sequences have a limited number of intersections, computation of these matching measures from raw data is more challenging and less reliable. By integrating registration into the ISISD framework, IReSISD allows the registration stage to exploit the shape information yielded by ISISD. Therefore, in cases with few intersections, the registration stage produces a better robustness than traditional registration methods that are based on similarity measures.

5.1 Overview of IReSISD

The registration method of IReSISD is inspired by the work of Yezzi et al. [62] for level set based registration and segmentation of two overlapping images or full volumes, presented in Section 3.3.3. A level set contour and registration functions are updated simultaneously. The contour is driven by image data towards the boundaries of the object, while, at the same time, the images are driven towards the contour by the registration. Thus, the sequences are progressively registered on a common frame which is the model of the object, while the model is refined as the sequences reach positions where they provide more coherent information to the segmenting level set contour. This is iterated until convergence of the segmenting contour on the object's boundary and of the registration functions (see overview schematic in Fig. 5.1).

IReSISD benefits from the advantages of ISISD, so it can process datasets having arbitrary spatial configurations and from any modality through the choice of an appropriate segmentation algorithm. Therefore, the registration method of IReSISD is more general than [62] and 1) can use any segmentation method, 2) can register any number of im-

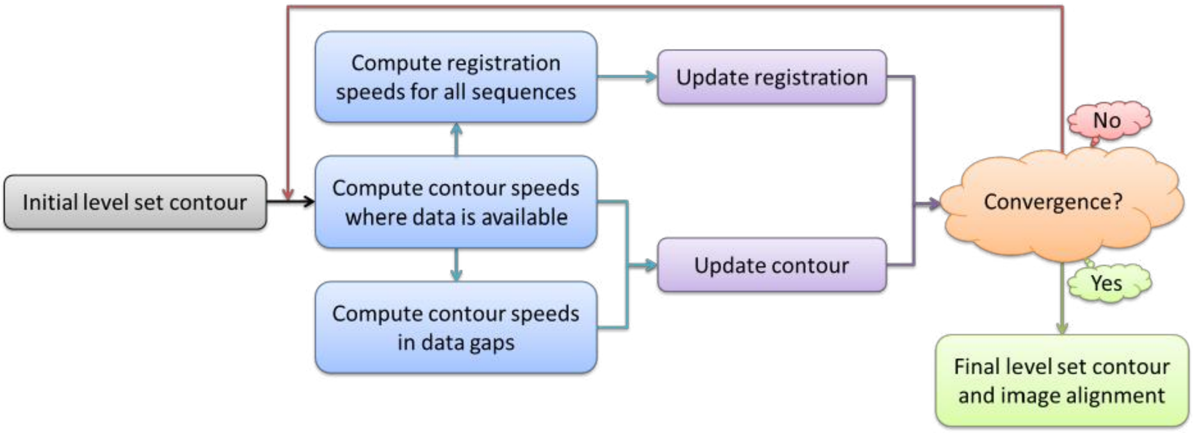

Initial level set contour
Compute registration speeds for all sequences
Compute contour speeds where data is available
Compute contour speeds in data gaps
Update registration
Update contour
Convergence?
No
Yes
Final level set contour and image alignment

Figure 5.1: *Overview of IReSISD.*

ages or volumes, 3) can register sparse data which do not span the whole volume, 4) can register intersecting, non-overlapping images, and 5) can handle arbitrary spatial configurations. In addition, the registration method of IReSISD is more robust to local minima than the registration of [62] due to the use of global information on the geometry of both the level set contour and the object in the images.

Next, in Section 5.2, we justify the validity of our adaptation of the method of Yezzi et al. [62] to multiple sparse and non-overlapping sequences having arbitrary spatial configurations. In Section 5.3, we present the formulation of our registration method and its integration into ISISD. Implementation details are provided in Section 5.4. In Section 5.5, we evaluate the quality of the registration and of the complete reconstruction process of IReSISD and we compare them against the results of a state-of-the-art method, *i.e.* a semi-sequential framework implemented with the registration of Lotjonen et al. [33] followed by ISISD.

5.2 Justification

The registration method of Yezzi et al. [62], presented in Section 3.3.3, applies to two dense and overlapping datasets only. However, we will show in this section that this approach has physical grounds which we will use in Section 5.3 to derive a new and more general registration method suited to multiple sparse and non-overlapping data.

Recall that [62] updates a segmenting contour according to (3.40a), while (3.40b) drives the evolution of a registration function g which transforms the coordinate system of a target image into the coordinate system of a registered image, such that $\hat{C} = g(C)$, with $\hat{C}$ and C being the contour in the registered and target images respectively.

Equation 3.40i makes the segmenting contour evolve under the sum of the contributions of the two images being processed. Its generalisation to an arbitrary number of sparse data having any spatial configuration was reached in Chapter 4, where, in (4.7), the speed S of the level set contour was computed using all the images which intersect it simultaneously. This resulted in the contour being evolved under the sum of the contributions of all its intersecting images.

Let us analyse the effect of (3.40b) in the case of internal registration of misaligned medical data. The misalignments between medical volumes, described in Chapter 2, are due to movements of the patient between the acquisition of the different sequences, or due to different volumes of air in the lungs. They can therefore be approximated by rigid motions made of translations and rotations. Thus, we examine in more detail the action of (3.42), the restriction of (3.40) to rigid alignment, in the case of translations and rotations.

Translation - When aligning by translations only, (3.42b) becomes:

$$\frac{\partial T_i}{\partial t} = \int_C \left\langle \mathbf{u}_i, \left(\hat{f} \circ g\right) \mathbf{N} \right\rangle , \tag{5.1}$$

where $\langle\rangle$ is the scalar product and $\circ$ is the function composition operator. T_i is the i^{th} component of the alignment vector $\mathbf{T}$, and $\mathbf{u}_i$ is the unit vector oriented in the i^{th} direction. $\left(\hat{f} \circ g\right)(\mathbf{x})\,\mathbf{N}(\mathbf{x})$ is the displacement imposed on the contour at point $\mathbf{x}$ by the image being registered.

A physical interpretation of (5.1) is that T_i evolves by the total displacement in the i^{th} direction that the segmentation of the image being registered imposes on the contour.

Rotation - Let us consider the alignment by rotation around one axis only, for example the z-axis, with the rotation matrix defined in the 3D case as

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} . \tag{5.2}$$

Then, using trigonometric identities, (3.42b) becomes:

$$\begin{aligned} \frac{\partial \theta}{\partial t} &= \int_C \left\langle R'_\theta \mathbf{x}, \left(\hat{f} \circ g\right) R_\theta \mathbf{N} \right\rangle \\ &= \int_C \left\langle R_{\frac{\pi}{2}+\theta} \mathbf{x}, \left(\hat{f} \circ g\right) R_\theta \mathbf{N} \right\rangle \\ &= \int_C \left\langle \mathbf{x}, \left(\hat{f} \circ g\right) R_{-\frac{\pi}{2}} \mathbf{N} \right\rangle . \end{aligned} \tag{5.3}$$

R'_θ refers to the derivative of R_θ with respect to θ.

This registration may also be seen as the effect of a physical force applied to the image. Indeed, $\left\langle \mathbf{x}, \left(\hat{f} \circ g\right)(\mathbf{x})\, R_{-\frac{\pi}{2}} \mathbf{N}(\mathbf{x}) \right\rangle$ may be seen as the torque of a force applied to the contour at point $\mathbf{x}$, directed tangentially, and with an intensity of $\left(\hat{f} \circ g\right)(\mathbf{x})$. The sum of these torques along the contour C makes it rotate towards a better alignment with the image to be registered.

Thus, the action of these two types of alignment (translation and rotation) can be understood as having physical grounds and it would make sense to apply them to more general cases than [62], such as different segmentation algorithms, and non-overlapping images and sparse volumes.

5.3 Formulations

Adopting this idea, we propose to align by translation and rotation any number of non-overlapping and intersecting sequences relative to each other, while segmenting an object in the 3D or 4D volume that contains them, and interpolating its shape where no data is available. We first generalise the formulation of [62], in Sections 5.3.1 and 5.3.2, to derive a new and more general registration method suited for multiple sparse and non-overlapping data. Note that our formulation applies to sequences of any type, both spatial and temporal. Then, in Section 5.3.3, we further improve our registration method and increase its robustness to local minima. This new registration method for sparse data suffering from local minima and cluttered background is a new contribution of this work. Another contribution is its integration, together with the segmentation and interpolation stages, into the same level set framework.

5.3.1 Integrated Registration, Segmentation, and Interpolation for Sparse Data

For each sequence n, we define a mapping function $g^n(\mathbf{x}) = R^n\mathbf{x} + \mathbf{T}^n$ (where $R = R_\gamma R_\beta R_\alpha$ is the 3D rotation matrix), which registers it with a common global reference. As in Chapter 4, the evolution of the level set function ϕ is driven by all the images that intersect the contour, through the use of their respective contour speeds S^n in the computation of the total contour speed S. The evolution of the mapping functions g^n is given by our generalisation of Yezzi et al. [62]'s evolution equations for the case of registration by translation and rotation:

$$\left\{\begin{aligned}
S &= \sum_n S^n \circ g^n && \text{(5.4a)}\\
\frac{\partial T_i^n}{\partial t} &= \frac{1}{P_n}\int_{\Omega_n\cap C} \left(S^n \circ g^n\right)\langle \mathbf{u}_i, R^n\mathbf{N}\rangle \\
&= \frac{1}{P_n}\int_{\Omega_n} \delta_\epsilon(\phi)\left(S^n \circ g^n\right)\langle \mathbf{u}_i, R^n\mathbf{N}\rangle && \text{(5.4b)}\\
\frac{\partial \alpha^n}{\partial t} &= \frac{1}{P_n}\int_{\Omega_n\cap C} \left(S^n \circ g^n\right)\left\langle R_{\gamma^n} R_{\beta^n} R_{\frac{\pi}{2}+\alpha^n}\mathbf{x} + \mathbf{T}^n, R^n\mathbf{N}\right\rangle \\
&= \frac{1}{P_n}\int_{\Omega_n} \delta_\epsilon(\phi)\left(S^n \circ g^n\right)\left\langle R_{\gamma^n} R_{\beta^n} R_{\frac{\pi}{2}+\alpha^n}\mathbf{x} + \mathbf{T}^n, R^n\mathbf{N}\right\rangle && \text{(5.4c)}\\
\frac{\partial \beta^n}{\partial t} &= \frac{1}{P_n}\int_{\Omega_n\cap C} \left(S^n \circ g^n\right)\left\langle R_{\gamma^n} R_{\frac{\pi}{2}+\beta^n} R_{\alpha^n}\mathbf{x} + \mathbf{T}^n, R^n\mathbf{N}\right\rangle \\
&= \frac{1}{P_n}\int_{\Omega_n} \delta_\epsilon(\phi)\left(S^n \circ g^n\right)\left\langle R_{\gamma^n} R_{\frac{\pi}{2}+\beta^n} R_{\alpha^n}\mathbf{x} + \mathbf{T}^n, R^n\mathbf{N}\right\rangle && \text{(5.4d)}\\
\frac{\partial \gamma^n}{\partial t} &= \frac{1}{P_n}\int_{\Omega_n\cap C} \left(S^n \circ g^n\right)\left\langle R_{\frac{\pi}{2}+\gamma^n} R_{\beta^n} R_{\alpha^n}\mathbf{x} + \mathbf{T}^n, R^n\mathbf{N}\right\rangle \\
&= \frac{1}{P_n}\int_{\Omega_n} \delta_\epsilon(\phi)\left(S^n \circ g^n\right)\left\langle R_{\frac{\pi}{2}+\gamma^n} R_{\beta^n} R_{\alpha^n}\mathbf{x} + \mathbf{T}^n, R^n\mathbf{N}\right\rangle \ , && \text{(5.4e)}
\end{aligned}\right.$$

where Ω_n is the domain of sequence n, P_n is the number of points used in the sums for sequence n and δ_ϵ is the approximation of the Dirac function, which was defined in (4.6).

S^n is the speed imposed on the level set contour by the data of sequence n and it depends on the chosen segmentation method. S is the total speed of the level set contour, and is provided for equation (4.7) of the ISISD framework in order to compute the evolution of the level set function ϕ both on the domains of the images and in the gaps between them.

Note that Yezzi et al. [62] with rigid registration is a special case of (5.4), with two images or full volumes n_1 and n_2, and $\Omega_{n_1} = \Omega_{n_2}$. Also, while Yezzi et al. [62] designed their method using a region based algorithm, the proposed registration method can be used with any segmentation technique, the term S^n being a general speed for the contour, similar to S in (4.7). The curvature term $\kappa\mathbf{N}$ in (3.40a) is dropped in (5.4a) because the level set of ISISD does not require contour smoothing due to the intrinsic smoothing of the convolutions by the RBF (see Chapter 4).

In medical imaging applications, all the images of a sequence are acquired in a single measure (and breath hold if needed). The patient should be still during this short acquisition time and we can reasonably assume that the images of the same sequence (either spatial or temporal) are spatially aligned with each other. Thus, all the images of the same sequence are treated as a rigid dataset and are moved in unison during registration, and we sum the values at all images of sequence n in the integrals in (5.4). In the case of 4D cardiac MRI made up of temporal sequences, the acquisitions of the time-frames of different slice positions are synchronised by an electrocardiogram (see Section 2.1.1), therefore these time-frames can also be assumed to be temporally aligned. Thus, to sum up, all the time-frames of one temporal sequence are considered to be mutually spatially registered, and all images at a given time-frame and from different temporal sequences are considered to be temporally (but not spatially) aligned.

5.3.2 Improvement of the Speed of Convergence

The registration algorithm converges more quickly if the amounts of translation and rotation reflect the degree of misalignment of the associated sequence n with the rest of the dataset. A large misalignment should result in a considerable amount of shift and rotation of the sequence, while smaller, finer shifts and rotations should ensue from a limited misalignment. However, contrary to that in [62], which uses dense and overlapping data, this condition is not automatically fulfilled with intersecting sparse data. This is because the intersection of the images of sequence n with images of other sequences usually represents only a small portion of their pixels, and only a small part of the points in the integrals in (5.4) belong to other intersecting sequences.

We solve this problem by modifying the norms of $\frac{\partial \mathbf{T}^n}{\partial t}$ and $\frac{\partial \boldsymbol{\theta}^n}{\partial t}$ (with $\boldsymbol{\theta}^n = \begin{bmatrix} \alpha^n, & \beta^n, & \gamma^n \end{bmatrix}^T$ being the vector of rotation angles that define R^n) according to an estimation of misalignment λ_n which exploits the dissimilarity between the contour's speeds S^n and S^m, where $m \neq n$, at intersection points between sequence n and all other sequences m, such that

$$\lambda_n = \frac{\left|\Omega_{(S^n \circ g^n)\times(S^m \circ g^m)<0} \cap C\right|}{|\Omega_n \cap \Omega_m \cap C|} , \tag{5.5}$$

where $\Omega_{(S^n \circ g^n)\times(S^m \circ g^m)<0}$ is the domain where images of sequence n and images of all other sequences m produce speeds of opposite signs for the contour C, and $|.|$ denotes cardinality.

Simple piecewise linear schemes were found to be sufficient to obtain satisfactory convergence speeds. We empirically established the following threshold values for λ_n and their associated amounts of transformation, which we applied in all our experiments to compute the amount of shift and rotation for sequence n, in pixels and degrees respectively, *i.e.*

$$\mathit{shift}^n = \begin{cases} 1 & \lambda_n \geq 0.6 \\ \text{linear from } 0.1 \text{ to } 1 & 0.3 \leq \lambda_n < 0.6 \\ \min\left\{\text{linear from } 0 \text{ to } 0.1, \left\|\frac{\partial \mathbf{T}^n}{\partial t}\right\|\right\} & 0 \leq \lambda_n < 0.3 \end{cases} , \tag{5.6}$$

$$\mathit{rotation}^n = \begin{cases} \sqrt{\mathbb{D} \times {a_1}^2} & \lambda_n \geq 0.6 \\ \text{linear from } \sqrt{\mathbb{D} \times {a_2}^2} \text{ to } \sqrt{\mathbb{D} \times {a_1}^2} & 0.2 \leq \lambda_n < 0.6 \\ \min\left\{\text{linear from } 0 \text{ to } \sqrt{\mathbb{D} \times {a_2}^2}, \left\|\frac{\partial \boldsymbol{\theta}^n}{\partial t}\right\|\right\} & 0 \leq \lambda_n < 0.2 \end{cases} , \tag{5.7}$$

with $a_1 = 1°$ and $a_2 = 0.1°$ being angles of rotation around each axis, and $\mathbb{D}$ the number of dimensions. The modified shift and rotation vectors are then

$$\frac{\widetilde{\partial \mathbf{T}^n}}{\partial t} = \frac{\frac{\partial \mathbf{T}^n}{\partial t}}{\left\|\frac{\partial \mathbf{T}^n}{\partial t}\right\|} \cdot \mathit{shift}^n , \tag{5.8}$$

$$\frac{\widetilde{\partial \boldsymbol{\theta}^n}}{\partial t} = \frac{\frac{\partial \boldsymbol{\theta}^n}{\partial t}}{\left\|\frac{\partial \boldsymbol{\theta}^n}{\partial t}\right\|} \cdot \mathit{rotation}^n . \tag{5.9}$$

Note that $\left\|\frac{\partial \boldsymbol{\theta}^n}{\partial t}\right\|$ only approximates the amount of rotation of R^n. However, its computation is simpler than finding the rotation axis and angle of rotation of R^n, and the scheme in (5.7) and (5.9) was found to provide fast enough convergence.

In order to avoid the modelled object drifting out of the boundaries of the volume, we subtract an offset from the update of each component of g^n, which is the minimum displacement $\widetilde{\frac{\partial g_i^n}{\partial t}}_{min}$ in absolute values amongst all sequences, *i.e.*

$$\mathit{offset}_i = \widetilde{\frac{\partial g_i^n}{\partial t}}_{min}, \tag{5.10}$$

with $\widetilde{\frac{\partial g_i^n}{\partial t}}_{min}$ such that:

$$\left\| \widetilde{\frac{\partial g_i^n}{\partial t}}_{min} \right\| = \min_n \left\| \widetilde{\frac{\partial g_i^n}{\partial t}} \right\|. \tag{5.11}$$

This also restricts the translations and rotations to their minimal useful parts and hence limits the amount of deformation that the level set contour must undergo at each iteration in order to adapt to the displacements of the sequences.

Finally, $\mathbf{T}^n$ and R^n are updated as:

$$\begin{cases} \mathbf{T}_{t+1}^n & = & \mathbf{T}_t^n + dt\left(\widetilde{\frac{\partial \mathbf{T}^n}{\partial t}} - \mathit{offset}_{\mathbf{T}}\right) \\ \boldsymbol{\theta}_{t+1}^n & = & \boldsymbol{\theta}_t^n + dt\left(\widetilde{\frac{\partial \boldsymbol{\theta}^n}{\partial t}} - \mathit{offset}_{\boldsymbol{\theta}}\right), \end{cases} \tag{5.12}$$

with $\mathit{offset}_{\mathbf{T}} = \begin{bmatrix} \mathit{offset}_{T_x}, & \mathit{offset}_{T_y}, & \mathit{offset}_{T_z} \end{bmatrix}^T$ and $\mathit{offset}_{\boldsymbol{\theta}} = \begin{bmatrix} \mathit{offset}_{\alpha}, & \mathit{offset}_{\beta}, & \mathit{offset}_{\gamma} \end{bmatrix}^T$.

5.3.3 Robustness to Local Minima

In (5.4b) to (5.4e), only the points located on the level set contour are used to update the position and orientation of a sequence. This implies that, before the contour converges on the edges of the object, the global geometry of the object is unknown. This is illustrated in Fig. 5.2b with the simple example of two 2D overlapping images. As a result, the registration is more likely to get trapped in local minima, especially in cases of high initial misalignments, such as in Fig. 5.2c where the level set contour is unable to continue its growth because of the contradictory information provided by the different images.

We address this issue by using the speeds S both on the contour and in the domain $\Omega_{\neq}^n$ where $(S^n \circ g^n) \times \phi < 0$. These additional speeds provide more information on the global shape of the object, as illustrated in Fig. 5.2d. Therefore, we refer to this variant of the proposed registration method as the “global variant”, as opposed to the previous

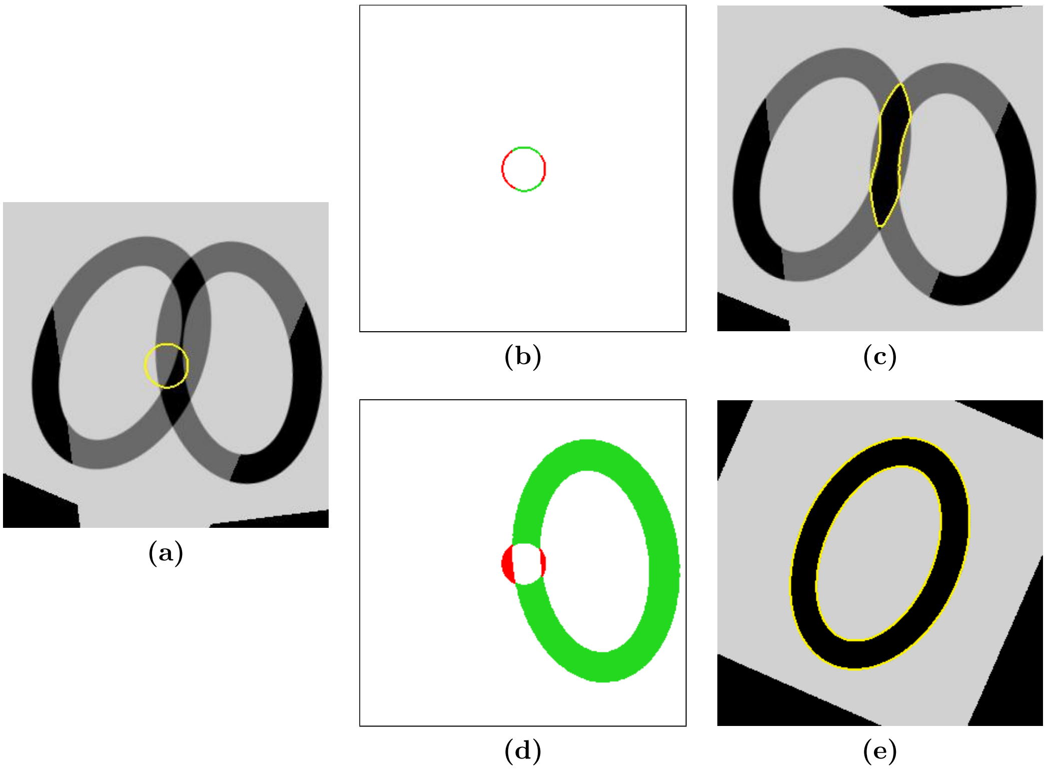


Figure 5.2: *Improvement of the robustness of registration to local minima using global information on the geometry of the object – local variant (top row) vs. global variant (bottom row) of the proposed registration method in the simple case of two identical 2D overlapping images suffering from a significant initial misalignment (left column, with initial segmenting contour superimposed in yellow). Middle column: initial segmentation speeds S, displayed for the right hand side image of the two overlapping images – red: negative speeds; green: positive speeds. Right column: registration and segmentation results. Note that the local variant is trapped in a local minimum, while the global variant is able to use the extra information about the geometry of the object provided by the segmentation speeds S to align the two images.*

“local variant” which uses local information only.

Equations (5.4b) to (5.4e) then become

$$\left\{ \begin{array}{lcl} \frac{\partial T_i^n}{\partial t} & = & \frac{1}{P_n} \int_{(\Omega_n \cap C) \cup \Omega^n_{\neq}} (S^n \circ g^n) \left\langle u_i^n, R^n \mathbf{N} \right\rangle \\ \frac{\partial \alpha^n}{\partial t} & = & \frac{1}{P_n} \int_{(\Omega_n \cap C) \cup \Omega^n_{\neq}} (S^n \circ g^n) \left\langle R_{\gamma^n} R_{\beta^n} R_{\frac{\pi}{2}+\alpha^n} \mathbf{x} + \mathbf{T}^n, R^n \mathbf{N} \right\rangle \\ \frac{\partial \beta^n}{\partial t} & = & \frac{1}{P_n} \int_{(\Omega_n \cap C) \cup \Omega^n_{\neq}} (S^n \circ g^n) \left\langle R_{\gamma^n} R_{\frac{\pi}{2}+\beta^n} R_{\alpha^n} \mathbf{x} + \mathbf{T}^n, R^n \mathbf{N} \right\rangle \\ \frac{\partial \gamma^n}{\partial t} & = & \frac{1}{P_n} \int_{(\Omega_n \cap C) \cup \Omega^n_{\neq}} (S^n \circ g^n) \left\langle R_{\frac{\pi}{2}+\gamma^n} R_{\beta^n} R_{\alpha^n} \mathbf{x} + \mathbf{T}^n, R^n \mathbf{N} \right\rangle \ . \end{array} \right. \tag{5.13}$$

The estimation of misalignment λ_n is changed accordingly into

$$\lambda_n^{global} = \frac{\left|\Omega_{(S^n \circ g^n)\times(S^m \circ g^m)<0} \cap \left(C \cup \Omega^n_{\neq} \cup \Omega^m_{\neq}\ \right)\right|}{\left|\Omega_n \cap \Omega_m \cap \left(C \cup \Omega^n_{\neq} \cup \Omega^m_{\neq}\right)\right|} \ . \tag{5.14}$$

Using the simple example of two 2D overlapping images, Fig. 5.2 illustrates how this global approach can greatly reduce the number of local minima which may bias the registration in the case of the local approach – which, in this 2D example with region-based segmentation, is effectively the method of Yezzi et al. [62]. The global variant of the registration method can exploit the geometry of the object *before* the level set contour matches its shape, through the use of more segmentation speeds S^n, as shown in Fig. 5.2d, thus avoiding the method getting trapped in local minima in Fig. 5.2e.

5.3.4 Extended Velocities and Normals

The global variant of the registration method (5.13) computes S^n for all data points of the contour C and the domain $\Omega^n_{\neq}$, rather than only on the contour as in (5.4). Therefore, an extended velocity scheme needs to be used for the computation of S^n. We implemented a simple extended velocity scheme where S^n is computed everywhere in the same way as on the contour.

In addition, the normal vector $\mathbf{N}$ is now used potentially in the whole volume rather than just on the contour, thus we must ensure that $\mathbf{N}$ is directed pointing outwards from the contour everywhere. This can be enforced by adding the classic level set normalisation term to (4.9):

$$\frac{\partial \phi}{\partial t}(\mathbf{x}) = \left(\frac{d\alpha}{dt} \star \psi\right)(\mathbf{x}) + sign\left(\phi(\mathbf{x})\right)\left(1 - \|\nabla\phi(\mathbf{x})\|\right) \ , \tag{5.15}$$

where $\star$ is the convolution operator.

Note that the normalisation term is used to update ϕ directly rather than being incorporated into S in (4.7), similarly to the geometric terms in (4.12). Indeed, this term can also be defined everywhere in the volume and does not depend on the availability of data, and therefore it does not require any interpolation. In addition, its smoothing by convolutions with the RBF is not desirable and may degrade its performance.

5.3.5 Complete Algorithm

As seen in Section 5.3.3 and illustrated in Fig. 5.2, the global variant is desirable in order to avoid local minima when there is significant misalignment. However, our experiments showed that the local variant is more robust to errors in the segmentation, such as spurious false object pixels in the background. Indeed, these spurious points would be used by the global variant and might bias its registration, while the local variant only uses contour points and would ignore them. The local variant is therefore more accurate and can take over when the deforming contour sufficiently matches the shape of the object in order to achieve best local fitting.

Thus, we recommend to use a combination of the two global and local variants in order to benefit from their complementary advantages. First, the global variant should be used in order to get out of possible local minima for the local method. Then, when the level set contour matches the shape of the object sufficiently well, thus indicating that the registration is not stuck in a local minimum, the algorithm should switch to the local variant. We use the following condition as a sufficient matching condition:

$$\left|\Omega_{(S^n \circ g^n)>0} \cap \Omega_{\phi>0}\right| > \left|\Omega^n_{\neq}\right| \ . \tag{5.16}$$

The overall algorithm for the IReSISD framework, which integrates the combined registration method and ISISD's segmentation and interpolation, is outlined in Algorithm 5.1.

Note, as in Chapter 4, that the IReSISD framework is not limited to sets of slices but can apply to more different spatial configurations, *e.g.* 3D point clouds. Indeed, similarly to ISISD, the registration method of IReSISD makes no assumption on the location of the data points in (5.4) and (5.13).

5.4 Implementation

In this section, we cover the main implementation aspects of the registration method and the complete IReSISD framework.

```
Initialize a contour $C$
Initialize a boolean $global$ to **true** if local minima are expected, **false** otherwise
repeat
    Compute contour speeds $S$ using (5.4a)
    if $global = $ true then
        Update $C$ using (4.7), (5.15) and (4.13)
    else
        Update $C$ using (4.7) and (4.8)
    end if
    for all sequence $n$ do
        if $global = $ true then
            Compute $\frac{\partial \mathbf{T}^n}{\partial t}$ and $\frac{\partial \boldsymbol{\theta}^n}{\partial t}$ using (5.13)
            Estimate misalignment $\lambda_n$ using (5.14)
            if condition (5.16) = true then
                Set $global$ to false
            end if
        else
            Compute $\frac{\partial \mathbf{T}^n}{\partial t}$ and $\frac{\partial \boldsymbol{\theta}^n}{\partial t}$ using (5.4b) to (5.4e)
            Estimate misalignment $\lambda_n$ using (5.5)
        end if
        Compute $shift^n$ and $rotation^n$ using (5.6) and (5.7)
        Compute $\widetilde{\frac{\partial \mathbf{T}^n}{\partial t}}$ and $\widetilde{\frac{\partial \boldsymbol{\theta}^n}{\partial t}}$ using (5.8) and (5.9)
    end for
    for all degree of freedom $i$ do
        Compute $offset_i$ using (5.10) and (5.11)
    end for
    for all sequence $n$ do
        Update $\mathbf{T}^n$ and $R^n$ using (5.12)
    end for
until convergence
```

Algorithm 5.1: – *Integrated Registration, Segmentation, and Interpolation for Sparse Data*

5.4.1 Choice of RBF

As in the ISISD framework, the flatness of the RBF has an influence on the segmentation and interpolation results. The RBF flatness selection schemes proposed in Section 4.4.3 can also be used in the IReSISD framework in order to obtain a better segmentation and interpolation.

Similarly, the flatness of the RBF has an impact on the registration results. Indeed, when the RBF is too sharp, the contour can sometimes fold around an image slice. This is illustrated in Fig. 5.3, where the contour folded around particularly badly misaligned slices (seen from the side in this figure) before they had time to move to better positions. When such a slice has a strong misalignment, it may get trapped in a local minimum. This problem can be solved by using a flatter RBF, which will yield a smoother contour, as explained in Section 4.4.2.

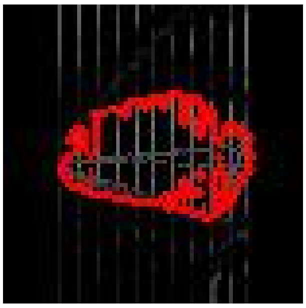


Figure 5.3: *Possible effect of a too sharp RBF on registration: contour folding around individual slices prevents the slices from being properly registered.*

Table 5.1 presents average registration errors obtained by the registration method of IReSISD when aligning artificial heart datasets in the SA plane using two different RBF flatness parameters. We see that the flat RBF (with $\gamma = 1$) avoids contour folding traps, and thus yields better results than the sharp RBF (with $\gamma = 0.5$), with average errors of (0.353, 0.331) pixels and (0.569, 0.466) pixels respectively.

Table 5.1: *Influence of RBF flatness on registration accuracy of intersecting 2D image slices (results are in pixels).*

RBF flatness	Average mean and std of registration error
$\gamma = 0.5$	(0.569, 0.466)±(0.448, 0.237)
$\gamma = 1$	(0.353, 0.331)±(0.245, 0.202)

Note that the small object or hole preserving scheme proposed in Section 4.4.3 is also suited for this application, since it uses a sharp RBF for a more accurate segmentation, then a flat RBF for refining the interpolation, successively. In this case, both registration and interpolation can be refined during the second step, with the flat RBF correcting possible foldings of the contour. This scenario will be illustrated in our experiments in Section 5.5.

5.4.2 Choice of Segmentation Algorithm and Time Step

The discussion on the choice of segmentation algorithm of Section 4.4.4 remains valid in the IReSISD framework. Thus, any segmentation algorithm may be used, provided that it is adapted to the data being processed. It should be stressed that this choice of algorithm impacts the registration stage in the same way that it impacts interpolation in ISISD (and IReSISD), since registration also depends on segmentation results through the terms S^n in (5.4) and (5.13).

In our implementation the same time step dt is used for updating both the registration functions and the level set function. As in the ISISD framework, dt is set to 1 at the beginning of the process and is reduced progressively when oscillations of the level set contour are detected in order to allow a finer convergence of both the registration, and the segmentation and interpolation.

5.4.3 Speed Selection

When registering cluttered images using the global variant of the registration method and an intensity based segmentation algorithm, one must ensure that any background objects which have similar intensities to the object of interest do not attract the contour and bias the registration. Indeed, if the segmentation algorithm confuses a background object with the object of interest when computing the extended velocities, then the resulting spurious speeds S^n may greatly degrade the registration results when used in (5.13). This is illustrated in the top row of Fig. 5.4, where the background speeds biased the registration, leading to the level set contour segmenting the wrong objects. Thus, in Fig. 5.4b, the SA slice was so much shifted by the registration using the speeds of Fig. 5.4a that the RV contour segmented the LV, and the LV contour segmented part of the background. In Fig. 5.4d, the segmentation of the LA slice was poor due to both spurious speeds in Fig. 5.4c that biased the registration, and conflicting information from the intersecting slices.

Therefore, it may be necessary to select the positive speeds S^n which are allowed to be used in (5.13). In our implementation, this is enforced by simply setting to zero any areas of positive speeds which are not connected to the interior of the level set contour. This way, only the object being segmented can have positive speeds and drives the registration, as in the bottom row of Fig. 5.4, where the two contours segmented

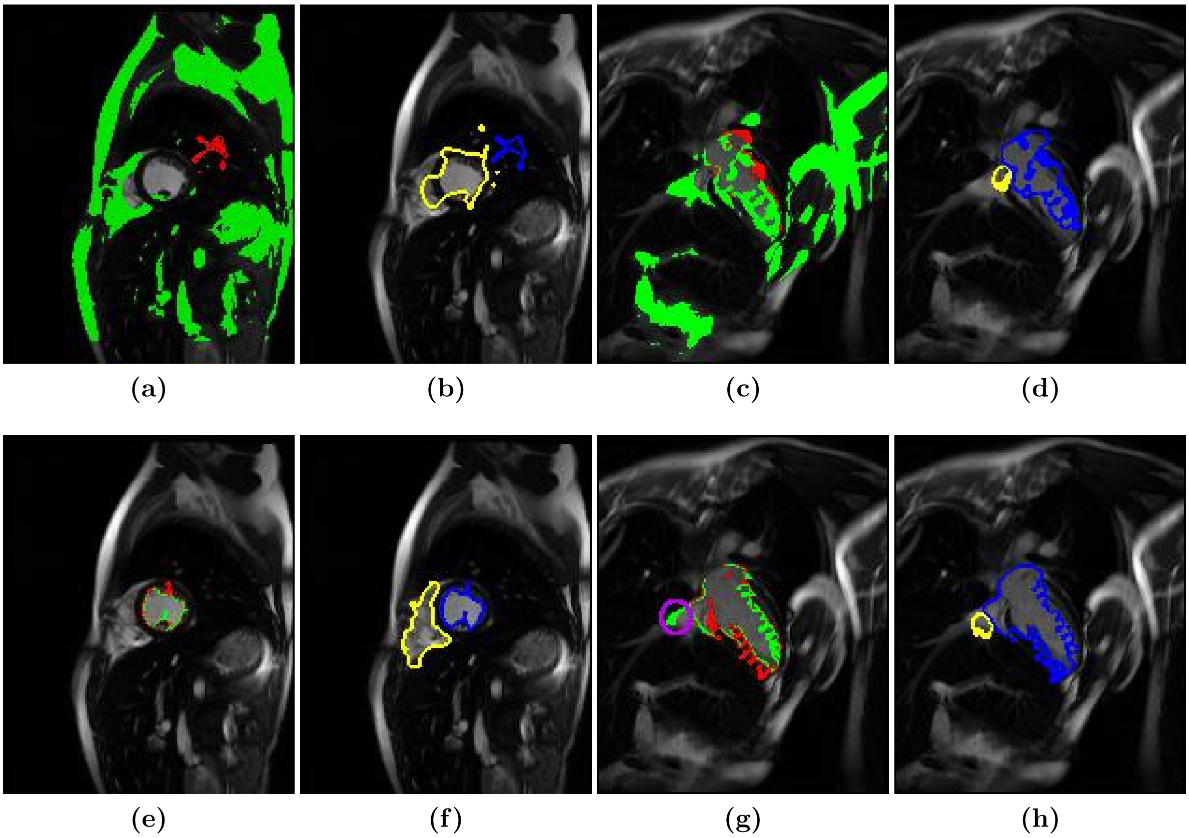


Figure 5.4: *Speed selection for preventing the effect of spurious speeds on registration results. Global variant of the registration method using: top row: all positive speeds in the images, bottom row: selected positive speeds. From left to right: speeds and final segmentation for a central SA slice, and speeds and final segmentation for an LA slice. Note that only speeds for the LV contour are displayed. Speeds for the RV contour have similar properties. Green and red: positive and negative speeds, respectively. Blue and yellow: LV and RV segmenting contours, respectively. The purple circle highlights some remaining spurious speeds that require the use of the local variant of the registration method to be discarded.*

the correct objects. As explained in Section 5.3.5, the accuracy of the registration and segmentation can be further improved by the local variant of the registration method, which would not use some remaining spurious positive speeds, such as that highlighted by a purple circle in Fig. 5.4g.

Note that this technique requires the initial level set contour to be positioned inside the object to be segmented. Alternatively, the user could be required to select one seed point inside the object of interest on every images being used, and only the positive areas of the speed maps which are connected to these seed points would be selected.

5.5 Evaluation of Registration and Modelling on Medical Tomographic Data

To assess and evaluate the IReSISD framework, we first concentrate on the accuracy and robustness of the *registration* contribution of the proposed framework, and then we assess the performance of the *complete framework*, integrating all stages of registration, segmentation, and interpolation. As in our evaluations of the interpolation of ISISD in Section 4.5, we do not evaluate the quality of the segmentation, and we do not recommend any specific segmentation algorithm.

5.5.1 Evaluation Protocol

Previous registration methods, *e.g.* those presented in [19, 25, 51, 60, 62], are not suitable for comparison, as they are designed to register dense images and can not handle several intersecting 2D images in a 3D volume. The methods presented in [17, 46] are limited to temporal sequences having the specific spatial configuration of one stack of parallel slices intersecting two approximately orthogonal slices. In addition, the similarity measures they use do not allow them to process multi-modal data. The integrated method of Zambal et al. [64] is designed for modelling the heart from cardiac MRI, therefore it cannot be used on our non-cardiac datasets. In addition, the authors informed us that their code and prior (AAM and 3D shape) models were no longer available. Therefore, we compare our registration method against the very popular NMI-based registration approach, which was presented in Section 3.2.1, for two reasons: (a) it is commonly used to register multi-modal medical images, and (b) we implement it using the algorithm described in [33] for registration of sparse data made up of two orthogonal stacks of slices (see Section 3.2.1). Lotjonen et al. [33] is the only method we know of that can handle sparse data made of intersecting slices, and which can be applied to arbitrary spatial configurations even if it was originally designed for two orthogonal stacks of parallel slices. It performs in two iterative steps: first, one sequence is chosen randomly, and second, it is shifted in the direction which increases its similarity measure with all of its intersecting sequences. We initially set the shift step to 2 pixels, then decrease it progressively to 1, 0.1, 0.01 and 0.001 pixels. This method was originally designed for temporal sequences, but the same algorithm can be applied to spatial sequences in our tests. Likewise, although in [33] the method is applied to translation only, it can also be applied to rotation, and it is therefore suitable for comparison. We shall refer to our

Table 5.2: *Values of parameter γ used in the registration experiments*

Experiment	**Modelled object**	**Spatial configuration**	γ
Artificial data	Brain ventricles model	All	2
	LV model A	All	0.7 then 3
Real data (quantitative evaluation)	Kidney	5 pixels spacing	3
		10 pixels spacing	4
		15 pixels spacing	5
		20 pixels spacing	6
	Acetabulum	5 pixels spacing	3
		10 pixels spacing	5
		15 pixels spacing	7
		20 pixels spacing	9
Real data (qualitative evaluation)	Brain ventricles	All	1.5
	Hip joint	All	1.5
	4D LV cavity	All	1 then 4

extended use of [33] as "SR[33]" where SR is for "Sequential Registration".

IReSISD uses the integrated segmentation and interpolation method of ISISD. Therefore, when assessing the performance of the *complete framework*, integrating all stages of registration, segmentation, and interpolation, for fairness we compare against the sequential approach which successively applies NMI based registration and the integrated segmentation and interpolation method of ISISD. We refer to this framework as "SR[33]+ISISD", or for brevity "SR+ISI", where ISI is for "Integrated Segmentation and Interpolation". We also use the same segmentation algorithm for the two compared methods: the PC model from [7] for our artificial data made up of piecewise constant regions, and the Parzen based method for real data. "SR+ISI" and IReSISD also use the same values of γ, which are summarised in Table 5.2.

The registration method of IReSISD is evaluated quantitatively as the registration error of artificially misaligned datasets. Artificial initial misalignments are induced in a dataset by initialising the registration vectors $\mathbf{T}^n$ and $\boldsymbol{\theta}^n$ of its individual sequences to random values. Then, after registration, we compute the registration error as the average and standard deviation of the distances of the individual sequences to their mean translation $\mathbf{T}$ and mean rotation angles $\boldsymbol{\theta}$, since a perfect registration should move all the sequences to the same position.

The complete framework is assessed by computing Jaccard coefficients[2] of the shapes reconstructed from the initial misalignments by both the SR+ISI framework and IReSISD. The recovered shape is first aligned with the original model shape using the mean registration transformation of the sequences of the associated dataset.

In our experiments, we distinguish between two cases:

- Spatial sequences, as in Figs. 2.1a and 2.1b, where all the slices of a sequence move together and no spatial deformation happens within the sequence. In the rest of this chapter, we will denote such registration as "stack-wise" registration.
- Temporal sequences made up of a number of time-frames of a *single* slice position and orientation, as in Fig. 2.1c. In this case, the different slices which span the volume are registered with each other. We denote such registration as "slice-wise" registration.

5.5.2 Quantitative Evaluation of Accuracy on Artificial Data

We evaluate the IReSISD framework on the artificial brain and cardiac datasets. As brain and cardiac MRIs are usually made up of spatial and temporal sequences respectively, the artificial brain datasets are used to evaluate the "stack-wise" registration, and the artificial cardiac datasets are for the "slice-wise" registration.

For both the SR[33] and the proposed registration methods, the interpolation and segmentation for the artificial brain datasets are performed using a single value of 2 for γ. The artificial heart datasets are processed in two steps using the small object or hole preserving approach proposed in Section 4.4.3. The RBF flatness parameter γ is set to 0.7 and 3 for the two steps respectively, as in the experiments on real cardiac MRIs in Section 4.5.6. This is necessary to obtain a detailed segmentation and smooth interpolation. In IReSISD, this also allows correcting any possible foldings of the segmenting contour around the image slices (see Section 5.4.1) during the second processing step. For both the brain and cardiac datasets, the initial level set surface is a small sphere placed at the centre of the volume to be registered.

Registration Error

[2] See Section 4.5.2 for a definition of the Jaccard coefficient.

Stack-wise registration – Table 5.3 shows registration errors, in pixels and degrees, for stack-wise registration, averaged on all 9 artificial brain datasets. In the first row, the sequences are initially randomly shifted by up to 10 pixels in all directions and randomly rotated by up to 10° along all axes, so $\widehat{T_i^n}=\text{rand}\,[-10,10]$ and $\widehat{\theta_i^n}=\text{rand}\,[-10°,10°]$ (with θ_i^n the i^{th} component of θ^n), where $\widehat{.}$ indicates the initial value of the registration vectors. In the second row, we have $\widehat{T_i^n}=\text{rand}\,[-20,20]$ and $\widehat{\theta_i^n}=\text{rand}\,[-20°,20°]$. In the first test, the SR[33] method is slightly more accurate than the proposed registration method, but both methods achieve sub-pixel and sub-degree accuracy, with average errors on all datasets of (0.02, 0.05, 0.08) pixels for vectors $\mathbf{T}^n$ and (0.05°, 0.08°, 0.05°) for vectors θ^n for the SR[33] method, against (0.05, 0.08, 0.09) pixels and (0.27°, 0.35°, 0.22°) for the proposed registration method.

For more considerable initial misalignments, such as in the second test where $\widehat{T_i^n} = \text{rand}\,[-20,20]$ and $\widehat{\theta_i^n}=\text{rand}\,[-20°,20°]$, the overall performance of the SR[33] method is lower than the proposed registration method, with average errors of (0.74, 2.61, 1.99) pixels and (2.15°, 4.70°, 2.14°). This is due to the SR[33] method being sometimes trapped in local minima, as illustrated on the top row of Fig. 5.5, where two different ventricles of the brain are confused as one single object and superimposed. Conversely, the proposed registration method exploits information on the global shape of the objects that are contained in the sequences, and therefore is more robust to such local minima. As a result, it maintains a good accuracy, with average errors similar to the first test of (0.05, 0.08, 0.07) pixels and (0.34°, 0.40°, 0.22°).

Table 5.3: *Stack-wise registration accuracy on artificial 3D data: mean and standard deviation of registration error (in pixels and degrees for translations and rotations respectively), averaged over all 9 artificial brain datasets.*

Initial misalignment	SR[33]	Proposed
$\widehat{T_i^n} = \text{rand}\,[-10,10]$	(0.02, 0.05, 0.08) ±(1.2e-4, 4.3e-3, 2.3e-3)	(0.05, 0.08, 0.09) ±(3e-3, 6.7e-3, 2.5e-3)
$\widehat{\theta_i^n} = \text{rand}\,[-10°,10°]$	(0.05°, 0.08°, 0.05°) ±(2e-5, 3.8e-5, 2.6e-5)	(0.27°, 0.35°, 0.22°) ±(2e-3, 2.8e-3, 7.0e-4)
$\widehat{T_i^n} = \text{rand}\,[-20,20]$	(0.74, 2.61, 1.99) ±(0.33, 6.81, 2.11)	(0.05, 0.08, 0.07) ±(2e-3, 4.6e-3, 3.4e-3)
$\widehat{\theta_i^n} = \text{rand}\,[-20°,20°]$	(2.15°, 4.70°, 2.14°) ±(0.05, 0.23, 19.47)	(0.34°, 0.40°, 0.22°) ±(2.7e-3, 3.9e-3, 6.9e-4)

Slice-wise registration – Table 5.4 presents the registration errors for slice-wise registration, averaged on all 14 artificial cardiac datasets. In the first two rows, the initial misalignements are in the SA plane, with $\widehat{T_{xy}^n}=\text{rand}\,[-5,5]$ and $\widehat{T_{xy}^n}=\text{rand}\,[-10,10]$ respectively, and $\widehat{T_z^n}=0$ and $\widehat{\theta_i^n}=0°$, and the images are registered by translation in the

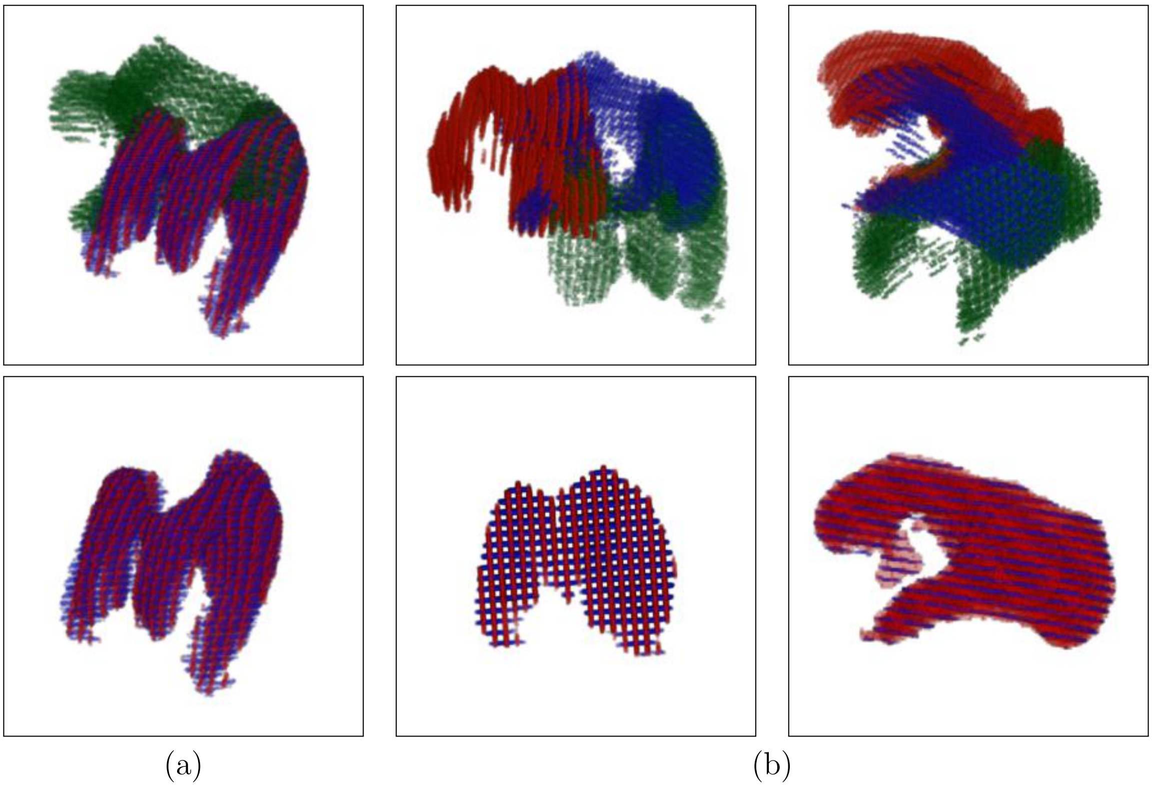


Figure 5.5: *Registration results on two artificial brain datasets which have local minima. Top row: the SR[33] method got trapped in the local minimum, while, bottom row: the proposed registration method achieved a satisfactory registration. (a) dataset 1, viewed from a top-front-left position. (b) dataset 2, viewed from the front (left) and the side (right). Colours correspond to different sequences: red: sagittal, blue and green: axial. Note that in the bottom row, the two axial sequences are nearly perfectly aligned, so only one colour is seen for them. For visualisation purposes, only pixels containing the object are displayed.*

SA plane only. This is motivated by the fact that real cardiac MRIs acquired during a breath hold at end-expiration tend to suffer mainly from shifts in the SA plane, due to small amounts of air remaining in the lungs, and have negligible shifts along the LA direction and rotations. For the first test, while both methods achieve sub-pixel accuracy, the proposed method obtains results that are more precise than the SR[33] method, with average registration errors of (0.36, 0.39) pixels for $\left(T_x^n, T_y^n\right)$ against (0.64, 0.77) pixels for the SR[33] method. The second test is an quite extreme case, with shifts of up to 10 pixels, equivalent to around 20 mm given the image resolution. However, the performance of the proposed registration method remains similar to that of the first test with smaller initial misalignments, with an average error of (0.36, 0.37) pixels. On the other hand, the accuracy of the SR[33] method is strongly degraded to an average error of (1.87, 1.99) pixels.

In the last two rows of Table 5.4, we shift the slices in all directions, with $\widehat{T^n_{xy}} = \text{rand}\,[-5, 5]$ and $\widehat{T^n_z} = \text{rand}\,[-3, 3]$ in the third row, and $\widehat{T^n_{xy}} = \text{rand}\,[-10, 10]$ and $\widehat{T^n_z} = \text{rand}\,[-4, 4]$ in the fourth row. At the same time, we also rotate the slices with $\widehat{\theta^n_i} = \text{rand}\,[-5°, 5°]$. Although the registration protocol of [33] is designed for registration by translation only, the same protocol can also be used with both translation and rotation. Again, the two methods achieve a similar accuracy in the third row, with the proposed registration method being slightly more accurate than the SR[33] method, with average registration errors of (1.05, 0.78, 0.95) pixels and (3.1°, 2.97°, 2.65°), and (1.29, 1.51, 2.02) pixels and (3.23°, 2.96°, 3.2°) respectively. In the fourth row, the proposed registration method maintains a good accuracy, with average errors of (1.2, 1.15, 1.13) pixels and (3.12°, 2.52°, 2.79°), while the average errors of the SR[33] method are higher at (2.8, 2.5, 2.08) pixels and (3.05°, 2.95°, 2.97°).

Table 5.4: *Slice-wise registration accuracy on artificial 3D data: mean and standard deviation of registration error (in pixels and degrees for translations and rotations respectively), averaged over all 14 artificial heart datasets.*

Initial misalignment	**SR[33]**	**Proposed**
$\widehat{T^n_{xy}} = \text{rand}\,[-5, 5]$	(0.64, 0.77)±(0.65, 0.87)	(0.36, 0.39)±(0.42, 0.2)
$\widehat{T^n_{xy}} = \text{rand}\,[-10, 10]$	(1.87, 1.99)±(3.72, 6.11)	(0.36, 0.37)±(0.44, 0.18)
$\widehat{T^n_{xy}} = \text{rand}\,[-5, 5]$ $\widehat{T^n_z} = \text{rand}\,[-3, 3]$ $\widehat{\theta^n_i} = \text{rand}\,[-5°, 5°]$	(1.29, 1.51, 2.02) ±(1.36, 2.16, 2.54) (3.23°, 2.96°, 3.2°) ±(0.09, 0.1, 0.09)	(1.05, 0.78, 0.95) ±(1.12, 0.68, 0.56) (3.1°, 2.97°, 2.65°) ±(0.09, 0.09, 0.07)
$\widehat{T^n_{xy}} = \text{rand}\,[-10, 10]$ $\widehat{T^n_z} = \text{rand}\,[-4, 4]$ $\widehat{\theta^n_i} = \text{rand}\,[-5°, 5°]$	(2.8, 2.5, 2.08) ±(4.6, 3.93, 1.96) (3.05°, 2.95°, 2.97°) ±(0.08, 0.06, 0.07)	(1.2, 1.15, 1.13) ±(3.65, 1.86, 0.78) (3.12°, 2.52°, 2.79°) ±(0.09, 0.09, 0.07)

This difference of results on the artificial cardiac datasets may be explained by the fact that the SR[33] method uses the intensity at the intersections of images to compute the NMI, and therefore requires a minimum number of intersection points in order to obtain meaningful similarity measures. Indeed, in [33] the datasets were made up of two stacks of 6 and 8 parallel SA and LA slices respectively, both spanning the whole left ventricle of the heart. On the contrary, the proposed registration method is still even in the presence of a low number of intersection points between the different sequences (*e.g.* 2 or 3 LA slices), thanks to its use of global segmentation results rather than local pixel intensity. Therefore, it can handle datasets where the LA slices are less numerous and cross the imaged object at various angles and locations, such as the datasets that we use in our experiments.

A difficulty, which sometimes arises when an image intersects very few other images, is that a position ambiguity can happen, as illustrated in Fig. 5.6, where two different positions can be viewed as equally correct. Such under-constrained situations happened a few times during our slice-wise registration tests, and a few datasets had one or two of their slices caught in a local minimum, at an ambiguous position, especially in the third and fourth tests. This was the case of LA slices suffering from a particularly high misalignment which places them on the opposite side of the rather symmetrical object, or of slices being moved to this position by the small, early stage segmentation contour, before its shape matches the shape of the object.

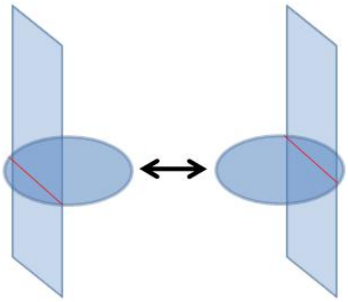


Figure 5.6: *Ambiguity in the registration of a slice due to insufficient constraints,* i.e. *too few intersections with other slices.*

This last scenario could be avoided by first inhibiting rotations and translations both out of the plane of the image and out of the SA plane. Then, when the shape of the contour was judged reliable enough, rotation and translation in all directions were authorised. In our implementation this condition is reached when the image and the majority of its intersecting images achieve condition (5.16).

The first ambiguity scenario, with slices suffering from an exceptionally high initial misalignment, can not be avoided since the correct registration of these slices is not known beforehand. This resulted in a wider distribution of the slices around their mean position in these datasets, which explains the large average standard deviations presented in Table 5.4. Most of the time, this did not prevent the contour to recover the proper shape of the object, as both ambiguous positions were acceptable in a topological point of view, as will be illustrated next.

Complete Framework

Table 5.5 shows the Jaccard coefficients averaged on all datasets for all tests and both methods.

Table 5.5: *Mean Jaccard coefficients obtained on artificial 3D data.*

Test	SR+ISI	IReSISD
Stack-wise 1	0.963	0.957
Stack-wise 2	0.848	0.962
Slice-wise 1	0.889	0.911
Slice-wise 2	0.817	0.910
Slice-wise 3	0.778	0.859
Slice-wise 4	0.667	0.847

Stack-wise registration – In the first test with small misalignments ($\widehat{T_i^n} = \text{rand}\,[-10, 10]$, $\widehat{\theta_i^n} = \text{rand}\,[-10°, 10°]$), both methods recover the shape of the brain model satisfactorily, with mean Jaccard coefficients of 0.963 and 0.957 respectively. For larger initial misalignments, the SR[33] method sometimes falls into local minima, as was illustrated in Fig. 5.5 and discussed during the previous evaluations of registration. In such cases, the resulting reconstructed shapes, shown in Fig. 5.7b, suffer from the contradictory information provided by the misaligned sequences, and as a result the Jaccard coefficient averaged on all datasets is decreased to 0.848. On the contrary, the registration of IReSISD is more robust to such local minima and therefore maintains the same accuracy, and IReSISD recovers satisfactory shapes (Fig. 5.7c) with a mean Jaccard coefficient of 0.962.

Slice-wise registration – The IReSISD framework recovers better shapes than the SR+ISI in all four tests, due to its more accurate registration. In particular, it is not impaired by larger initial misalignments in the second and fourth tests (with $\widehat{T_{xy}^n} = \text{rand}\,[-10, 10]$), and yields mean Jaccard coefficients of 0.910 and 0.847 respectively, similar to those of the first (0.911) and third (0.859) tests with smaller misalignments ($\widehat{T_{xy}^n} = \text{rand}\,[-5, 5]$ in these cases). In addition, in tests 3 and 4, the more significant number of degrees of freedom (3 translations and 3 rotations) makes the registration problem less constrained than in tests 1 and 2 and causes some slices to have several acceptable positions, as was discussed previously when evaluating registration and illustrated in Fig. 5.6. However, the IReSISD framework recovers quite accurate shapes even in such cases, as shown in Fig. 5.8c, with average Jaccard coefficients of 0.859 and 0.847 respectively. Indeed, it manages to place the slices in positions which make them contribute to the overall shape of the object while limiting any conflicts with other slices. The SR+ISI method however, attempts to align the slices without taking into account the global shape of the object, and therefore yields poor registration and shape recovery in such cases, as illustrated in Fig. 5.8b, with average Jaccard coefficients of 0.778 and 0.667 respectively.

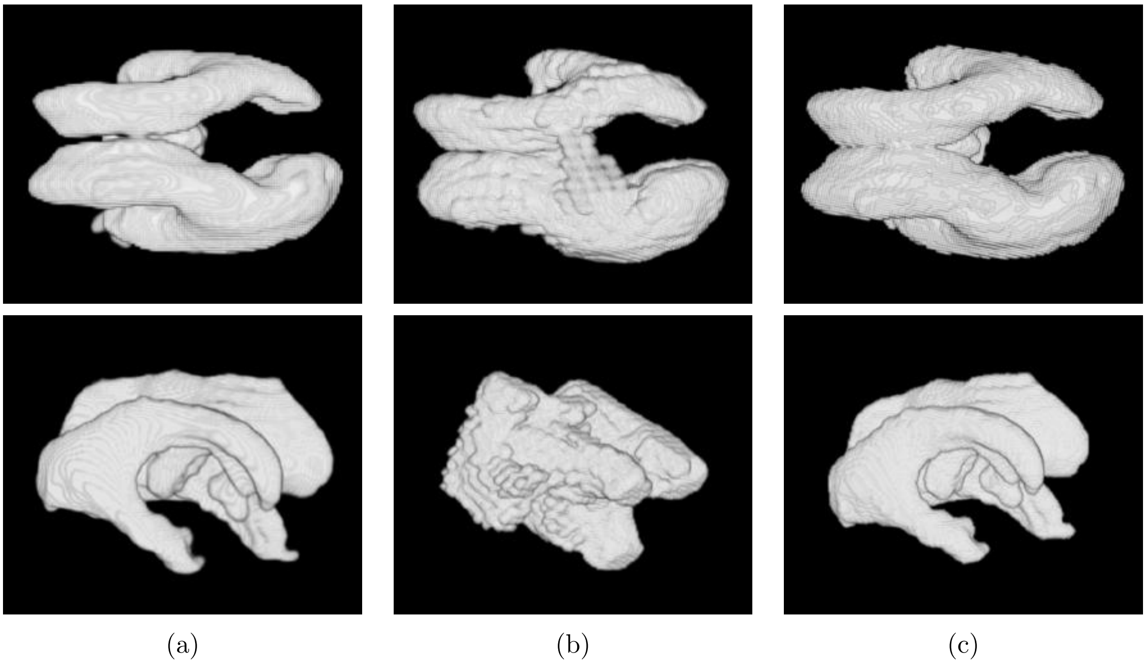


Figure 5.7: *Shape reconstructed from the two datasets of Fig. 5.5 with local registration minima. (a) original shape, (b) reconstruction by SR+ISI, (c) reconstruction by IReSISD. Top row: dataset 1, view from the top. Bottom row: dataset 2, view from a front-right position.*

5.5.3 Quantitative Evaluation of Accuracy on Real Data

In this experiment we show that IReSISD is not limited to images consisting of piecewise constant regions, but can also deal with real images from a variety of modalities that may exhibit large variations in appearance and various levels of noise. We evaluate the quality of registration and of shape reconstruction on the kidney and acetabulum datasets described in Section 2.2.2, using the Parzen estimator based segmentation algorithm described in Section 4.5.1.

The RBF flatness coefficient γ is set to 3, 4, 5 and 6 for the kidney datasets, and to 3, 5, 7 and 9 for the acetabulum datasets, in both cases for data with 5, 10, 15 and 20 pixels spacings respectively. The speed selection strategy presented in Section 5.4.3 is used to segment the kidney datasets because they contain several objects of similar intensities which, otherwise, would bias the intensity based segmentation algorithm.

Registration Error

Stack-wise registration – We perform two tests with $\widehat{T_i^n} = \text{rand}\,[-10, 10]$ and $\widehat{\theta_i^n} = \text{rand}\,[-10°, 10°]$, and $\widehat{T_i^n} = \text{rand}\,[-30, 30]$ and $\widehat{\theta_i^n} = \text{rand}\,[-20°, 20°]$ respectively. In the first test, only a

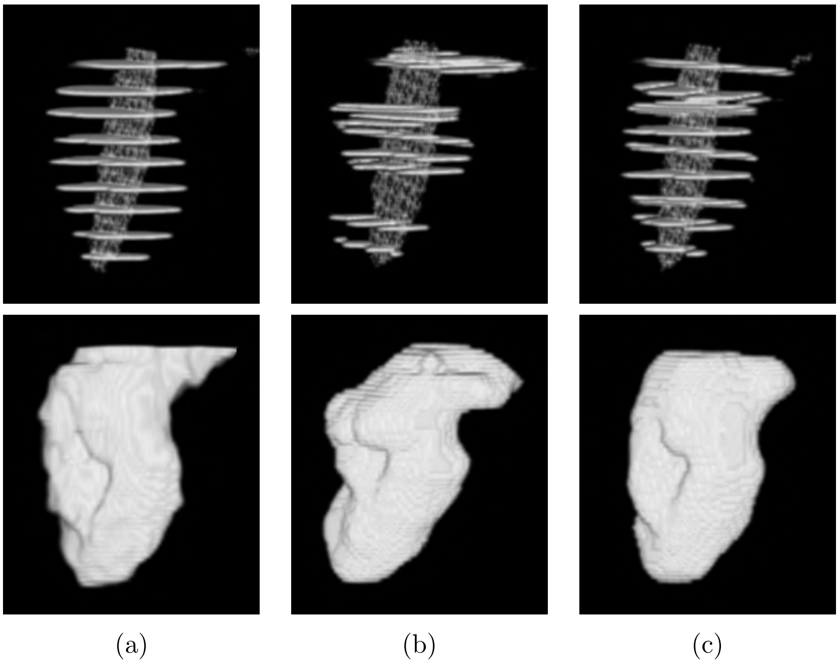


Figure 5.8: *Shape reconstruction from an artificial cardiac dataset containing slices with ambiguous poses. (a) original dataset before artificial misalignment of test 3, dataset processed (b) by SR+ISI, and (c) by IReSISD. Top row: slices of the dataset. Note that only one LA (vertical) slice is available, hence the numerous ambiguities in the pose of the SA slices. Bottom row: reconstructed 3D shape. IReSISD attempted to recover a globally smooth shape for the object, therefore it reconstructed a more plausible and accurate shape than SR+ISI.*

small Region of Interest (ROI) around the object of interest is used by both methods in order to avoid possible local minima where background objects and the object of interest have similar intensities. This allows testing the accuracy of both methods. In the second test, the entire images are used, in order to evaluate the robustness to local minima. Table 5.6 presents the average registration errors sorted by slice spacings.

The results show the same trend as the corresponding experiments on artificial data. For small initial misalignments and slice spacings, both methods generally achieve sub-pixel and sub-degree accuracy, with an average registration error for the first test of (0.03, 0.14, 0.19) pixels and (0.84°, 0.12°, 0.49°) for the SR[33] method, against (0.05, 0.1, 0.14) pixels and (0.4°, 0.13°, 0.33°) for the proposed registration method. However, for larger spacings or misalignments, the SR[33] method falls into local minima, especially when the image contains several objects of similar intensities, as illustrated in Fig. 5.9a. Therefore, it only achieves an average error of (2.94, 4.9, 7.58) pixels and (3.37°, 1.04°, 3.03°) in the second test. The proposed registration method avoids such issues, due to its global approach. It yields satisfactory registration for all our datasets except one,

Table 5.6: *Stack-wise registration accuracy on real 3D data: mean registration error (in pixels and degrees for translations and rotations respectively), averaged over all datasets for each test.*

Initial mis-alignment	Slice spacing	SR[33]	Proposed
$\widehat{T_i^n}$= rand$\,[-10, 10]$, $\widehat{\theta_i^n}$= rand$\,[-10°, 10°]$	5 pixels	(0, 0, 0.11) (0.01°, 0.001°, 0.001°)	(0.01, 0.01, 0.12) (0.02°, 0.02°, 0.02°)
	10 pixels	(0, 0, 0.11) (0.002°, 0.001°, 0.001°)	(0.01, 0.06, 0.13) (0.06°, 0.09°, 0.1°)
	15 pixels	(0.001, 0.001, 0.11) (0.01°, 0.003°, 0.001°)	(0.09, 0.05, 0.08) (0.11°, 0.16°, 0.1°)
	20 pixels	(0.11, 0.56, 0.42) (3.33°, 0.47°, 1.95°)	(0.08, 0.28, 0.25) (1.39°, 0.26°, 1.09°)
	Average	**(0.03, 0.14, 0.19)** **(0.84°, 0.12°, 0.49°)**	**(0.05, 0.1, 0.14)** **(0.4°, 0.13°, 0.33°)**
$\widehat{T_i^n}$= rand$\,[-30, 30]$, $\widehat{\theta_i^n}$= rand$\,[-20°, 20°]$	5 pixels	(0.001, 0.004, 0.12) (0.02°, 0.004°, 0.003°)	(0.03, 0.03, 0.14) (0.09°, 0.04°, 0.11°)
	10 pixels	(0.001, 0.002, 0.11) (0.01°, 0.001°, 0.003°)	(0.05, 0.07, 0.16) (0.1°, 0.08°, 0.02°)
	15 pixels	(8.66, 12.04, 5.15) (5.39°, 1.14°, 4.07°)	(0.91, 5.88, 1.17) (4.67°, 0.26°, 8.15°)
	20 pixels	(3.08, 7.54, 24.95) (8.07°, 3.02°, 8.04°)	(0.3, 0.69, 0.85) (0.32°, 0.41°, 0.49°)
	Average	**(2.94, 4.9, 7.58)** **(3.37°, 1.04°, 3.03°)**	**(0.32, 1.67, 0.58)** **(1.3°, 0.2°, 2.2°)**

which contains an object with a rather symmetrical shape (Fig. 5.9b). This symmetry causes an ambiguity on the poses of the sequences, similar to the ambiguities discussed in Section 5.5.2. Thus, the method achieves a poor registration for this dataset, with errors of (1.76, 11.75, 2.16) pixels and (9.25°, 0.23°, 16.23°), but the recovered shape is acceptable, as shown in Fig. 5.9e. This case increases the overall error of the proposed registration method in test 2 to (0.32, 1.67, 0.58) pixels and (1.3°, 0.2°, 2.2°). If we discount it, the error for test 2 becomes (0.08, 0.34, 0.49) pixels and (0.41°, 0.23°, 0.35°), which is similar to the results of the first test.

Slice-wise registration – We shift all the slices of a dataset independently in the (x,y)-directions by a random amount bounded by the slice spacing of the dataset: $\widehat{T_{xy}^n} =$ rand$\,[-SP, SP]$ with SP the slice spacing, and $\widehat{T_z^n} = 0$ and $\widehat{\theta_i^n} = 0°$. Registration results are presented in Table 5.7.

This test is different to the slice-wise registration tests of Section 5.5.2 with artificial data,

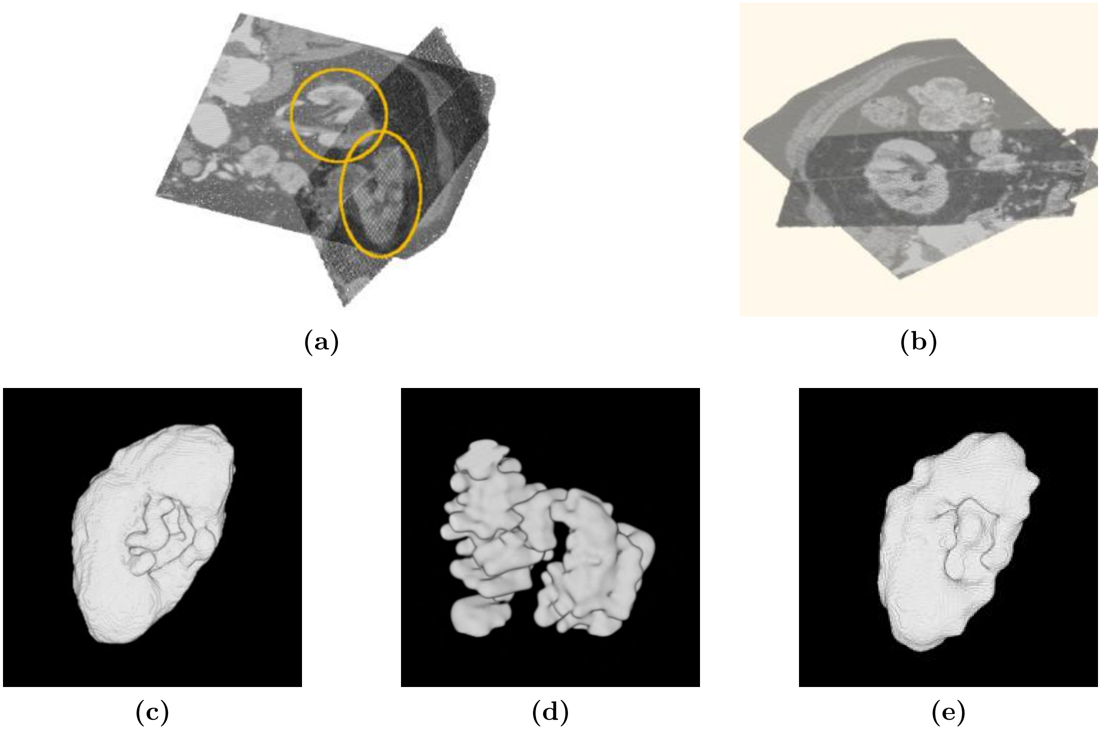


Figure 5.9: *Registration and shape reconstruction from a kidney dataset made up of two spatial sequences containing large (15 pixels) gaps, and suffering from a severe initial misalignment. Top row: registration result – for visualisation purposes, only one central slice of each sequence is displayed. (a) SR[33] method, (b) proposed registration method. The orange circles highlight a failure of the SR[33] method to match together two cross-sections of the kidney, due to a confusion with background objects. Bottom row: reconstructed shape – (c) original shape, (d) SR+ISI, (e) IReSISD.*

since in Section 5.5.2 the datasets were made up of one stack plus a varying small number of roughly orthogonal slices having arbitrary positions and orientations, whilst here the datasets are composed of two orthogonal stacks of parallel slices. This spacial configuration contains more intersections between the slices and is the configuration for which the method in [33] was originally designed. Therefore, both SR[33] and the proposed registration method perform similarly well for slice spacings and initial misalignments of up to 15 pixels, and achieve sub-pixel accuracy.

In the last test, with 20 pixel spacings and maximum misalignments, the SR[33] method encounters local minima problems when objects of the background are confused with the object of interest. At the same time, the number of intersection points becomes less suitable to compute similarity measures. As a result, its average registration error increases to (4.22, 3.45) pixels. The proposed registration method does not suffer from these problems and results in an error of (0.48, 0.84) pixels, similar to the errors of the three other tests.

Table 5.7: *Slice-wise registration accuracy on real 3D data: mean and standard deviation of registration error (in pixels), averaged over all datasets for each test.*

Slice spacing & shift max	**SR[33]**	**Proposed**
5 pixels	(0.34, 0.54)±(0.22, 0.70)	(0.12, 0.20)±(0.01, 0.03)
10 pixels	(0.34, 0.92)±(0.14, 3.58)	(0.17, 0.35)±(0.02, 0.06)
15 pixels	(0.27, 0.27)±(0.02, 0.02)	(0.26, 0.77)±(0.13, 0.71)
20 pixels	(4.22, 3.45)±(27.84, 27.25)	(0.48, 0.84)±(0.19, 0.56)
Average	**(1.29, 1.30)±(7.05, 7.89)**	**(0.26, 0.54)±(0.09, 0.34)**

Complete Framework

The ground-truth shapes used to compute the Jaccard coefficients are the shapes recovered by level set segmentation of the original, full volume datasets. The Jaccard coefficients for both methods, averaged on all datasets for each test, are presented in Table 5.8. When the registration yielded by the SR+ISI and IReSISD are similarly accurate, as in the first stack-wise registration test, both frameworks reconstruct satisfactory shapes, with mean Jaccard coefficients of 0.937 and 0.942 respectively.

Table 5.8: *Mean Jaccard coefficients yielded on real 3D data.*

Test	**SR+ISI**	**IReSISD**
Serie-wise 1	0.937	0.942
Serie-wise 2	0.721	0.929
Slice-wise	0.882	0.934

However, when the SR[33] method falls into local minima because it confuses background objects with the object of interest, as in Fig. 5.9a, the two sequences which make the dataset provide incoherent information to the segmenting level set surface. As a result, the level set attempts to segment the object at two different locations, obtaining a Jaccard coefficient of 0.118 in the case of Fig. 5.9d, which decreases the mean Jaccard coefficient for the second stack-wise registration test to 0.721.

IReSISD does not suffer from this problem. As explained previously, the proposed registration method was trapped in a local minimum in the case of Fig. 5.9, caused by the symmetry of the object. However, the method takes into account the global shape of the object when aligning the images, and attempts to reduce the discrepancy in the segmentation information that they provide. Therefore, in such cases, the proposed framework still recovers a plausible shape, as illustrated in Fig. 5.9e, where it obtained a Jaccard coefficient of 0.832. Thus, the mean Jaccard coefficient for the second stack-wise registration test remains high at 0.929, similar to the first stack-wise registration test with smaller initial misalignments.

Similarly, in the slice-wise registration test, the SR[33] method encountered local minima problems for the largest initial misalignments, and the reconstructions of the full framework suffer from the resulting badly aligned sequences, with an average Jaccard measure of only 0.882. On the contrary, the greater robustness of the proposed registration method to local minima offered a better support to the shape reconstruction of the full framework, and provides an average Jaccard measure of 0.934.

5.5.4 Qualitative Evaluation on Real Data

We qualitatively evaluated the stack-wise registration on the brain and hip MRI datasets, and the slice-wise registration on the cardiac MRI datasets. The same RBF flatness parameter $\gamma = 1.5$ was used for all brain and hip datasets, and it was set to 1 then 4 when processing the 4D cardiac MRIs in two steps. The hip dataset was segmented using a combination of the edge-based model of [61] and the intensity-based model of [7].

The brain and hip datasets had only very small initial misalignments of less than 5 pixels and a few degrees. The registration, segmentation, and interpolation results (Fig. 5.10 and top row of Fig. 5.11) were inspected visually, and we subjectively assessed that both methods perform similarly well on these datasets.

In the bottom of Fig. 5.11, the difficult segmentation of the diagonal sequence in the hip dataset, due to some fine surrounding tissues being of a similar intensity than the bone, biased the reconstruction of the femur head where this sequence was predominant, *i.e.* in the gaps of the other sequences. This resulted in the bumps seen in Fig. 5.11i. This issue can be solved by selecting a more robust segmentation algorithm. When discarding this sequence, the reconstruction from the remaining sequences was visually satisfactory (Fig. 5.11d).

The range of initial misalignments of the temporal sequences of the cardiac MRIs was more important, with shifts of 1 to 10 pixels. We found that registration by rotation and by translation along the long-axis direction did not improve the quality of alignment

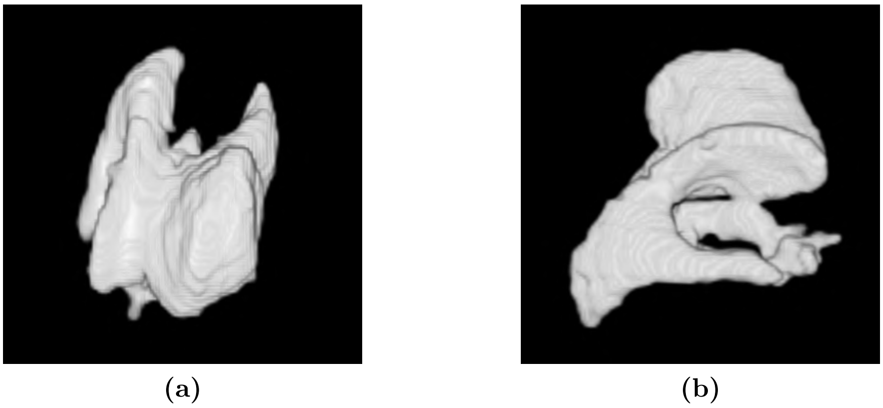


Figure 5.10: *Registration, segmentation, and interpolation of brain ventricles from T1 and T2 MRI viewed (a) from the top, and (b) from the side.*

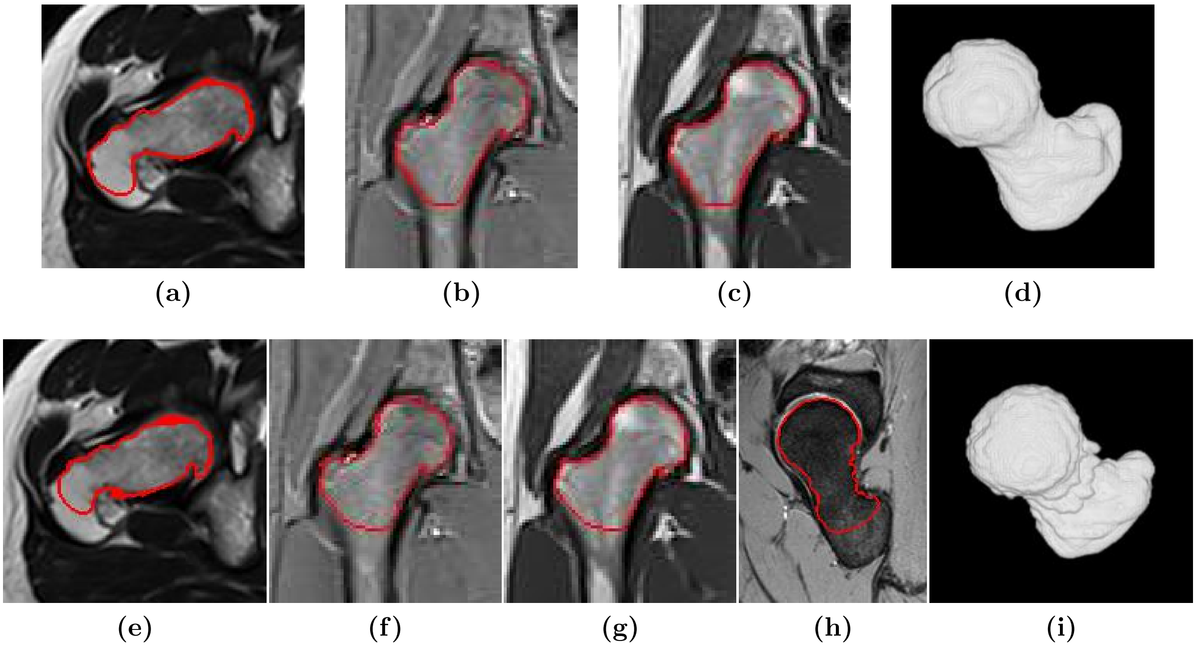


Figure 5.11: *Registration, segmentation, and interpolation of a femur head from T1, T2 and STIR MRI. Top row: modelling from one axial and two coronal sequences. Bottom row: modelling from the same sequences plus a diagonal sequence that is badly segmented, thus biasing the reconstruction in the gaps of the other sequences. (a) to (c) and (e) to (h): central slices of Fig. 2.8 with segmentation in red. (d) and (i): 3D reconstructions.*

on these data. We suspect that this is due to the fact that they were all acquired at end-expiration, which is a position where the heart undergoes negligible or no rotation and shift along its long-axis. Therefore, the results we show have been obtained using translation in the SA plane only. For datasets acquired at end-inhalation or during free breathing, translation along the long-axis and rotation may be added as in Section 5.5.2.

Fig. 5.12 presents some results, by both methods, of slice-wise registration on real cardiac MRIs which had severe initial misalignments of 5 to 10 pixels. The SR+ISI method performs less satisfactorily in such cases, and yields even poorer results than in Section 5.5.2 with the artificial heart datasets. We explain this by the higher complexity of the data, whose background contained several objects, which can confuse the SR[33] registration method. In addition, most of the datasets contained only a small number of LA slices (3 by average) while the method proposed in [33] was designed for datasets made up of two stacks of 6 SA and 8 LA slices, so less intersections between slices were available to compute similarity measures. On the contrary, the proposed registration method registers the datasets very well in all cases, and the complete framework recovers plausible shapes of the cavity of the left ventricle of the heart.

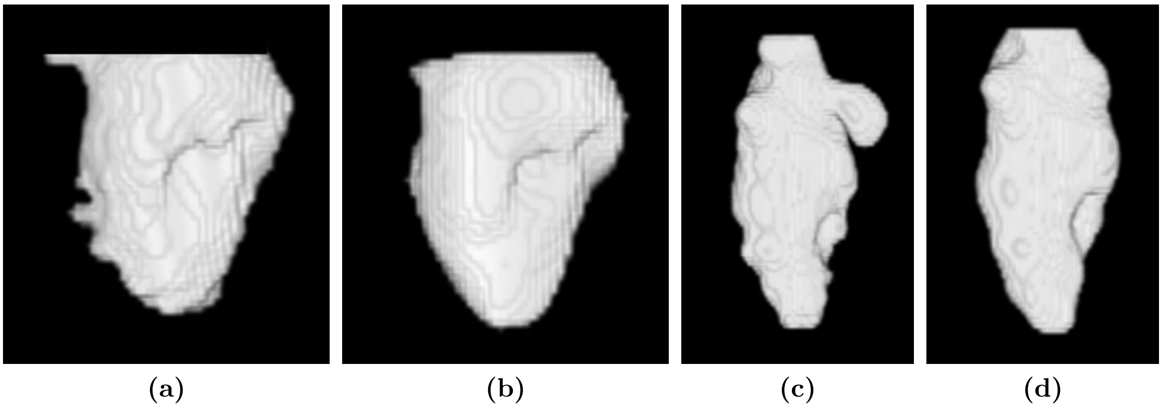


Figure 5.12: *Registration, segmentation, and interpolation of the cavity of the left ventricle of the heart from real 4D MRIs. Only one time-frame is displayed. (a) and (c) SR+ISI, (b) and (d) IReSISD. The protrusions in shapes (a) and (c) are due to the badly aligned SA and LA slices which provide conflicting information to the subsequent integrated interpolation and segmentation of ISISD.*

5.5.5 Timing

The considerations on the timing of ISISD given in Section 4.5.8 also apply to IReSISD, since it uses the same segmentation and interpolation stages than ISISD. When registering the datasets at the same time than segmenting them and interpolating, the IReSISD framework was moderately slower than ISISD, due to the additional computations of the extended velocities S^n, normals $\mathbf{N}$, and normalisation term in (5.15).

As in Section 4.5.8, timing comparisons against SR+ISI mainly highlight differences in the registration times, since ISISD was used for the segmentation and interpolation stages of SR+ISI. In our implementation, IReSISD was always faster than SR+ISI by an order of magnitude. However, this running time can not be fairly compared with SR[33], since we used our own implementation of SR[33], which was performing using a simple iterative gradient descent and was not optimised for speed. Note however that IReSISD does not require an additional registration step like SR+ISI or other sequential methods.

5.6 Conclusion

In this chapter, we extended the ISISD framework of Chapter 4 to a fully unified framework for integrated registration, segmentation, and interpolation for sparse and misaligned data. To this end, we introduced a new registration method based on the level set method and which is adapted to sparse and non-overlapping data, and we integrated it to the ISISD framework.

We have demonstrated the use of this new IReSISD framework to modelling from sparse and misaligned medical tomographic data, and we compared against a state-of-the-art semi-sequential method implemented with the sequential registration of Lotjonen et al. [33], and the integrated segmentation and interpolation of ISISD. We particularly focused on handling misalignements, even in the ill-constrained case of very sparse data having only a few intersections.

Overall, IReSISD proved to be as flexible as ISISD in the variety of shapes, spatial configurations, imaging modalities, and segmentation algorithms that it can handle. The registration method of IReSISD was more robust to small number of intersections, cluttered background, and local minima than the NMI based method of Lotjonen et al. [33]. Its integration to the previous ISISD framework allowed processing simultaneously misaligned data that were produced by different modalities.

The integration of all three stages of registration, segmentation, and interpolation was more robust than sequential processing, due to the three stages benefiting from each other. Note however that in such integration, registration and interpolation rely heavily on segmentation results, and therefore segmentation has to be completely reliable. This should be ensured by designing segmentation algorithms adapted to the data being processed, as will be highlighted by our application of IReSISD to cardiac cine MRIs in the next chapter.

Chapter 6

Modelling of the Heart from Cardiac Cine MRI

In this chapter, we demonstrate the application of IReSISD to modelling the heart from cine MRI. As explained in Section 2.1.1, the modelling of the heart from cine MRI is traditionally challenging in both the registration and interpolation aspects, due to relatively large gaps and a difficult and ill-constrained registration. This application is therefore well suited to evaluate the potential of the IReSISD framework in real situations.

Several heart chambers and the myocardium are modelled simultaneously, thanks to the possibility to extend the framework to handle multiple regions. Indeed, multi-region segmentation is a common feature of level set methods that was proposed *e.g.* in [53, 66]. The proposed level set framework naturally inherits from this feature, as will be demonstrated later. We take advantage of the clear separation between the bright intensities of the blood pools and the darker colours of the surrounding tissues in cardiac cine MRIs to segment the two cavities of the LV and RV using the simple PC model described in Section 4.5.1. The segmentation of the myocardium is more difficult due to the very weak contrast of its external boundary, thus it often requires the use of prior knowledge. Several methods have been proposed in the past to segment the myocardium using prior knowledge, *e.g.* [35]. In this chapter, we will demonstrate the possibility of integrating such prior knowledge with IReSISD in order to model both cavities and the myocardium.

As in Sections 4.5 and 5.5, our evaluations focus on the accuracy of the registration and interpolation, and we compare qualitatively with the popular sequential approach implemented with the NMI based registration of Lotjonen et al. [33], the image interpolation of Cordero-Grande et al. [14], and the same segmentation as in ISISD and IReSISD. Although it would have been desirable to also compare against the fully integrated method of Zambal et al. [64], which uses deformable models to reconstruct the heart, it was not possible, as explained earlier in Section 3.1.2. Therefore, we only compared against the traditional sequential approach.

We first demonstrate in Section 6.1 the possibility to use prior knowledge in IReSISD to segment the myocardium. Then, in Section 6.2, we evaluate the quality of the modelling of the heart cavities based on image data only.

6.1 Joint 3D Modelling of the LV and RV Cavities and Myocardium Using Prior Data

In this experiment, we show that it is possible to integrate prior knowledge to the IReSISD framework in order to constrain the difficult segmentation of the myocardium, whose intensities are very similar to these of the background. In such an application, the use of prior knowledge mainly aims to prevent the contour which segments the external boundary of the myocardium (epicardium) from leaking into the background. For simplicity and clarity of presentation, we use a simple form of prior knowledge, *i.e.* a constraint on the thickness of the myocardium. Note that more elaborate forms and uses of prior knowledge were proposed in the past for the level set framework, *e.g.* by Zhang et al. [67]. The design of an optimum prior model to constrain the active contour based segmentation of cardiac MRI is still an open research subject, and the investigation of such an optimum solution is not in the scope of this work.

Our method is similar to what was presented in [35] for edge based segmentation, but adapted to the PC segmentation algorithm. An initial segmentation of the two LV and RV cavities is computed in a first step, using two level sets which are updated simultaneously and do not need to intersect and interact as a result of the natural separation of the two cavities. Note that, when using the small object or hole preserving approach described in Section 4.4.3, only the first segmentation step with a sharp RBF can be used at that point. Indeed, the aim is to delineate the internal boundary of the myocardium (endocardium), which is the border of the LV cavity.

Then, the segmentation of the myocardium is added to the process, driven by both a coupling term $S_{coupling}$ between the endocardium and the epicardium, and an image derived term S_{image} limited to the area of expected location of the epicardium:

$$S = \nu S_{coupling} + (1 - \nu) S_{image} \ . \tag{6.1}$$

ν is a weighting coefficient and is set to 0.5 in our test. An example of the output of this second step is displayed in Fig. 6.1.

The coupling and image based terms differ from those in [35], since they have been adapted to the PC model. The coupling term is computed as:

$$S_{coupling} = \begin{cases} -1 & \text{if } \phi_1 \leq -d - w \ , \\ \sqrt[3]{\frac{d-|\phi_1|}{w}} & \text{if } -d - w < \phi_1 < -d + w \ , \\ & \text{ and } S_{PC} > 0 \ , \\ S_{PC} & \text{if } -d - w < \phi_1 < -d + w \ , \\ & \text{ and } S_{PC} \leq 0 \ , \\ S_{PC} & \text{if } \phi_1 \geq -d + w \ . \end{cases} \tag{6.2}$$

ϕ_1 is the value of the level set function associated with the LV cavity contour (or endocardium) at the end of the initial segmentation stage, and therefore it represents the (negative) distance to this contour. d and w are the nominal distance between the endo- and epicardium, and the transition width respectively. In [35] they are derived from an *a priori* model, but for simplicity, they are set to 6 and 4 in our test. S_{PC} is the speed yielded by the PC segmentation model. The image derived term is defined as:

$$S_{image} = \begin{cases} \exp\left(-\frac{(x-d)^2}{2w}\right) S_{PC} & \text{if } \phi_1 \leq -d \text{ and } S_{PC} > 0 \ , \\ S_{PC} & \text{otherwise} \ . \end{cases} \tag{6.3}$$

During the second step, the two LV and RV cavities and the myocardium are segmented concurrently by two level sets, using the multi-region level set scheme proposed by Vese and Chan [53]. Note that other multi-region methods may be used, such as the scheme proposed by Zhang and Matuszewski [66] which uses one level set per object and includes a prior term for both contour exclusion and topology preservation that ensures that the LV and RV cavities are surrounded by the outer contour of the heart and exclude each other. For simplicity, in this experiment, we demonstrate the possibility to segment multiple regions using the simpler method of Vese and Chan [53] that is independent of the prior knowledge. This scheme uses a number N of level sets simultaneously in order to segment 2^N piecewise constant or piecewise smooth regions. The regions are defined as all the possible intersections of the areas bounded by the different level set contours, and by construction they cannot overlap or create vacuums. Following this idea, in our implementation two level sets are jointly used to segment the LV and RV cavities (Figs. 6.1b and 6.1c respectively), which are defined as the two areas which are inside one level set contour and outside of the other (blue and red areas in Fig. 6.1a). The myocardium (Fig. 6.1d) is the area where both level set functions are positive (green area in Fig. 6.1a).

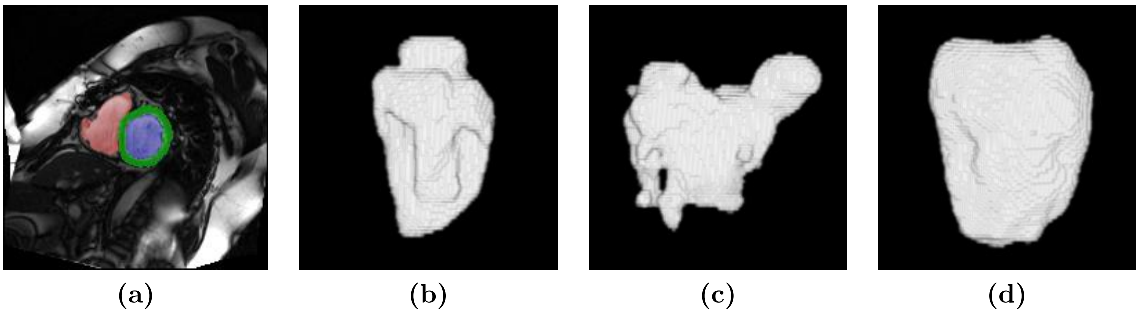


Figure 6.1: *Modelling of multiple regions and using prior knowledge – reconstruction of the two cavities of the heart and myocardium from a cine MRI. (a) SA slice with the LV, RV, and myocardium segmentations coloured in blue, red, and green respectively. 3D reconstructions of (b) the LV, (c) the RV, and (d) the myocardium.*

As in the experiment on cardiac MRIs of Section 4.5.6, IReSISD is used with the small object or hole preserving approach in two steps, as described in Section 4.4.3, with γ set to 0.7 then 3 in order to delineate the papillary muscles.

Note that other, more elaborate prior knowledge may be used with IReSISD and would certainly improve the results presented in Fig. 6.1d. In particular, they may include models of the shape and dynamics of the myocardium. Their integration into IReSISD can be done, as in this experiment, through the definition of a new contour speed $S_{coupling}$.

6.2 4D Modelling of the LV and RV Cavities

This experiment demonstrates how IReSISD can be applied to the 4D modelling of the LV and RV cavities from real cardiac cine MRIs. As in Section 4.5.7, the inherent ability of ISISD and IReSISD to process periodic data is exploited to interpret the cyclical nature of cardiac motion.

Similarly to Section 6.1, the simple PC segmentation algorithm [7] is used in order to separate the bright blood pools of the LV and RV cavities from the dark myocardium and background, and two level sets jointly segment both cavities. Since the myocardium is not segmented in this experiment, a low intensity is artificially assigned to its associated area (where both level set functions are positive) in order to prevent it from segmenting parts of the cavities.

Contrary to Sections 4.5.6 and 6.1, where the small object or hole preserving approach described in Section 4.4.3 had to be used in order to capture the papillary muscles in 3D, we use here a simple scheme in 4D with a single value of 3 for the RBF flatness parameter γ, for all datasets. Indeed, in 4D the papillary muscles are no longer isolated features since they are connected to the myocardium at some point in the cardiac cycle, as illustrated, for example, in Fig. 17 of Appendix C. Therefore, this simple scheme provides the required level of robustness to noise while being able to delineate the papillary muscles.

The initialisation is provided manually by drawing on one image, chosen by the user, the centre and radius of a sphere inside the cavity to segment. This sphere is defined by the user in one time-frame only, but, in order to accelerate the convergence, it also serves as an initialisation for all the other time points.

We perform a slice-wise registration on the cardiac MRIs, with all 25 time-frames being used for computing the registration. The range of initial misalignments was visually assessed to be shifts of 0 to 10 pixels. As in Section 5.5.4, we register by translation in the SA plane only.

We compare qualitatively against the sequential approach implemented with the registration of SR[33] followed by the image interpolation of Cordero-Grande et al. [14] and a 4D level set segmentation. Segmentation is implemented with ISISD in order to benefit from the same robustness to noise as that in IReSISD and therefore to produce comparable segmentations. The interpolation of the volume is performed in 3D for each time-frame separately, since [14] can not perform interpolation across the time domain in an inter-frame fashion. The segmentation stage is performed in 4D by ISISD on the full, interpolated volumes produced by [14]. Therefore, the segmentation stage exploits the periodicity of the data, but not the interpolation stage. The interpolated volumes are treated as 3D images, therefore the c_{in} and c_{out} coefficients of the PC model in (4.19) are computed for the full 3D volumes rather than for each slice separately as in IReSISD. The RBF flatness coefficient γ was tuned to obtain a similar robustness to noise and segmentation quality as in IReSISD. Indeed, the same value for γ on the sparse and the interpolated data would not provide the same segmentation, since the higher density of data points in the interpolated case results in an increased smoothing effect of the RBF convolutions. A value of 2 for γ was found to provide a similar segmentation to IReSISD, and was used for all datasets.

We performed the comparisons on 14 of our cardiac MRI datasets which contain at least one LA slice in order to allow registration. Two examples of segmentation by both the sequential method and IReSISD are shown in Figs. 15 to 21 and 22 to 25 of Appendix C, and Figs. 6.2 and 6.3 present the associated reconstructions. In the case of the sequential method, only the segmentations of the original, non-interpolated slices are shown, because of space constraints. The example dataset 1 is representative of our datasets which were obtained from real patients, and it has a typical level of noise and a standard spatial configuration of 11 SA slices (but only 10 contain the heart and are displayed here) spaced at 10 mm (*i.e.* ∼5 pixels) and 6 LA slices. It is also one of our datasets that suffers from relatively high initial misalignments, with shifts of its individual slices of about 2 to 10 pixels. Example dataset 2 is interesting because of its particularly large gaps of ∼10 pixels (20 mm) between its SA slices, which allow assessing the robustness of interpolation. This dataset is made up of 6 SA slices (again, only 5 slices contain the heart and are displayed here), and 3 LA slices.

The reconstructions obtained by IReSISD contain parts of the atria and great vessels, which are modelled from the LA slices. Note that the surface of the atria and great vessels are less detailed when modelled from datasets having very few (2 or 3) LA slices, such as the example dataset 2. In order to separate the ventricles from the atria in the models, the valves between them would need to be detected. The automatic detection and tracking of the valves are not in the scope of this work and could be the object of a future work, so we present the full models containing both the ventricles and parts of the atria. Note also that the SA slices do not span the atria and stop at the approximate position of the valve between the LV and left atrium. Therefore, the sequential method, which only uses these SA slices for the interpolation and segmentation stages, only modelled the ventricles.

When subjectively compared, we observe that the segmentations of the 2D images of the cardiac datasets are similarly accurate for both methods. Due to the use of the simple

PC segmentation model, the fuzziness of the boundaries of the RV is ignored. As a result, the segmentation of the RV is often incomplete or includes parts of the background. In our worse segmentation case, the RV was not found at the correct place, and this led to the failure of the registration stage. This case had to be handled by modelling the LV alone in a first step, then segmenting and interpolating the RV using ISISD in a second step, without updating the registration. A more elaborate segmentation algorithm would be needed to model the RV cavity more accurately. To this end, investigating dynamic texture based algorithms and the use of prior knowledge could be considered in the future.

The results of the registration stage is found to be more satisfactory by the IReSISD framework than by the sequential approach. The NMI based registration method SR[33] performs less accurately than the registration method of IReSISD on all 14 datasets, and yields poorer results than in the experiments on artificial heart datasets of Section 5.5.2, as was already discussed in Section 5.5.4. When the registration is particularly bad, it offers a poor support to the subsequent interpolation and segmentation stages. In our implementation of the sequential approach with the interpolation of Cordero-Grande et al. [14] which uses SA slices only, this resulted in “twisted” shapes, *e.g.* as highlighted by the blue circle in Fig. 6.2. If both SA and LA slices were to be used, then the bad registration would also lead to the creation of protrusions, due to the conflicting information that the misaligned SA and LA slices would provide to the interpolation and segmentation stages, as was illustrated in Figs. 5.12a,c where the registration of Lotjonen et al. [33] was followed by the integrated interpolation and segmentation of ISISD.

The interpolation is also more satisfactory with the IReSISD framework. Notably, the apex of the LV is better segmented and interpolated by the proposed method than by [14] followed by segmentation, thanks to the method taking into account the global shape of the modelled object and to the use of the LA slices. In the case of the example dataset 2, IReSISD also produced smoother shapes, while [14] interpolated straight sections in the large gaps between the SA slices.

6.3 Conclusion

In this chapter, we have applied the IReSISD framework to modelling cavities of the heart and the myocardium from cardiac cine MRI. First, we demonstrated the use of prior knowledge to constrain a difficult segmentation. Then, we assessed the quality of registration and interpolation qualitatively, and compared against a popular state-of-the-art sequential method.

Overall, IReSISD provided a more robust registration and interpolation than the NMI based method of [33] and the image interpolation of [14], thus confirming the results of our evaluations in Sections 4.5 and 5.5. The inherent ability of the framework to handle periodic data was also exploited and allowed making sense of the cyclical nature of the cardiac cycle.

The quality of segmentation was found to be a limiting factor in this application, that may hinder the registration and interpolation stages in the worse cases. Therefore, more robust segmentation algorithms should be investigated, including the use of texture as was done *e.g.* in [23] for tagged MRIs, as well as prior knowledge based methods.

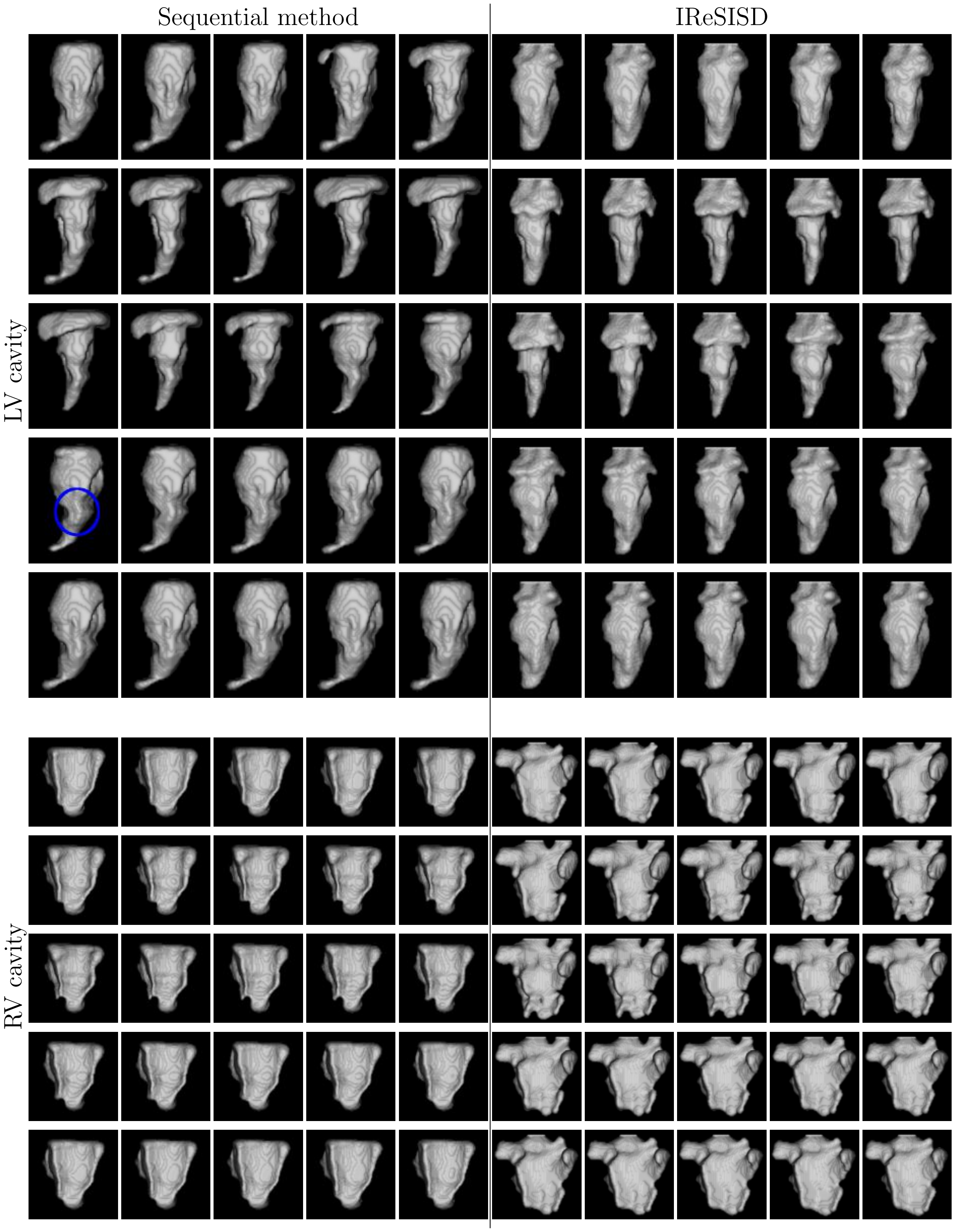


Figure 6.2: *4D modelling of the LV and RV cavities – dataset 1, all 25 time-frames. The blue circle highlights the "twisted" shape of the reconstructed LV cavity, due to the poor alignment of the SA slices yielded by the SR[33] registration of the sequential method.*

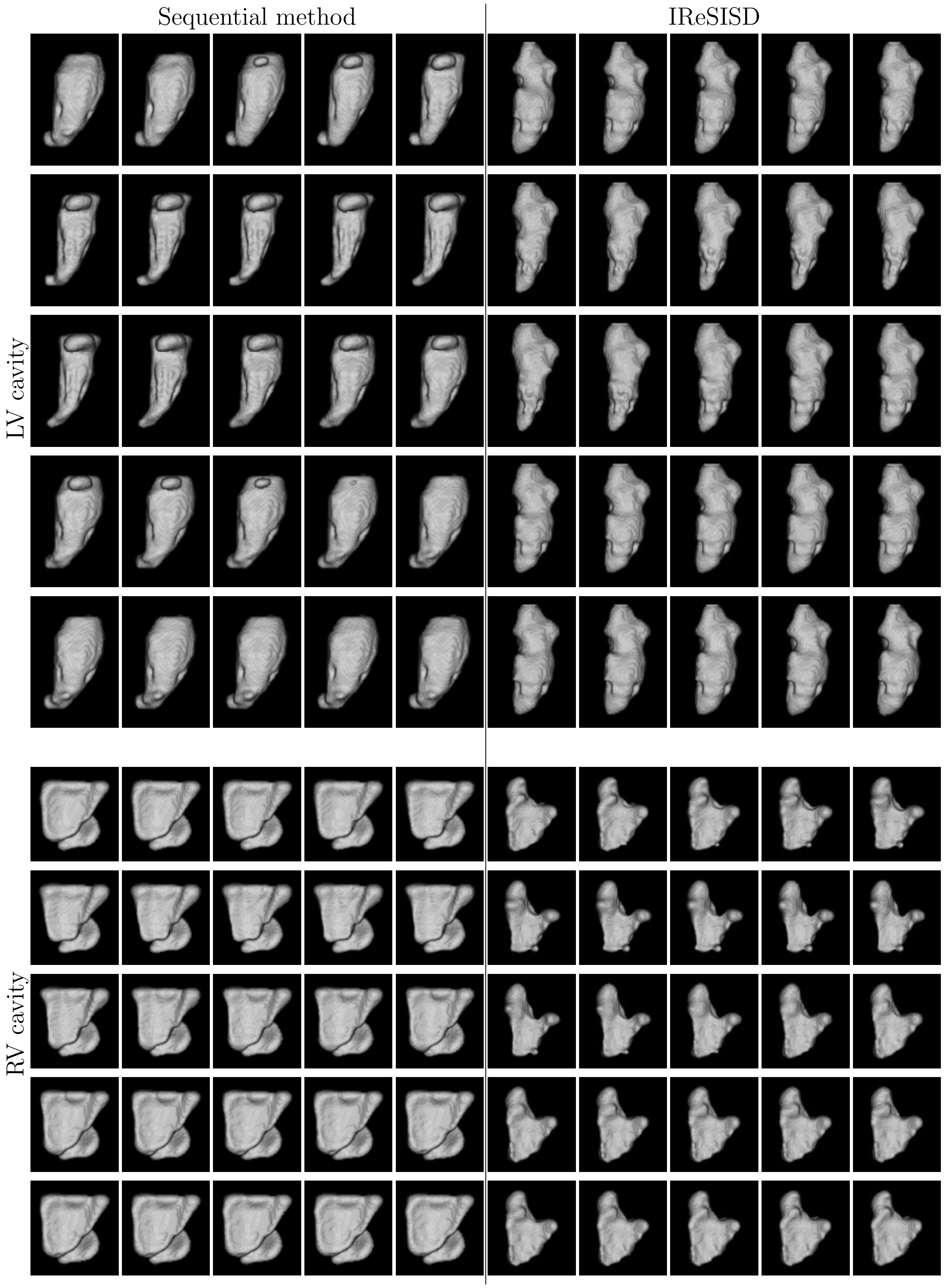


Figure 6.3: *4D modelling of the LV and RV cavities – dataset 2, all 25 time-frames.*

Chapter 7

Conclusion

The aim of this work was to develop a method for modelling 3D and 4D objects from sparse and misaligned data. Such a method needs to *register*, *segment*, and *interpolate* the data volume in order to recover a closed shape of the object. Several issues were addressed, notably:

- **the possibly large gaps and missing information** – resulting from a compromise being frequently sought by radiologists between detailed examinations and fast acquisitions.
- **the diverse appearance of medical images** – either due to physical limitations of the scanner, which may produce images having different gains and contrasts even within the same dataset, or due to different imaging modalities being used concurrently in the same examination. This makes the handling of the gaps in the data more difficult.
- **the great variety of possible spatial configurations of the data** – even though we focused on medical tomographic data, many medical acquisition protocols exist, which produce sets of slices that have arbitrary positions and orientations.
- **the wide variety of shapes being modelled** – since this work aimed at being general enough to be employed for any medical application, *e.g.* imaging any organ or tumour. In addition, such a high flexibility would allow imaging of objects that have significant shape variations, such as organs of patients suffering from deforming pathologies, or evolving tumours.

- **the difficulty of registration** – due to the limited intersections available in sparse datasets.
- **the interdependency of registration, segmentation, and interpolation** – which should be taken into account and exploited.

Interpolation of the object's shape allowed handling the gaps in the data. The issue of different gains and contrasts in the same dataset has been addressed by interpolating the object's shape through the segmenting contour or surface, rather than the image intensity, as is done traditionally in medical image analysis. Interpolation of the object's shape, regardless of the spatial configuration of the data, was provided by a new level set scheme which we introduced in Chapter 4. In this new scheme, the level set function was interpolated using RBFs. In addition to being able to handle different gains and contrasts, and sparse data of arbitrary spatial configurations, this new level set scheme was numerically stable and more robust to noise than a classical level set. It benefited from the flexibility of level set methods that enables it to model a great variety of shapes and to segment the data using any segmentation algorithm, the choice of which depends on the data.

We also presented a new registration method in Chapter 5, which was adapted to sparse data, and we integrated it into the above level set framework. This registration method was based on the mutual convergence of the segmenting contour and the images towards each other. In addition to being able to handle any spatial configuration and imaging modality, this registration method was robust to having only a small number of intersections between the different sequences to align, thanks to the use of segmentation results. Further, its robustness to local minima was increased by taking into account global information on the geometry of both the segmenting contour and the object. Its integration, together with segmentation and registration, into our level set framework allowed exploiting the interdependency of these three stages. The integrated framework could process sparse data made up of different sequences acquired in turn and independently, and which may suffer from misalignments.

Our fully integrated framework was evaluated on 3D and 4D medical tomographic images in Sections 4.5 and 5.5, using artificial data, MRI, and CT scans. We demonstrated the flexibility of our framework by modelling shapes of various topologies and complexities, using different segmentation algorithms that can exploit both image data and prior knowledge. Registration was assessed and compared against a popular NMI-based registration method [33]. Although, for relatively dense datasets and low misalignments, the method in [33] was slightly more accurate (but with both methods achieving sub-pixel and sub-degree accuracy), our proposed registration method proved to be more robust to just a few intersections between the slices which make up a dataset, and to local minima in case of significant misalignments. Interpolation was compared against a state-of-the-art image interpolation method [14] and obtained a more global interpolation of the shape of the object, especially in large gaps and where the object disappears between two slices. It could also handle all spatial configurations and different modalities simultaneously, while the image interpolation method of [14] was limited to parallel and equally spaced images having the same gain and contrast. However, not all spatial configurations obtained the same quality of interpolation, and the accuracy was reduced in cases where the planes of the image slices were oriented roughly tangentially to the surface of the object. Note that this is rarely the case in medical imaging, where the orientation of the slices are adapted to the morphology of the patient by the radiologist. Finally, the full integration of all three processes into the same level set framework increased their respective robustness and accuracy, and produced very detailed shapes.

Note however that this full integration also implies that the success of all three processes depends strongly on the robustness of the chosen segmentation algorithm, as was illustrated by our application of IReSISD to the modelling of the heart from cine MRI in Chapter 6. Since the proposed framework is general enough to employ any segmentation method, it is possible to choose the most well adapted algorithm for a given application and type of data. Segmentation and modelling may also be improved continuously by integrating the latest segmentation algorithms into the framework as and when they are developed.

7.1 Contributions

Our contributions are summarised below.

- We introduced the new concept of integrating segmentation and interpolation by interpolating a level set segmenting surface.
- We presented a new level set scheme based on the interpolation of its implicit function by RBFs.
- We proposed a novel level set based registration method for sparse and non-overlapping data.

- We integrated the three processes of registration, segmentation, and interpolation in a level set framework. To the best of our knowledge, this is the first time that all three processes are integrated in a level set framework.
- We introduced a general modelling framework that can handle arbitrary imaging modality, spatial configuration of the data, and shape of modelled object.

7.2 Future Work

The careful choice of a segmentation algorithm that is adapted to the data, *i.e.* the modality and imaged organ, is highly recommended for any practical usage of IReSISD. The design of robust segmentation algorithms that are suited to specific data should be the subject of future work. Since IReSISD can use any segmentation algorithm, extra robustness may be obtained by combining several segmentation criteria, such as edges, intensity, and texture, in order to exploit all available information. The use of prior knowledge may also be investigated to produce more robust segmentations.

A possible extension of this work is the application of IReSISD to different types of data, such as 3D point clouds. Indeed, as highlighted before, IReSISD is a general framework and was designed to be used on data having any spatial configuration, through the choice of an adapted segmentation algorithm for the computation of the contour speed S. In Appendix A, we propose a segmentation algorithm for S in the case of 3D point clouds with line-of-sight information, and we present some preliminary results of the application of IReSISD to the modelling of objects from RGB-D data produced by the Kinect sensor. This application remains to be evaluated and compared against state-of-the-art methods. Other segmentation algorithms, that could be more robust and accurate, may also be investigated.

References

[1] O. Bernard, D. Friboulet, P. Thevenaz, and M. Unser. Variational B-spline level-set: A linear filtering approach for fast deformable model evolution. *IEEE Trans. Image Processing*, 18:1179–1191, 2009.

[2] A. Boulch and R. Marlet. Fast and robust normal estimation for point clouds with sharp features. *Computer Graphics Forum*, 31:1765–1774, 2012.

[3] O. Camara, E. Oubel, G. Piella, S. Balocco, M. De Craene, and A. Frangi. Multi-sequence registration of cine, tagged and delay-enhancement MRI with shift correction and steerable pyramid-based detagging. In *Proc. Int. Conf. Functional Imaging and Modeling of the Heart*, pages 330–338, 2009.

[4] cardiachealth.org. Heart diagram, 2013. URL `http://www.cardiachealth.org/heart-disease-diagnosis/cardiac-mri`.

[5] J. Carr, R. Beatson, J. Cherrie, T. Mitchell, W. Fright, B. McCallum, and T. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *Proc. SIGGRAPH Int. Conf. Computer Graphics and Interactive Techniques*, pages 67–76, 2001.

[6] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Int. Journal of Computer Vision*, 22:61–79, 1997.

[7] T. Chan and L. Vese. Active contours without edges. *IEEE Trans. Image Processing*, 10:266–277, 2001.

[8] A. Chandler, R. Pinder, T. Netsch, J. Schnabel, D. Hawkes, D. Hill, and R. Razavi. Correction of misaligned slices in multi-slice MR cardiac examinations by using slice-to-volume registration. In *Proc. IEEE Int. Symposium on Biomedical Imaging: Nano to Macro*, pages 474–477, 2006.

[9] D. Chopp and J. Sethian. Flow under curvature: Singularity formation, minimal surfaces, and geodesics. *Experimental Mathematics*, 2(4):235–255, 1993.

[10] D. Chopp and J. Sethian. Motion by intrinsic Laplacian of curvature. *Interfaces and Free Boundaries*, 1(1):107–123, 1999.

[11] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal. Automated multimodality image registration using information theory. In *Proc. Int. Conf. Information Processing in Medical Imaging*, pages 263–274, 1995.

[12] A. Collignon, D. Vandermeulen, P. Suetens, and G. Marchal. 3D multi-modality medical image registration using feature space clustering. In *Proc. Int. Conf. Computer Vision, Virtual Reality and Robotics in Medicine*, pages 195–204, 1995.

[13] L. Cordero-Grande, G. Vegas-Sanchez-Ferrero, P. Casaseca-de-la Higuera, J. San-Roman-Calvar, A. Revilla-Orodea, M. Martin-Fernandez, and C. Alberola-Lopez. Unsupervised 4D myocardium segmentation with Markov random field based deformable model. *Medical Image Analysis*, 15:283–301, 2011.

[14] L. Cordero-Grande, G. Vegas-Sanchez-Ferrero, P. Casaseca-de-la Higuera, and C. Alberola-Lopez. A Markov random field approach for topology-preserving registration: Application to object-based tomographic image interpolation. *IEEE Trans. Image Processing*, 21, 2012.

[15] D. Cremers. Nonlinear dynamical shape priors for level set segmentation. *Journal of Scientific Computing*, 35(2-3):132–143, 2008.

[16] B. Delhay, J. Lotjonen, P. Clarysse, T. Katila, and I.E. Magnin. A dynamic 3-D cardiac surface model from MR images. In *Proc. Computers in Cardiology*, pages 423–426, 2005.

[17] A. Elen, F. Maes, J. Bogaert, and P. Suetens. 3D breath hold related motion correction of multi-view 2D functional MRI. In *Proc. MICCAI Workshop on Analysis of Functional Medical Images*, pages 49–56, 2008.

[18] R. Franke. Scattered data interpolation: Tests of some methods. *Mathematics of Computation*, 38:181–200, 1982.

[19] Y. Gong and M. Brady. Texture-based simultaneous registration and segmentation of breast DCE-MRI. In *Proc. Int. Workshop on Digital Mammography*, pages 174–180, 2008.

[20] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73:325–348, 1987.

[21] G. Grevera and J. Udupa. Shape-based interpolation of multidimensional grey-level images. *IEEE Trans. Medical Imaging*, 15:881–892, 1996.

[22] J. Hammersley and P. Clifford. Markov fields on finite graphs and lattices. 1971.

[23] A. Histace, B. Matuszewski, and Y. Zhang. Segmentation of myocardial boundaries in tagged cardiac MRI using active contours: A gradient-based approach integrating texture analysis. *Int. Journal of Biomedical Imaging*, 2009.

[24] J. Koikkalainen, M. Pollari, J. Lotjonen, S. Kivisto, and K. Lauerma. Segmentation of cardiac structures simultaneously from short- and long-axis MR images. In *Proc. Int. Conf. Medical Image Computing and Computer-Assisted Intervention*, pages 427–434, 2004.

[25] C. Le Guyader and L. Vese. A combined segmentation and registration framework with a nonlinear elasticity smoother. *Computer Vision and Image Understanding*, 115(12):1689–1709, 2011.

[26] M. Leventon, W.E.L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 316–323, 2000.

[27] M. Levoy. The Digital Michelangelo Project, 2003. URL http://graphics.stanford.edu/projects/mich/.

[28] G. Li, S. Lu, Z. Min, J. Li, X. Xu, and E. Song. 3D prostate boundary reconstruction from 2D TRUS images. In *Proc. Int. Conf. Bioinformatics and Biomedical Engineering*, pages 940–943, 2007.

[29] T. Li and T. Denney. Registration of short and long-axis images in cine cardiac MRI. In *Proc. Annual Meeting of Int. Society for Magnetic Resonance in Medicine*, page 1208, 2006.

[30] Y. Li, C. Li, and Y.Z. Zhang. Novel level set model for the image segmentation based on parzen-window. In *Proc. IEEE Int. Conf. Computer Science and Automation Engineering*.

[31] L. Liu, C. Bajaj, J. O. Deasy, D. A. Low, and T. Ju. Surface reconstruction from non-parallel curve networks. *Computer Graphics Forum*, 27:155–163, 2008.

[32] J. Lotjonen, S. Kivisto, J. Koikkalainen, D. Smutek, and K. Lauerma. Statistical shape model of atria, ventricles and epicardium from short- and long-axis MR images. *Medical Image Analysis*, 8:371–386, 2004.

[33] J. Lotjonen, M. Pollari, S. Kivisto, and K. Lauerma. Correction of movement artifacts from 4-D cardiac short- and long-axis MR data. In *Proc. Int. Conf. Medical Image Computing and Computer-Assisted Intervention*, pages 405–412, 2004.

[34] J. Lotjonen, V. Jarvinen, B. Cheong, E. Wu, S. Kivisto, J. Koikkalainen, J. Mattila, H. Kervinen, R. Muthupillai, F. Sheehan, and K. Lauerma. Evaluation of cardiac biventricular segmentation from multiaxis MRI data: A multicenter study. *Journal of Magnetic Resonance Imaging*, 28:626–636, 2008.

[35] M. Lynch, O. Ghita, and P.F. Whelan. Left-ventricle myocardium segmentation using a coupled level-set with a priori knowledge. *Computerized Medical Imaging and Graphics*, 30:255–262, 2006.

[36] K. McLeish, D. Hill, D. Atkinson, J. Blackall, and R. Razavi. A study of the motion and deformation of the heart due to respiration. *IEEE Trans. Medical Imaging*, 21: 1142–1150, 2002.

[37] S. Morigi and F. Sgallari. 3D long bone reconstruction based on level sets. *Computerized Medical Imaging and Graphics*, 28:377–390, 2004.

[38] Olegalexandrov. Level set method, 2007. URL http://en.wikipedia.org/wiki/File:Level_set_method.jpg.

[39] S. Osher and J.A. Sethian. Front propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[40] Z. Pan, X. Yin, and G. Wu. Segmentation-based interpolation of 3D medical images. In *Proc. Int. Conf. Computational Science and Applications*, pages 731–740, 2004.

[41] A. Pihuit, O. Palombi, and M.-P. Cani. Reconstruction implicite de surfaces 3D à partir de régions 2D dans des plans parallèles. In *Proc. Journées de l'AFIG*, 2009.

[42] S. Rahman and S. Wesarg. Combining short-axis and long-axis cardiac MR images by applying a super-resolution reconstruction algorithm. In *Proc. SPIE Medical Imaging: Image Processing*, volume 7623, page 76230I, 2010.

[43] J.-K. Shen, B. Matuszewski, L.-K. Shark, and C. Moore. Deformable registration using spring mass system with cross-section correction. In *Proc. Int. Conf. Medical Information Visualisation - BioMedical Visualisation*, pages 9–14, 2007.

[44] W. Shi, J. Caballero, C. Ledig, X. Zhuang, W. Bai, K. Bhatia, A. M. S. Monteiro de Marvao, T. Dawes, D. O'Regan, and D. Rueckert. Cardiac image super-resolution with global correspondence using multi-atlas PatchMatch. In *Proc. Int. Conf. Medical Image Computing and Computer-Assisted Intervention*, pages 9–16, 2013.

[45] G. Slabaugh, Q. Dinh, and G. Unal. A variational approach to the evolution of radial basis functions for image segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[46] P. Slomka, D. Fieno, A. Ramesh, V. Goyal, H. Nishina, L. Thompson, R. Saouaf, D. Berman, and G. Germano. Patient motion correction for multiplanar, multi-breath-hold cardiac cine MR imaging. *Journal of Magnetic Resonance Imaging*, 25 (5):965–973, 2007.

[47] P. Smereka. Semi-implicit level set methods for curvature and surface diffusion motion. *Journal of Scientific Computing*, 19:439–456, 2003.

[48] R. Smolikova-Wachowiak, M.P. Wachowiak, A. Fenster, and M. Drangova. Registration of two-dimensional cardiac images to preprocedural three-dimensional images for interventional applications. *Journal of Magnetic Resonnance Imaging*, 22:219–228, 2005.

[49] C. Studholme, D. Hill, and D. Hawkes. Multiresolution voxel similarity measures for MR-PET registration. In *Proc. Int. Conf. Information Processing in Medical Imaging*, pages 287–298, 1995.

[50] C. Studholme, D. Hill, and D. Hawkes. An overlap invariant entropy measure of 3D medical image alignment. *Pattern Recognition*, 32:71–86, 1999.

[51] Y. Sun, M.-P. Jolly, and J. Moura. Integrated registration of dynamic renal perfusion MR images. In *Proc. IEEE Int. Symposium on Image Processing*, pages 1923–1926, 2004.

[52] H. van Assen, M. Danilouchkine, M. Dirksen, J. Reiber, and B. Lelieveldt. A 3-D active shape model driven by fuzzy inference: Application to cardiac CT and MR. *IEEE Trans. Information Technology in Biomedicine*, 12:595–605, 2008.

[53] L. Vese and T. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *Int. Journal of Computer Vision*, 50(3):271–293, 2002.

[54] P. Viola and W. Wells. Alignment by maximisation of mutual information. In *Proc. Int. Conf. Computer Vision*, pages 15–23, 1995.

[55] S. Wang and M. Wang. Radial basis functions and level set method for structural topology optimization. *Int. Journal for Numerical Methods in Engineering*, 65: 2060–2090, 2006.

[56] S. Wang, K. Lim, B. Khoo, and M. Wang. An extended level set method for shape and topology optimization. *Journal of Computational Physics*, 221:395–421, 2007.

[57] X. Wang and M. Mirmehdi. Archive film restoration based on spatiotemporal random walks. In *Proc. European Conf. Computer Vision*, pages 478–491, 2010.

[58] R. Whitaker. A level-set approach to 3D reconstruction from range data. *Int. Journal of Computer Vision*, 29:203–231, 1998.

[59] J. Woo, E.Z. Murano, M. Stone, and J.L. Prince. Reconstruction of high-resolution tongue volumes from MRI. *IEEE Trans. Biomedical Engineering*, 59:3511–3524, 2012.

[60] C. Xiaohua, M. Brady, and D. Rueckert. Simultaneous segmentation and registration for medical image. In *Proc. Int. Conf. Medical Image Computing and Computer-Assisted Intervention*, pages 663–670, 2004.

[61] R. Yang, M. Mirmehdi, and X. Xie. A charged active contour based on electrostatics. In *Proc. Advanced Concepts for Intelligent Vision Systems Conf.*, pages 173–184, 2006.

[62] A. Yezzi, L. Zollei, and T. Kapur. A variational framework for integrating segmentation and registration through active contours. *Medical Image Analysis*, 7:171–185, 2003.

[63] X. Yuan and X. Yuan. Fusion of multi-planar images for improved three-dimensional object reconstruction. *Computerized Medical Imaging and Graphics*, 35:373–382, 2011.

[64] S. Zambal, J. Hladuvka, and K. Buhler. Improving segmentation of the left ventricle using a two-component statistical model. In *Proc. Int. Conf. Medical Image Computing and Computer-Assisted Intervention*, pages 151–158, 2006.

[65] S. Zambal, J. Hladuvka, K. Buhler, and A. Neubauer. A fully automatic system for segmentation and analysis of the left and right ventricles of the heart using a bi-temporal two-component model. In *Proc. Computer Assisted Radiology and Surgery Conf.*, pages 93–94, 2007.

[66] Y. Zhang and B. Matuszewski. Multiphase active contour segmentation constrained by evolving medial axes. In *Proc. IEEE Int. Conf. Image Processing*, pages 2993–2996, 2009.

[67] Y. Zhang, B. Matuszewski, A. Histace, and F. Precioso. Statistical shape model of Legendre moments with active contour evolution for shape detection and segmentation. In *Proc. Int. Conf. Computer Analysis of Images and Patterns*, volume 6854 of *Lecture Notes in Computer Science*, pages 51–58, 2011.

[68] H.-K. Zhao, S. Osher, B. Merriman, and M. Kang. Implicit and nonparametric shape reconstruction from unorganized data using a variational level set method. *Computer Vision and Image Understanding*, 80:295–314, 2000.

[69] ZooFari. Heart diagram, 2010. URL http://en.wikipedia.org/wiki/File:Heart_diagram-en.svg.

Appendix A

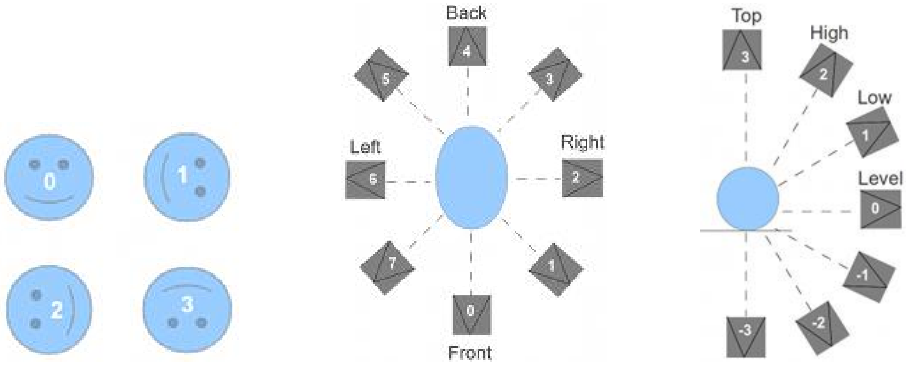
Application to Object Modelling from RGB-D Data

This thesis focused on the application of the proposed IReSISD framework to modelling from medical tomographic data. In this appendix, we demonstrate the possibility to also use IReSISD on RGB-D data. This is possible since, as highlighted at the end of Section 4.3.1, Algorithm 4.2 does not make any assumption on the type of data, and 3D point clouds may be processed by choosing an appropriate segmentation algorithm for the computation of S. We propose a segmentation algorithm adapted to 3D point clouds with line-of-sight information, and we show some early results of modelling from both artificial and Kinect RGB-D data.

Proposed Method

This section presents how we propose to apply IReSISD to object modelling from multiple sparse and misaligned 3D point clouds. Our experiments were carried out from a small number of RGB-D images, acquired to cover as many different sides of the object as possible, with some overlapping between them. Typically, we used one image viewed from each of the 8 cardinal directions, and occasionally one additional image viewed from the top. The data were processed off-line.

The first step is to define a rough initialisation for the registration. This is currently done by asking the user to select the type of orientation of the image (see Fig. 1 for the different orientation types available) and a small ROI inside the object to segment, which is drawn on the RGB image. This ROI is used to initialise both the registration, by aligning the centres of gravity of the 3D points inside the ROIs, and the segmentation model, using the RGB values of these points. So, this first step ensures that the 3D points belonging to the object are placed roughly in the same area of the volume and with relatively coherent orientations of the depth maps. In addition, the colours of the ROI points in the RGB images initialise a colour-based segmentation model which will be used to classify the 3D points between object and background.



(a) (b) (c)

Figure 1: *Available options for a rough initialisation of the orientation of the 3D point clouds by the user. (a) orientation of the object, (b) horizontal orientation of the camera, (c) vertical orientation of the camera.*

Secondly, a 3D level set surface is initialised as a small sphere at the centre of the ROIs. It will be evolved by the IReSISD framework at the same time as the point clouds, using a segmentation algorithm for the computation of S that will be described next. This evolution is done in a multi-resolution manner in order to speed up the computation and to reduce the number of possible local minima for the registration. The initialisation of each resolution is provided by the result of the previous coarser resolution.

During our tests, we found that registration using only the global variant of our registration method was more accurate for registering point clouds than the combination of global and local variants presented in Section 5.3.5. Therefore, all the results presented in this Appendix are obtained using the global variant of our registration method only.

Segmentation Algorithm

Our proposed segmentation algorithm for 3D point clouds with line-of-sight information is adapted from the method of Whitaker [58] who uses a level set method to model objects from dense and aligned point clouds with line-of-sight information. A similar approach can be used for sparse and misaligned data by integrating an extended version of this segmentation algorithm into our IReSISD framework.

Contrary to the scenario of [58] where all data points are assumed to belong to the modelled object, we propose to make use of RGB and depth information to classify them between the object to segment and the background. In our implementation, we chose two simple classification criteria: the first one is based on colour only and, similarly to [7] for grey-level images, it assumes that the object is a region of a constant colour in the RGB image. The second scheme is based on depth only and it assumes that the object is disjoint from its surrounding points – this is the case, for example, of a bust of a person if this person stands in the middle of an open space and the field-of-view of the camera does not include the lower part of the body which connects to the floor (see example in the right column of Fig. 5). Other, more elaborate and robust segmentation criteria may be used as well.

If a data point is classified as a background point, then all level set surface points which are on its associated line-of-sight between the camera and the data point are given a negative speed $S = -1$. Indeed, the object can not be in this empty area and, since the contour is initialised around or inside of the object, it should therefore shrink towards it. All level set surface points located on the line-of-sight behind the data point are also given a negative speed, but it is weighted by a confidence term which decreases away from the level set surface. This reflects the increasing probability of the background surface occluding the object. Thus, in this area, we use the speeds $S = -\mathfrak{C}$, where

$$\mathfrak{C} = \min\left(\frac{3}{1+\mathfrak{d}}, 1\right) \tag{1}$$

expresses our confidence in the choice of speed, with $\mathfrak{d}$ the unsigned distance to the data point in voxels. This confidence function is equal to 1 between 0 and 2 voxels, then it decreases to reach 0 at infinity.

If a data point is recognised as an object point, then the level set surface points located on its line-of-sight are given speeds S according to a modified version of [58]. This modification of [58] aims at obtaining a better registration with IReSISD, and it makes use of the robustness to noise of IReSISD. For simplicity, we consider a parametric coordinate τ along a line-of-sight, with $\tau = -1$ at the position of the camera and $\tau = 0$ at the data point. Lets τ_{ls} be the first intersection of the level set surface with the line-of-sight, *i.e.* the smaller τ at which the level set function ϕ becomes positive. If no level set surface intersects a line-of-sight, we set its corresponding τ_{ls} to infinity.

For $\tau < 0$, *i.e.* between the camera and the object, we want to push the level set surface towards the data point. As in [58], we must choose the speed carefully in order to take into account possible foldings of the surface. When $\tau > \tau_{ls}$, we set the speed S to $S_{\mathfrak{d}} \times \left(\overrightarrow{CP} \cdot \mathbf{N}(\tau)\right)$, with

$$S_{\mathfrak{d}} = \min\left(\frac{\mathfrak{d}(\tau)}{2}, 1\right) . \tag{2}$$

$\overrightarrow{CP}$ is the unit vector from the camera to the data point, $\mathbf{N}$ is the normal vector of the level set surface, and $\cdot$ is the scalar product. Thus, $\overrightarrow{CP} \cdot \mathbf{N}(\tau)$ is the projection of the level set normal on the line-of-sight and is in effect the cosine of the angle between these two unit vectors. If $\left(\overrightarrow{CP} \cdot \mathbf{N}(\tau)\right) < 0$, then the level set surface faces the camera, and $S < 0$ and shrinks the level set surface towards the data point. If $\left(\overrightarrow{CP} \cdot \mathbf{N}(\tau)\right) > 0$, then the level set surface faces away from the camera at that point τ. In such a case, [58] sets the data term of the speed to 0. However, we know that the point is between the camera and the object, so instead we push it with a positive speed towards the data point. Note that this speed will change once the level set surface point reaches the data point and the corresponding τ becomes positive. When $\tau \leq \tau_{ls}$, we use $S = -S_{\mathfrak{d}}$, which is negative. Although the value $S_{\mathfrak{d}} \times \left(\overrightarrow{CP} \cdot \mathbf{N}(\tau)\right)$ could also produce a correct segmentation, the use of the global variant of our registration method requires S to always be negative in the area where $\tau \leq \tau_{ls}$. Otherwise, after the level set contour converges to the boundaries of the object, some positive speeds due to $\left(\overrightarrow{CP} \cdot \mathbf{N}(\tau)\right)$ being positive close to surface foldings could create an artificial $\Omega_{\neq}$ area that could bias the registration.

For $\tau \geq 0$, *i.e.* beyond the data point, our confidence in the choice of the speed decreases away from the data point because we do not know how thick is the object and what is beyond this part of the surface of the object. Therefore, we use the confidence term $\mathfrak{C}$ defined in (1). Before the first level set surface point, *i.e.* where $\tau < \tau_{ls}$, we need to push the level set iso-contours towards the data point. Thus, S is set to $-S_{\mathfrak{d}} \times \left(\overrightarrow{CP} \cdot \mathbf{N}(\tau)\right)$. Otherwise, we give a positive speed to the contour. Where $\tau = \tau_{ls}$, *i.e.* on the first intersection of the level set surface with the line-of-sight, S is set to $S_{\mathfrak{d}}$, and where $\tau > \tau_{ls}$, S is equal to $S_{\mathfrak{d}} \times \mathfrak{C}$ inside the level set surface where $\phi \geq 0$, and to 0 where $\phi < 0$. Thus, contrary to [58], S is positive inside the level set surface even far away from the data point (although its value is quite low due to the confidence term) and when the level set function faces away from the camera. These positive speeds inside the level set surface avoid the creation of an artificial $\Omega_{\neq}$ area and, at the same time, provide a balloon force which tends to inflate the object and prevents the different point clouds to shrink to a central position during registration. Note that these positive speeds are weak enough to be overran by the negative speeds that point clouds with a different camera orientation may produce, so they should not hinder the registration.

Table 1 summarises the computation of S in our proposed segmentation algorithm.

The curvature term and the Gaussian model of noise in the depth measurements in [58] are not necessary with IReSISD since its RBF interpolated level set scheme can handle this noise and produces smooth surfaces, as discussed in Section 4.4.2. This will be demonstrated in the next section.

Preliminary Results

We apply IReSISD to both artificial and real RGB-D data. The artificial datasets are produced by rendering artificial scenes in RGB-D using Blender. They will be used to

Table 1: *Proposed segmentation algorithm for the computation of the contour speed S on 3D point clouds*

Classification of data point	**Sign of τ**	**Value of τ**	**Sign of ϕ**	**Contour speed S**
Background	Negative			-1
	Positive			$-\mathfrak{C}$
Object	Negative	$\tau \leq \tau_{ls}$		$-S_{\mathfrak{d}}$
		$\tau > \tau_{ls}$		$S_{\mathfrak{d}} \times \left(\overrightarrow{CP} \cdot \mathbf{N}(\tau)\right)$
	Positive	$\tau < \tau_{ls}$		$-S_{\mathfrak{d}} \times \left(\overrightarrow{CP} \cdot \mathbf{N}(\tau)\right)$
		$\tau = \tau_{ls}$	$\phi = 0$	$S_{\mathfrak{d}}$
		$\tau > \tau_{ls}$	Negative	0
			Positive	$S_{\mathfrak{d}} \times \mathfrak{C}$

evaluate the accuracy and robustness of the modelling of IReSISD, and to establish its limits of performance in the case of perfectly dense and aligned data. Our real datasets are acquired using a Kinect sensor.

Artificial Data

The models obtained by ISISD from our ideally dense and aligned artificial data are used as ground-truth for our on-going evaluations of the registration, segmentation, and interpolation of IReSISD. The ground-truth reconstructions are displayed in Fig. 2. Note that the flexible level set method used in ISISD could segment all the different topologies, and the smoothing property of our RBF interpolated level set scheme did not prevent it from capturing fine details and sharp corners.

Fig. 3 presents preliminary results on assessing the robustness of ISISD to noise in the depth measurements. They were obtained by adding an artificial Gaussian noise to the depth values of our artificial data. Overall, we find that our RBF interpolated level set scheme is as effective at handling noise in depth measurements as it was for noise in the images of tomographic datasets. In addition, similarly to tomographic data, we note that different levels of RBF flatness produce various degrees of robustness to noise.

A registration result is also presented in Fig. 4, where the different point clouds that make up the dataset were artificially translated and rotated by random amounts (Fig. 4a). In Fig. 4b, IReSISD achieved a very accurate registration of the point clouds and modelled the object with a similar precision to the ground-truth model of Fig. 2.

More tests and comparisons against state-of-the-art methods will be performed in the future in order to better evaluate all three stages of registration, segmentation, and interpolation of IReSISD on 3D point clouds.

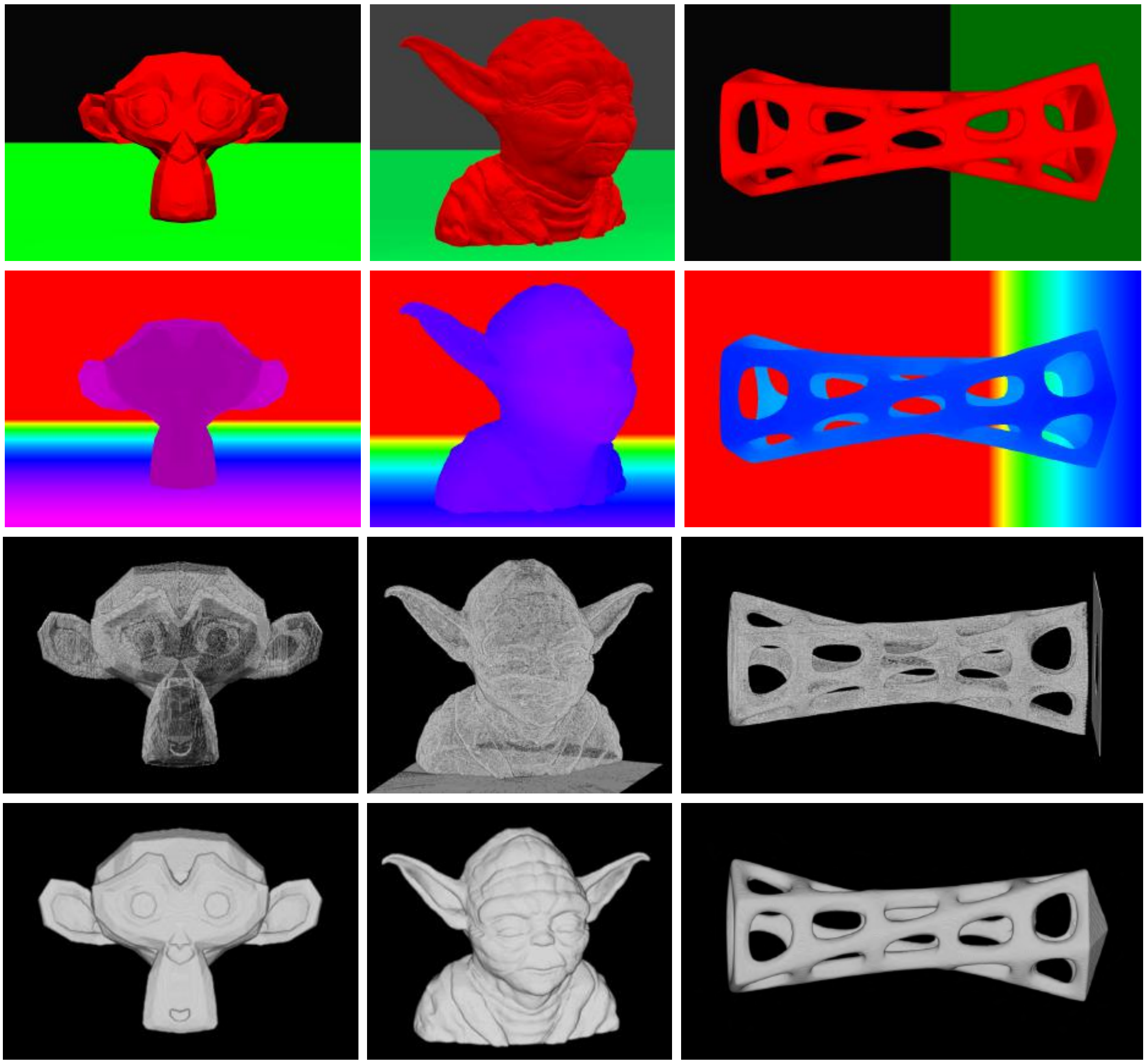


Figure 2: *Ground-truth modelling from dense and aligned artificial point clouds. First two rows: RGB and depth views of the artificial objects. Third row: 3D point clouds. Bottom row: 3D models obtained by ISISD.*

Kinect Data

We also present some preliminary results on real 3D point clouds produced with the Kinect sensor. In Fig. 5, a small plush toy of approximately 7x10x15 cm is modelled from 13 Kinect RGB-D images having various view points. It is segmented based on its blue colour. The bust of a person is also modelled from 14 Kinect RGB-D images, using depth information for the segmentation. The initial registration estimates of the point clouds are displayed in the third row of Fig. 5. The final registrations, in the fourth row of Fig. 5, are visually satisfactory. The two resulting models, in the last row of Fig. 5, are also reasonably accurate and detailed, in spite of the noise and holes in the depth measurements (see second row of Fig. 5).

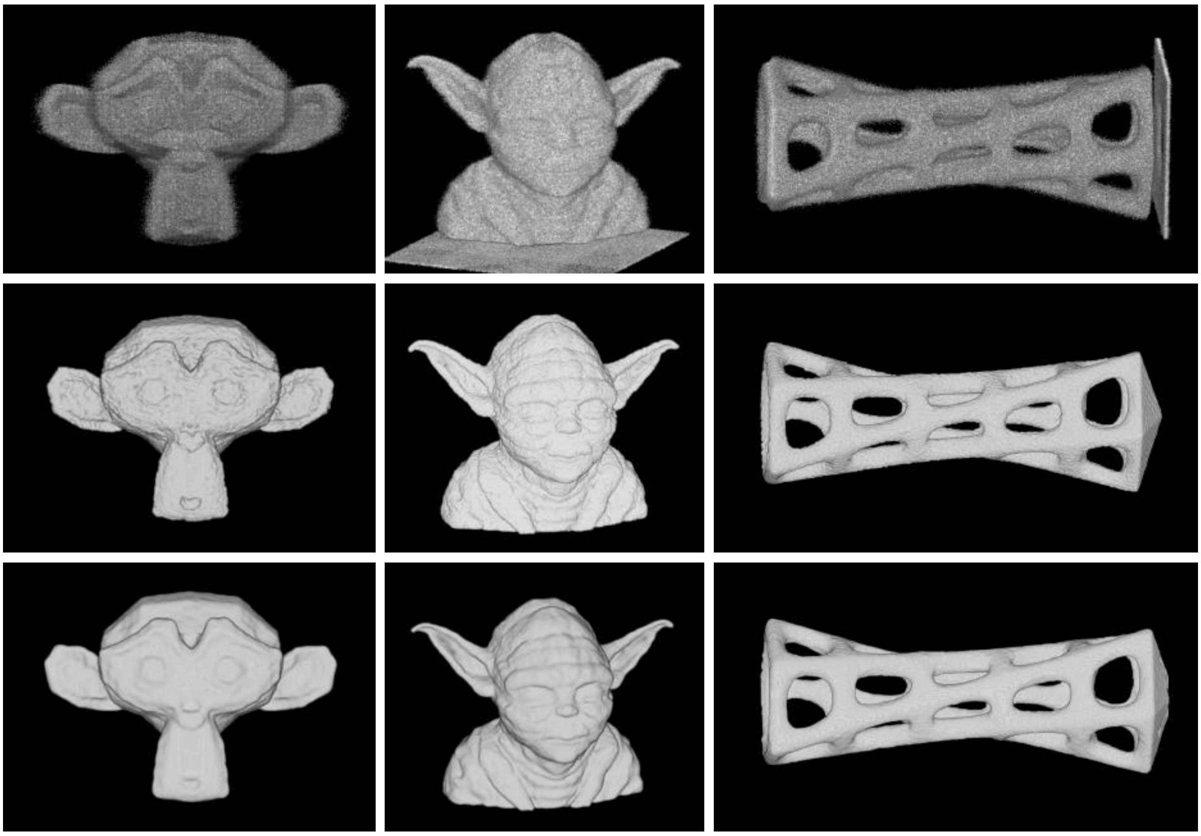
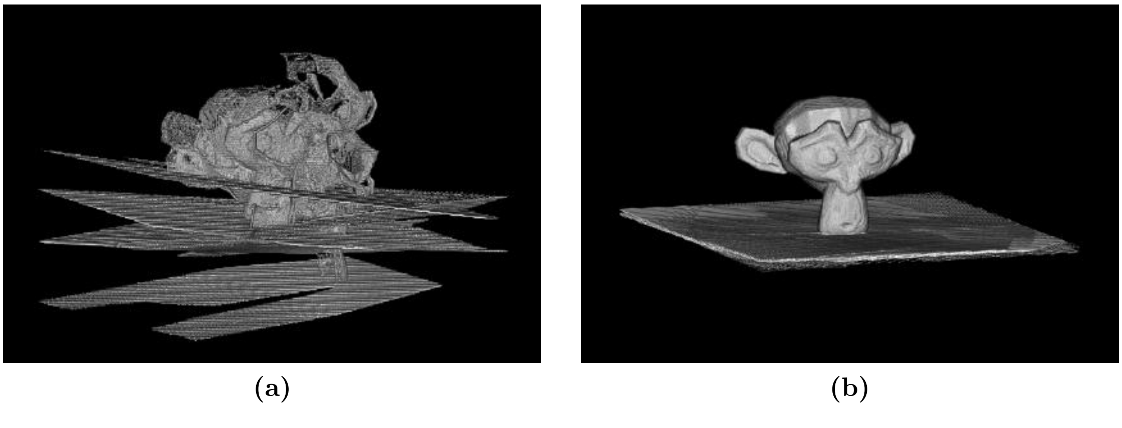


Figure 3: *Modelling from noisy artificial point clouds. Top row: noisy point clouds. Model by ISISD using: middle row: a sharp RBF, and bottom row: a flat RBF.*



(a) (b)

Figure 4: *Registration and modelling from misaligned artificial 3D point clouds. (a) initial misaligned point clouds, and (b) final re-aligned point clouds and superimposed object's model.*

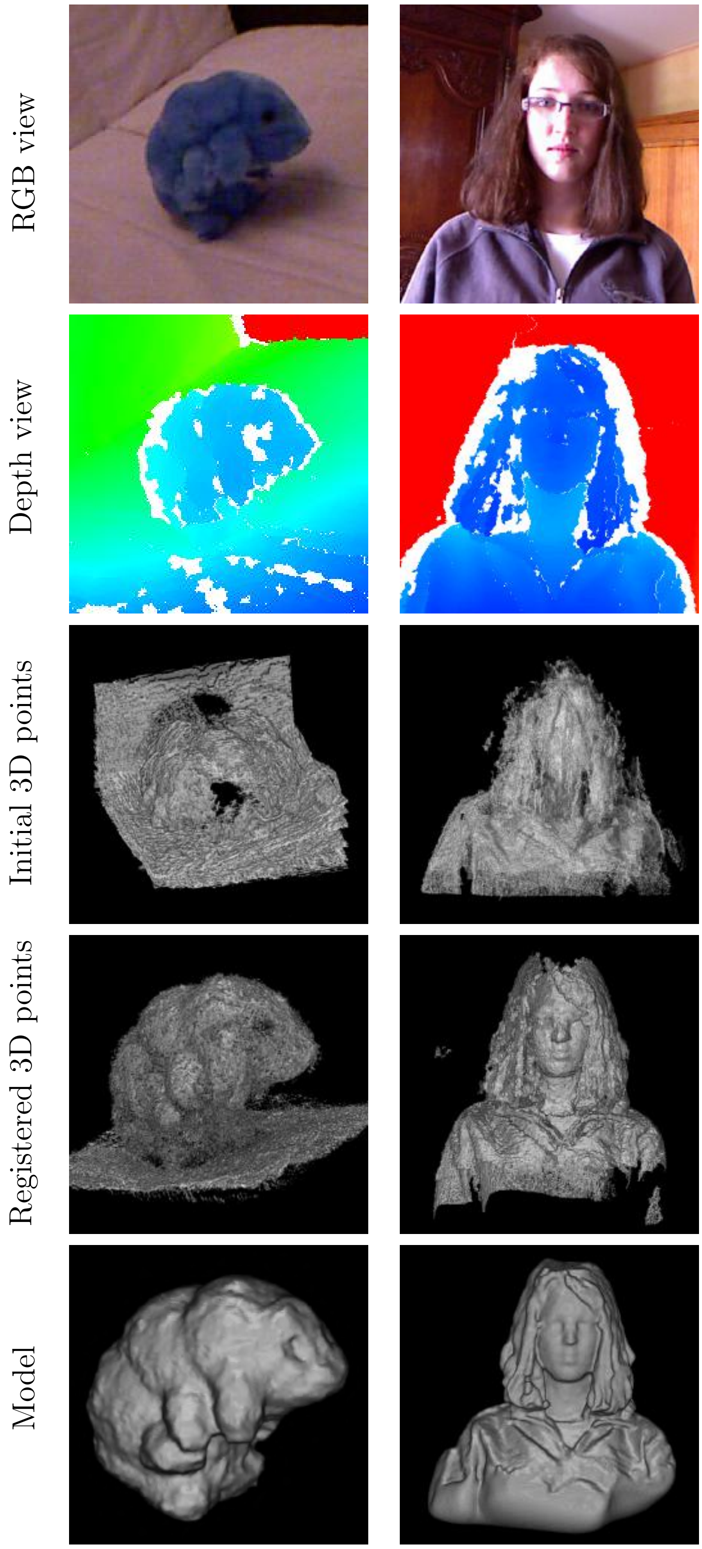


Figure 5: *Modelling from real Kinect RGB-D data. Left: blue plush toy, right: human bust.*

Appendix B

Application to Image Inpainting and Video Restoration

In this appendix, we propose a new application of the ISISD framework to image inpainting and video restoration. We will see that this problem is equivalent to filling holes in a surface. This hole filling stage can be achieved by the interpolation property of the ISISD framework.

Proposed Method

A grey-level image can be represented as a closed surface in a 3D space whose dimensions are the two spatial dimensions of the image plus a grey-level dimension[1]. Using such a representation, image inpainting and video restoration are effectively hole filling in this image derived surface. Note that, in this case, segmentation is known beforehand and is given by the location of the surface in the grey-level dimension where data is available, *i.e.* where the image does not require inpainting. Thus, the interpolation stage of ISISD may be used alone to fill holes in the surface.

Incomplete surfaces may be closed using ISISD by first initialising the level set contour on the surface where data is available and setting the level set function to 0 elsewhere. Then, Algorithm 4.2 may be applied until convergence, with S set to the signed distance to the surface where this information is available.

Attention should be given to preventing the surface from folding. This is necessary to

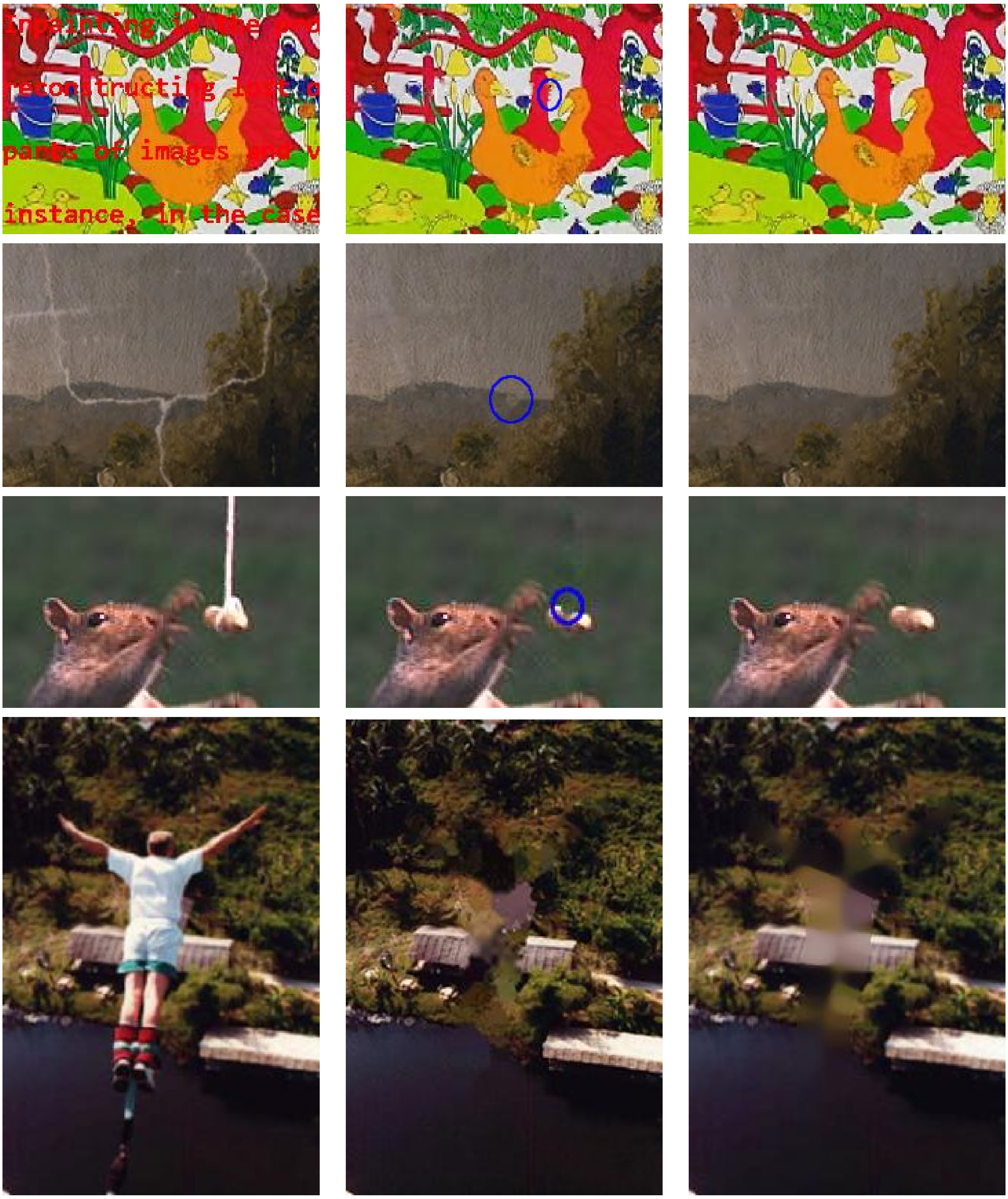
[1] The different channels of a colour image may be processed separately.

ensure that the surface crosses the “grey-level-axis” once per pixel position only. In our experiments, we found that computing S on the border of the gaps, or even everywhere in the 3D space, rather than only on the contour, ensured this condition was respected. This means that we do not use δ_ε in (4.7) and therefore the RBF is required to be sharp enough to prevent the background from having an overwhelming effect. This is fine since only sharp RBFs were found to produce clear object boundaries.

Results

Examples of image inpainting are provided in Fig. 6 and are compared against the Random Walks based Defect Removal (RWDR) method of Wang and Mirmehdi [57]. For untextured images or small holes, the proposed method better reconstructed edges than the RWDR method [57]. However, for large holes, *e.g.* in the bottom row of Fig. 6, only the slowly varying components of textured images are reconstructed while higher frequency texture components are missing. This may be addressed in a future work, possibly by also ”diffusing“ the higher frequency components of the hole border areas.

Similarly, videos may be restored in 4D, thus taking into account the neighbouring frames, as in Fig. 7. For videos with slowly varying scene, the proposed method produced clearer edges than RWDR when restoring large defects. However, for videos showing fast moving scenes such as the background in the bus sequence (bottom row of Fig. 7), the very sharp RBF used here prevented the method from binding correctly narrow objects from one frame to the other. This prevented the lamp-posts in the bottom row of Fig. 7 from being properly reconstructed. In future work, this problem may be addressed using, for example, registration.



(a) (b) (c)

Figure 6: *Application of ISISD to image inpainting. (a) original image, inpainted image by (b) RWDR [57], and (c) ISISD. The blue circles highlight some examples of edges being better preserved by ISISD than by RWDR [57].*

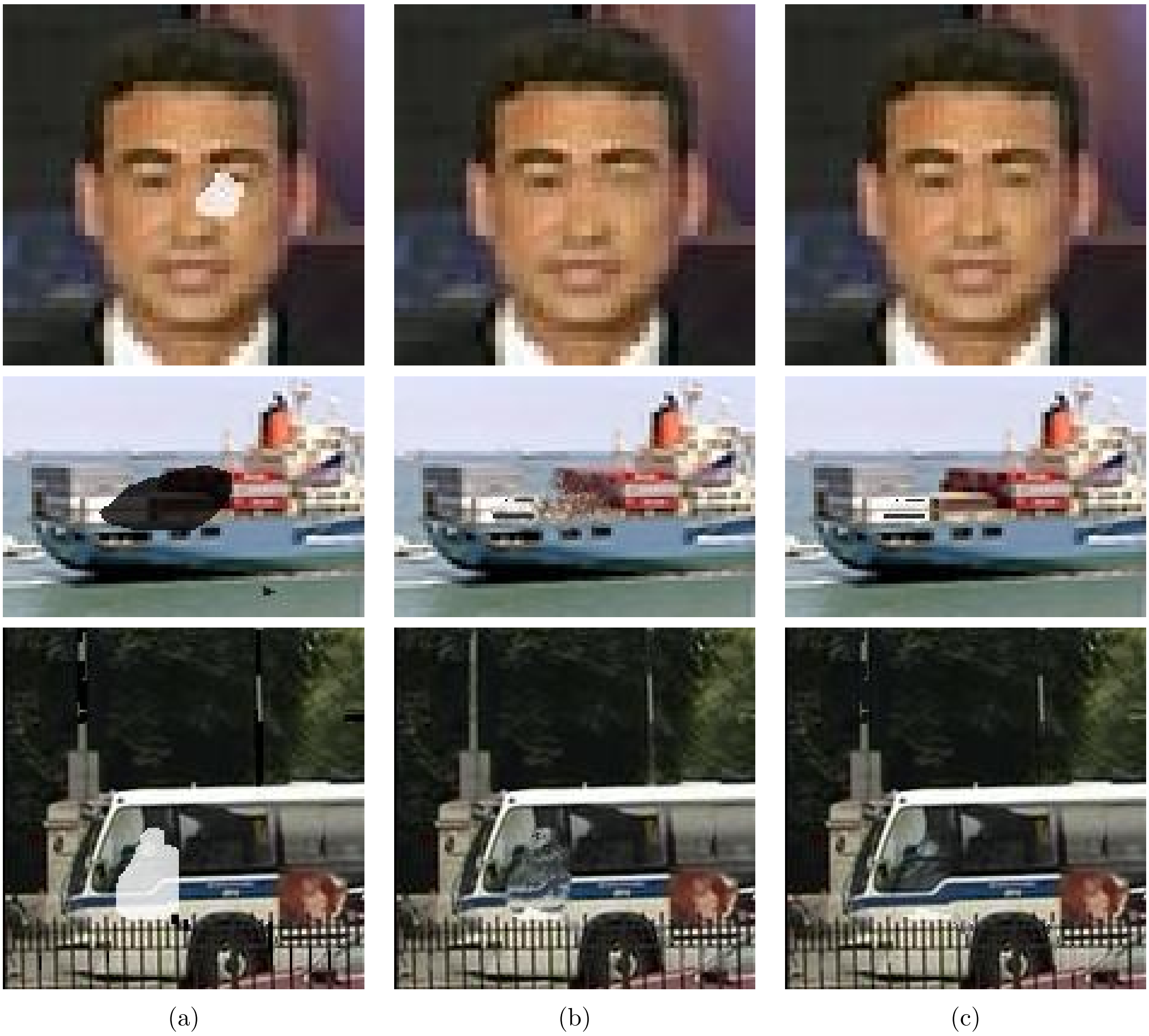


Figure 7: *Application of ISISD to video restoration. (a) degraded frame, restored frame by (b) RWDR [57], and (c) ISISD.*

Appendix C

Examples of Raw and Segmented Cardiac Cine MRI Datasets

This appendix presents examples of datasets and segmentations of cardiac cine MRI. Figs. 8 to 14 show the images of a typical dataset that is made of a stack of SA images and a few LA images, all having 25 time-frames. This type of dataset is presented with more details in Section 2.2.2. Figs. 15 to 21 and Figs. 22 to 25 present two datasets, zoomed around the heart, and with the segmentations produced by the sequential method and IReSISD superimposed in colour. See Section 6.2 for a description and a discussion of these segmentation results.

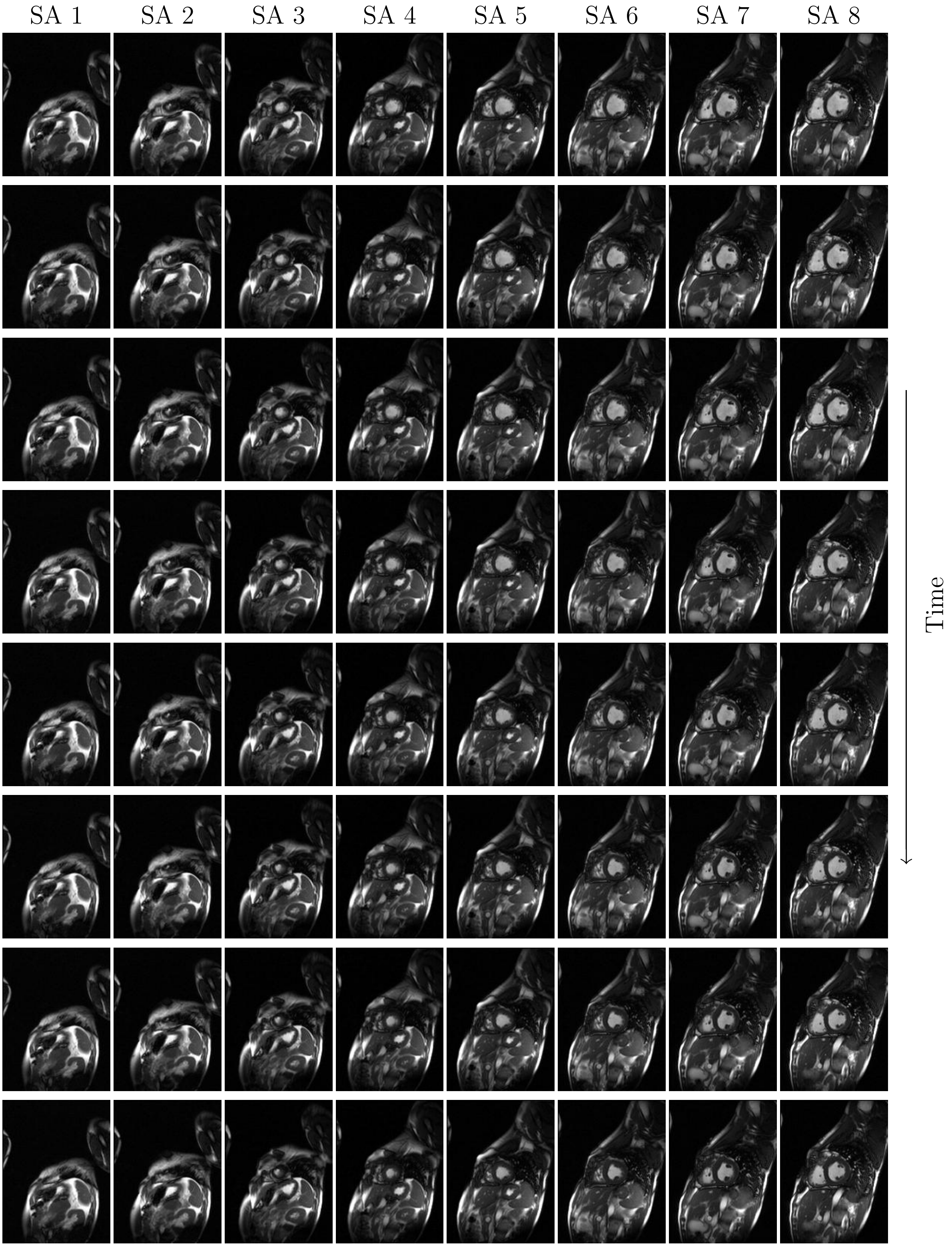


Figure 8: *Example of cardiac cine MRI made up of a stack of SA slices plus 3 LA slices – time-frames 1 to 8 (vertical axis) of slices 1 (apex) to 8 of the SA stack (horizontal axis).*

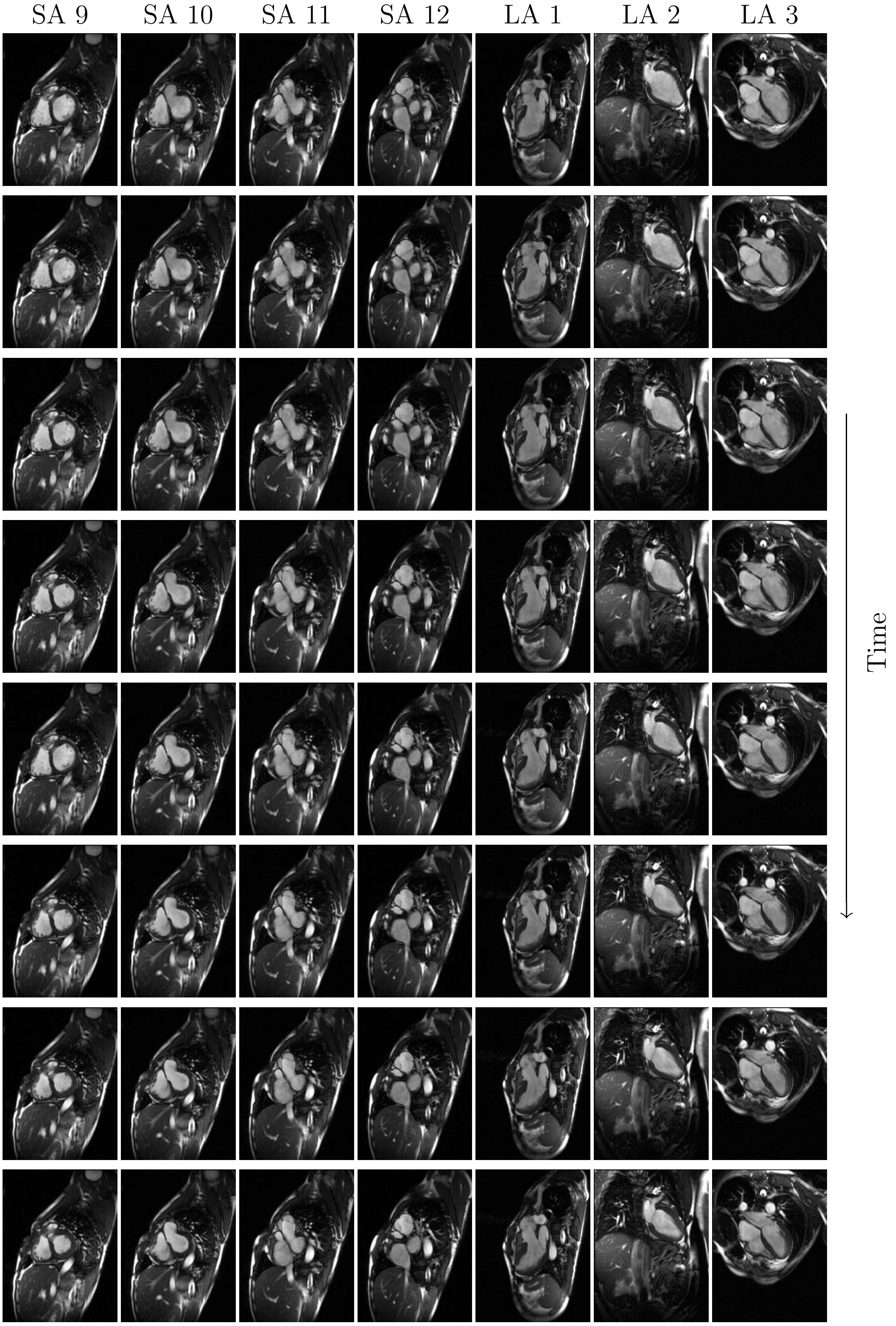


Figure 9: *Example of cardiac cine MRI made up of a stack of SA slices plus 3 LA slices (continued) – time-frames 1 to 8 (vertical axis) of slices 9 to 12 of the SA stack and of the LA slices (horizontal axis).*

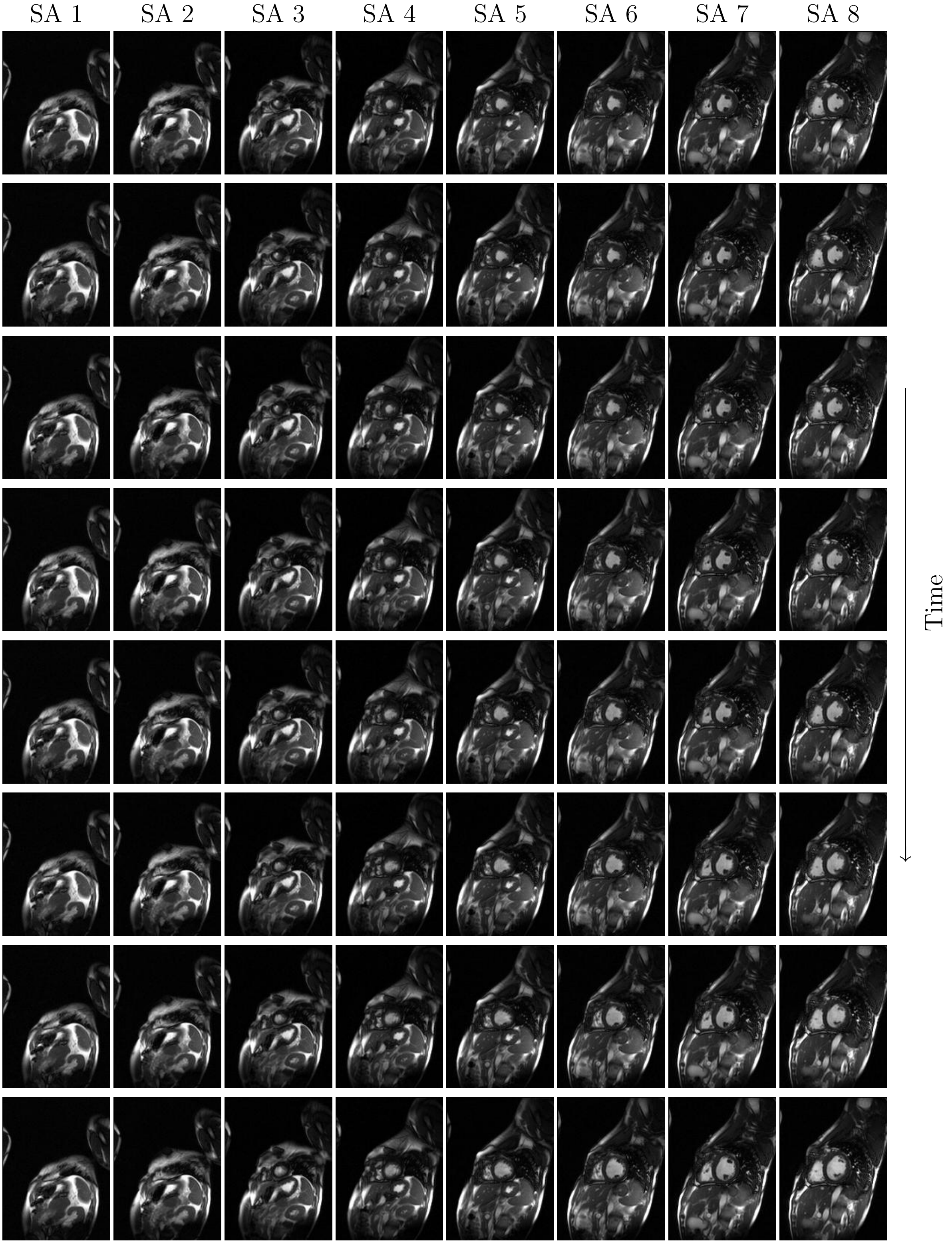


Figure 10: *Example of cardiac cine MRI made up of a stack of SA slices plus 3 LA slices (continued) – time-frames 9 to 16 (vertical axis) of slices 1 (apex) to 8 of the SA stack (horizontal axis).*

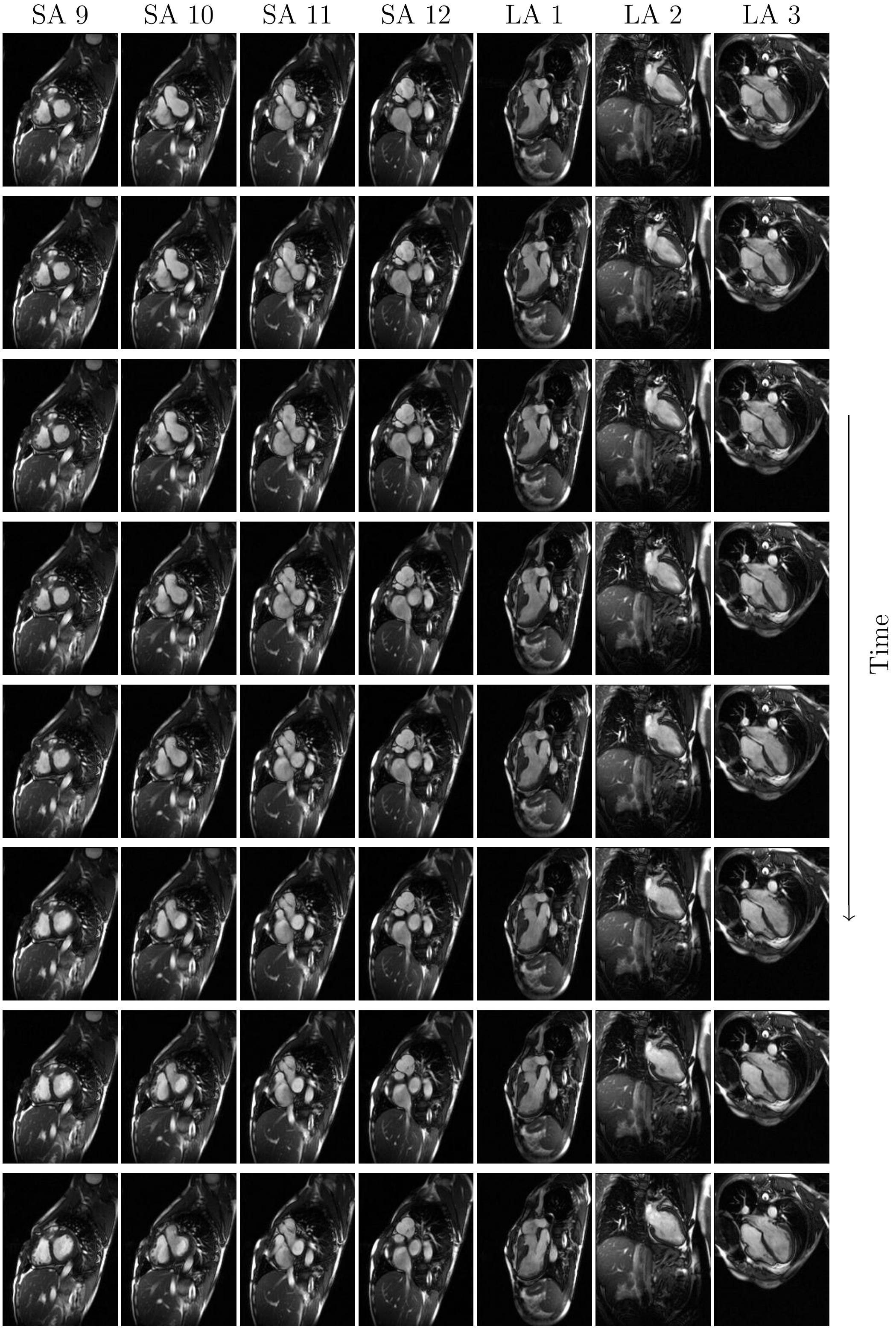


Figure 11: *Example of cardiac cine MRI made up of a stack of SA slices plus 3 LA slices (continued) – time-frames 9 to 16 (vertical axis) of slices 9 to 12 of the SA stack and of the LA slices (horizontal axis).*

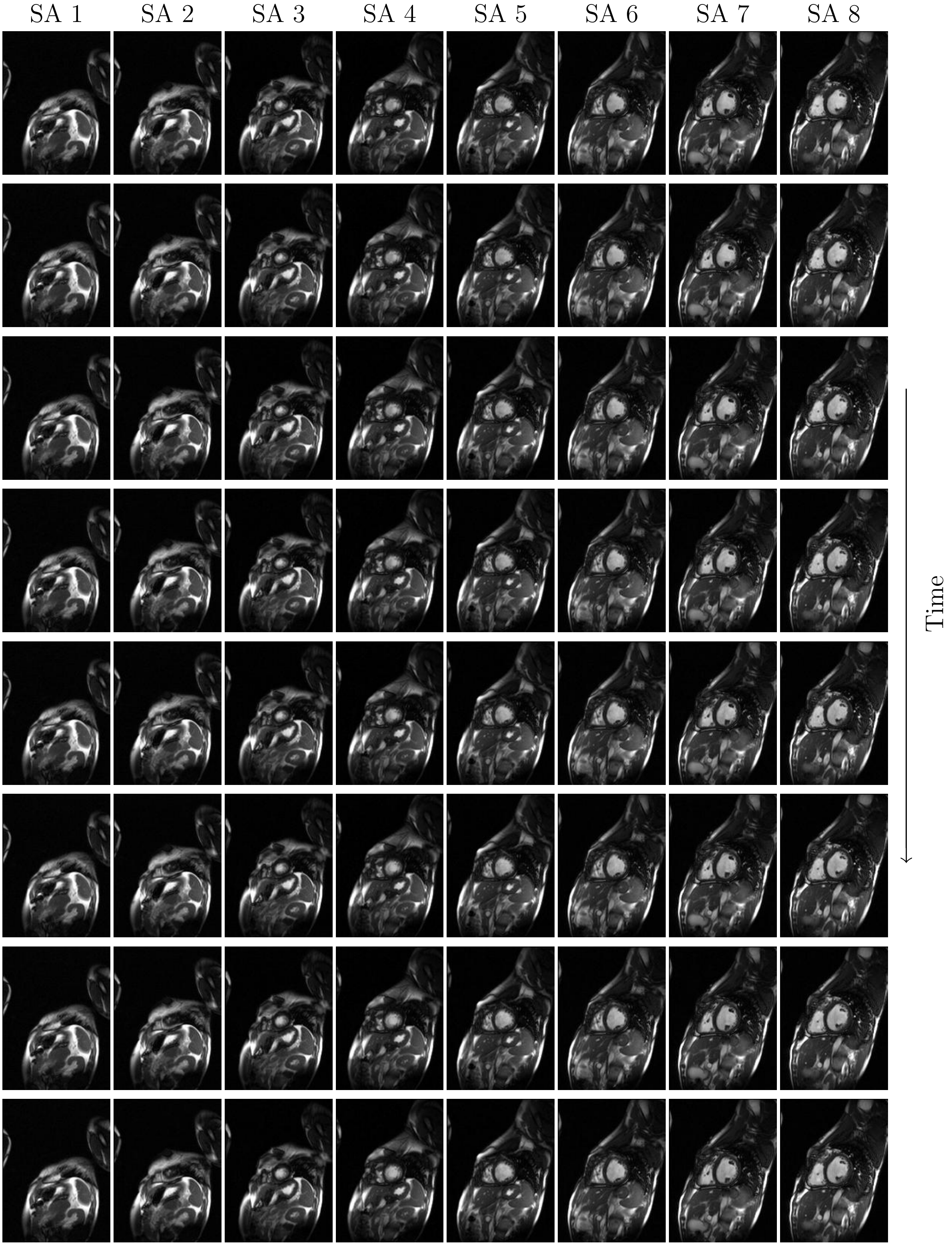


Figure 12: *Example of cardiac cine MRI made up of a stack of SA slices plus 3 LA slices (continued) – time-frames 17 to 24 (vertical axis) of slices 1 (apex) to 8 of the SA stack (horizontal axis).*

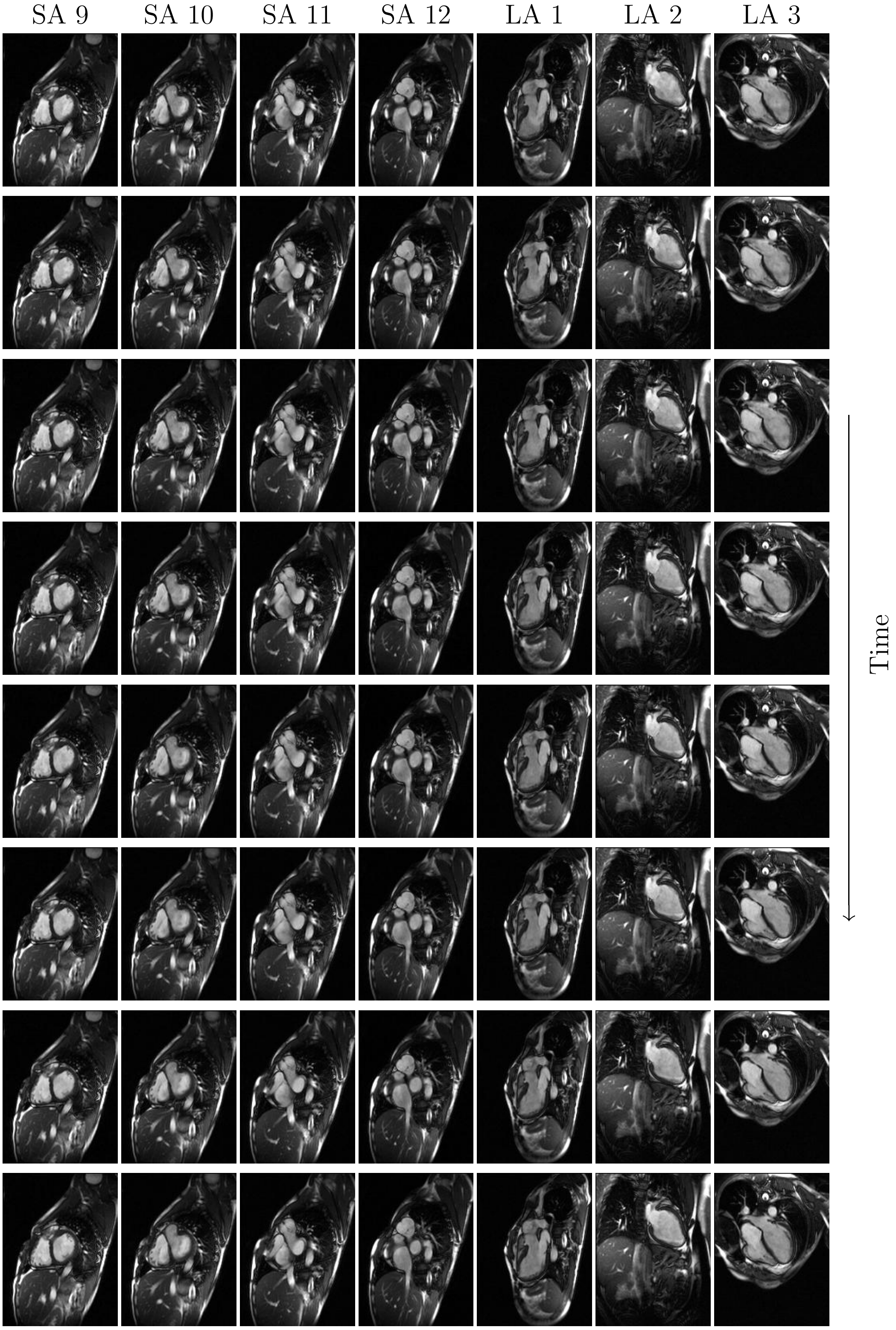


Figure 13: *Example of cardiac cine MRI made up of a stack of SA slices plus 3 LA slices (continued) – time-frames 17 to 24 (vertical axis) of slices 9 to 12 of the SA stack and of the LA slices (horizontal axis).*

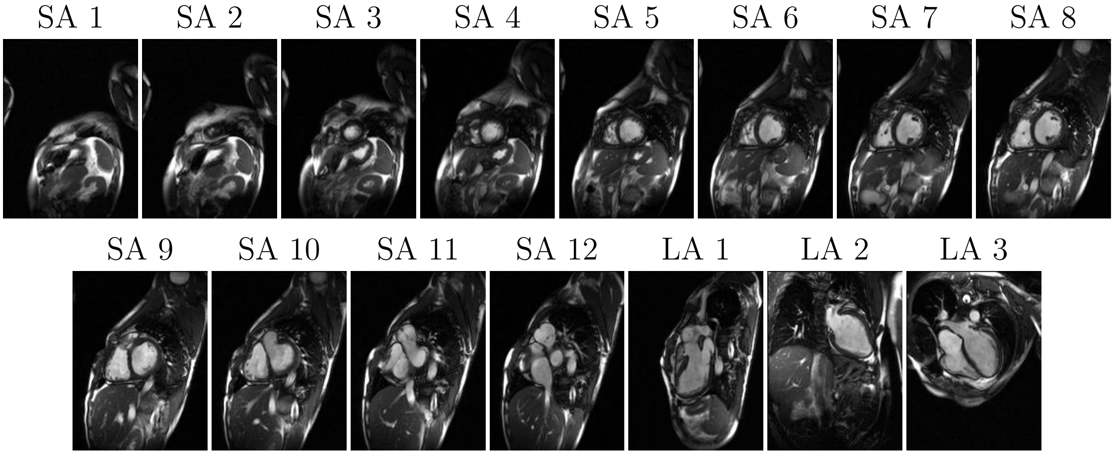


Figure 14: *Example of cardiac cine MRI made up of a stack of SA slices plus 3 LA slices (continued) – time-frame 25 of the SA stack and the LA slices.*

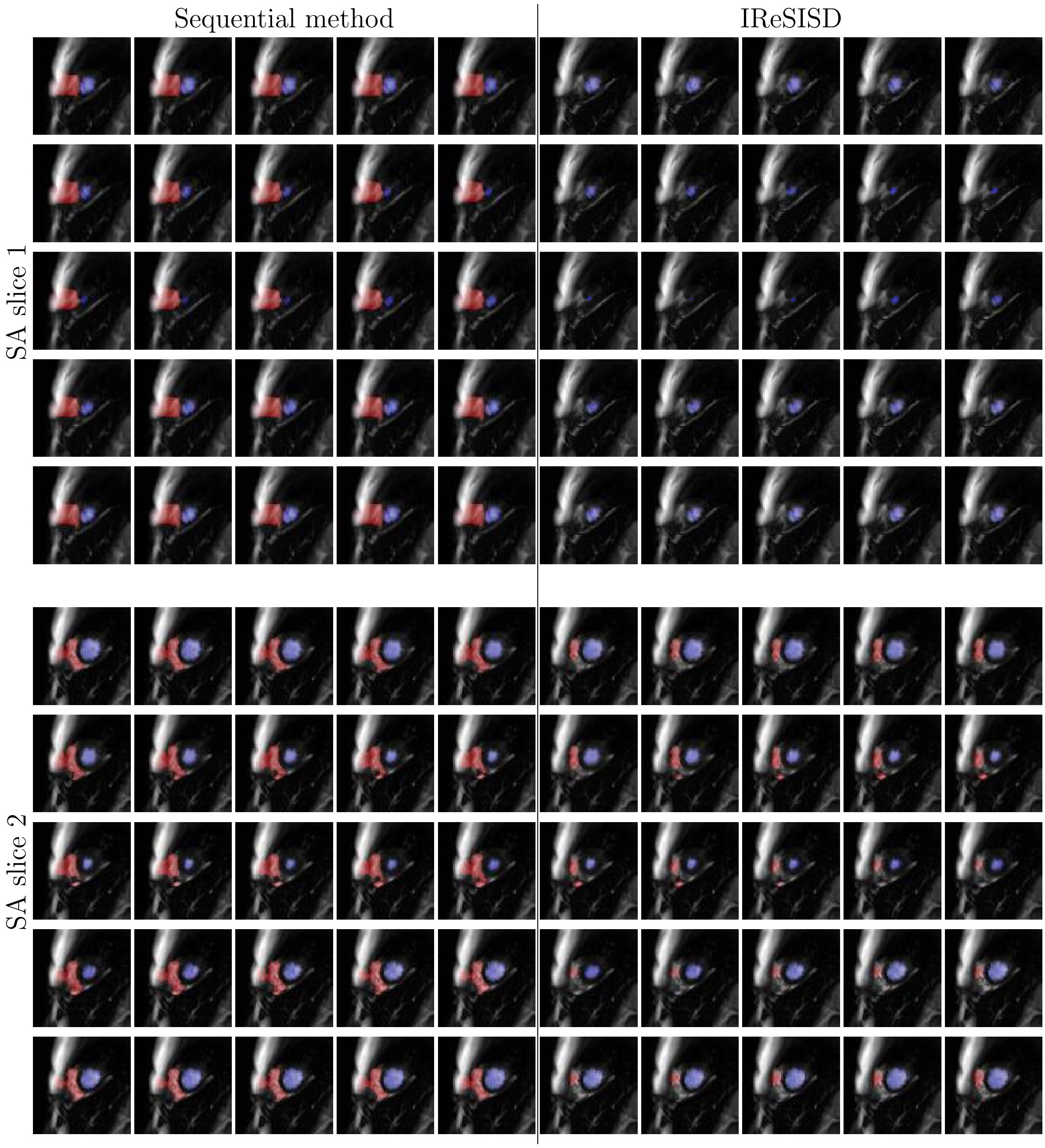


Figure 15: *Segmentation of cardiac cine MRI – dataset 1, all 25 time-frames of SA slices 1 and 2. Blue: LV cavity segmentation, red: RV cavity segmentation.*

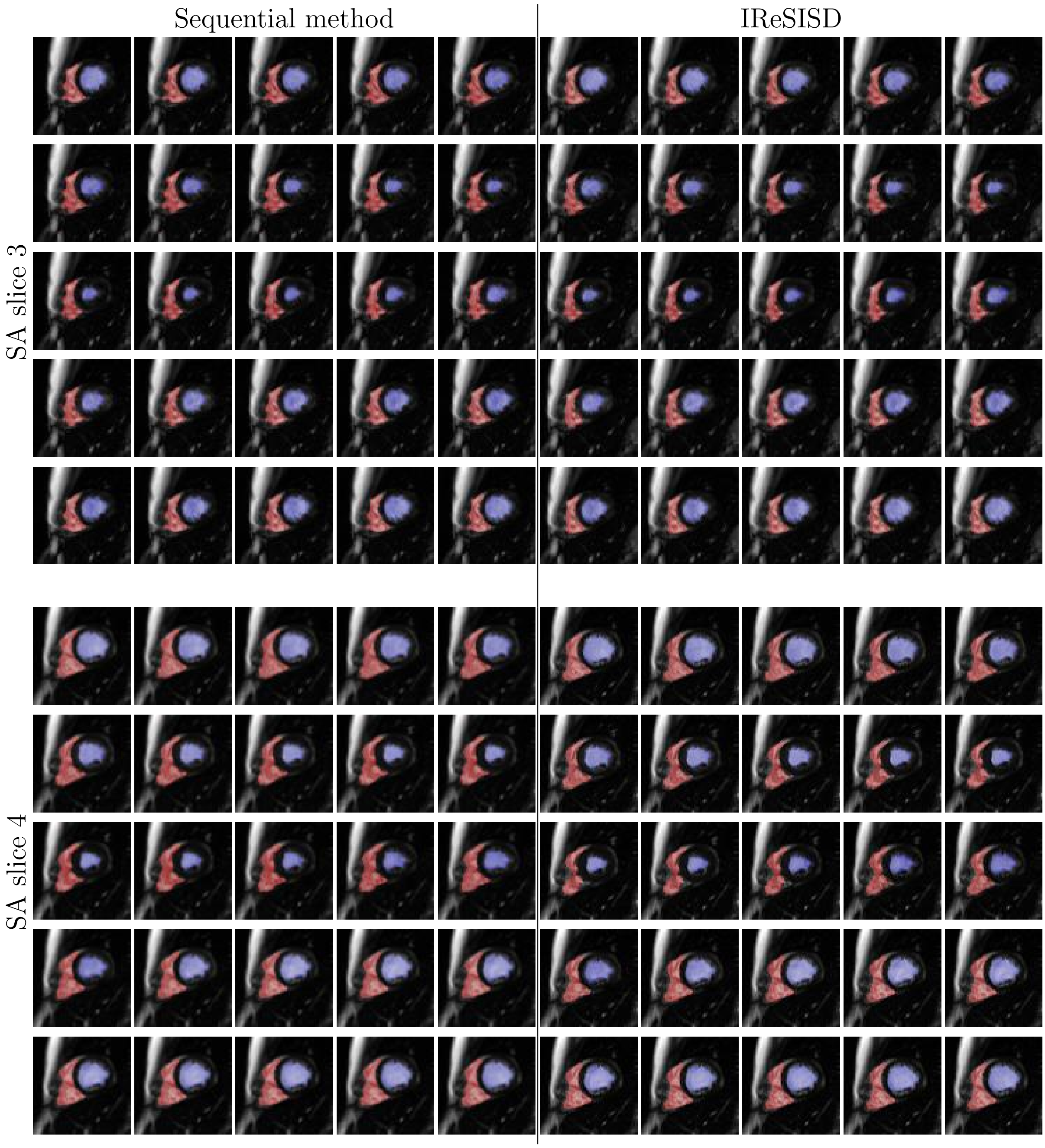


Figure 16: *Segmentation of cardiac cine MRI – dataset 1, all 25 time-frames of SA slices 3 and 4. Blue: LV cavity segmentation, red: RV cavity segmentation.*

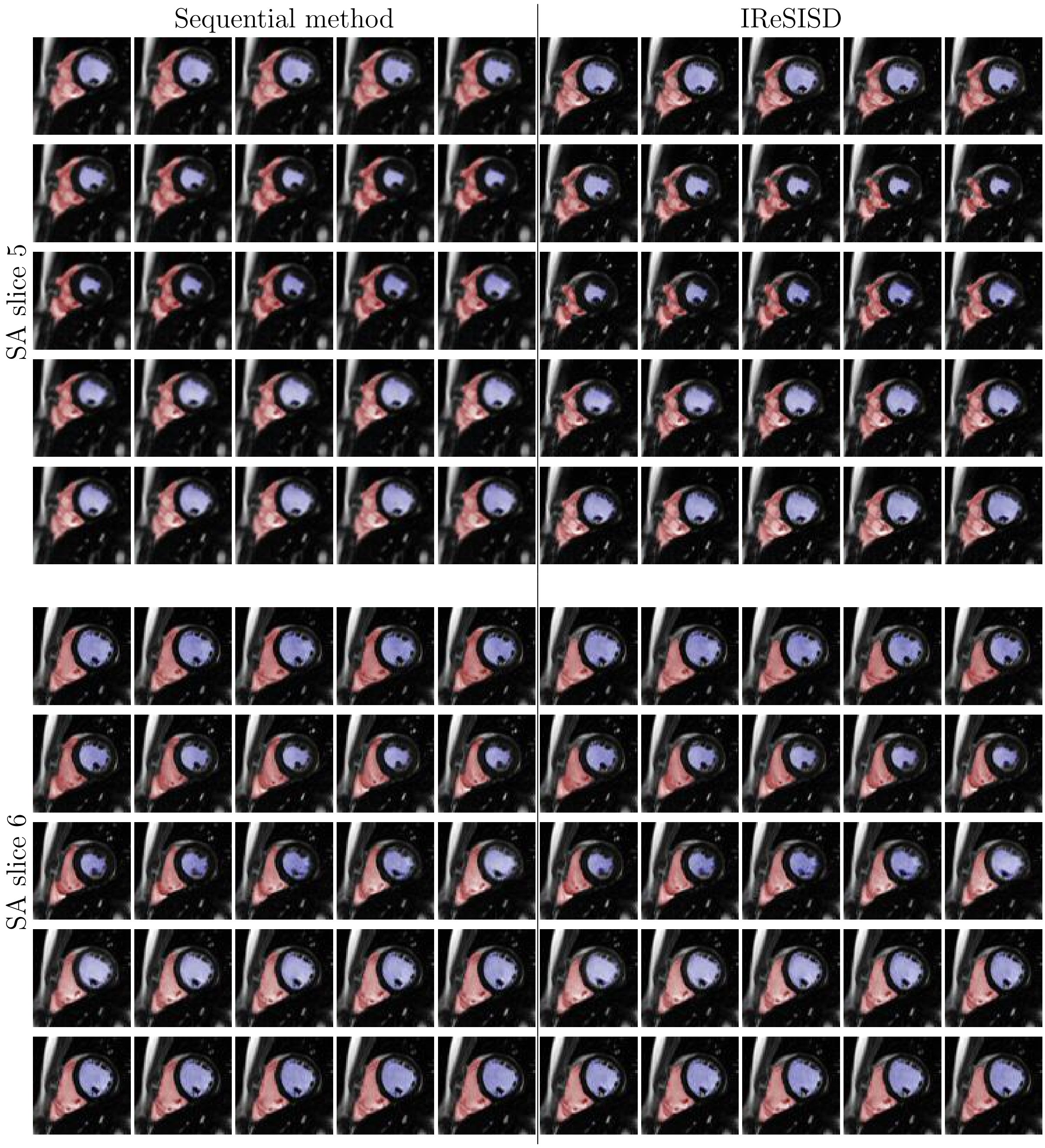


Figure 17: *Segmentation of cardiac cine MRI – dataset 1, all 25 time-frames of SA slices 5 and 6. Blue: LV cavity segmentation, red: RV cavity segmentation.*

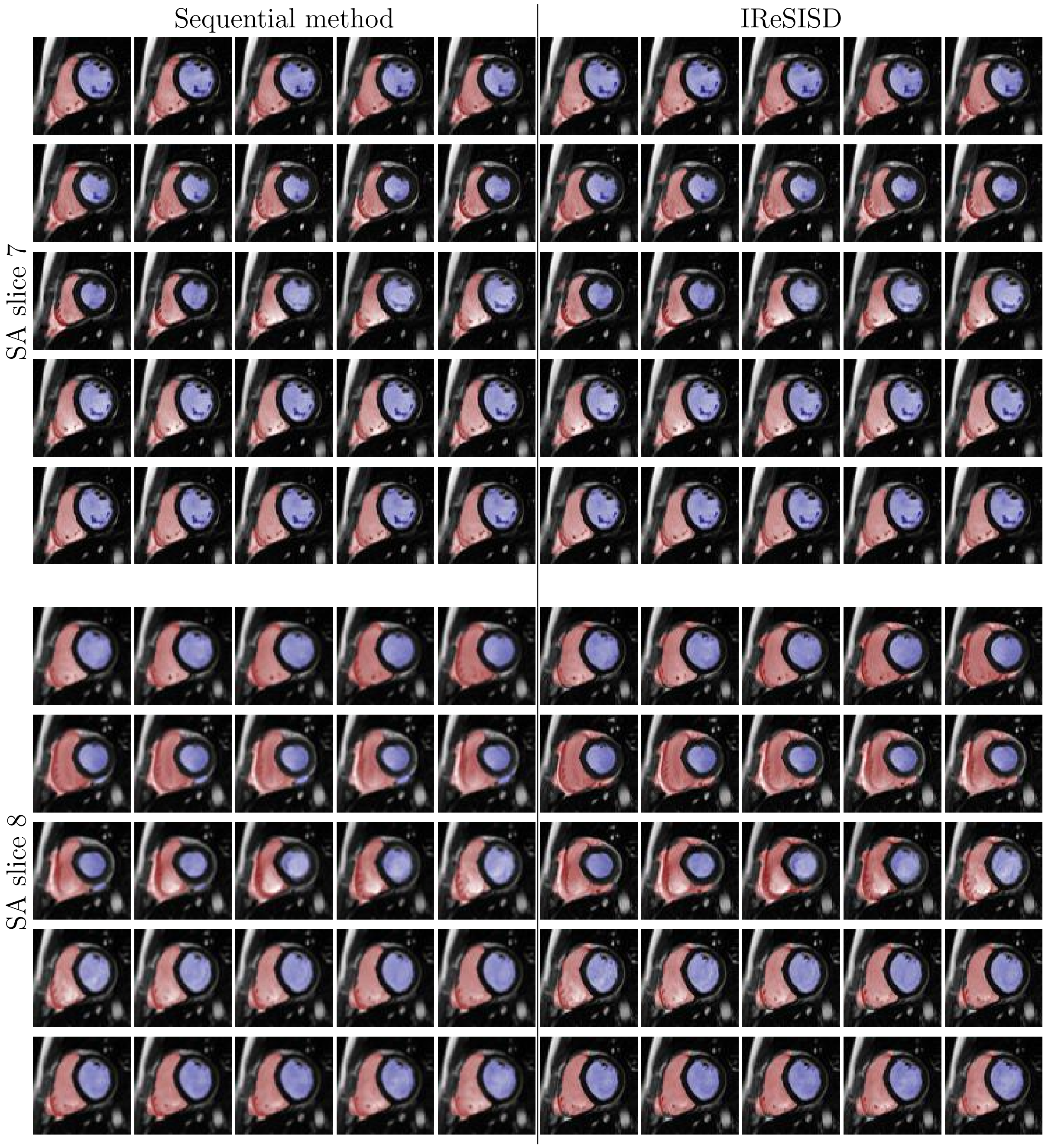


Figure 18: *Segmentation of cardiac cine MRI – dataset 1, all 25 time-frames of SA slices 7 and 8. Blue: LV cavity segmentation, red: RV cavity segmentation.*

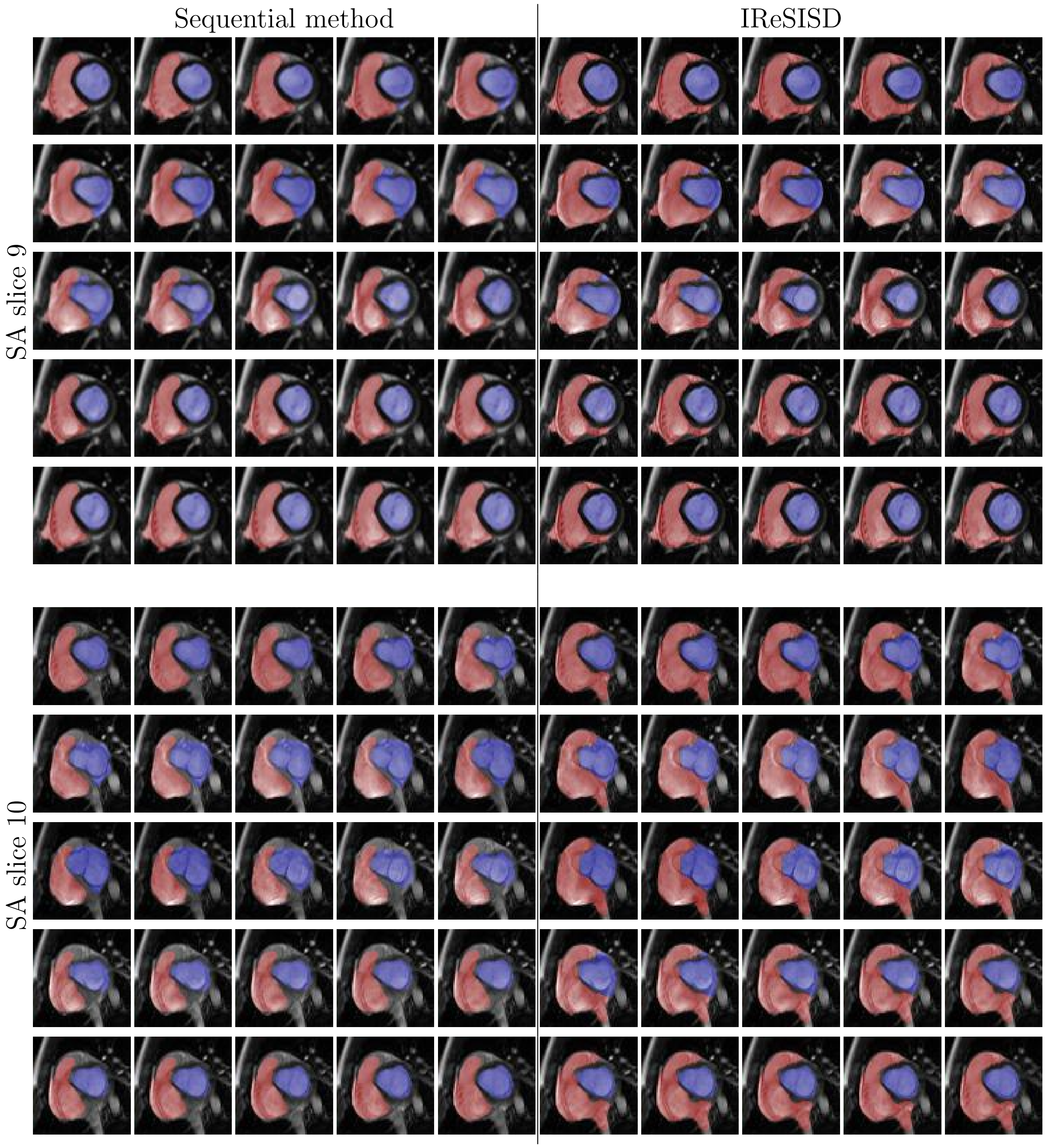


Figure 19: *Segmentation of cardiac cine MRI – dataset 1, all 25 time-frames of SA slices 9 and 10. Blue: LV cavity segmentation, red: RV cavity segmentation.*

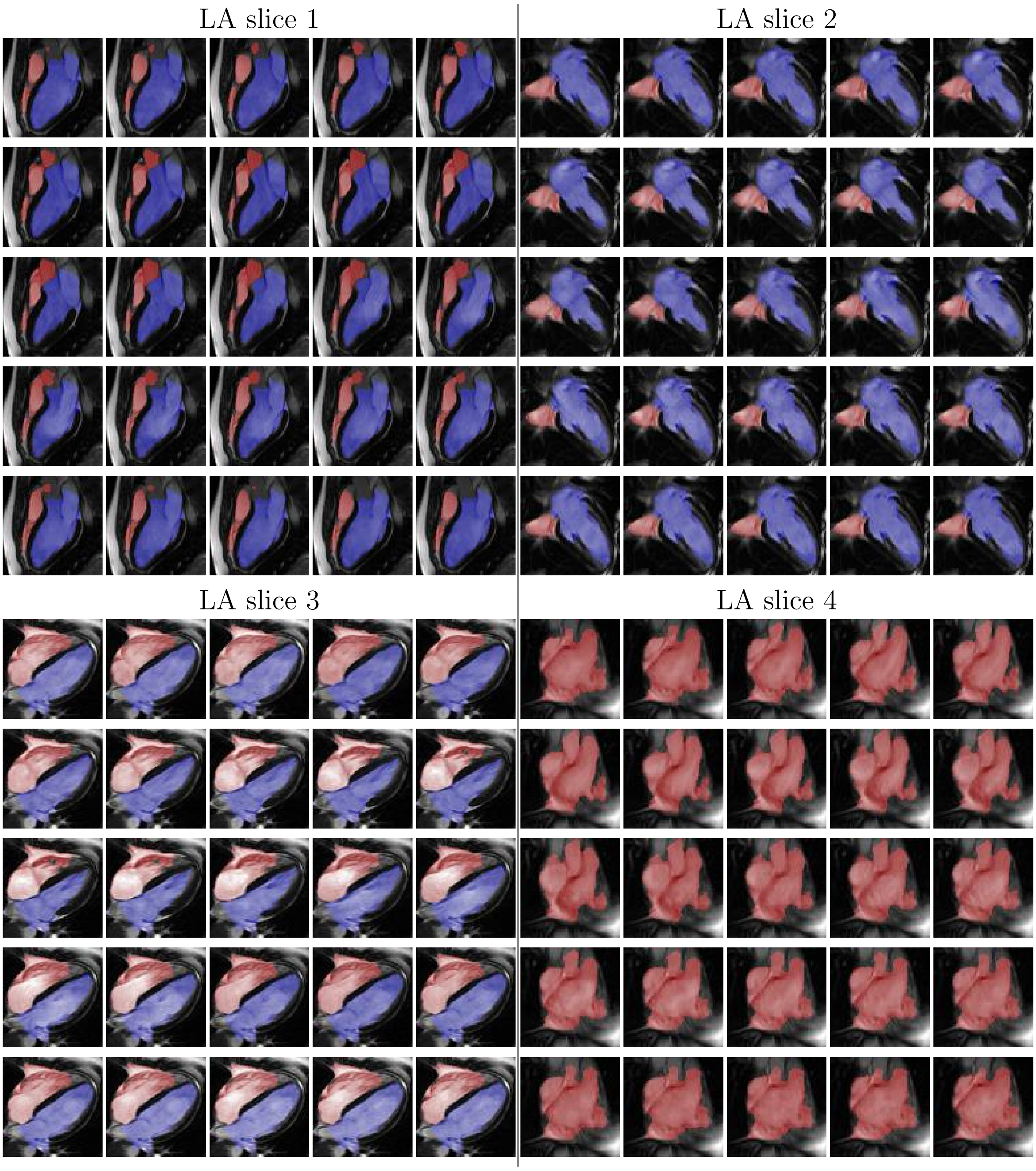


Figure 20: *Segmentation of cardiac cine MRI – dataset 1, all 25 time-frames of LA slices 1 to 4. Blue: LV cavity segmentation, red: RV cavity segmentation. Only the results of IReSISD are shown, because LA slices are not used by the interpolation stage [14] of the sequential method.*

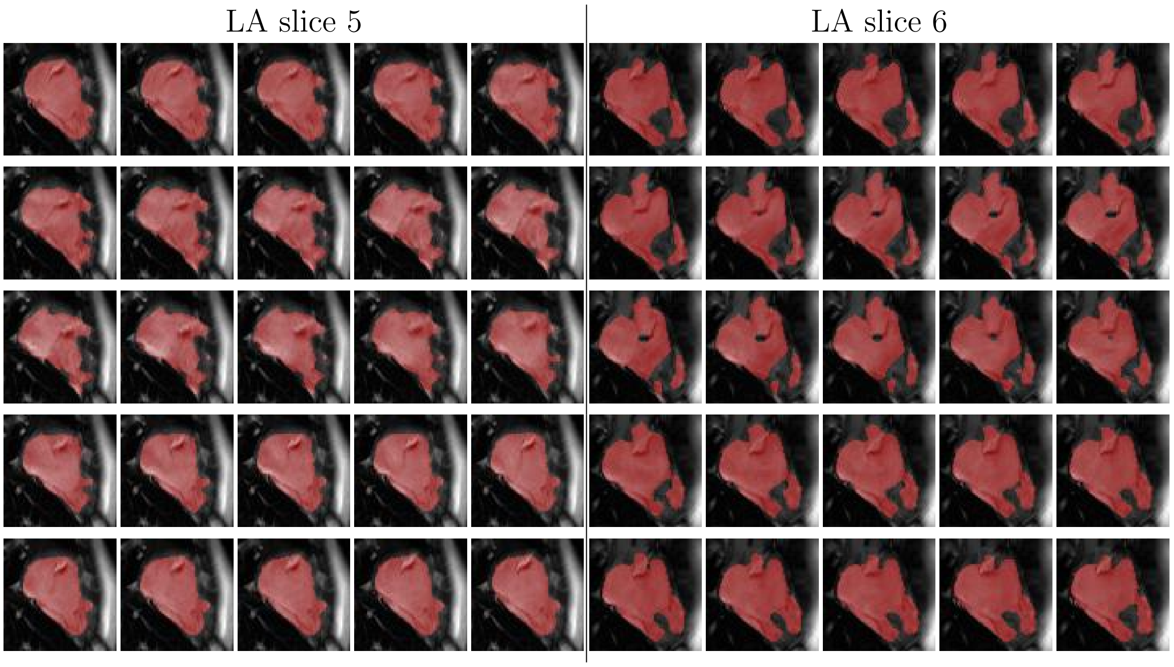


Figure 21: *Segmentation of cardiac cine MRI – dataset 1, all 25 time-frames of LA slices 5 and 6. Blue: LV cavity segmentation, red: RV cavity segmentation. Only the results of IReSISD are shown, because LA slices are not used by the interpolation stage [14] of the sequential method.*

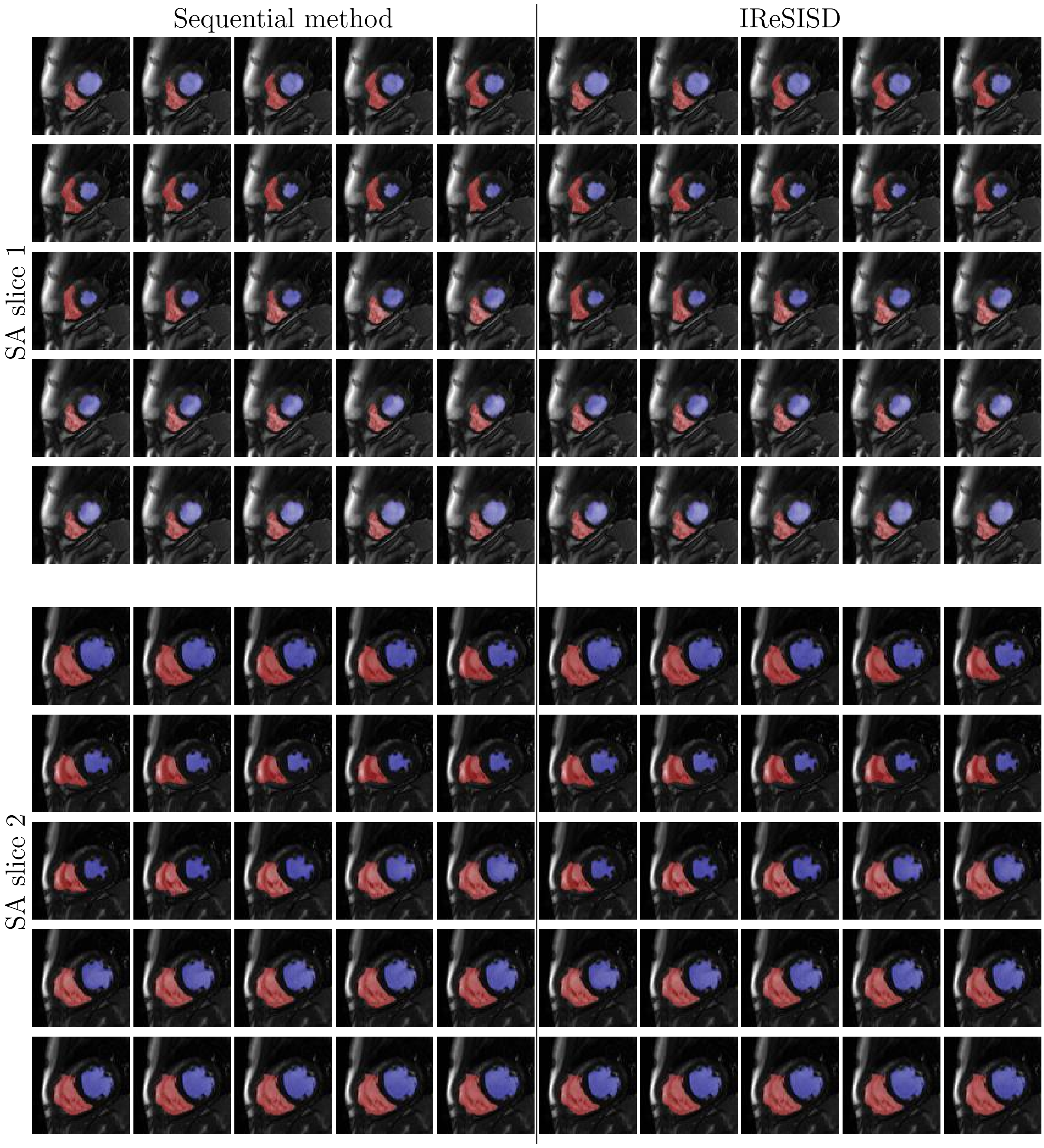


Figure 22: *Segmentation of cardiac cine MRI – dataset 2, all 25 time-frames of SA slices 1 and 2. Blue: LV cavity segmentation, red: RV cavity segmentation.*

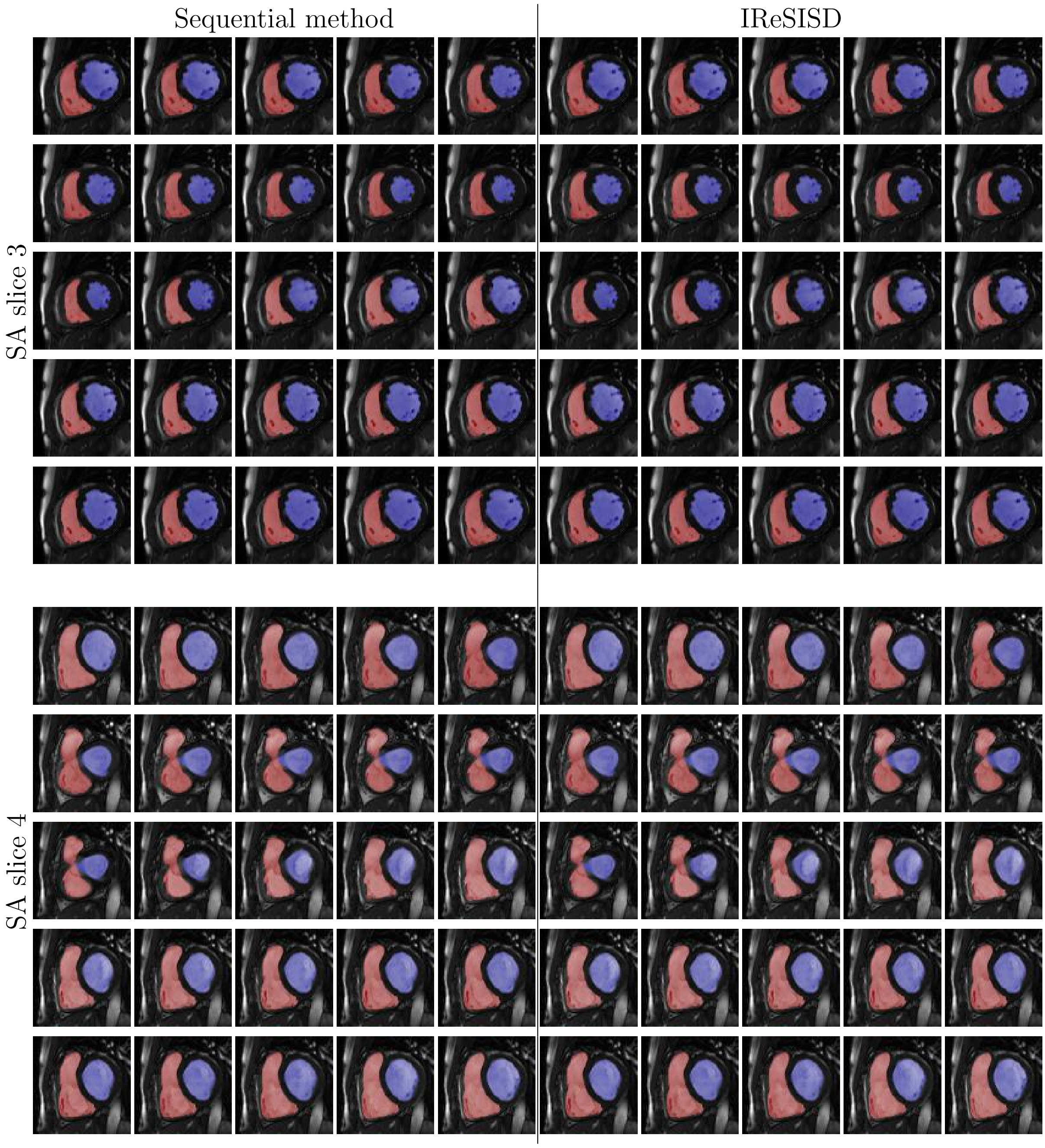


Figure 23: *Segmentation of cardiac cine MRI – dataset 2, all 25 time-frames of SA slices 3 and 4. Blue: LV cavity segmentation, red: RV cavity segmentation.*

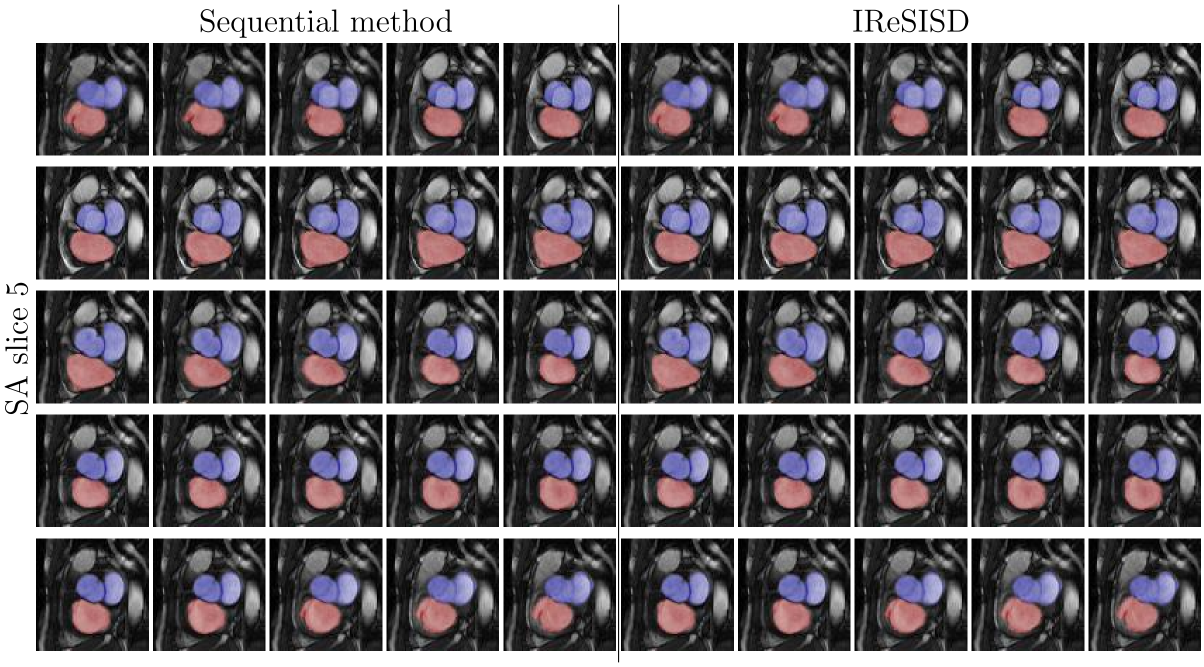


Figure 24: *Segmentation of cardiac cine MRI – dataset 2, all 25 time-frames of SA slice 5. Blue: LV cavity segmentation, red: RV cavity segmentation.*

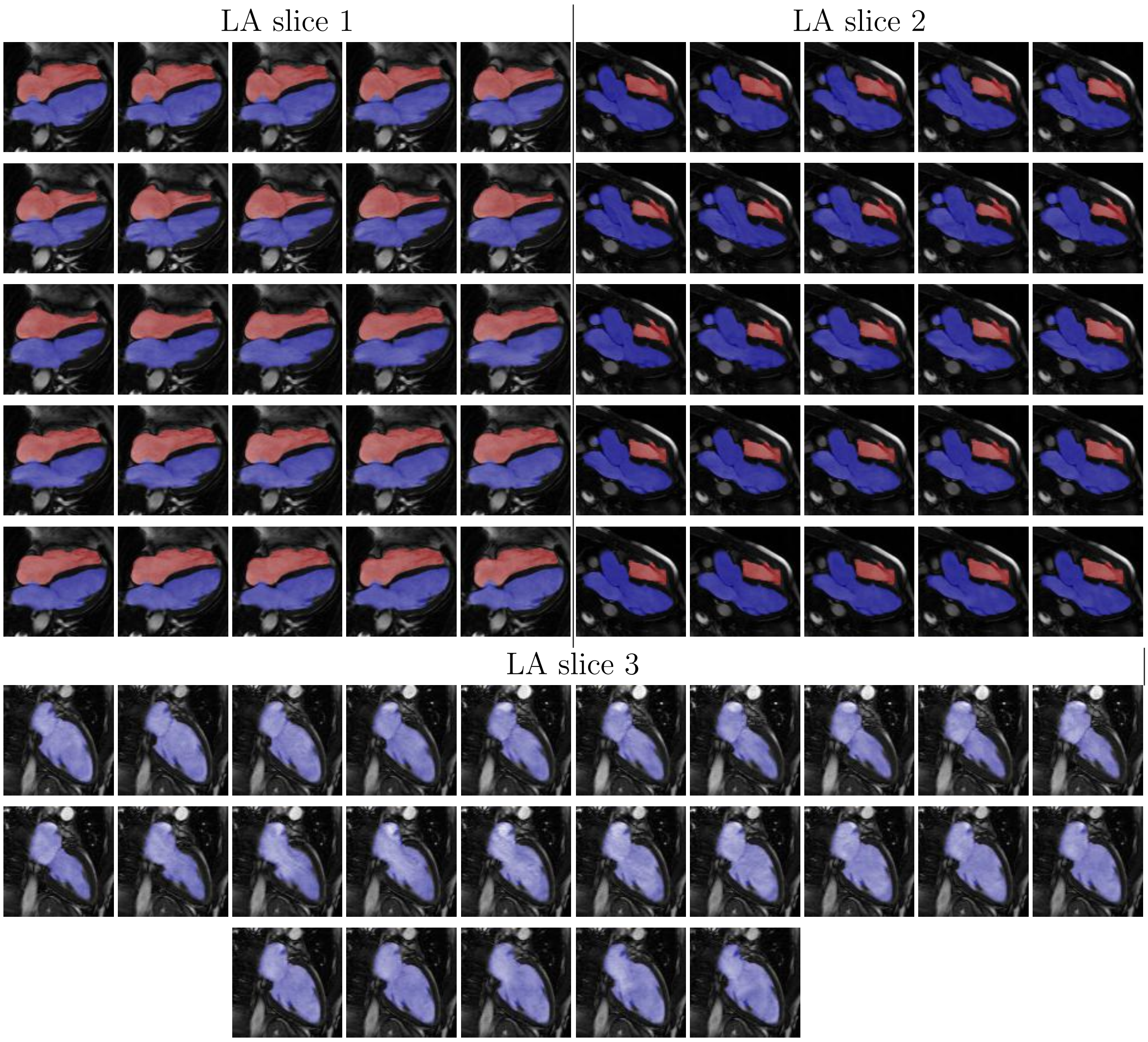


Figure 25: *Segmentation of cardiac cine MRI – dataset 2, all 25 time-frames of LA slices 1 to 3. Blue: LV cavity segmentation, red: RV cavity segmentation. Only the results of IReSISD are shown, because LA slices are not used by the interpolation stage [14] of the sequential method.*