MITT: Medical Image Tracking Toolbox

Sandro Queirós, Pedro Morais, Daniel Barbosa, Jaime C. Fonseca, João L. Vilaça, and Jan D'hooge

***Abstract*— Over the years, medical image tracking has gained considerable attention from both medical and research communities due to its widespread utility in a multitude of clinical applications, from functional assessment during diagnosis and therapy planning to structure tracking or image fusion during image-guided interventions. Despite the ever-increasing number of image tracking methods available, most still consist of independent implementations with specific target applications, lacking the versatility to deal with distinct end-goals without the need for methodological tailoring and/or exhaustive tuning of numerous parameters. With this in mind, we have developed the Medical Image Tracking Toolbox (MITT) - a software package designed to ease customization of image tracking solutions in the medical field. While its workflow principles make it suitable to work with 2D or 3D image sequences, its modules offer versatility to set up computationally efficient tracking solutions, even for users with limited programming skills. MITT is implemented in both C/C++ and MATLAB, including several variants of an object-based image tracking algorithm and allowing to track multiple types of objects (i.e. contours, multi-contours, surfaces and multi-surfaces) with several customization features. In this work, the toolbox is presented, its features discussed, and illustrative examples of its usage in the cardiology field provided, demonstrating its versatility, simplicity and time efficiency.**

This work was funded by projects "NORTE-01-0145-FEDER-000013" and "NORTE-01-0145-FEDER-024300", supported by the Northern Portugal Regional Operational Programme (Norte2020), under the Portugal 2020 Partnership Agreement, through the European Regional Development Fund (FEDER), and also by FEDER funds, through the Competitiveness Factors Operational Programme (COMPETE), and by national funds, through the FCT – Fundação para a Ciência e Tecnologia, under the scope of the project POCI-01-0145-FEDER-007038. The authors acknowledge support by FCT and the European Social Found, through Programa Operacional Capital Humano (POCH), in the scope of the PhD grants SFRH/BD/93443/2013 (S. Queirós) and SFRH/BD/95438/2013 (P. Morais). J. L. Vilaça and J. D'hooge are joint last authors. *Asterisk indicates corresponding author.*

*S. Queirós is with the Life and Health Sciences Research Institute (ICVS), School of Medicine, University of Minho, 4710-057, Braga, Portugal, with ICVS/3B's – PT Government Associate Laboratory, Braga/Guimarães, Portugal, with Algoritmi Center, School of Engineering, University of Minho, Guimarães, Portugal, and with Lab on Cardiovascular Imaging and Dynamics, KU Leuven, 3000 Leuven, Belgium (email: sandroqueiros@med.uminho.pt).

P. Morais is with the Life and Health Sciences Research Institute and ICVS/3B's – PT Government Associate Laboratory, University of Minho, Portugal, with Lab on Cardiovascular Imaging and Dynamics, KU Leuven, Belgium, and with Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial, Universidade do Porto, Porto, Portugal.

D. Barbosa and J. L. Vilaça are with the Life and Health Sciences Research Institute and ICVS/3B's – PT Government Associate Laboratory, University of Minho, Portugal, and with 2Ai – Polytechnic Institute of Cávado and Ave, 4750-810 Barcelos, Portugal.

J. C. Fonseca is with the Algoritmi Center, School of Engineering, University of Minho, 4800-058 Guimarães, Portugal.

J. D'hooge is with the Lab on Cardiovascular Imaging and Dynamics, KU Leuven, 3000 Leuven, Belgium.

***Index Terms*—anatomical affine optical flow, image processing toolbox, medical image tracking toolbox; motion estimation.**

I. Introduction

MEDICAL imaging is nowadays firmly established within the medical community, being vastly used during routine clinical practice to help diagnose and treat a multitude of diseases. Alongside, medical image processing has gained considerable attention from industrial and research communities, with the development of an ever-increasing number of automated solutions [1-9]. Since the beginning, the majority of clinical imaging applications, and hence also software, focused on the anatomical evaluation of the subject/patient through static image acquisitions (or dynamic ones but considering one time point only), aiming at assessing the morphology (*e.g.*, shape, structure, size, location, etc.) of the internal body [9]. Nonetheless, over the years, medical and research communities started directing their efforts towards analyzing dynamic/functional aspects of our body (*e.g.*, cardiac motion and deformation, blood flow, brain activity, respiratory movements, among others), given its incremental value for diagnosis and treatment [10-12]. While the former relies on processing techniques such as image segmentation, shape analysis, image classification or image enhancement [1, 2, 7-9], the latter is usually based upon techniques like image registration and tracking [3-5].

In broad terms, whereas image registration focuses on finding the relationship between two or more images, image tracking aims at following the motion of specific patterns along an image sequence. Although registration can be used to perform tracking (through the quantification of the relative motion between pairs of images), its generic principle allows its application in many other fields (extensive reviews on the subject can be found in Markelj, et al. [3], Sotiras, et al. [4]). With respect to image-based tracking, potential clinical applications include functional assessment during diagnosis (e.g., motion/deformation assessment of cardiac structures [10, 11, 13, 14], quantification of tendon displacement [15, 16], among many others) and therapy planning (e.g., dynamic assessment of heart valves prior to percutaneous interventions [17-19]), structure tracking during image-based surgeries, interventions and radiotherapy (e.g., chamber/valve tracking during cardiac interventions [20, 21], tumor-tracking during radiotherapy or biopsy interventions [22-24], instrument tracking during surgeries [25, 26], among others [3, 27]) and tracking-based multimodality image fusion [28-30], being also used as a tool to potentiate other image-based assessments (e.g., motion correction for free-breathing imaging [31]).

Overall, while image tracking techniques have a tremendous applicability, most existing solutions consist of independent implementations with very specific target applications. Taking the cardiology field as an example [17, 32-38], image tracking solutions are typically aimed at a certain modality or sequence (e.g., ultrasound, computed tomography, cine or tagged magnetic resonance sequences) with a particular image dimensionality (bi- or tridimensional sequences) and to assess a single chamber (left or right ventricle or atrium, the myocardium or a cardiac valve), thus lacking the versatility to deal with distinct end-goals without the need for methodological tailoring and/or exhaustive tuning of numerous and complex parameters.

A notable exception is the elastix toolbox, which provides a vast set of algorithmic options for image registration in a fairly easy and customizable manner [39]. Nonetheless, aiming at tracking applications, its translation into the clinical environment can still be challenging, with limitations including its computational burden (due to the principles behind image registration), the need for high-level programming skills to extract meaningful information from full sequences (as it is built upon the principle of aligning a pair of images only) and the need to customize a wide range of parameters when tuning for a specific tracking goal (given its extremely generic applicability).

In light of this, we developed the Medical Image Tracking Toolbox (MITT) – a novel toolbox designed with the ultimate goal of providing in a single software package the tools required to easily tailor an image tracking solution for a particular medical application. While its workflow principles make it suitable to deal with full 2D or 3D image sequences, its underlying modules and algorithms, together with a command-line interface, permit an easy customization of computationally efficient tracking solutions, even for users with limited programming skills.

This paper is structured as follows. In Section II, the overall structure of MITT is described, key features are presented, and an overview of the underlying tracking module and its key components is given. Section III presents several application scenarios, demonstrating the versatility and computational efficiency of MITT for each, as well as providing evidence of its clinical relevance. Section IV discusses MITT's current implementation and compares it with other toolboxes. The main conclusions are given in Section V.

II. Medical Image Tracking Toolbox

A. Toolbox characteristics

The MITT is structured according to the scheme of Fig. 1. Currently, its implementation includes an object-based image tracking module based on the previously described anatomical affine optical flow (AAOF) algorithm [40], including 2D+t and 3D+t implementations of both global and localized versions of AAOF, and permitting the tracking of multiple types of objects (*i.e.* contours, multi-contours, surfaces and multi-surfaces) with several customization features. The software package can then be complemented with post-processing and visualization add-ons/plugins for a specific clinical application, currently including a cardiology-oriented module (focused on strain analysis and motion assessment).

MITT is currently implemented in two programming languages, C/C++ and MATLAB (Mathworks, USA). The former is built upon widely used open source libraries, namely OpenCV (opencv.org), the Insight Toolkit (ITK; itk.org) and the Visualization Toolkit (VTK; vtk.org) [41], existing both CPU- and GPU-based implementations. While MATLAB is inherently a scripting language, the C/C++ version has a command-line interface, enabling easy processing of data sets by users with limited programming skills, independently of the version used. With advanced users in mind, a MATLAB-to-C++ interface is also available to combine the computational efficiency of C/C++ with the advantages of MATLAB for code prototyping. The MITT executables can be accessed online[1] (a demo version is available in supplementary data).

B. Toolbox workflow overview

MITT is a modular toolbox that comprises three modules (Fig. 1): the input handling; the image tracking algorithms and associated components; and the output handling, in which the tracked object is saved into the disk or used in add-ons/plugins (created with a specific purpose in mind).

With respect to the input, the software accepts images in the MetaImage file format, with the user having to indicate the input images' file names. To facilitate, the image sequence can be stored as independent files in an ordered fashion (indicated in the file name with a wildcard character) or as a single file with a higher-dimensional data (*e.g.*, a volume for a 2D image sequence). The former is appropriate for 3D+t applications, while the latter can be useful for 2D+t ones (namely ultrasound acquisitions in which the sequence is usually extracted in a single file). Optionally, the user may also supply a mask image (same format and entering the file names), indicating the regions of interest (*i.e.* any point outside the mask will be ignored when processing). If the same mask is to be applied to all frames, the user can indicate a single image/volume and the software will replicate as needed. These inputs will form the data object within MITT.

Besides providing the images, the user can configure the tracking algorithm and its components through a parameter text file (an example is provided in supplementary material 1). These settings allow to select the suitable configuration for a particular application, thus controlling the algorithm's pipeline and optionally setting any additional parameter.

Taking into account the tracking module, the last required input is the object to be tracked, which must be provided as a structured grid in the VTK file format. This format supports 1D, 2D and 3D structured grids, thus being able to represent any object of interest (*i.e.* contours, multi-contours, surfaces or multi-surfaces; Section II-D4). Any grid point will consist of x-y-z values in world coordinates (example files are provided in supplementary data), even for objects in a 2D image (*i.e.* all z values must be equal to 0). Note that the object is not

[1] Available at http://2ai.ipca.pt/mitt

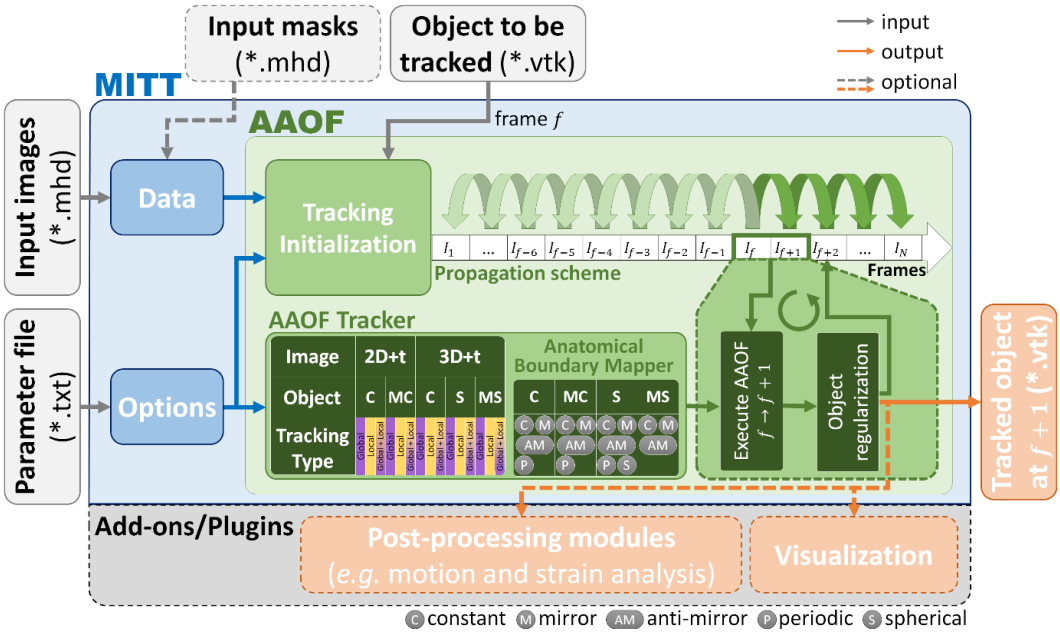

Fig. 1 – Schematic workflow of MITT. For a given input image sequence (2D or 3D) and according to a set of options defined through a parameter text file, an object of interest is propagated using the anatomical affine optical flow (AAOF) algorithm. Such object-based tracking module includes distinct AAOF variants, permitting the tracking of multiple type of objects (contours [C], multi-contours [MC], surfaces [S] and multi-surfaces [MS], with several boundary conditions available) using either global or localized principles (or a combination of them). After each propagation step, an object regularization step is optionally applied, and the process repeated between adjacent frames according to a user-defined propagation scheme. In the end, the tracked object is saved to the disk and/or used in add-ons/plugins to extract meaningful clinical information.

necessarily defined at the first frame of the sequence, being the user able to indicate which frame (f) it represents.

Based on the user inputs, the associated tracker is then defined and initialized (Section II-D), and the object tracked from the initial frame to all others according to a user-defined propagation scheme (option set in the parameter file; see Section II-C). After each propagation step, the tracked object at the new frame is saved to the disk in the same VTK file format. At the end, there will be one file per input frame.

Besides outputting the tracked object, these results can be used in add-ons/plugins to extract relevant clinical information or to be visualized. Currently, two MATLAB add-ons have been implemented: a cardiology-oriented module for strain analysis and a set of visualization routines (supporting both 2D and 3D images, and any type of object).

In the following sub-sections, more information is given about each component of MITT, describing their overall features and associated options. Finally, an example of usage for the C/C++ version is given in the Appendix A, with several example cases (used in Section III) being additionally provided in supplementary data.

C. *Tracking propagation scheme*

Since the reference frame in which to start the tracking can be set to an arbitrary frame within the input sequence, there are a few alternatives on how to apply the frame-to-frame propagation to the entire sequence. In this regard, the following propagation schemes are currently supported by MITT: (1) bidirectional propagation - *i.e.* from the reference frame to the last frame and from the reference frame to the first frame, without assuming a complete cycle (Fig. 2A); (2) unidirectional propagation - *i.e.* always forward and, if needed, assumes cyclic propagation (Fig. 2B); (3) 50/50 bidirectional propagation - *i.e.* it propagates half of the frames forward, and half backward, assuming cyclic propagation if needed, minimizing the number of propagation steps and thus reducing error accumulation (Fig. 2C).

D. *Tracking module – Anatomical affine optical flow (AAOF)*

1) *Algorithm overview*

The AAOF is an object-based image tracking algorithm, whose key principle is to estimate an object's motion between adjacent frames using optical flow (OF) [40]. While traditional OF strategies compute the motion across a predefined regular region using the brightness constancy assumption, the AAOF algorithm uses the a priori knowledge about the object of interest (mandatorily known) to anatomically constrain the motion estimation step to the region of interest (ROI) around the tracked object (*i.e.*, defining a window function wrt. the object). Besides reducing the method's computational burden, the constrained estimation preserves the constant motion assumption over the defined ROI by avoiding the influence of surrounding structures. The motion estimated at the anatomical ROI is then integrated into an affine motion model (or a reduced version of it, Section II-D2), using either global or localized principles (Section II-D3). To be able to capture large displacements, the algorithm also includes an iterative displacement refinement scheme. The idea is to progressively accumulate the estimated displacement fields over a fixed number of iterations, warping the target image after each iteration and thus iteratively reducing the amount of motion left to be estimated. After a fixed number of iterations, the

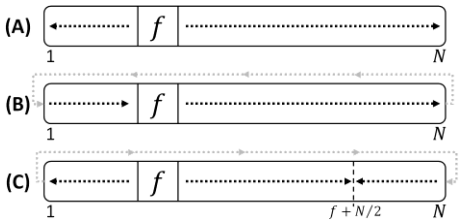

Fig. 2 – MITT supports three propagation schemes: (A) bidirectional; (B) unidirectional; (C) 50/50 bidirectional.

estimated motion is used to propagate the object to the subsequent frame and the whole process is repeated following the defined propagation scheme (Section II-C).

Besides choosing the adequate motion model and tracking type (Sections II-D2 and II-D3, respectively), a few other parameters must be carefully set according to the application's object and imaging modality to achieve an accurate tracking, namely the size of the window function used in the anatomical ROI, the number of iterations of the refinement scheme and the standard deviation of the Gaussian derivative kernel used to compute the optical flow's image spatial gradients. Indeed, the window size controls the amount of object's neighborhood used to estimate the motion. If set too large, the motion of surrounding structures may influence the estimated motion, while a too small window might not be sufficient to capture enough information to accurately estimate the underlying motion. In its turn, the number of iterations must be set based on the expected amount of motion for the object under study, being also dependent on the defined window size. Finally, the Gaussian kernels' width must be set according to the signal-to-noise ratio of the image, being large enough to reduce the method's sensitivity to noise but sufficiently small to retain enough information within the defined ROI. Note that all these parameters also influence the method's computational burden. Moreover, all can be defined through the parameter text file (see example in supplementary material 1, available in the supplementary files/multimedia tab).

2) Motion model

The motion model sets the degrees of freedom for the object's motion between subsequent frames. Generally, an affine transformation model is used, allowing to describe the motion within the anatomical ROI as a combination of four components: translation, rotation, scaling and shearing. However, for a given application, it might be interesting to re-parameterize the affine transformation to only include a subset of its components [42] – this is known as model reduction. Besides the affine model (default option), three reduced motion models are currently supported by MITT: rigid (*i.e.*, translation plus rotation only); similarity (*i.e.*, rigid plus isotropic scaling); and anisotropic similarity (*i.e.*, rigid plus non-isotropic scaling).

3) Tracking type

Originally, AAOF relied on a global motion model [43, 44]. In this version, the goal was to integrate the motion estimated within the entire anatomical ROI into a single affine transformation, which would describe the propagation of the entire object to the adjacent frame.

More recently, a localized version of this algorithm was proposed [40], in which a local affine motion model is independently estimated for each object's point via a weighted anatomical ROI. In other words, the motion is estimated per point using its anatomically constrained neighborhood. Such anatomical weighting is made possible by describing the object based on its anatomical topology (*i.e.*, mapping it in a regular mesh, namely by assuming a polar, spherical, prolate or cylindrical topology, among others). For example, a contour in a 2D image can be described using a polar topology, while a sphere in 3D can be represented with a spherical topology. Note that, if required, other representations (not based on the anatomy) can be used to describe the object or points of interest, as long as it is given as a regular mesh to enable a localized weighting/smoothing. Equally important is the setting of the desired support for the localized estimation. In this regard, the user can control the size of the anatomically-constrained neighborhood used for each point through the definition of a Gaussian-based weight, ultimately controlling the smoothness of the estimated motion – it can be defined from a single point estimate (highly irregular and noisy tracking) to a global support estimate (as in the global counterpart, with less degrees of freedom to capture local motion/deformation). Note however that, in opposition to other methods that regularize a motion field estimated *a priori* [45-47], the AAOF algorithm embeds such anatomical information directly into the OF estimation (through the direct computation of a local motion model per point). The localized support parameter is thus paramount in achieving an accurate and robust tracking. The reader is kindly referred to Queirós, *et al.* [40] for further details regarding this parameter and its influence on the tracking result.

MITT supports both global and localized versions of AAOF. Within the localized one, the user can set the localized support independently for each dimension of the object's anatomical mesh. This parameter sets the Gaussian kernel's radius used (sigma equal to a third of the radius and size equal to 2×radius+1) and is given as a percentage of the object's grid size (number of points) along the corresponding dimension. To further increase the toolbox's flexibility, a combined version is also available (named *Global+Local*). In this version, the algorithm applies a global estimate and proceeds with a localized refinement (using the global affine transformation as the starting point for the second stage). This option makes it possible to track an object even if its movement combines both global motion (such as a large translation and/or rotation due to surrounding structures) and local deformation (*i.e.* within the object itself). When this option is used, all abovementioned parameters (number of iterations, window size, etc.) can be independently set for both global and localized stages.

4) Mesh topology/dimensionality

According to the dimensionality/topology of the mesh chosen to represent the object to be tracked and the image dimensionality, MITT defines four types of objects: (1) contours, used to represent objects with a single parametric coordinate (*i.e.*, objects entered as a 1D structured grid; Fig.

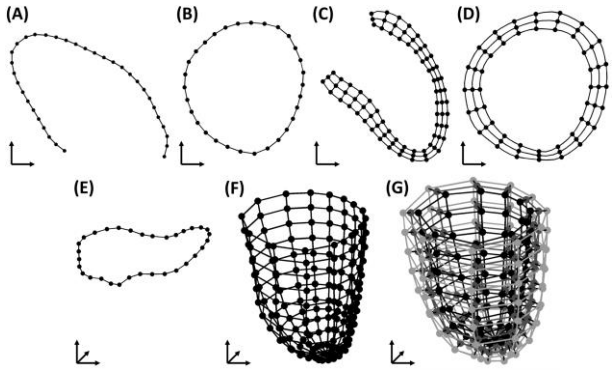

Fig. 3 – Four type of objects are currently defined by MITT according to the dimensionality/topology of the tracked object: contours (A, B and E); multi-contours (C and D); surfaces (F); and multi-surfaces (G). All types can be further customized based on the used boundary conditions (e.g., to represent open or closed contours as in (A) and (B), respectively).

3A, B and E); (2) multi-contours, used to represent a set of related contours in a dual parametric coordinate system (*i.e.* objects entered as a 2D structured grid) and whose anatomical weighting is jointly applied across both dimensions (Fig. 3C and D); (3) surfaces, representing objects in a three-dimensional space but described using a topology with two coordinates only (*i.e.*, objects entered as a 2D structured grid; Fig. 3F); (4) multi-surfaces, representing a set of surfaces related with each other (*i.e.* objects entered as a 3D structured grid) and whose motion should be jointly estimated (Fig. 3G). Although multi-contours and surfaces have the same mesh dimensionality, they are here described separately given their distinct geometric meaning. Note that both contours and multi-contours may be used in 2D or 3D applications, while surfaces and multi-surfaces are intended for 3D applications only and enable a three-dimensional assessment of the object despite its description on a lower-dimensional space (*e.g.*, assuming closed poles in a spherical topology).

5) Anatomical boundary mapping

As mentioned above, in the localized AAOF, each object point is tracked by integrating the estimated motion within an anatomically-constrained neighborhood. Since a weighted integration is applied across the anatomical mesh during OF computation, it is crucial to deal with the mesh's boundaries. Indeed, besides setting the object's mesh dimensionality, one must also define appropriate boundary conditions to match the anatomical topology used to represent the object of interest. In this sense, MITT supports multiple options within its anatomical boundary mapper, allowing the user to quickly adapt the tracker to the needs of its application. Currently supported boundary conditions include constant boundaries (Fig. 4B), mirror boundaries (Fig. 4C), anti-mirror boundaries (Fig. 4D), periodic boundaries (Fig. 4E) and spherical boundaries (*i.e.*, 180º-shifted mirrored boundaries; Fig. 4E). While the first three are commonly used in open objects (such as open contours, or across the second dimension of multi-contours, etc.), the fourth is applicable for closed objects (either closed contours or across the circumferential direction of a surface described with a spherical or cylindrical topology; see Fig. 3B and F, respectively). In its turn, the fifth is very useful for a spherical topology (*i.e.*, for an anatomical

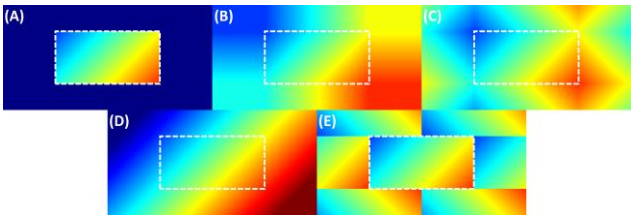

Fig. 4 – (A) An object described with a 2D mesh topology (delimited by the white dashed line) was colored based on a monotonically increasing value (representing an intermediate quantity computed in the optical flow algorithm). For this given object, and to correctly embed its topology, MITT supports the use of constant boundaries (B), mirror boundaries (C), anti-mirror boundaries (D), periodic boundaries (E; at the left and right sides) or spherical boundaries (E; at the top and bottom sides).

weighting during motion estimation while assuming a closed sphere-like object; see Fig. 3F-G). Note that the user can choose the boundary per mesh dimension and per side (*i.e.*, two distinct boundary conditions per mesh dimension, unless periodic is assumed). In Section III, we demonstrate possible application scenarios for each one of the aforementioned type of objects (with distinct mesh topologies), exemplifying the setting of these boundary conditions for each.

E. Object regularization

In some applications, one might want to make sure the object keeps a specific structure throughout the propagation (*e.g.*, ventricle with a closed apex in a 3D space, with a row of grid points representing the same Cartesian position, *i.e.* the apex; a sphere-like 3D left atrial shape, which is closed in both ends; or a blind-ended left atrial appendage shape). These regularizations are presently supported by MITT (*i.e.*, single or double closed surface-like object) and are applied, if needed, after each propagation step.

III. Experiments and Results

In this section, several application scenarios for MITT are described, aiming to exemplify its suitability for quick customization of object-based tracking solutions. Examples are taken from cardiac imaging, although it is not restricted to this application only. To illustrate its versatility, the image data targeted multiple cardiac structures imaged with two of the most clinically used modalities for motion assessment (ultrasound and magnetic resonance imaging, MRI), and considering the software applicability for both healthy subjects and diseased patients (Section III-A). Given the computational attractiveness of the underlying algorithm, the computational burden for each application was also studied (Section III-B). All MITT commands, parameter files and data used in the following experiments are available as supplementary data (except 3D image data, shared upon request), together with a demo version of MITT executables for Microsoft Windows (R) (x64) operating system [48].[2]

A. Application scenarios

The application scenarios, henceforth named demos, include the tracking of: (1) the left atrium (LA) endocardium

[2] https://data.mendeley.com/datasets/9y35mnt56v/1

TABLE I
SETTINGS USED TO EXECUTE EACH DEMO, AND ASSOCIATED IMAGE AND OBJECT'S CHARACTERISTICS

Description	Demo 1	Demo 2	Demo 3	Demo 4	Demo 5	Demo 6	Demo 7	Demo 8	Demo 9
Imaging modality	cMRI (4CH)	cMRI (SAx)	tMRI (SAx)	cMRI (4CH)	US (4CH)	US	US	US	US
Object to be tracked	LA_{endo}	RV_{endo}	LV_{myo}	LV_{myo}	LV_{myo}	$MV_{annulus}$	AV_{wall}	LV_{endo}	LV_{myo}
General parameters:									
Image/Mesh dimensionality	2D/1D	2D/1D	2D/2D	2D/2D	2D/2D	3D/1D	3D/2D	3D/2D	3D/3D
Mesh size	62×1	80×1	60×5	80×5	60×5	64×1	32×32	40×20	40×20×3
Propagation scheme	UP	UP	BP	BP	BP	BP	BP	50/50 BP	50/50 BP
Mask usage	No	No	No	No	Yes	Yes	Yes	Yes	Yes
AAOF parameters:									
Motion model	Affine	Affine	Affine	Affine	Affine	Affine	Affine	Affine	Affine
Tracking type	Local	Local	Local	Local	Local	Local	Global + Local	Local	Local
Window radii (mm)[§]	9×9	9×9	6×6	9×9	5×5	3×3×3	2×2×2	4×4×4	4×4×4
Gaussian kernel's sigma (mm)	1.0×1.0	1.0×1.0	0.6×0.6	1.0×1.0	0.6×0.6	0.5×0.5×0.5	0.4×0.4×0.4	0.6×0.6×0.6	0.6×0.6×0.6
Localized support (%)*	25	30	30×30	35×30	35×30	25	50×50	30×30	30×30×30
Anatomical boundary mapper:									
Mesh dimension 1 (side 1/side 2)	Constant	Periodic	Periodic	Mirror	Mirror	Periodic	Periodic	Periodic	Periodic
Mesh dimension 2 (side 1/side 2)	-	-	Constant	Constant	Constant	Constant	Constant	Spherical/Mirror	Spherical/Mirror
Mesh dimension 3 (side 1/side 2)	-	-	-	-	-	-	-	-	Constant
Object regularization	-	-	-	-	-	-	-	Apex closing	Apex closing

BP: bidirectional propagation; UP: unidirectional propagation; 50/50 BP: 50/50 bidirectional propagation; cMRI: cine magnetic resonance imaging; tMRI: tagged magnetic resonance imaging; US: ultrasound; 4CH: 4-chamber view; SAx: short-axis view; AV: aortic valve; endo: endocardium; LA: left atrium; LV: left ventricle; MV: mitral valve; myo: full myocardium; RV: right ventricle.
[§]The window size is given by 2× radii+1; *The localized support is defined through the setting of the kernel's radius (size = 2×radius+1; sigma = radius/3), given as a percentage of the object's mesh size.

of a healthy subject and a patient with amyloidosis, imaged using 2D cine-MRI in a 4-chamber (4CH) view; (2) the right ventricle (RV) endocardium of a patient with hypertrophic cardiomyopathy (HCM), imaged using 2D cine-MRI in a short-axis (SAx) view; (3) the left ventricle (LV) myocardium of a patient with suspected chronic myocardial ischemic heart disease, imaged using 2D tagged-MRI in a SAx view; (4) the LV myocardium of a healthy subject and a patient with HCM, imaged using 2D cine-MRI in a 4CH view; (5) the LV myocardium of a patient with myocardial infarction (MI), imaged using 2D transthoracic echocardiography (TTE) in a 4CH view; (6) the mitral valve (MV) annulus of a patient with a functionally normal MV, imaged using 3D transesophageal echocardiography (TEE); (7) the aortic valve (AV) wall of a patient with severe aortic stenosis, imaged using 3D-TEE; and (8) and (9) the LV endocardial and myocardial walls, respectively, of a patient with MI, imaged using 3D-TTE.

Importantly, all these anatomical structures are definable through the type of objects available in MITT, namely contours to represent both LA and RV endocardium in 2D and the MV annulus in 3D, multi-contours for LV myocardium in 2D images, surfaces for AV and LV endocardium walls, and multi-surfaces for the myocardial wall. For each one of these demos, adequate settings were then chosen to execute MITT (from tracking type and motion model to boundary conditions and object regularization), as summarized in Table I.

Fig. 5 illustrates the tracking results (at distinct time points across an image sequence) for all 2D demos. The automatically extracted strain curves for each case (global, circumferential, longitudinal and/or radial [10]) are also depicted to demonstrate their potential clinical use. Note, *e.g.*, the lower absolute peak strain values obtained for the diseased patient compared to the healthy subject in demos 1 (Fig. 5B vs. A; right panel) and 4 (Fig.5F vs. E; right panel). Movies of all demos are available as supplementary material (Movie 1 to 9), available in the supplementary files/multimedia tab.

B. Computational time

Besides versatility, MITT intends to simultaneously offer computational efficiency for tracking solutions. In this regard, the computational time required to execute the aforementioned demos (including input/output operations) was registered for all MITT versions (CPU- and GPU-based implementations in C/C++, and MATLAB), using code running on an Intel® Core™ i7 CPU at 2.8 GHz, with 16 GB of RAM, a NVIDIA Quadro® K2100M, and a Microsoft Windows® 8.1 operating system. For the sake of representativeness, the code was executed 5 times for each demo and version. Table II presents the computational performance both in terms of average computational time (and standard deviation) and number of frames processed per second (fps). Note that, due to the computational burden of a multi-surface tracking, this option is currently not available in the MATLAB version and thus no result is provided in demo 9 for this version.

IV. DISCUSSION

In recent years, there has been a growing body of evidence

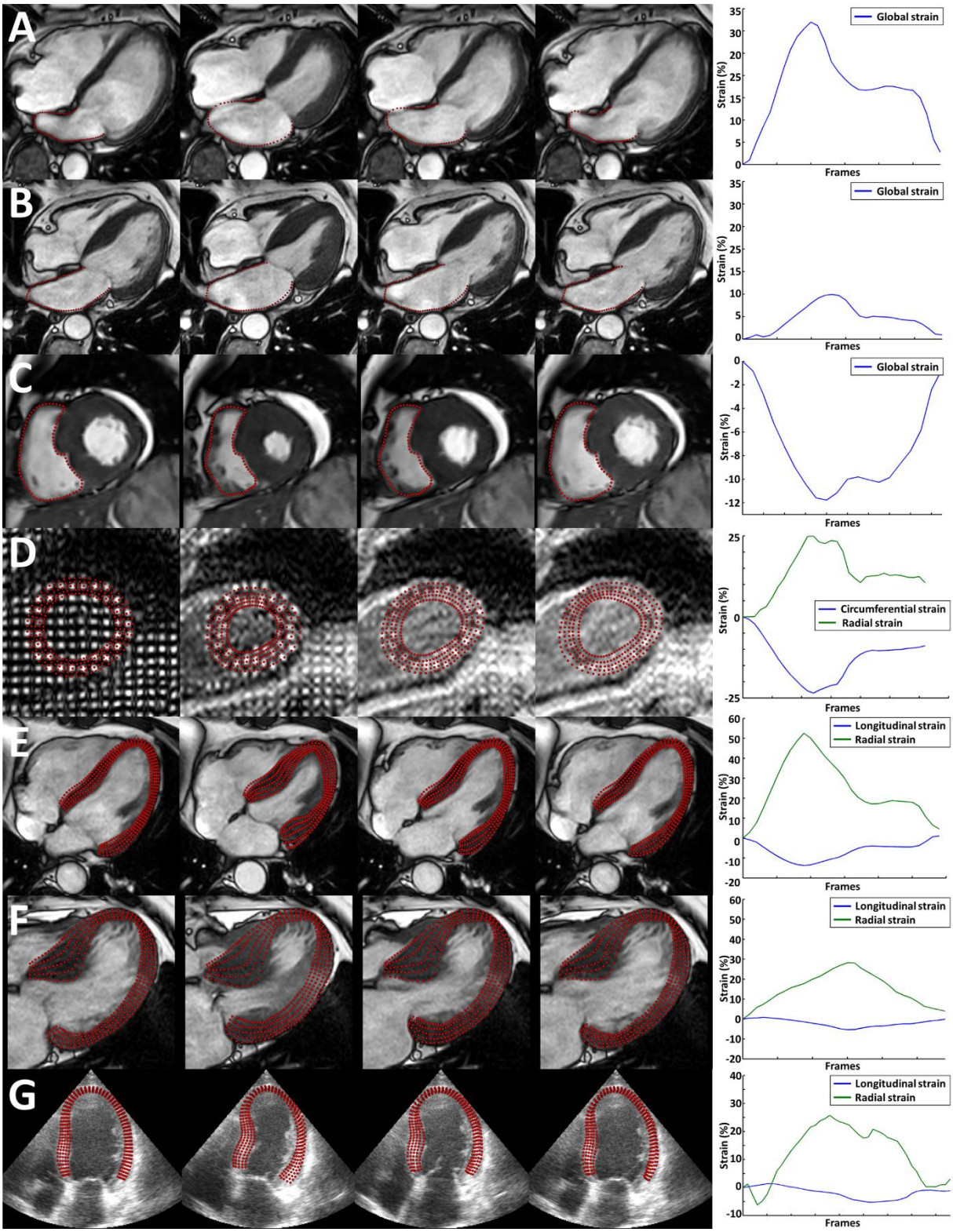

Fig. 5 – Tracking results for the 2D demos, with associated strain curves. (A) and (B) Demo 1 (healthy subject and patient with amyloidosis, respectively); (C) Demo 2 (patient with hypertrophic cardiomyopathy, HCM); (D) Demo 3 (patient with suspected myocardial ischemia); (E) and (F) Demo 4 (healthy subject and patient with HCM, respectively); (G) Demo 5 (patient with myocardial infarction).

demonstrating the substantial information provided by image tracking applications for patients' diagnosis and treatment [5, 10, 11, 20]. For this reason, we have developed and presented MITT - a software package designed for easy customization of tracking solutions in the medical field. Indeed, as described in Section II and demonstrated by the experiments in Section III, MITT offers the following key features: (1) versatility: the currently embedded object-based tracking algorithm permits the adaptation towards multiple applications (Table I); (2) simplicity: MITT is designed to simplify the customization

TABLE II
COMPUTATIONAL PERFORMANCE FOR EACH DEMO AND VERSION

Demo	# frames	GPU-based C/C++		CPU-based C/C++		MATLAB	
		Time (s)	FPS	Time (s)	FPS	Time (s)	FPS
1	30	0.53 ± 0.02	56.8	0.26 ± 0.01	113.3	1.13 ± 0.05	26.6
2	20	0.46 ± 0.02	43.1	0.21 ± 0.01	93.0	0.92 ± 0.01	21.8
3	32	1.10 ± 0.01	29.0	1.29 ± 0.02	24.9	4.97 ± 0.07	6.4
4	30	1.24 ± 0.04	24.1	1.94 ± 0.09	15.5	6.27 ± 0.06	4.8
5	44	1.32 ± 0.07	33.2	3.10 ± 0.11	14.2	4.59 ± 0.04	9.6
6	23	1.67 ± 0.02	13.8	2.89 ± 0.07	8.0	8.28 ± 0.17	2.8
7	71	56.79 ± 1.75	1.3	117.22 ± 1.22	0.6	428.61 ± 13.91	0.2
8	44	16.17 ± 0.39	2.7	33.07 ± 0.55	1.3	91.26 ± 1.59	0.5
9	44	125.90 ± 3.00	0.3	138.79 ± 2.72	0.3	-	-

FPS: frames per second (Hz). Time given as mean ± standard deviation.

and setting of the underlying algorithms towards a given end-goal, even for users with limited programming skills (see Appendix and supplementary data); (3) speed: the C/C++ version permits the development of computationally efficient solutions (Table II); (4) modularity and transparency: the toolbox components were designed to be modular, enabling the comparison of distinct settings for each component, while being transparent with respect to its internal processes to simplify customization and reduce mistakes; and (5) multi-language implementation: MITT is currently implemented in two programming languages, allowing to take advantage of the computational efficiency of C/C++ for end-user applications (namely the GPU version), but also for easy and quick code prototyping through its MATLAB implementation (with the possibility of creating additional add-ons/plugins).

With regard to the AAOF implementation and with respect to previous works [40, 43], MITT also offers numerous novel features, including: (1) the option to describe structures using multi-contour and multi-surface objects; (2) the support for reduced motion models (and not only an affine model; Section II-D2); (3) the possibility to provide an image mask (a single one or one per frame), allowing to indicate regions considered unfeasible for motion estimation (*i.e.* object points positioned within these regions are estimated based on the remaining points); (4) the addition of a configurable boundary mapper, easing the customization and increasing the algorithm's versatility; and (5) the option to combine global and localized implementations (with independent parameters) to capture complex motions.

Taking the cardiology field as example, the presented demos clearly proved the versatility and ease of adaptation of the software, namely with respect to the possible objects to be tracked. Indeed, although the LA endocardium is typically described as an open structure (demo 1, Fig. 5A and 5B, Movie 1a and 1b) and RV as a closed one (demo 2, Fig. 5C and Movie 2), both were easily described as 1D grid-like objects (a contour). To adapt the contour-based object to each case, the user must specify suitable boundary conditions (Fig. 4), using open ones for LA (such as constant, mirror or anti-mirror boundaries, according to the expected motion) and closed ones for RV (namely periodic boundaries). If trying to simultaneously represent multiple boundaries (namely the LV myocardium, as in demos 3, 4 and 5, Fig. 5D-G and Movies 3 to 5), the structure of interest can be described as a multi-contour object. Once again, the adequate setting of the boundary conditions allows to adapt this type of object to different structures. While demo 3 used periodic contours to represent the ventricle imaged from a SAx view (closed along the circumferential direction), mirror boundary conditions were used in demos 4 and 5 to represent the open-surface LV when imaged from a 4-chamber view (opened along the longitudinal direction). A similar analysis is possible in the 3D demos: the closed MV annulus (demo 6, Movie 6) was represented as a contour with periodic boundaries, while the AV wall (demo 7, Movie 7) was described using a surface with mixed boundary conditions (periodic over the circumferential direction and constant along the valve's axis). In demo 8 (Movie 8), a surface was also used to represent the LV endocardium, with the anatomical boundary mapping modified accordingly. In this case, and since a spherical topology was used to represent the object, three distinct boundaries were used: a periodic one to represent the circumferential direction; a spherical boundary along the longitudinal direction at the apex side to provide a closed-type 3D anatomical ROI; and a mirror boundary at the basal side (shown increased accuracy during experimentation). Finally, and as an extension of demos 3, 4 and 5 towards 3D tracking, the full myocardial wall (demo 9, Movie 9) was straightforwardly represented as a multi-surface object, with the motion of all layers being jointly estimated during OF computation.

Besides providing adaptability in terms of represented objects, MITT offers the possibility to easily design tracking solutions for multiple modalities. In this study, the demos included several ultrasound acquisitions (both 2D and 3D TTE and TEE, imaged from different views), but also multiple MRI sequences (namely tagged MRI and cine MRI, in both SAx and 4CH views). Interestingly, the adaptation from one modality to another is performed simply through the customization of a few parameters. Taking demos 4 and 5 as an example (LV tracking in a 4CH view, Table I and Fig. 5E-G), the adaptation from MRI to ultrasound required the modification of two parameters only: the Gaussian kernel's size (based on the overall modality's signal to noise ratio) and the window size (given the image resolution and overall image quality – *i.e.* MRI presents more homogeneous regions, requiring larger windows to capture enough information to estimate the motion using our differential OF strategy, while ultrasound has more features to track, namely the speckle patterns, with smaller windows being enough). Moreover, due to the ultrasound restricted field-of-view, demo 5 also required the use of a mask to limit the regions used during motion estimation (easily included through the command-line interface). Note that the object's points that fall outside the provided mask (or near its edges, see Supplementary Material 1 for details about the available options) will have their motion estimated based on the remaining points within the anatomical ROI. This strategy works for both global (same transformation applied to all points) and local AAOF (in

which estimations are local, but rely on a user-sized anatomical neighborhood). A similar analysis is possible when comparing MRI sequences (demos 3 and 4), and is expected if one would consider other imaging modalities (such as computed tomography or fluoroscopy). Importantly, within the same modality/sequence, most parameters are largely preserved across scenarios (*e.g.*, demos 1, 2 and 4, and demos 6, 8 and 9), ascertaining the simplicity in customizing MITT towards different end-goals.

Since a software's computational burden remains a relevant constraint towards clinical translation, MITT aims at providing an efficient base framework for tracking solutions. According to the experiments (Table II), the C/C++ version demonstrated to be viable for online processing, achieving over 24 fps for all 2D cases (reaching over 90 fps for single contours). Even for 3D tracking, it showed potential towards fast assessment of volumetric sequences, processing contours at ~14 fps and surfaces at 1 to 3 fps. The only exception remains for multi-surface objects, which is associated to computational limitations of the underlying libraries (no GPU implementation is available in the ITK library for convolution of 4D matrices). Nonetheless, if one would intend to increase performance, namely for potential applications like image-based guidance in valvular transcatheter interventions (demo 6 and 7), one could decrease the number of points used to represent the structure of interest or down-sample the volumetric sequence (in the experiments, full resolution images were used – Movies 6 and 7). Indeed, by subsampling the object by 2 in each dimension and the image volume by 3, one achieves ~42 and 16 fps for demos 6 and 7, respectively (Movies 6_sub and 7_sub). Lastly, when comparing CPU- and GPU-based versions, GPU consistently presented a lower burden, being outperformed only for single contours in 2D sequences. Such result is related to the faster processing of a contour-based tracking together with the necessity of transferring data between the graphics and central processing units in the GPU implementation.

Although none of the experiments focused on the global AAOF version, one would expect that setting it as such would lead to a better computational performance (tests revealed a time reduction up to 40%, and achieving an 80% reduction for the multi-surface case) at the cost of a reduced ability to capture regional motion and deformation. In opposition, the combined '*Global+Local*' tracking improves the ability to capture large and complex motions (*i.e.* large displacements due to motion of surrounding structures combined with local deformation of the object under study; demo 7), but at the expense of an inferior time performance with respect to either a pure global or localized strategy.

With respect to the object's regularization, although the presented demos only use one-sided regularization (LV apex in these scenarios), MITT also allows to apply a double-sided one (Section II.E). Nonetheless, more complex regularization models (namely using statistical shape models provided by the user to regularize the surface after each propagation step) are yet to be implemented.

Interestingly, while being an object-based tracking method, the AAOF algorithm implemented in MITT together with the available inputs/parameters permit its customization towards the estimation of a dense motion field between frames (see supplementary material 2 available in the supplementary files/multimedia tab, and Movie 10). Indeed, if we consider a grid-like object in the Cartesian space (instead of defining an anatomical mesh topology) and set adequate boundaries (namely constant ones), one can estimate the motion locally across the entire image domain, which ultimately can be used to deform images, propagate single landmarks or track multiple objects simultaneously (Movies 11 and 12). In simple terms, such grid-like object transforms the anatomical principle behind AAOF back to the original Cartesian optical flow strategy, thus enabling the development of registration-like solutions (namely global or local unimodal data alignment or landmark tracking, [3, 4]). Nonetheless, full support for this type of applications is yet to be incorporated.

When compared to previously presented toolboxes, such as Elastix [39], NiftyReg [49], ANTs [50], MIRTK [51], ITK[4] [50], among many others [51], MITT presents key differences. While these toolboxes mostly implement image registration methods (from classic free-form deformation to diffeomorphic, sparse or shape-based one) with the key goal of aligning a pair of images using dense motion field estimation principles, MITT embeds an optical flow algorithm focused on object-based image tracking over a full image sequence. This brings potential advantages, but also limitations. On the one hand, image registration methods are typically more robust to the image content than OF-based methods, given the differential principles of the latter and the ability to include numerous regularization terms in the former. However, since MITT restricts the OF computation to the anatomical ROI (rather than the entire image domain) defined based on the object to be tracked (which usually partly presents some contrast with neighbor regions), this sensitivity is minimized. Nonetheless, this object-based principle can also be a limitation for certain applications. On the other hand, OF methods tend to be computationally more attractive, which may be interesting in online scenarios. In this regard, MITT's GPU version may be an added advantage. Finally, in opposition to most of the aforementioned toolboxes, MITT also provides a MATLAB-to-C++ interface, allowing easy code prototype in MATLAB with the optimization benefits of the C++ implementation.

Altogether, although demonstrated using examples from cardiac imaging only, the application scenarios shown are a lesser set of all the potential applications that can be tailored with MITT. Given the versatility in inputs and parameters, the toolbox may be used for a multitude of clinical applications, including: functional assessment during diagnosis (e.g., besides cardiac motion assessment, it may be employed for tendon or vessel displacement quantification); dynamic assessment during therapy planning; structure (e.g., devices, tumor, neighbor organs, prosthesis, etc.) or landmark (e.g., medical annotations) tracking during image-guided interventions or radiotherapy; tracking-based multimodality image fusion; among others. To do it, one must always define an object to be tracked *a priori* (either based on the object's

geometry or using a grid-like object) and set the parameters accordingly, modifying them to suit the application but also to tune the algorithm and thus improve accuracy.

V. Conclusion

In this study, we presented the Medical Image Tracking Toolbox (MITT) - a versatile, easy to use, computationally efficient and publicly available toolbox based on the AAOF algorithm, which may potentially be useful for researchers and clinicians interested in medical image tracking.

Appendix
MITT usage example for the C/C++ version

Object-based image tracking using the C/C++ version of MITT is performed through a command-line interface. An example command to run MITT is as follows:

MITT -i img.mhd -t obj.vtk -p param.txt -o outDir

Here, the user mandatorily indicates the input images' file name (*img.mhd*, which includes the full image sequence), the object's file name (*obj.vtk*), the parameter text file (*param.txt*) and the output folder name (*outDir*, in which the VTK files of the tracked object will be saved). Optional arguments may include the mask images' file name (*-m*) or the reference frame (*-r*, if the object to be tracked was not given at the first frame of the sequence). A complete list of mandatory and optional arguments, with a small description of their function, can be generated by calling *MITT --help*.

In its turn, the parameter file includes additional settings to configure the tracking algorithm and its components. A simplified version of it might include the following:

(ImageDimensionality 2) // 2D image sequence
(MeshDimensionality 1) // contour object
(TrackingType "Global") // global AAOF
(TrackingDirection 0) // bidirectional propagation
(MotionType "Affine") // affine motion model
(BoundaryDimension_1 "constant" "constant")

More detailed examples of parameter files are given in supplementary data, together with usage examples.

Acknowledgment

The authors acknowledge Dr. Alexandros Papachristidis (King's College Hospital, London, UK), Professor Jan Bogaert (UZ Leuven, Leuven, Belgium) and Dr. Philippe B. Bertrand (Ziekenhuis Oost-Limburg, Genk, Belgium) for sharing some of the data used in the application scenarios.

References

[1] K. Doi, "Computer-aided diagnosis in medical imaging: historical review, current status and future potential," *Computerized Medical Imaging and Graphics*, vol. 31, pp. 198-211, 2007.

[2] T. Heimann and H.-P. Meinzer, "Statistical shape models for 3D medical image segmentation: a review," *Medical Image Analysis*, vol. 13, pp. 543-563, 2009.

[3] P. Markelj, D. Tomaževič, B. Likar, and F. Pernuš, "A review of 3D/2D registration methods for image-guided interventions," *Medical Image Analysis*, vol. 16, pp. 642-661, 2012.

[4] A. Sotiras, C. Davatzikos, and N. Paragios, "Deformable medical image registration: A survey," *IEEE Transactions on Medical Imaging*, vol. 32, pp. 1153-1190, 2013.

[5] A. P. James and B. V. Dasarathy, "Medical image fusion: A survey of the state of the art," *Information Fusion*, vol. 19, pp. 4-19, 2014.

[6] S. K. Zhou, H. Greenspan, and D. Shen, *Deep Learning for Medical Image Analysis*, 1st ed.: Academic Press, 2017.

[7] J. A. Noble and D. Boukerroui, "Ultrasound image segmentation: a survey," *IEEE Transactions on Medical Imaging*, vol. 25, pp. 987-1010, 2006.

[8] S. K. Setarehdan and S. Singh, *Advanced algorithmic approaches to medical image segmentation: state-of-the-art applications in cardiology, neurology, mammography and pathology*: Springer Science & Business Media, 2012.

[9] I. Bankman, *Handbook of medical image processing and analysis*, 2nd ed.: Academic Press, 2008.

[10] J.-U. Voigt, G. Pedrizzetti, P. Lysyansky, T. H. Marwick, H. Houle, R. Baumann, *et al.*, "Definitions for a common standard for 2D speckle tracking echocardiography: consensus document of the EACVI/ASE/ Industry Task Force to standardize deformation imaging," *European Heart Journal - Cardiovascular Imaging*, vol. 16, pp. 1-11, 2014.

[11] R. M. Lang, L. P. Badano, J. Afilalo, V. Mor-Avi, A. Armstrong, L. Ernande, *et al.*, "Recommendations for cardiac chamber quantification by echocardiography: An update from the American Society of Echocardiography and the European Association of Cardiovascular Imaging," *European Heart Journal - Cardiovascular Imaging*, vol. 16, pp. 233-271, 2015.

[12] D. A. Gusnard and M. E. Raichle, "Searching for a baseline: functional imaging and the resting human brain," *Nature reviews. Neuroscience*, vol. 2, p. 685, 2001.

[13] H. Wang and A. A. Amini, "Cardiac motion and deformation recovery from MRI: a review," *IEEE Transactions on Medical Imaging*, vol. 31, pp. 487-503, 2012.

[14] H. Geyer, G. Caracciolo, H. Abe, S. Wilansky, S. Carerj, F. Gentile, *et al.*, "Assessment of myocardial mechanics using speckle tracking echocardiography: fundamentals and clinical applications," *Journal of the American Society of Echocardiography*, vol. 23, pp. 351-369, 2010.

[15] J.-W. H. Korstanje, R. W. Selles, H. J. Stam, S. E. Hovius, and J. G. Bosch, "Development and validation of ultrasound speckle tracking to quantify tendon displacement," *Journal of Biomechanics*, vol. 43, pp. 1373-1379, 2010.

[16] O. R. Seynnes, J. Bojsen-Møller, K. Albracht, A. Arndt, N. J. Cronin, T. Finni, *et al.*, "Ultrasound-based testing of tendon mechanical properties: a critical evaluation," *Journal of Applied Physiology*, vol. 118, pp. 133-141, 2015.

[17] F. Veronesi, C. Corsi, L. Sugeng, V. Mor-Avi, E. G. Caiani, L. Weinert, *et al.*, "A study of functional anatomy of aortic-mitral valve coupling using 3D matrix transesophageal echocardiography," *Circulation: Cardiovascular Imaging*, vol. 2, pp. 24-31, 2008.

[18] F. Veronesi, C. Corsi, L. Sugeng, E. G. Caiani, L. Weinert, V. Mor-Avi, *et al.*, "Quantification of mitral apparatus dynamics in functional and ischemic mitral regurgitation using real-time 3-dimensional echocardiography," *Journal of the American Society of Echocardiography*, vol. 21, pp. 347-354, 2008.

[19] K. R. Khabbaz, F. Mahmood, O. Shakil, H. J. Warraich, J. H. Gorman, R. C. Gorman, *et al.*, "Dynamic 3-dimensional echocardiographic assessment of mitral annular geometry in patients with functional mitral regurgitation," *The Annals of Thoracic Surgery*, vol. 95, pp. 105-110, 2013.

[20] T. M. Peters and C. A. Linte, "Image-guided interventions and computer-integrated therapy: Quo vadis?," *Medical Image Analysis*, vol. 33, pp. 56-63, 2016.

[21] Z. Luo, J. Cai, T. M. Peters, and L. Gu, "Intra-operative 2-D ultrasound and dynamic 3-D aortic model registration for magnetic navigation of transcatheter aortic valve implantation," *IEEE Transactions on Medical Imaging*, vol. 32, pp. 2152-2165, 2013.

[22] Q. Xu, R. J. Hamilton, R. A. Schowengerdt, B. Alexander, and S. B. Jiang, "Lung tumor tracking in fluoroscopic video based on optical flow," *Medical Physics*, vol. 35, pp. 5351-5359, 2008.

[23] V. De Luca, T. Benz, S. Kondo, L. König, D. Lübke, S. Rothlübbers, *et al.*, "The 2014 liver ultrasound tracking benchmark," *Physics in Medicine and Biology*, vol. 60, p. 5571, 2015.

[24] M. Baumann, P. Mozer, V. Daanen, and J. Troccaz, "Prostate biopsy tracking with deformation estimation," *Medical Image Analysis*, vol. 16, pp. 562-576, 2012.

[25] X. Wu, J. Housden, Y. Ma, B. Razavi, K. Rhode, and D. Rueckert, "Fast catheter segmentation from echocardiographic sequences based on segmentation from corresponding X-ray fluoroscopy for cardiac catheterization interventions," *IEEE Transactions on Medical Imaging*, vol. 34, pp. 861-876, 2015.
[26] M. E. Karar, D. R. Merk, V. Falk, and O. Burgert, "A simple and accurate method for computer-aided transapical aortic valve replacement," *Computerized Medical Imaging and Graphics*, vol. 50, pp. 31-41, 2014.
[27] K. Cleary and T. M. Peters, "Image-guided interventions: technology review and clinical applications," *Annual Review of Biomedical Engineering*, vol. 12, pp. 119-142, 2010.
[28] J. J. Thaden, S. Sanon, J. B. Geske, M. F. Eleid, N. Nijhof, J. F. Malouf, *et al.*, "Echocardiographic and fluoroscopic fusion imaging for procedural guidance: an overview and early clinical experience," *Journal of the American Society of Echocardiography*, vol. 29, pp. 503-512, 2016.
[29] P. Lang, M. W. Chu, D. Bainbridge, G. M. Guiraudon, D. L. Jones, and T. M. Peters, "Surface-Based CT–TEE Registration of the Aortic Root," *IEEE Transactions on Biomedical Engineering*, vol. 60, pp. 3382-3390, 2013.
[30] P. Lang, P. Seslija, M. W. Chu, D. Bainbridge, G. M. Guiraudon, D. L. Jones, *et al.*, "US–Fluoroscopy Registration for Transcatheter Aortic Valve Implantation," *IEEE Transactions on Biomedical Engineering*, vol. 59, pp. 1444-1453, 2012.
[31] J. Ehrhardt, R. Werner, D. Säring, T. Frenzel, W. Lu, D. Low, *et al.*, "An optical flow based method for improved reconstruction of 4D CT data sets acquired during free breathing," *Medical Physics*, vol. 34, pp. 711-721, 2007.
[32] M. Alessandrini, B. Heyde, S. Queiros, S. Cygan, M. Zontak, O. Sophome, *et al.*, "Detailed Evaluation of Five 3D Speckle Tracking Algorithms using Synthetic Echocardiographic Recordings," *IEEE Transactions on Medical Imaging*, vol. 35, pp. 1915-1926, 2016.
[33] P. Morais, S. Queirós, B. Heyde, J. Engvall, J. D'hooge, and J. L. Vilaça, "Fully automatic left ventricular myocardial strain estimation in 2D short-axis tagged magnetic resonance imaging," *Physics in Medicine and Biology*, vol. 62, pp. 6899-6919, 2017.
[34] B. Heyde, M. Alessandrini, J. Hermans, D. Barbosa, P. Claus, and J. D'hooge, "Anatomical image registration using volume conservation to assess cardiac deformation from 3D ultrasound recordings," *IEEE Transactions on Medical Imaging*, vol. 35, pp. 501-511, 2016.
[35] T. Dietenbeck, D. Barbosa, M. Alessandrini, R. Jasaityte, V. Robesyn, J. D'hooge, *et al.*, "Whole myocardium tracking in 2D-echocardiography in multiple orientations using a motion constrained level-set," *Medical Image Analysis*, vol. 18, pp. 500-514, 2014.
[36] C. Tobon-Gomez, M. De Craene, K. Mcleod, L. Tautz, W. Shi, A. Hennemuth, *et al.*, "Benchmarking framework for myocardial tracking and deformation algorithms: An open access database," *Medical Image Analysis*, vol. 17, pp. 632-648, 2013.
[37] R. I. Ionasec, I. Voigt, B. Georgescu, Y. Wang, H. Houle, F. Vega-Higuera, *et al.*, "Patient-specific modeling and quantification of the aortic and mitral valves from 4-D cardiac CT and TEE," *IEEE Transactions on Medical Imaging*, vol. 29, pp. 1636-1651, 2010.
[38] N. Almeida, A. Papachristidis, P. Pearson, S. I. Sarvari, J. Engvall, T. Edvardsen, *et al.*, "Left atrial volumetric assessment using a novel automated framework for 3D echocardiography: a multi-centre analysis," *European Heart Journal - Cardiovascular Imaging*, vol. 18, pp. 1008-1015, 2017.
[39] S. Klein, M. Staring, K. Murphy, M. A. Viergever, and J. P. Pluim, "Elastix: a toolbox for intensity-based medical image registration," *IEEE Transactions on Medical Imaging*, vol. 29, pp. 196-205, 2010.
[40] S. Queirós, J. L. Vilaça, P. Morais, J. C. Fonseca, J. D'hooge, and D. Barbosa, "Fast left ventricle tracking using localized anatomical affine optical flow," *International Journal for Numerical Methods in Biomedical Engineering*, vol. 33, p. e2871, 2017.
[41] W. J. Schroeder, B. Lorensen, and K. Martin, *The visualization toolkit: an object-oriented approach to 3D graphics*: Kitware, 2004.
[42] J.-Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel Corporation*, vol. 5, p. 4, 2001.
[43] D. Barbosa, B. Heyde, T. Dietenbeck, D. Friboulet, J. D'hooge, and O. Bernard, "Fast left ventricle tracking in 3D echocardiographic data using anatomical affine optical flow," in *Functional Imaging and Modeling of the Heart (FIMH2013)*, 2013, pp. 191-199.
[44] S. Queirós, D. Barbosa, B. Heyde, P. Morais, J. L. Vilaça, D. Friboulet, *et al.*, "Fast automatic myocardial segmentation in 4D cine CMR datasets," *Medical Image Analysis*, vol. 18, pp. 1115-1131, 2014.
[45] Y. Zhou, O. Bernard, E. Saloux, A. Manrique, P. Allain, S. Makram-Ebeid, *et al.*, "3D harmonic phase tracking with anatomical regularization," *Medical Image Analysis*, vol. 26, pp. 70-81, 2015.
[46] K. McLeod, M. Sermesant, P. Beerbaum, and X. Pennec, "Spatio-Temporal Tensor Decomposition of a Polyaffine Motion Model for a Better Analysis of Pathological Left Ventricular Dynamics," *Medical Imaging, IEEE Transactions on*, vol. 34, pp. 1562-1575, 2015.
[47] J. Crosby, B. H. Amundsen, T. Hergum, E. W. Remme, S. Langeland, and H. Torp, "3-D speckle tracking for assessment of regional left ventricular function," *Ultrasound in Medicine & Biology*, vol. 35, pp. 458-471, 2009.
[48] S. Queirós, P. Morais, D. Barbosa, J. C. Fonseca, J. L. Vilaça, and J. D' hooge, "MITT - Medical Image Tracking Toolbox," Mendeley Data, v1 [doi:10.17632/9y35mnt56v.1].
[49] M. Modat, J. McClelland, and S. Ourselin, "Lung registration using the NiftyReg package," *Medical image analysis for the clinic-a grand challenge*, vol. 2010, pp. 33-42, 2010.
[50] B. B. Avants, N. J. Tustison, M. Stauffer, G. Song, B. Wu, and J. C. Gee, "The Insight ToolKit image registration framework," *Frontiers in neuroinformatics*, vol. 8, p. 44, 2014.
[51] A. P. Keszei, B. Berkels, and T. M. Deserno, "Survey of non-rigid registration tools in medicine," *Journal of digital imaging*, vol. 30, pp. 102-116, 2017.