Signal Processing 202 (2023) 108771

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Efficient convolution-based pairwise elastic image registration on three multimodal similarity metrics

Rosa-María Menchón-Lara*, Federico Simmross-Wattenberg, Manuel Rodríguez-Cayetano, Pablo Casaseca-de-la-Higuera, Miguel Á. Martín-Fernández, Carlos Alberola-López

Laboratorio de Procesado de Imagen. ETSI de Telecomunicación, Universidad de Valladolid, Valladolid, Spain

ARTICLE INFO

Article history:
Received 31 January 2022
Revised 31 August 2022
Accepted 3 September 2022
Available online 6 September 2022

Keywords:
Multimodal registration
Free-form deformation
Convolution
Non-rigid registration
Efficient implementation

ABSTRACT

This paper proposes a complete convolutional formulation for 2D multimodal pairwise image registration problems based on free-form deformations. We have reformulated in terms of discrete 1D convolutions the evaluation of spatial transformations, the regularization term, and their gradients for three different multimodal registration metrics, namely, normalized cross correlation, mutual information, and normalized mutual information. A sufficient condition on the metric gradient is provided for further extension to other metrics. The proposed approach has been tested, as a proof of concept, on contrast-enhanced first-pass perfusion cardiac magnetic resonance images. Execution times have been compared with the corresponding execution times of the classical tensor product formulation, both on CPU and GPU. The speed-up achieved by using convolutions instead of tensor products depends on the image size and the number of control points considered, the larger those magnitudes, the greater the execution time reduction. Furthermore, the speed-up will be more significant when gradient operations constitute the major bottleneck in the optimization process.

© 2022 The Author(s). Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license
(http://creativecommons.org/licenses/by-nc-nd/4.0/)

1. Introduction

Image registration plays a crucial role in the field of image processing and constitutes the basis for many higher level applications. This technique consists in aligning a pair or a set of images, for which spatial transformations have to be determined. A popular approach to define these transformations in elastic registration, specially for medical imaging applications, is the free-form deformations (FFD) paradigm [1,2]. This powerful geometric modeling technique can be used to represent arbitrary deformations applied to objects [3]. An FFD model inherently gives rise to smooth deformation fields with appropriate selection of its basis functions. However, these models are limited by large memory requirements and high execution times needed to register large-scale images [4].

Recently, a novel and highly efficient implementation of groupwise image registration based on FFDs has been proposed [5]. This approach replaces the classical FFD implementation based on tensor products by a convolutional formulation, achieving large execution time reduction not only for evaluation of transformations, but also for gradient calculation during the optimization process. The average runtime reduction reported is above 90% and the registration accuracy was properly demonstrated on different datasets.

Interpolation using convolutions is a well-known topic [6–8]; however, the implementation of FFDs has been based on tensor products since its onset [1,9] and, to the best of our knowledge, reported implementations [10–14] have not made use of convolutions. As for the convolution-based implementation in [5], it was fully developed for a monomodal registration metric only, namely, the sum of squared differences.

In this paper, we propose the convolutional formulation of elastic pairwise registration on multimodal metrics and we focus on three of them, specifically, normalized cross correlation (NCC), mutual information (MI) and normalized mutual information (NMI). Interestingly, the requirement stated in [5] for the metrics turns out not to be a necessary condition for convolution-based problem formulation. Albeit it is satisfied by NCC, it does not hold for either MI or NMI but they both adhere to this formulation because of the analytical expression of their gradients, for which a sufficient condition has been determined. Furthermore, both the regularization

* Corresponding author.
E-mail address: rmenchon@lpi.tel.uva.es (R.-M. Menchón-Lara).

https://doi.org/10.1016/j.sigpro.2022.108771
0165-1684/© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license
(http://creativecommons.org/licenses/by-nc-nd/4.0/)

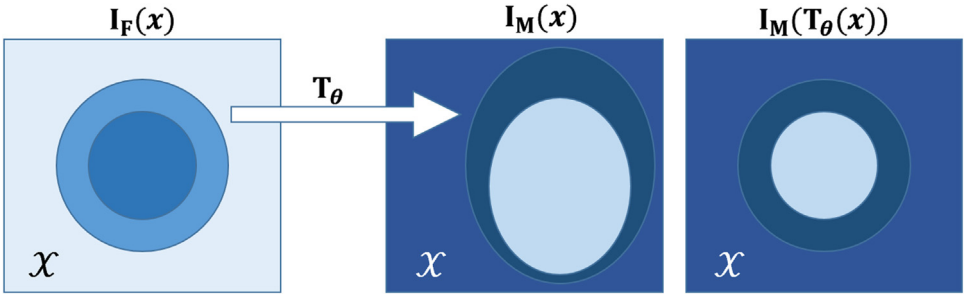

Fig. 1. Scheme of spatial transformations in 2D pairwise image registration.

term and its gradient are also posed in terms of simple convolutions.

The regularization term was not analyzed in [5]. The simple regularizer proposed in [15] was applied directly on the deformation coefficients. This term is a quadratic-like penalty function and does not require additional B-spline interpolations. Therefore, there are no grounds for its convolutional reformulation. However, in this paper we have resorted to a popular regularization term, namely, the transformation Laplacian, as a simplified version of the 2D transformation bending energy [1,16,17]. It turns out that the convolutional implementation of this term provides a significant reduction of the computing time with respect to its direct implementation. Analytical expressions are derived for both the metrics and the regularization gradients.

In order to accelerate the registration procedure, several efficient implementations on graphics processing units (GPU) have been proposed [18–23], reporting significant speed-ups over CPU implementations. Moreover, Deep Learning approaches have also been applied to the image registration problem [24–26], providing fast solutions once the networks have been trained for a specific application. Our proposal leads to important execution time reductions for the reformulated functions, both in CPU and GPU, as evidenced by the empirical tests carried out. Since these functions are repeated at each iteration of the optimization process, the speed-up reached by the proposed approach in the overall registration procedure will be more significant whenever those functions —and more specifically, operations in which gradients are involved— constitute a major bottleneck.

The aim of this work is to provide an efficient alternative to compute FFDs for any application of elastic image registration, including multimodal scenarios, based on the convolution operator —which in turn is very efficiently implemented in many platforms— instead of on the tensor product operator. Hence, we will not focus on the comparative analysis of accuracy —or advantages and disadvantages— of different metrics. As a proof of concept, 2D pairwise (PW) image registration has been explored on contrast-enhanced first-pass perfusion cardiac magnetic resonance images (MRI). Our formulation applies naturally on 3D applications, as we showed in [5].

The remainder of this paper is structured as follows. Section 2 describes the methods. First, in Sect. 2.1, 2D PW image registration based on FFDs is revised and spatial transformations are reformulated in terms of convolutions. Then, Section 2.2 introduces NCC, MI and NMI. As stated above, the efficient convolutional approach can also be applied to calculate the regularization term, as Section 2.3 describes. Gradient evaluation is described in Sect. 2.4). Results are shown and discussed in Section 3. Section 4 summarizes the main conclusions of the paper. Finally, Appendix A provides a detailed description of gradient calculations for the three multimodal metrics analyzed in this work.

2. Methods

2.1. Convolutional pairwise registration

In the case of PW image registration, two images are to be spatially aligned. In particular, a moving image (I_M) is registered to a fixed image (I_F) —or reference image— by means of a spatial transformation $\mathbf{T}_{\boldsymbol{\theta}}$, parameterized by $\boldsymbol{\theta}$, that maps each point in I_F into the corresponding point in I_M (see Fig. 1).

The registration process consists in finding the parameters $\boldsymbol{\theta}$, which define the optimal spatial transformation $\mathbf{T}_{\boldsymbol{\theta}}$. Therefore, the problem is posed as an optimization problem where a cost function is minimized. Typically, the registration cost function is defined as

$$\mathcal{C}(\boldsymbol{\theta}) = -\mathcal{S}(I_F(\mathbf{x}), I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}))) + \lambda \cdot \mathcal{R}(\boldsymbol{\theta}) \tag{1}$$

The first term in Eq. (1) represents the cost associated with image similarity, i.e. the degree of alignment between both the fixed image and the transformed moving image; the minus sign converts similarity into a cost, so higher similarity implies less cost. In addition, to constrain the deformation to be realistic, a regularization term $\mathcal{R}(\boldsymbol{\theta})$, weighted by a λ factor, is included. The solution to this equation in $\boldsymbol{\theta}$ is usually found by iterative optimization procedures.

This paper focuses on elastic registration based on the FFD paradigm, which consists in locally deforming a given image by manipulating a grid of control points distributed across the image at an arbitrary mesh resolution [1]. B-spline functions are commonly used to define transformations; specifically, third-order B-spline basis functions provide a good balance between function smoothness and region support [14]. Thus, FFD-based registration defines $\mathbf{T}_{\boldsymbol{\theta}}$ through cubic B-spline interpolation from a lattice of control points $\mathbf{u}$ by the following expression [1,9]

$$\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{x} + \sum_{\mathbf{u}\in\mathcal{N}(\mathbf{x})} \left(\prod_{l=1}^{L} B\left(\frac{x_l - p_{u_l}}{\Delta_l} \right) \right) \cdot \boldsymbol{\theta}_{\mathbf{u}} \tag{2}$$

where $\mathbf{x}$ represents a spatial point in the image domain $\mathcal{X}$ of dimension L, $\mathcal{N}(\mathbf{x})$ is the set of all indices of control points in the vicinity of $\mathbf{x}$, $\boldsymbol{\theta}_{\mathbf{u}}$ represents the control point displacements, p_{u_l} is the location of a given control point in $\mathcal{X}$ along dimension l, Δ_l is the spacing in pixels between control points along dimension l, and B stands for the third order B-spline function obtained through the Cox-DeBoor recursion formula as defined in [27].

As introduced in [5], on the basis of the compact support of B-spline functions and by assuming the FFD defined on a discrete Cartesian coordinate system, the classical tensor product formulation (Eq. (2)) can be implemented by means of convolutions. Thus, for a 2D domain problem, the spatial coordinates $\mathbf{x} = (x_1, x_2) \in \mathcal{X}$ take on integer values, $1 \le x_1 \le N_1$ and $1 \le x_2 \le N_2$, with N_l the image size along dimension l. Furthermore, we assume the grid of control points $\mathbf{u} = (u_1, u_2)$, with $\boldsymbol{\Delta} = (\Delta_1, \Delta_2) \in \mathbb{Z}^{2^+}$ spacing, lo-

cated on a subset of the (N_1, N_2) points. Then, the 2D transformation can be expressed as

$$\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{x} + \sum_{q_1=1}^{N_1} B\left(\frac{x_1 - q_1}{\Delta_1}\right) \sum_{q_2=1}^{N_2} B\left(\frac{x_2 - q_2}{\Delta_2}\right) \cdot \boldsymbol{\Pi}(q_1, q_2) \tag{3}$$

where

$$\boldsymbol{\Pi}(q_1, q_2) = \delta(q_1 - \rho_1(q_1)) \cdot \delta(q_2 - \rho_2(q_2)) \cdot \boldsymbol{\theta}_{(\eta_1(q_1), \eta_2(q_2))} \tag{4}$$

with δ denoting the Kronecker delta. Functions ρ_l and η_l are defined for convenience to cancel the contribution of any point q_l that is not a control point and to select the appropriate displacement

$$\rho_l(q_l) = \begin{cases} q_l & \text{if } \exists u_l : q_l = p_{u_l} \\ 1/2 & \text{otherwise} \end{cases} \tag{5}$$

$$\eta_l(q_l) = \begin{cases} u_l & \text{if } q_l = p_{u_l} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Therefore, the 2D transformation can be reformulated as two consecutive 1D discrete convolutions with zero extension along each spatial dimension

$$\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{x} + B[x_1/\Delta_1] * B[x_2/\Delta_2] * \boldsymbol{\Pi}[x_1, x_2] \tag{7}$$

Furthermore, because of the definition of $\boldsymbol{\Pi}$ (see Eq. (7)), the first convolution implemented can be evaluated only in those rows (or columns) containing control points. The computational complexity of the proposed convolutional formulation is detailed in [5] for 3D transformations.

2.2. Multimodal image similarity metrics

Three different image similarity metrics are described here for multimodal pairwise registration, namely, NCC, MI and NMI.

2.2.1. Normalized cross correlation

The normalized cross correlation (NCC) between the fixed image, $I_F(\mathbf{x})$, and the transformed moving image, $I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}))$, is defined by

$$\text{NCC} = \frac{\sum_{\mathbf{x}\in\mathcal{X}} (I_F(\mathbf{x}) - \overline{I_F})(I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})) - \overline{I_M})}{\sqrt{\sum_{\mathbf{x}\in\mathcal{X}} (I_F(\mathbf{x}) - \overline{I_F})^2 \sum_{\mathbf{x}\in\mathcal{X}} (I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})) - \overline{I_M})^2}} = \frac{\mathcal{A}}{\mathcal{B}} \tag{8}$$

where $\overline{I_F}$ and $\overline{I_M}$ stand for the average gray-values of the corresponding images respectively, i.e., $\overline{I_F} = 1/|\mathcal{X}| \sum_{\mathbf{x}\in\mathcal{X}} I_F(\mathbf{x})$ and $\overline{I_M} = 1/|\mathcal{X}| \sum_{\mathbf{x}\in\mathcal{X}} I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}))$, where $|\mathcal{X}|$ is the number of pixels of the images.

2.2.2. Mutual information

In the PW approach, mutual information (MI) is defined from three entropy functions

$$\text{MI} = H(I_F(\mathbf{x})) + H(I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})) - H(I_F(\mathbf{x}), I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})) \tag{9}$$

where $H(\cdot)$ is the Shannon's entropy[1] Then, MI can be calculated by normalized histograms of the images,

$$\text{MI} = \sum_{f\in\mathcal{L}_F} \sum_{m\in\mathcal{L}_M} p(f, m|\boldsymbol{\theta}) \cdot log_2\left[\frac{p(f, m|\boldsymbol{\theta})}{p_F(f) \cdot p_M(m|\boldsymbol{\theta})}\right] \tag{10}$$

[1] Given the histogram, $p(i)$, of an image I, the Shannon's entropy is defined by $H(I) = -\sum_i p(i) log_2(p(i))$.

where $\mathcal{L}_F$ and $\mathcal{L}_M$ are discrete sets (bins) of intensity values of the fixed and moving images, respectively, with bin centres (f, m) regularly spaced; $p(f, m|\boldsymbol{\theta})$ stands for the joint histogram, which can by expressed as follows based on the Parzen-window method

$$p(f, m|\boldsymbol{\theta}) = \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x}\in\mathcal{X}} \left[K\left(\frac{f - I_F(\mathbf{x})}{\sigma_F}\right) \cdot K\left(\frac{m - I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}))}{\sigma_M}\right)\right] \tag{11}$$

where $K(\cdot)$ is a kernel function. In this case, we use the cubic B-spline kernel as in [28]. The scaling factors σ_F and σ_M in Eq. (11) are the intensity bin widths in $\mathcal{L}_F$ and $\mathcal{L}_M$. Note that if the number of bins is too small, the estimated joint histogram will have poor resolution, whereas computational cost increases with the number of bins. A good trade-off, empirically selected for medical images, is to use 64 bins to normalize both ranges of intensities [29].

The fixed and moving marginal histograms, $p_F(f)$ and $p_M(m|\boldsymbol{\theta})$ in Eq. (10), can be estimated by summing $p(f, m|\boldsymbol{\theta})$ on the intensity bins of the other image, i.e., over m and f, respectively.

$$p_F(f) = \sum_{m\in\mathcal{L}_M} p(f, m|\boldsymbol{\theta}) = \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x}\in\mathcal{X}} K\left(\frac{f - I_F(\mathbf{x})}{\sigma_F}\right) \tag{12}$$

$$p_M(m|\boldsymbol{\theta}) = \sum_{f\in\mathcal{L}_F} p(f, m|\boldsymbol{\theta}) = \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x}\in\mathcal{X}} K\left(\frac{m - I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}))}{\sigma_M}\right) \tag{13}$$

2.2.3. Normalized mutual information

The normalized mutual information (NMI) is defined as

$$\begin{aligned} \text{NMI} &= \frac{H(I_F(\mathbf{x})) + H(I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}))}{H(I_F(\mathbf{x}), I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}))} \\ &= \frac{\sum_{f\in\mathcal{L}_F} \sum_{m\in\mathcal{L}_M} p\left(f, m|\boldsymbol{\theta}\right) \cdot log_2\left[p_F(f) \cdot p_M\left(m|\boldsymbol{\theta}\right)\right]}{\sum_{f\in\mathcal{L}_F} \sum_{m\in\mathcal{L}_M} p\left(f, m|\boldsymbol{\theta}\right) \cdot log_2\left[p\left(f, m|\boldsymbol{\theta}\right)\right]} = \frac{\mathcal{D}}{\mathcal{E}} \end{aligned} \tag{14}$$

with the aforementioned joint and marginal histograms described in Eqs. (11)-(13). As proposed in [29], we use a specifically defined second order polynomial kernel function for NMI metric:

$$K(x) = \begin{cases} -1.8|x|^2 - 0.1|x| + 1, & \text{if } 0 \le |x| < 0.5 \\ 1.8|x|^2 - 3.7|x| + 1.9, & \text{if } 0.5 \le |x| \le 1 \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

2.3. Regularization term

As stated above, it is advisable to include a regularization term in the cost function to obtain realistic deformations. In this paper, the second order spatial derivatives have been used to define $\mathcal{R}(\boldsymbol{\theta})$ as

$$\begin{aligned} \mathcal{R}\left(\boldsymbol{\theta}\right) &= \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x}\in\mathcal{X}} \left(\left|\frac{\partial^2 \mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_1^2}\right|^2 + \left|\frac{\partial^2 \mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_2^2}\right|^2 \right) \\ &= \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x}\in\mathcal{X}} \left(\left|\mathrm{R}_1\left(\mathbf{x}; \boldsymbol{\theta}\right)\right|^2 + \left|\mathrm{R}_2\left(\mathbf{x}; \boldsymbol{\theta}\right)\right|^2 \right) \\ &= \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x}\in\mathcal{X}} \left\{ \mathrm{R}_{1,1}^2\left(\mathbf{x}; \boldsymbol{\theta}_{\boldsymbol{u},1}\right) + \mathrm{R}_{1,2}^2\left(\mathbf{x}; \boldsymbol{\theta}_{\boldsymbol{u},2}\right) + \mathrm{R}_{2,1}^2\left(\mathbf{x}; \boldsymbol{\theta}_{\boldsymbol{u},1}\right) \right. \\ &\quad \left. + \mathrm{R}_{2,2}^2\left(\mathbf{x}; \boldsymbol{\theta}_{\boldsymbol{u},2}\right) \right\} \end{aligned} \tag{16}$$

with $\mathrm{R}_{i,l}$ the l-th component of the vector function $\mathbf{R}_i(\mathbf{x}; \boldsymbol{\theta}) = \frac{\partial^2 \mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_i^2}$. Thus, we can use the same procedure applied for the reformulation of the transformation in Eqs. (3)-(7) to the partial derivatives in Eq. (16)

$$\mathrm{R}_{1,l}(\mathbf{x}; \boldsymbol{\theta}_{\boldsymbol{u},\boldsymbol{l}}) = \frac{1}{\Delta_1^2} \sum_{u_1=C_1^1}^{C_1^2} B''\left(\frac{x_1 - p_{u_1}}{\Delta_1}\right) \sum_{u_2=C_2^1}^{C_2^2} B\left(\frac{x_2 - p_{u_2}}{\Delta_2}\right) \cdot \boldsymbol{\theta}_{\mathbf{u},l} =$$

$$= \frac{1}{\Delta_1^2} B''[x_1/\Delta_1] * B[x_2/\Delta_2] * \mathbf{\Pi}_l[x_1, x_2] \tag{17}$$

where B'' stands for the second order derivative of the B-spline function, which can be calculated analytically. A similar expression can be obtained for the components of the partial derivative $\mathrm{R}_2(\mathbf{x}; \boldsymbol{\theta})$ in Eq. (16)

$$\mathrm{R}_{2,l}(\mathbf{x}; \boldsymbol{\theta}_{\boldsymbol{u},\boldsymbol{l}}) = \frac{1}{\Delta_2^2} B[x_1/\Delta_1] * B''[x_2/\Delta_2] * \mathbf{\Pi}_l[x_1, x_2] \tag{18}$$

Therefore, the convolutional formulation has been extended here for evaluating efficiently not only the transformations but also the regularization term.

2.4. Gradient evaluation

As previously stated, the optimal set of deformation parameters $\boldsymbol{\theta} = \{\theta_{\mathbf{u},l}\} = \{\theta_{(u_1,u_2),l}\}$, with $l = \{1, \ldots, L\}$ —see Eq. (2)—, are found by minimizing the cost function in Eq. (1)

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \mathcal{C}(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left(-\mathcal{S}(\boldsymbol{\theta}) + \lambda \cdot \mathcal{R}(\boldsymbol{\theta})\right) \tag{19}$$

Therefore, $\nabla\mathcal{C}(\boldsymbol{\theta})$ must be evaluated at each iteration of the optimization process. The derivative of the cost function with respect to each component $\theta_{u,l}$ is defined as

$$\frac{\partial \mathcal{C}(\boldsymbol{\theta})}{\partial \theta_{u,l}} = -\frac{\partial \mathcal{S}(\boldsymbol{\theta})}{\partial \theta_{u,l}} + \lambda \cdot \frac{\partial \mathcal{R}(\boldsymbol{\theta})}{\partial \theta_{u,l}} \tag{20}$$

Our formulation also permits gradient evaluation with convolutions. However, only the gradient of the registration metric was addressed in [5], i.e., the first term in Eq. (20), and it was fully developed only for the sum of squares differences, which is a monomodal metric. Here, the convolutional approach to compute the gradient is applied to other multimodal registration metrics that comply with a sufficient condition. Furthermore, we show how the same procedure can be used to evaluate the gradient of the regularization term (second term in Eq. (20)) as well.

2.4.1. Metric gradient

The use of convolutions as a way to implement the gradient of the similarity function ($\mathcal{S}(\boldsymbol{\theta})$) between the images to be registered is possible whenever the derivative of the registration metric with respect to each component $\theta_{u,l}$ of the transformation parameters includes terms with the following structure

$$\sum_{\mathbf{x}\in\mathcal{X}} \underbrace{\mathbf{\Lambda}_{\mathcal{S}}(I_F(\mathbf{x}), I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})), \mathbf{\Phi}_{\boldsymbol{\theta}}) \cdot \frac{\partial I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}))}{\partial T_{\theta_l}(\mathbf{x})}}_{\mathcal{V}_{\theta_l}(\mathbf{x})} \cdot \frac{\partial T_{\theta_l}(\mathbf{x})}{\partial \theta_{u,l}} \tag{21}$$

where $\mathbf{\Lambda}_{\mathcal{S}} = \frac{\partial \mathrm{S}}{\partial I_M}$, and $\mathbf{\Phi}_{\boldsymbol{\theta}}$ stands for a set of terms that are common $\forall x \in \mathcal{X}$. This is straightforwardly satisfied when the similarity function can be expressed as a function of terms that take the form

$$\sum_{\mathbf{x}\in\mathcal{X}} \mathrm{S}(I_F(\mathbf{x}), I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})), \mathbf{\Phi}_{\boldsymbol{\theta}}) \tag{22}$$

which is the case of SSD and NCC, as stated in [5]. However, other functions, such as MI and NMI do not satisfy Eq. (22), although Eq. (21) is indeed satisfied. A detailed formulation for NCC, MI and NMI is included in Appendix A to arrive at (21). From this expression, as we introduced in [5], it is possible to obtain the following convolutional formulation:

$$\frac{\partial \mathcal{S}(\boldsymbol{\theta})}{\partial \theta_{u,l}} = \sum_{\mathbf{x}\in\mathcal{X}} \mathcal{V}_{\theta_l}(\mathbf{x}) B\left(\frac{x_1 - p_{u_1}}{\Delta_1}\right) B\left(\frac{x_2 - p_{u_2}}{\Delta_2}\right) =$$

$$\sum_{x_1=s_{11}}^{s_{12}} B\left(\frac{x_1 - p_{u_1}}{\Delta_1}\right) \underbrace{\sum_{x_2=s_{21}}^{s_{22}} B\left(\frac{x_2 - p_{u_2}}{\Delta_2}\right) \mathcal{V}_{\theta_l}(x_1, x_2)}_{\mathbf{\Psi}} =$$

$$\underbrace{\sum_{x_1=s_{11}}^{s_{12}} B\left(\frac{x_1 - p_{u_1}}{\Delta_1}\right) \mathbf{\Psi}(x_1, p_{u_2})}_{\mathbf{\Omega}} \tag{23}$$

where $s_{l1} = p_{u_l} - R_l$, and $s_{l2} = p_{u_l} + R_l$, with R_l the influence radius of the control points. Functions $\mathbf{\Psi}$ and $\mathbf{\Omega}$ correspond to discrete 1D convolutions at control point $\mathbf{u}$, in coordinates x_2 and x_1 respectively. Function $\mathcal{V}_{\theta_l}(\mathbf{x})$ has been defined in Eq. (21). Hence:

$$\frac{\partial \mathcal{S}(\boldsymbol{\theta})}{\partial \theta_{u,l}} = B\left[\frac{x_1}{\Delta_1}\right] * B\left[\frac{x_2}{\Delta_2}\right] * \mathcal{V}_{\theta_l}[x_1, x_2]\Big|_{(p_{u_1}, p_{u_2})} \tag{24}$$

2.4.2. Regularization term gradient

As for the gradient of the regularization term $\mathcal{R}(\boldsymbol{\theta})$, similarly to the metric gradient, it can also be implemented with respect to the whole parameter set at once by means of convolutions. From the expressions defined in Eqs. (16)-(18)

$$\frac{\partial \mathcal{R}(\boldsymbol{\theta})}{\partial \theta_{u,l}} = \frac{2}{|\mathcal{X}|} \sum_{\mathbf{x}\in\mathcal{X}} \left(\mathrm{R}_{1,l}(\mathbf{x}; \boldsymbol{\theta}_{\boldsymbol{u},\boldsymbol{l}}) \frac{\partial \mathrm{R}_{1,l}(\mathbf{x}; \boldsymbol{\theta}_{\boldsymbol{u},\boldsymbol{l}})}{\partial \theta_{u,l}} + \mathrm{R}_{2,l}(\mathbf{x}; \boldsymbol{\theta}_{\boldsymbol{u},\boldsymbol{l}}) \frac{\partial \mathrm{R}_{2,l}(\mathbf{x}; \boldsymbol{\theta}_{\boldsymbol{u},\boldsymbol{l}})}{\partial \theta_{u,l}}\right) =$$

$$= \frac{2}{|\mathcal{X}|} \left(\sum_{\mathbf{x}\in\mathcal{X}} \mathrm{D}_{1,l}(\mathbf{x}; \boldsymbol{\theta}_{\mathbf{u},l}) + \sum_{\mathbf{x}\in\mathcal{X}} \mathrm{D}_{2,l}(\mathbf{x}; \boldsymbol{\theta}_{\mathbf{u},l})\right) \tag{25}$$

The components of the term $\sum \mathrm{D}_{1,l}$ can be expressed as

$$\sum_{\mathbf{x}\in\mathcal{X}} \mathrm{D}_{1,l}\left(\mathbf{x}; \boldsymbol{\theta}_{\mathbf{u},l}\right) =$$

$$\frac{1}{\Delta_1^2} \sum_{x_2=s_{21}}^{s_{22}} B\left(\frac{x_2 - p_{u_2}}{\Delta_2}\right) \underbrace{\sum_{x_1=s_{11}}^{s_{12}} B''\left(\frac{x_1 - p_{u_1}}{\Delta_1}\right) \mathrm{R}_{1,l}\left(x_1, x_2; \boldsymbol{\theta}_{\mathbf{u},l}\right)}_{\mathbf{Y}} =$$

$$\frac{1}{\Delta_1^2} \underbrace{\sum_{x_2=s_{21}}^{s_{22}} B\left(\frac{x_2 - p_{u_2}}{\Delta_2}\right) \mathbf{Y}(p_{u_1}, x_2)}_{\boldsymbol{\zeta}} =$$

$$\frac{1}{\Delta_1^2} B''\left[\frac{x_1}{\Delta_1}\right] * B\left[\frac{x_2}{\Delta_2}\right] * \mathrm{R}_{1,l}[x_1, x_2]\Big|_{(p_{u_1}, p_{u_2})} \tag{26}$$

Functions $\boldsymbol{\Upsilon}$ and $\boldsymbol{\zeta}$ are the results of 1D convolutions evaluated at control point locations, each of which along a different spatial dimension. A similar expression can be obtained for the components of the term $\sum \mathrm{D}_{2,l}$ in Eq. (25)

$$\sum_{\mathbf{x}\in\mathcal{X}} \mathrm{D}_{2,l}(\mathbf{x}; \boldsymbol{\theta}_{\mathbf{u},l}) = \frac{1}{\Delta_2^2} B\left[\frac{x_1}{\Delta_1}\right] * B''\left[\frac{x_2}{\Delta_2}\right] * \mathrm{R}_{2,l}[x_1, x_2]\Big|_{(p_{u_1}, p_{u_2})} \tag{27}$$

3. Results and discussion

All experiments have been run using *MATLAB R2020a* on a standard PC workstation with *Intel Core i7-4790 @ 3,6 GHz* processor and 16 GB RAM. GPU executions were carried out on a *nVIDIA Quadro RTX6000* using MATLAB CUDA capabilities with no further parallelization work from our implementation. Throughout the experiments, our aim is to compare FFD-based convolutional PW registration with the implementation based on tensor products, on both CPU and GPU.

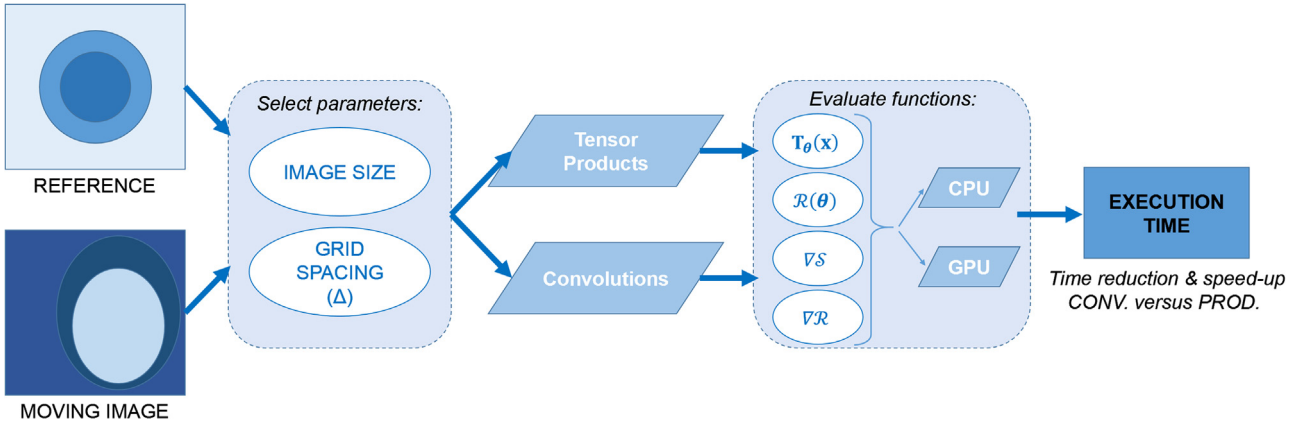

Fig. 2. Graph of the experiment for empirical measurement of execution times on CPU and GPU as a function of image size and control point spacing for both the tensor products approach and our convolutional proposal.

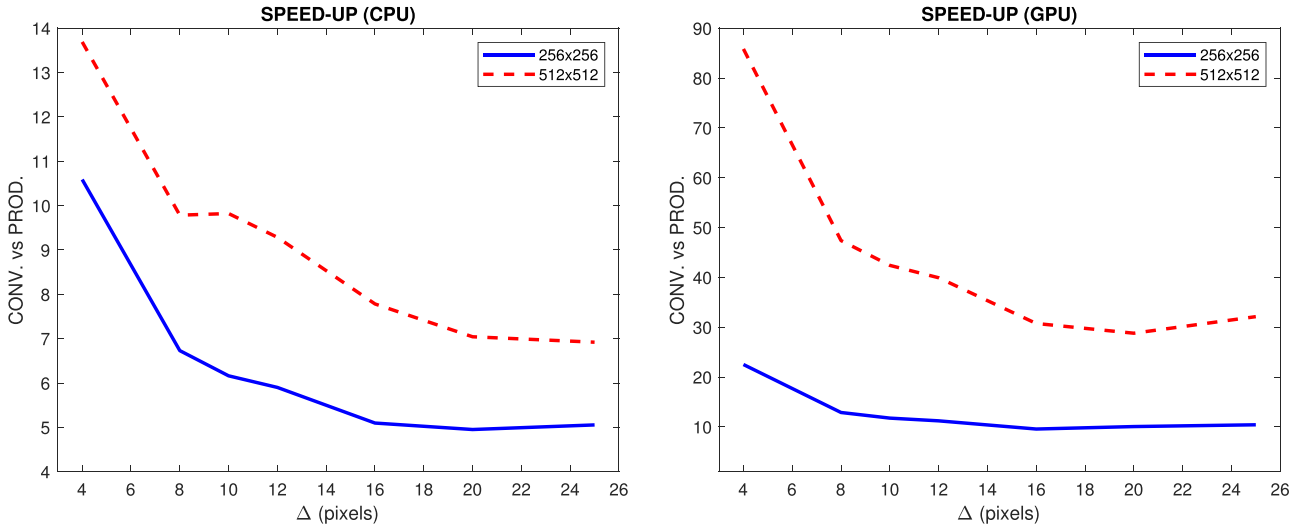

Fig. 3. Speed-up achieved using the proposed convolutional approach versus the tensor product approach (time prod./time conv.) for image sizes 256 × 256 (solid line) and 512 × 512 (dashed line) for different resolutions of control point grid: (a) speed-up from Matlab executions on CPU; (a) speed-up from Matlab executions on GPU.

3.1. Analysis of execution time as a function of image size and control point grid resolution

In this section we mean to gain insight on the computational savings that can be obtained from each component of our convolutional formulation. The core of our proposal consists of the spatial transformation, $\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})$, described in Eqs. (2)-(7), the regularization term, $\mathcal{R}(\boldsymbol{\theta})$, —Eqs. (16)-(18)—, the metric gradient, $\nabla\mathcal{S}$, —Eqs. (21)-(23)— and the gradient of the regularization term, $\nabla\mathcal{R}$, expressed in Eqs. (25)-(27). The actual metric used is involved in $\nabla\mathcal{S}$ through the term $\boldsymbol{\Lambda}_{\mathcal{S}}$.

In [5], the computational cost in terms of number of operations of both the tensor product and the convolutional approaches was analytically derived. The obtained cost showed that the number of products needed for the proposed approach depends on the image size as well as on the control point grid resolution and, consequently, on the number of control points. Bearing this in mind, the execution times of the reformulated functions have been empirically measured on CPU and GPU for different values of the involved parameters (see the graph of the experiment in Fig. 2). To this end, arbitrary images and control point grids are used to evaluate the functions reformulated by convolutions in isolation. In order to make our analysis independent of the actual metric used, we have used a random value for $\boldsymbol{\Lambda}_{\mathcal{S}}$. Only one execution of the functions involved has been carried out, i.e., no iterations until convergence of the cost functions are measured in this experiment.

Table 1 includes the measured execution times for the tensor product formulation (prod.) and the convolutional formulation (conv.) for one execution of functions $\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})$, $\mathcal{R}(\boldsymbol{\theta})$, $\nabla\mathcal{S}$ —with $\boldsymbol{\Lambda}_{\mathcal{S}}$ as indicated above— and $\nabla\mathcal{R}$. Different image sizes (256 and 512 pixels per dimension) and different control point spacings ($\Delta = \{8, 12\}$ pixels) —which leads to different sizes of the control point grid— have been considered. The execution times have been obtained by means of the *timeit* and *gputimeit* Matlab functions in CPU and GPU, respectively. Both functions call the specified function multiple times and returns the median of the time measurements. The figures in this table indicate that our proposal significantly reduces execution times of the four functions, both on CPU and GPU, showing a dependence on the problem magnitude, i.e., on the image and grid sizes. As can be seen, evaluations of both $\nabla\mathcal{R}$ and the $\nabla\mathcal{S}$ constitute the most time-consuming functions in the optimization, excluded metric calculation. According to these data, the greatest time reduction is achieved for $\nabla\mathcal{R}$, which also represents the slowest function.

Table 1
Empirical CPU and GPU execution times in milliseconds for those functions that can be reformulated in terms of convolution operations. Experiments include two sizes of image and two control point resolutions (Δ, in pixels) leading to four different grids of control points. Times for the classical tensor product formulation (prod.) and the proposed convolutional approach (conv.) are shown, as well as the percentage of time reduction (R) and speed-up achieved using convolutions.

Function	Img. Size	Δ	Grid	CPU Execution				GPU Execution			
				prod. (ms)	conv. (ms)	R	Speed-up	prod. (ms)	conv. (ms)	R	Speed-up
$\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})$	256 × 256	8	21 × 21	4.68	0.81	82.7%	×6	2.06	0.44	78.6%	×5
			31 × 31	5.06	0.90	82.2%	×6	2.10	0.43	79.5%	×5
	512 × 512	8	43 × 43	27.42	2.72	90.1%	×10	2.50	0.45	82.0%	×6
			63 × 63	28.18	3.18	88.7%	×9	2.47	0.45	81.8%	×5
$\mathcal{R}(\boldsymbol{\theta})$	256 × 256	8	21 × 21	6.95	1.59	77.1%	×4	2.30	0.90	60.9%	×3
			31 × 31	7.15	1.83	74.4%	×4	2.28	0.96	57.9%	×2
	512 × 512	8	43 × 43	36.83	4.53	87.7%	×8	3.07	0.99	67.8%	×3
			63 × 63	37.70	5.34	85.8%	×7	3.09	1.04	66.3%	×3
$\nabla\mathcal{S}$	256 × 256	8	21 × 21	9.61	2.93	69.5%	×3	6.47	0.63	90.3%	×10
			31 × 31	10.78	2.89	73.2%	×4	7.63	0.64	91.6%	×12
	512 × 512	8	43 × 43	45.57	10.31	77.4%	×4	34.21	0.67	98.0%	×51
			63 × 63	50.88	9.90	80.5%	×5	39.94	0.65	98.4%	×61
$\nabla\mathcal{R}$	256 × 256	8	21 × 21	14.60	1.06	92.7%	×14	11.35	0.63	94.4%	×18
			31 × 31	15.90	1.24	92.2%	×13	12.75	0.61	95.2%	×21
	512 × 512	8	43 × 43	79.20	3.96	95.0%	×20	63.87	0.62	99.0%	×103
			63 × 63	85.40	3.44	96.0%	×25	70.93	0.62	99.1%	×114

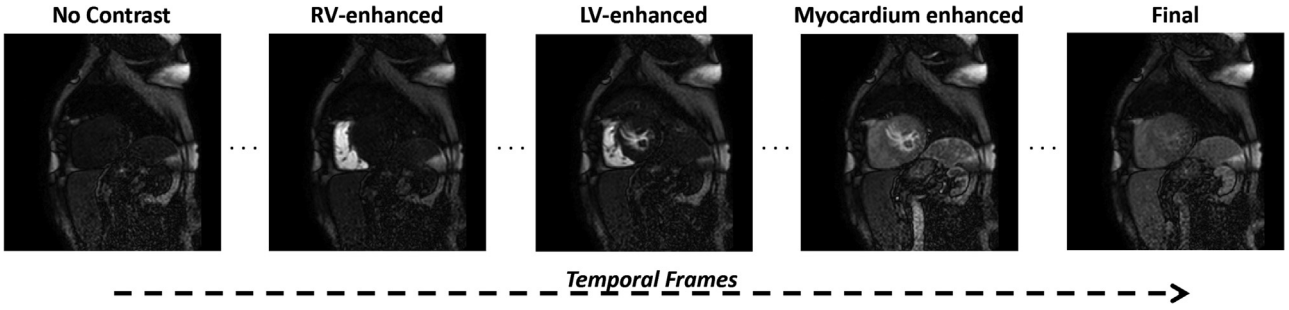

Fig. 4. Example of cardiac first-pass perfusion images. Short-axis view of a medial slice of the heart. From left to right: previous to contrast arrival; contrast reaches the right ventricle (RV) cavity; contrast reaches the left ventricle (LV) cavity; myocardium enhanced; and final frame of the dynamic perfusion sequence.

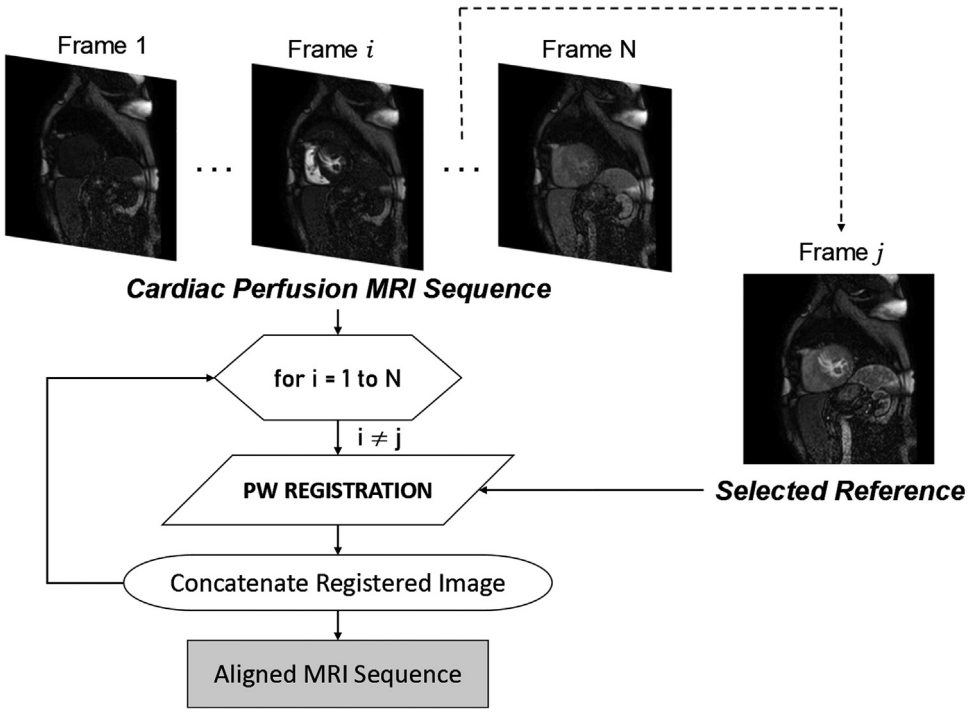

Fig. 5. Graph of the experiment for the alignment of 2D cardiac perfusion MRI sequences by pairwise image registration.

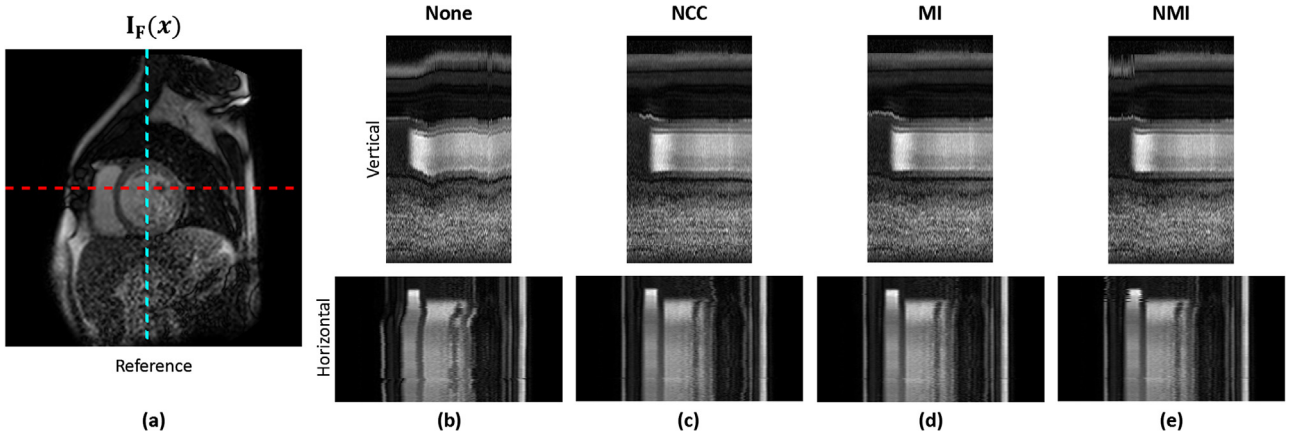

Fig. 6. Visual results for 2D sequential elastic pairwise (PW) registration of dynamic cardiac perfusion MRI: (a) Frame selected as optimal reference with vertical and horizontal profiles (dashed lines); (b) vertical and horizontal temporal profiles of the original sequence; (c) temporal profiles of the registered sequence using NCC metric; (d) temporal profiles of the registered sequence using MI metric; (e) temporal profiles of the registered sequence using NMI metric.

Additionally, Fig. 3 shows the speed-up achieved using convolutions instead of tensor products on CPU (Fig. 3a) and on GPU (Fig. 3b). These execution time accelerations have been empirically calculated from executions for different resolutions of the control point grid by considering all the functions included in Table 1. In Fig. 3, the variation of the convolutional improvement with respect to the image size and resolution of control point grid can be clearly appreciated. For an image size of 256 $\times$ 256 pixels, again without taking into account the time for metric evaluation, each iteration of the optimizer is at least 5 times faster using the proposed approach on CPU. The speed-up is even higher on GPU, with execution time improvements above $\times$10. These values are significantly increased for smaller values of $\boldsymbol{\Delta}$, i.e., as a denser control point grid is considered and, therefore, the computational load of the tensor product approach increases. A similar trend is observed for the case of an image size of 512 $\times$ 512 pixels. In this case, the execution time improvement is greater, with figures above $\times$7 and $\times$30 faster on CPU and GPU, respectively.

Although calculation of the registration metric has been excluded in the analysis presented here, it does play a key role in the computational efficiency of a registration procedure, as we show next.

3.2. 2D Cardiac perfusion magnetic resonance images

The multimodal scenario used in this paper to analyze the proposed convolutional formulation for FFD based registration is the alignment of contrast-enhanced first-pass perfusion cardiac MRI, hereinafter referred to as cardiac perfusion MRI.

Perfusion MRI uses a T_1-weighted acquisition sequence and requires the injection of a contrast agent that gives rise to a shortening of the T_1 relaxation time of the tissues. Thus, signal intensity increases when the contrast agent uptake by a given tissue takes place. Therefore, this translates to changes in the image intensities of the acquired image sequence (see Fig. 4). In particular, cardiac perfusion MRI is used to study the blood supply to the myocardium in order to detect and evaluate myocardial ischemia, for which a precise analysis of the temporal curves of contrast concentration in different regions of the myocardium is carried out. The evaluation of these perfusion curves requires the correct alignment of the image sequence, since several sources of motion, such as patient breathing and inaccuracies in the cardiac triggering during acquisition, may cause misalignment among the images.

The test dataset used here includes 12 cardiac perfusion MRI sequences acquired with a *Philips Intera* 1.5 T scanner using a fast field echo MAG prepared gradient recalled sequence with spatial presaturation. All images have 288 $\times$ 288 matrix size, whereas the number of frames ranges from 132 to 188 with spatial resolution from 1.250 to 1.840 mm^2, depending on the subject.

For each perfusion MRI study, the alignment procedure is carried out by registering each image frame to a preselected reference image sequentially. Therefore, a total of $N-1$ PW registrations are executed for each one of the perfusion sequences, with N the number of frames (see Fig. 5). For the reference selection, we use the unbiased template estimation method proposed in [30]. The spacing between control points is set to 15 mm, since this spacing showed the best results with this dataset in a previous work [31].

Furthermore, the optimizer used is the iterative non-linear conjugate gradient algorithm [32], with the maximum number of iterations set to 100. It is worth mentioning that other alternatives may be used to reduce the computational cost of the optimization, such as undersampling strategies and the adaptive stochastic gradient descent algorithm. However, these additional optimizations have not been considered in this work since we focus on the comparative analysis of the efficiency achieved using the proposed convolutional approach with respect to the formulation based on tensor products.

Figure 6 shows an example of the results of the sequential elastic PW registration of a cardiac perfusion MRI using the three multimodal metrics analyzed in this work: NCC, MI and NMI. Two temporal intensity profiles, marked with dashed lines on the reference image (Fig. 6a), are shown for the registered sequence using each metric (Figs. 6c-6 e). Furthermore, the same profiles from the original non registered sequence are also included, in which the misalignments between consecutive frames can be appreciated (Fig. 6b). Moreover, as Fig. 7 illustrates with an example, there are no significant differences between the results for the convolution-based PW registration and the classical implementation; the resulting aligned images, as well as deformations and the corresponding pixel displacements, are virtually identical. Fig. 8 supports this statement by showing boxplots of these differences; as can be seen, in all cases the interquartile ranges of the differences are below 0.2 pixels, and only one of the whiskers extends to the vicinity of 0.45 pixels, i.e., far less than half of a pixel.

Table 2 includes specific execution times for the sequential PW registration of each one of the cardiac perfusion MRI sequences using NCC, MI and NMI with the classical formulation based on tensor products and the proposed reformulation based on con-

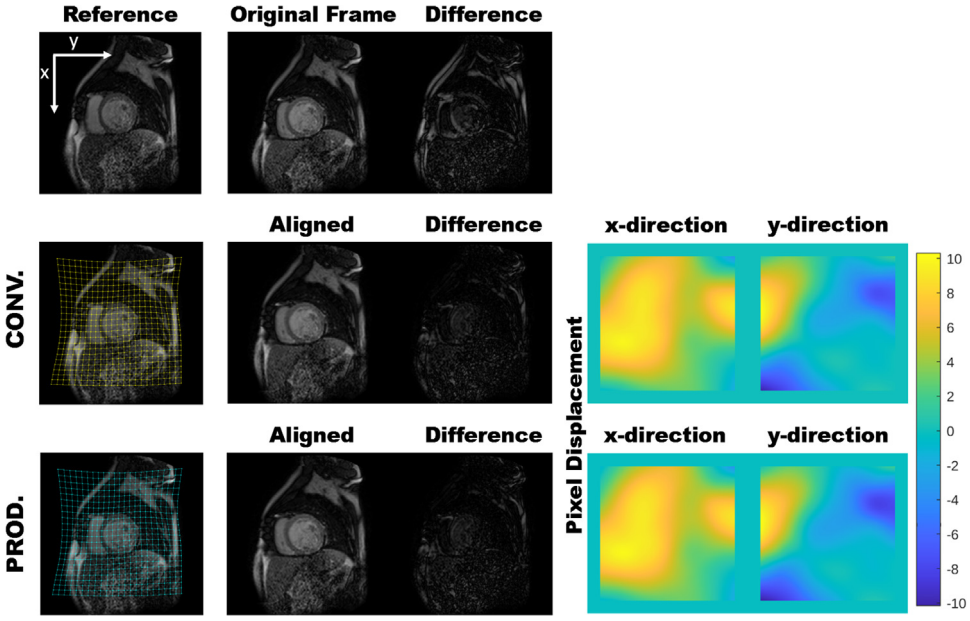

Fig. 7. Comparison between the proposed convolution-based registration and the classical tensor product methodology.

Table 2
Results for 2D sequential elastic pairwise (PW) registration of each cardiac perfusion MRI study (ID) by using three multimodal similarity metrics: NCC, MI and NMI. Matlab execution times in minutes on CPU and GPU devices (Dev.) for the proposed convolutional approach (conv.) and the tensor product formulation (prod.) of the overall registration procedure. Number of frames of each sequence (N); resolution of images in mm/pixel (Res.); grid size of control points (Grid).

ID	N	Res.	Grid	Dev.	NCC Metric		MI Metric		NMI Metric	
					prod.	conv.	prod.	conv.	prod.	conv.
1	13	91.250	23×23	CPU	4.29	1.12	75.41	69.45	173.3	165.0
				GPU	2.79	0.69	9.49	6.00	15.06	10.31
2	13	21.667	31 × 31	CPU	6.63	1.52	78.87	64.05	236.3	226.0
				GPU	4.44	0.94	10.70	6.68	21.32	14.89
3	15	91.424	25 × 25	CPU	5.51	1.42	90.82	86.59	209.5	197.3
				GPU	3.54	0.84	11.66	7.14	19.05	13.04
4	15	51.250	23 × 23	CPU	6.47	1.70	58.30	57.16	257.6	246.7
				GPU	4.39	1.07	7.61	4.81	22.68	16.00
5	15	91.389	25 × 25	CPU	9.00	2.24	119.1	112.2	253.7	204.4
				GPU	6.24	1.40	14.58	8.91	24.61	16.37
6	18	81.250	23 × 23	CPU	8.14	2.14	133.6	121.8	333.4	317.9
				GPU	5.39	1.29	16.73	10.17	26.47	18.49
7	13	91.354	23 × 23	CPU	4.42	1.12	85.53	82.27	206.9	200.8
				GPU	2.99	0.72	11.10	6.78	17.53	12.06
8	14	91.840	31 × 31	CPU	8.12	1.92	98.97	90.16	358.1	294.7
				GPU	5.70	1.19	12.28	7.67	32.35	21.75
9	18	81.562	29 × 29	CPU	12.50	2.81	94.40	85.58	375.4	356.8
				GPU	8.72	1.76	12.60	7.21	36.54	24.39
10	18	81.458	25 × 25	CPU	5.76	1.52	106.3	102.7	274.3	266.7
				GPU	3.93	0.95	13.46	8.49	24.44	16.40
11	13	51.424	25 × 25	CPU	5.11	1.26	79.61	68.96	206.0	184.9
				GPU	3.45	0.82	10.30	5.88	16.46	11.70
12	18	81.389	25 × 25	CPU	6.30	1.61	112.0	105.5	278.4	227.6
				GPU	4.21	0.99	14.15	8.66	25.49	17.98
Conv. Time Reduction				**CPU**	**75.23%**		**7.64%**		**8.67%**	
				GPU	**77.90%**		**38.90%**		**31.43%**	

volution operations, both on CPU and GPU. As can be seen, our convolutional approach accomplishes lower registration times for all the perfusion sequences and metrics on both CPU and GPU. However, the execution time reduction is not uniform among the three multimodal metrics used. The improvement of the convolutional approach in the overall registration time is lower for MI and NMI metrics, as indicated by the mean reduction percentages highlighted in the last two rows of the table. As discussed below, this is due to the fact that the bottleneck in registration using MI and NMI is the calculation of the metric itself. However, as for NCC, the bottleneck is the evaluation of those functions that can be efficiently reformulated by means of convolutions and, hence, a great speed-up is observed. On the other hand, the time reduction percentages of the convolutional approach with respect to the tensor product formulation (see last two rows in Table 2) indicate that the gain achieved on GPU is greater than the gain on CPU.

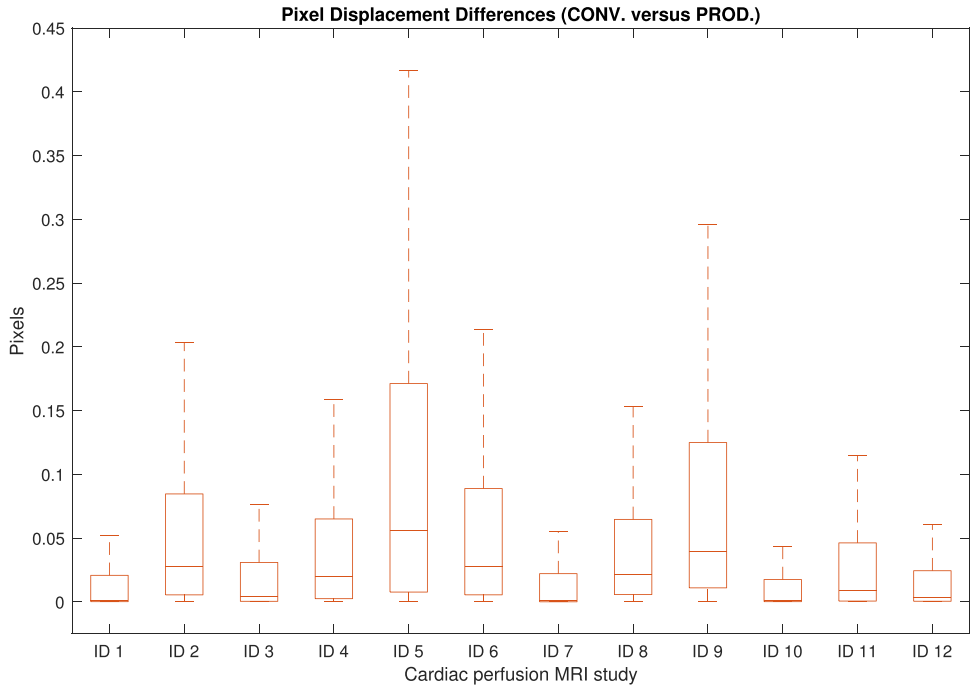

Fig. 8. Boxplot of differences (in pixels, for the set of frames in each sequence) between pixel displacements obtained with the proposed convolution-based algorithm and the classical tensor product methodology.

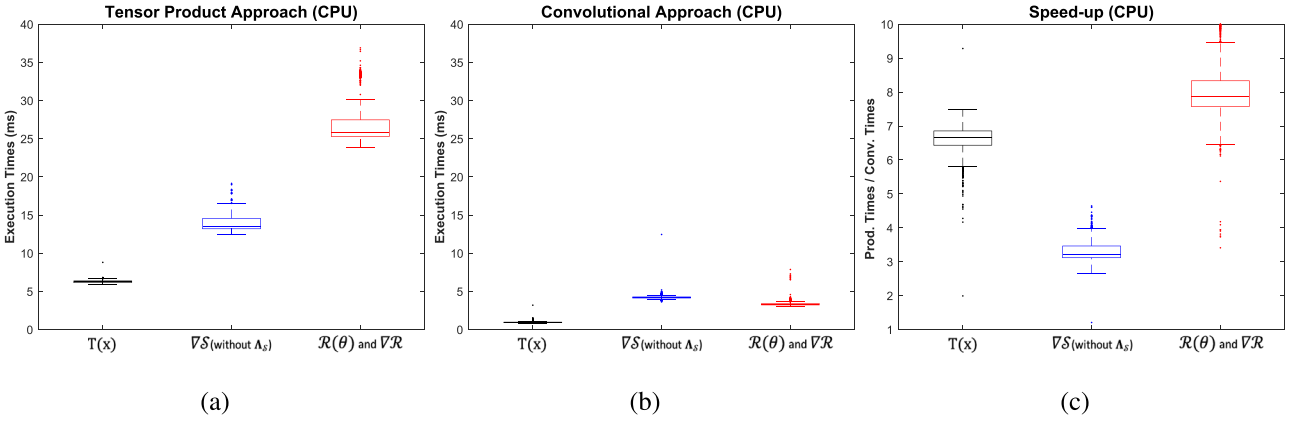

Fig. 9. Boxplots of CPU execution times in milliseconds for evaluating the spatial transformation (black), the metric gradient (blue) and the regularization term together with its gradient (red): (a) execution times for the tensor product approach; (b) execution times for the proposed convolutional approach; (c) speed-up achieved by the proposed reformulation (time prod. / time conv.). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

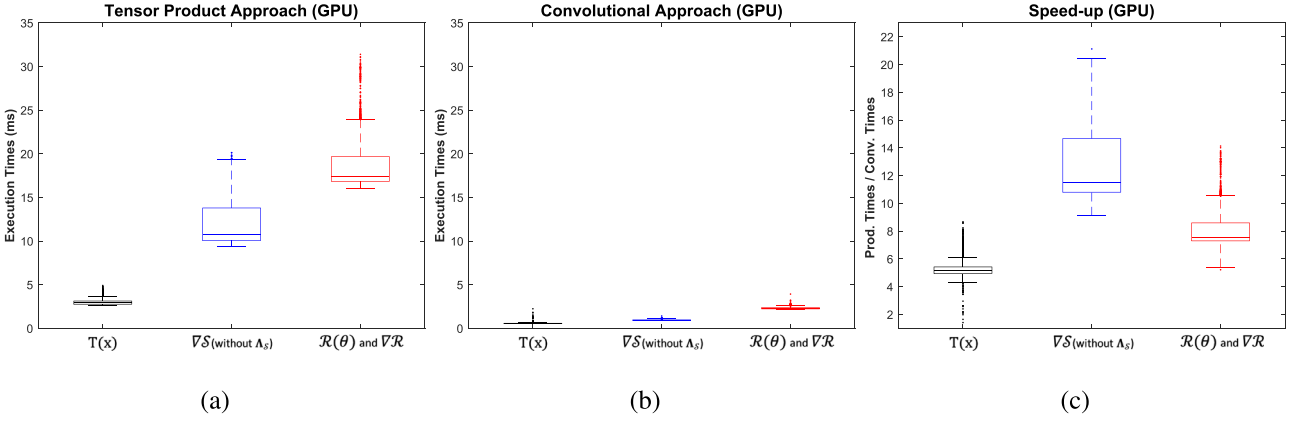

Fig. 10. Boxplots of GPU execution times in milliseconds for evaluating the spatial transformation (black), the metric gradient (blue) and the regularization term together with its gradient (red): (a) execution times for the tensor product approach; (b) execution times for the proposed convolutional approach; (c) speed-up achieved by the proposed reformulation (time prod. / time conv.). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

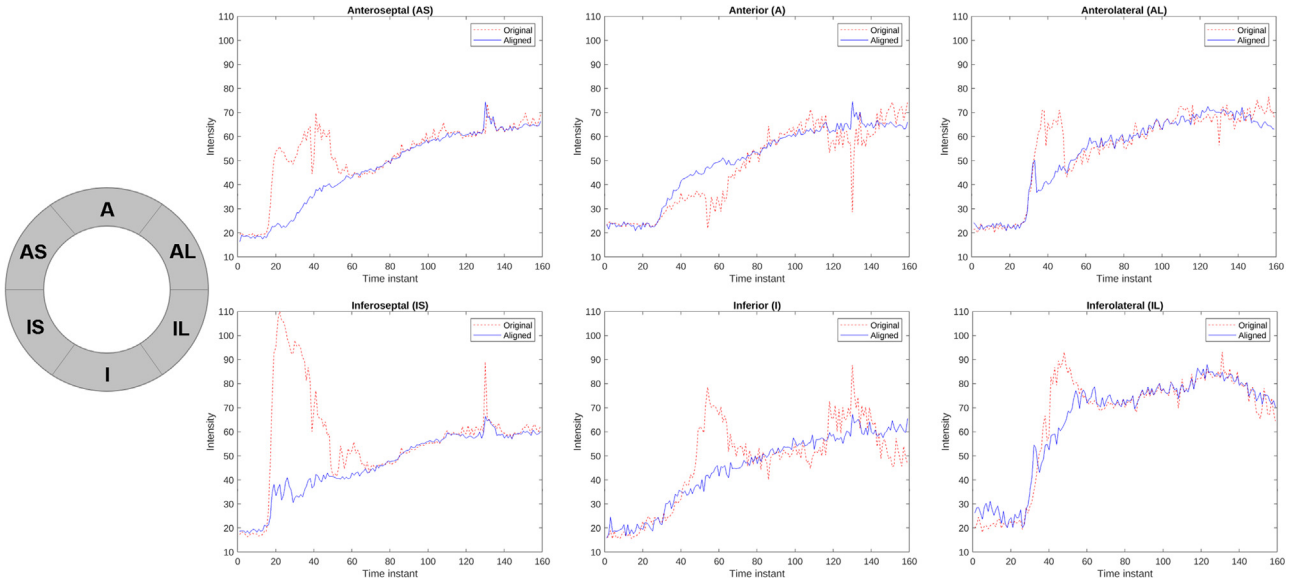

Fig. 11. Myocardial perfusion curves for the image in Fig. 6a. The left scheme shows the standard zones considered for myocardial perfusion analysis: anteroseptal (AS), anterior (A), anterolateral (AL), inferoseptal (IS), inferior (I), and inferolateral (IL). The figures on the right show the mean perfusion curves for each zone. The original curves (red dashed lines) are included, as well as the curves obtained after image registration using the proposed method with NCC metric (blue solid lines). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

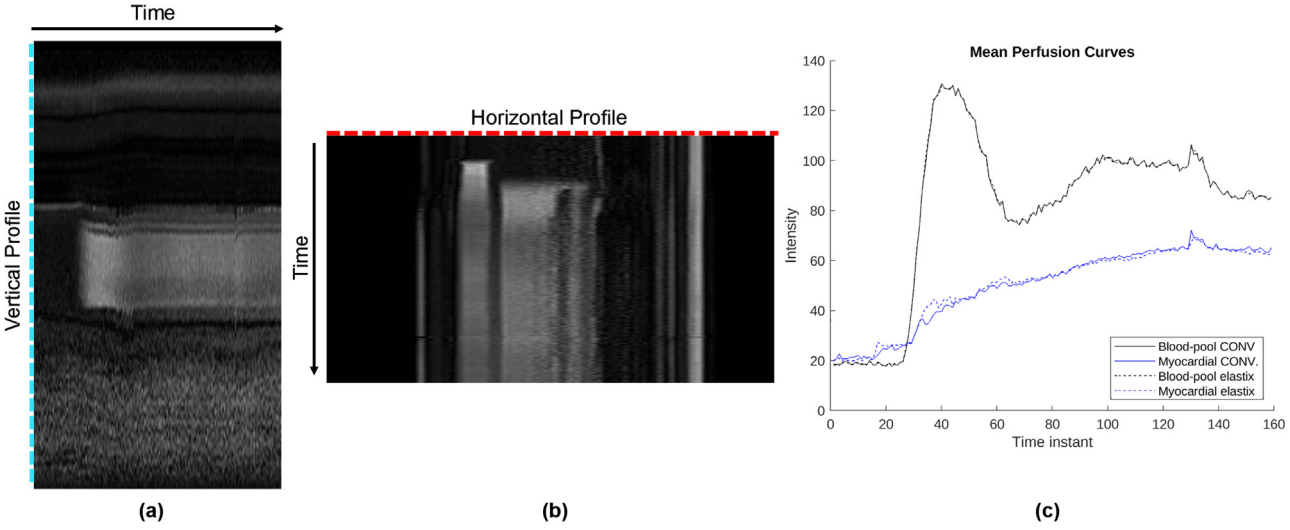

Fig. 12. Results for 2D sequential pairwise (PW) registration with the *Elastix* software using NCC metric: Temporal profiles of the registered sequence for comparison with the results in Fig. 6, (a) vertical profile and (b) horizontal profile; (c) mean perfusion curves at blood-pool (black) and myocardium (blue) obtained after registration using the proposed method (solid lines) and *Elastix* (dashed lines). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Moreover, Table 3 shows the mean times per PW image registration with the tensor product approach, using NCC, MI and NMI on CPU and GPU. The table includes the time percentages dedicated to evaluating the metric —including $\mathcal{S}$ and $\mathbf{\Lambda}_{\mathcal{S}}$ — (Metric), the spatial transformations, $\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})$, (Transf.), the metric gradients, $\nabla\mathcal{S}$ —excluding calculation of $\mathbf{\Lambda}_{\mathcal{S}}$, since it is included in the metric time, as commented above— (Grad.), the regularization term $\mathcal{R}(\boldsymbol{\theta})$, and its gradient $\nabla\mathcal{R}$, (Reg.). The percentage of total time that can be reduced by means of the proposed convolutional formulation is indicated in each case (Reducible). As stated above, these data show that the weight of the metric evaluation in the overall registration time is larger for MI and NMI, specially on CPU.

Figures 9 and 10 show boxplots of the corresponding execution times per call to the functions of interest. These functions are the following: (i) evaluation of spatial transformations (black boxes), i.e., $\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})$; (ii) calculation of the metric gradient with respect to the set of control point displacements (blue boxes), except for the term $\mathbf{\Lambda}_{\mathcal{S}}$, corresponding to the above mentioned $\nabla\mathcal{S}$; (iii) and the estimation of the regularization term $\mathcal{R}$ and its gradient $\nabla\mathcal{R}$ with respect to $\boldsymbol{\theta}_{\boldsymbol{u}}$ (red boxes). Note that for this analysis the mean execution times of the aforementioned functions for each PW registration done during the registration process of all images (ID = 1, ..., 12) using the three metrics under study have been considered. Therefore, each boxplot has been created using 5721 samples of time data, corresponding to $\sum_{i=1}^{12} 3 \times (N_i - 1)$. Figs. 9(a) and 10(a) show the function execution times for the tensor product approach in CPU and GPU executions, respectively; whereas Figs. 9(b) and 10(b) show the corresponding times with the proposed convolutional approach. Additionally, Figs. 9(c) and 10(c) illustrate the speed-up achieved by the proposed reformulation based on simple convolution operations with respect to the tensor product approach, i.e., prod. running time / conv. running

Table 3
Mean times per pairwise (PW) registration for the cardiac perfusion MRI dataset using the classical tensor product formulation. Total PW registration time in seconds (Total); percentage of total time dedicated to evaluating the registration metric $\mathcal{S}$ and its gradient with respect to the transformed image $\mathbf{\Lambda}_{\mathcal{S}}$ (Metric); percentage of total time for spatial transformations $\mathbf{T}(\mathbf{x})$ (Transf.); percentage of total time for calculation of metric gradient $\nabla\mathcal{S}$ with respect to the set of control point displacements, $\boldsymbol{\theta}_{\boldsymbol{u}}$, excluding $\mathbf{\Lambda}_{\mathcal{S}}$ (Grad.); percentage of total time for evaluating regularization term $\mathcal{R}$ and its gradient $\nabla\mathcal{R}$ with respect to $\boldsymbol{\theta}_{\boldsymbol{u}}$ (Reg.); percentage of total time that can be reduced using the proposed convolutional approach (Reducible). Percentages do not sum to 100 since additional operations (e.g., interpolations) have not been accounted for.

Mean Times / Frame	NCC		MI		NMI	
	CPU	GPU	CPU	GPU	CPU	GPU
Total (seconds)	2.588	1.755	35.643	4.552	99.513	8.873
% Metric ($\mathcal{S}$, $\mathbf{\Lambda}_{\mathcal{S}}$)	1.8	5.0	91.6	52.1	96.2	65.0
% Transf. ($\mathbf{T}(\mathbf{x})$)	12.6	7.9	1.1	4.4	0.5	2.6
% Grad. ($\nabla\mathcal{S}$)	27.5	29.5	2.3	14.7	1.0	12.5
%Reg. ($\mathcal{R}$, $\nabla\mathcal{R}$)	51.9	49.0	4.4	24.3	2.0	17.3
Reducible	**92.0%**	**86.4%**	**7.8%**	**43.4%**	**3.5%**	**32.4%**

time. As expected, values depicted in these boxplots agree with those previously showed in Sect. 3.1 (see Table 1), and execution time reduction using convolutions is extremely significant for each task separately. However, it is important to note that efficient implementations of MI and NMI are crucial to take full advantage of the proposed approach when these registration metrics are used, although those optimizations are not the goal of this paper.

As we have already mentioned throughout this paper, a major limitation of the conventional image registration methods is the execution time, and our proposal is aimed at lightening the computational cost. The above tests prove the efficiency of the convolutional formulation. However, registration accuracy is also an important aspect to take into account for the validation of the proposed methodology. Thus, the perfusion curves corresponding to the image in Fig. 6(a) have also been analyzed. Fig. 11 shows the curves representing mean image intensities along time at the six standard zones considered for myocardial perfusion analysis, namely, anteroseptal (AS), anterior (A), anterolateral (AL), inferoseptal (IS), inferior (I), and inferolateral (IL). These regions have been defined on a segmentation of the reference image. The figures include both the original curves obtained before registration and the curves obtained from the registered sequence for their comparison. As can be seen, the appearance of both curves differs significantly. The motion reduction can be observed in the smoother curves obtained after image registration (see blue solid lines). The remaining fluctuations in the curves after alignment are due to the image noise.

In addition, the perfusion MRI sequences were also registered using Elastix[2] in order to compare our results with those obtained by a robust and widely validated registration tool. In this case, the Elastix parameter file has been adapted to perform PW registration of 2D images by using NCC metric, random coordinate undersampling strategy and adaptive stochastic gradient descent optimization with a single resolution level, and without any regularization term. Fig. 12 shows an example of the results obtained by the Elastix registration. As can be observed, there are no visible differences with the results of the proposed registration method. Therefore, we can conclude that both approaches achieve similar registration accuracy. As for the registration times, the comparison of the proposed method with Elastix may be not fair, since it is executed from a compiled language, while our routines are executed in Matlab. Having said that, the mean Elastix runtime per each image pair registered is 16.88 seconds, whereas the mean registration time of the proposed convolutional CPU implementation in Matlab is 0.64 seconds, which represents a reduction in registration time above 96% for this configuration. However, further optimizations could probably be done for the Elastix experiments.

[2] http://elastix.isi.uu.nl/

4. Conclusions

This paper describes a multimodal convolution-based formulation for efficient pairwise image registration with FFDs. In this case, we been explored this type of registration on contrast-enhanced first-pass perfusion cardiac MRI. In particular, we show how the gradient formulation by convolution operations can be adapted to three different multimodal metrics: NCC, MI and NMI. This, added to the convolutional formulation of the spatial transformations as well as the regularization term and its gradient —functions that are independent of the metric used— allow us to achieve significant execution time reductions on the reformulated functions on CPU and GPU. Therefore, the speed-up achieved by the proposed convolutional FFD approach in the overall registration procedure is specially outstanding when the bottleneck of the optimization process is the calculation of gradients. However, the improvement will be smaller when the major part of registration time is spent in the evaluation of the registration metric itself, as it is the case with metrics based on mutual information. Despite this, the proposed approach constitutes an alternative of interest for platforms in which the convolution operator is highly optimized (as it is the case of Matlab) for any application of elastic registration, also for multimodal images, and specially for processing large datasets or time series.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Rosa-María Menchón-Lara: Conceptualization, Methodology, Software, Investigation, Writing – original draft, Visualization. **Federico Simmross-Wattenberg:** Resources, Data curation, Writing – review & editing. **Manuel Rodríguez-Cayetano:** Writing – review & editing. **Pablo Casaseca-de-la-Higuera:** Visualization, Writing – review & editing. **Miguel Á. Martín-Fernández:** Writing – review & editing. **Carlos Alberola-López:** Conceptualization, Formal analysis, Writing – original draft, Supervision, Project administration.

Acknowledgments

This work is partially supported by the Spanish '*Ministerio de Economía, Industria y Competitividad*' under grants TEC2017-82408-R and PID2020-115339RB-I00, and by ESAOTE Ltd under grant 18IQBM.

Appendix A. Gradients of Multimodal Registration Metrics

A1. NCC Gradient

From the definition of NCC in Eq. (8)

$$\frac{\partial \mathrm{NCC}}{\partial \theta_{u,l}} = \frac{\frac{\partial \mathcal{A}}{\partial \theta_{u,l}}\mathcal{B} - \mathcal{A}\frac{\partial \mathcal{B}}{\partial \theta_{u,l}}}{\mathcal{B}^2} = \frac{1}{\mathcal{B}}\frac{\partial \mathcal{A}}{\partial \theta_{u,l}} - \frac{\mathcal{A}}{\mathcal{B}^2}\frac{\partial \mathcal{B}}{\partial \theta_{u,l}} \tag{A.1}$$

The partial derivative of $\mathcal{A}$ can be expressed as

$$\frac{\partial \mathcal{A}}{\partial \theta_{u,l}} = \sum_{\mathbf{x}\in\mathcal{X}} (I_F(\mathbf{x}) - \overline{I_F}) \frac{\partial I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}))}{\partial T_{\theta_l}(\mathbf{x})} \frac{\partial T_{\theta_l}(\mathbf{x})}{\partial \theta_{u,l}} - \sum_{\mathbf{x}\in\mathcal{X}} (I_F(\mathbf{x}) - \overline{I_F}) \overline{\nabla I_M} \frac{\partial T_{\theta_l}(\mathbf{x})}{\partial \theta_{u,l}} =$$

$$= \sum_{\mathbf{x}\in\mathcal{X}} (I_F(\mathbf{x}) - \overline{I_F}) \frac{\partial I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}))}{\partial T_{\theta_l}(\mathbf{x})} \frac{\partial T_{\theta_l}(\mathbf{x})}{\partial \theta_{u,l}} - \sum_{\mathbf{x}\in\mathcal{X}} I_F(\mathbf{x}) \overline{\nabla I_M} \frac{\partial T_{\theta_l}(\mathbf{x})}{\partial \theta_{u,l}} + |\mathcal{X}| \overline{I_F} \cdot \overline{\nabla I_M} \frac{\partial T_{\theta_l}(\mathbf{x})}{\partial \theta_{u,l}}$$

where the last two terms are cancelled out. Whereas the derivative of $\mathcal{B}$ is obtained by

$$\frac{\partial \mathcal{B}}{\partial \theta_{u,l}} = \frac{\mathcal{F}}{\mathcal{B}} \sum_{\mathbf{x}\in\mathcal{X}} (I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})) - \overline{I_M}) \frac{\partial I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}))}{\partial T_{\theta_l}(\mathbf{x})} \frac{\partial T_{\theta_l}(\mathbf{x})}{\partial \theta_{u,l}}$$

where $\mathcal{F} = \sum_{\mathbf{x}\in\mathcal{X}} (I_F(\mathbf{x}) - \overline{I_F})^2$. Therefore, the gradient of the NCC metric takes on the following form

$$\frac{\partial \mathrm{NCC}}{\partial \theta_{u,l}} = \sum_{\mathbf{x}\in\mathcal{X}} \boldsymbol{\Lambda}_{\mathrm{NCC}}(I_F(\mathbf{x}), I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})), \mathcal{A}, \mathcal{B}, \mathcal{F}) \frac{\partial I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}))}{\partial T_{\theta_l}(\mathbf{x})} \frac{\partial T_{\theta_l}(\mathbf{x})}{\partial \theta_{u,l}} \tag{A.2}$$

with

$$\boldsymbol{\Lambda}_{\mathrm{NCC}}(I_F(\mathbf{x}), I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})), \mathcal{A}, \mathcal{B}, \mathcal{F}) = \frac{1}{\mathcal{B}}(I_F(\mathbf{x}) - \overline{I_F}) - \mathcal{A}\frac{\mathcal{F}}{\mathcal{B}^3}(I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})) - \overline{I_M}) \tag{A.3}$$

A2. MI Gradient

Departing from the definition of MI similarity metric in Eq. (10)

$$\mathrm{MI} = \underbrace{\sum_{f,m}\left[p(f,m|\boldsymbol{\theta})\log_2(p(f,m|\boldsymbol{\theta}))\right]}_{\mathrm{MI}_1} - \underbrace{\sum_{f,m}\left[p(f,m|\boldsymbol{\theta})\log_2(p_F(f))\right]}_{\mathrm{MI}_2} - \underbrace{\sum_{f,m}\left[p(f,m|\boldsymbol{\theta})\log_2(p_M(m|\boldsymbol{\theta}))\right]}_{\mathrm{MI}_3}$$

And the partial derivatives of the terms in the expression above with respect to a component $\theta_{u,l}$ of the transformation parameters $\boldsymbol{\theta}$ are obtained by the following expressions:

$$\frac{\partial \mathrm{MI}_1}{\partial \theta_{u,l}} = \sum_{f,m}\left[\log_2(p(f,m|\boldsymbol{\theta})) + \mu\right]\frac{\partial p(f,m|\boldsymbol{\theta})}{\partial \theta_{u,l}}$$

$$\frac{\partial \mathrm{MI}_2}{\partial \theta_{u,l}} = \sum_{f,m}\log_2(p_F(f))\frac{\partial p(f,m|\boldsymbol{\theta})}{\partial \theta_{u,l}} = \sum_{f}\log_2(p_F(f))\sum_{m}\frac{\partial p(f,m|\boldsymbol{\theta})}{\partial \theta_{u,l}} =$$

$$= \sum_{f}\log_2(p_F(f))\frac{\partial \sum_m p(f,m|\boldsymbol{\theta})}{\partial \theta_{u,l}} = \sum_{f}\log_2(p_F(f))\frac{\partial p_F(f)}{\partial \theta_{u,l}} = 0$$

$$\frac{\partial \mathrm{MI}_3}{\partial \theta_{u,l}} = \sum_{f,m}\left[\log_2(p_M(m|\boldsymbol{\theta}))\frac{\partial p(f,m|\boldsymbol{\theta})}{\partial \theta_{u,l}} + \mu\frac{p(f,m|\boldsymbol{\theta})}{p_M(m|\boldsymbol{\theta})}\frac{\partial p_M(m|\boldsymbol{\theta})}{\partial \theta_{u,l}}\right] =$$

$$= \sum_{f,m}\left[\log_2(p_M(m|\boldsymbol{\theta}))\frac{\partial p(f,m|\boldsymbol{\theta})}{\partial \theta_{u,l}}\right]$$

$$+\mu\sum_{m}\frac{1}{p_M(m|\boldsymbol{\theta})}\frac{\partial p_M(m|\boldsymbol{\theta})}{\partial \theta_{u,l}}\sum_{f}p(f,m|\boldsymbol{\theta}) =$$

$$= \sum_{f,m}\left[\log_2(p_M(m|\boldsymbol{\theta}))\frac{\partial p(f,m|\boldsymbol{\theta})}{\partial \theta_{u,l}}\right] + \mu\sum_{m}\frac{\partial \sum_f p(f,m|\boldsymbol{\theta})}{\partial \theta_{u,l}} =$$

$$= \sum_{f,m}\left[\log_2(p_M(m|\boldsymbol{\theta})) + \mu\right]\frac{\partial p(f,m|\boldsymbol{\theta})}{\partial \theta_{u,l}}$$

where $\mu = \log_2(e)$. Therefore,

$$\frac{\partial \mathrm{MI}}{\partial \theta_{u,l}} = \sum_{f,m}\underbrace{\log_2\left(\frac{p(f,m|\boldsymbol{\theta})}{p_M(m|\boldsymbol{\theta})}\right)}_{\mathcal{G}}\frac{\partial p(f,m|\boldsymbol{\theta})}{\partial \theta_{u,l}} \tag{A.4}$$

Furthermore, the partial derivative of the joint histogram $p(f,m|\boldsymbol{\theta})$ given by Eq. (11) can be obtained by

$$\frac{\partial p(f,m|\boldsymbol{\theta})}{\partial \theta_{u,l}} = \alpha\sum_{\mathbf{x}\in\mathcal{X}} K\left(\frac{f - I_F(\mathbf{x})}{\sigma_F}\right)K'\left(\frac{m - I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}))}{\sigma_M}\right)\frac{\partial I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}))}{\partial T_{\theta_l}(\mathbf{x})}\frac{\partial T_{\theta_l}(\mathbf{x})}{\partial \theta_{u,l}} \tag{A.5}$$

with $\alpha = -\frac{1}{\sigma_M|\mathcal{X}|}$. Then, the derivative of mutual information can be expressed as

$$\frac{\partial \mathrm{MI}}{\partial \theta_{u,l}} = \sum_{\mathbf{x}\in\mathcal{X}} \boldsymbol{\Lambda}_{\mathrm{MI}}(I_F(\mathbf{x}), I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})), \mathcal{G}, \sigma_F, \sigma_M)\frac{\partial I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}))}{\partial T_{\theta_l}(\mathbf{x})}\frac{\partial T_{\theta_l}(\mathbf{x})}{\partial \theta_{u,l}} \tag{A.6}$$

where

$$\boldsymbol{\Lambda}_{\mathrm{MI}}(I_F(\mathbf{x}), I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})), \mathcal{G}, \sigma_F, \sigma_M) =$$

$$= \alpha\sum_{f\in\mathcal{L}_F}\sum_{m\in\mathcal{L}_M}\mathcal{G}\cdot K\left(\frac{f - I_F(\mathbf{x})}{\sigma_F}\right)\cdot K'\left(\frac{m - I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}))}{\sigma_M}\right) \tag{A.7}$$

A3. NMI Gradient

From the expression of NMI in Eq. (14), its gradient is obtained by

$$\frac{\partial \mathrm{NMI}}{\partial \theta_{u,l}} = \frac{1}{\mathcal{E}}\frac{\partial \mathcal{D}}{\partial \theta_{u,l}} - \frac{\mathcal{D}}{\mathcal{E}^2}\frac{\partial \mathcal{E}}{\partial \theta_{u,l}} \tag{A.8}$$

where

$$\frac{\partial \mathcal{D}}{\partial \theta_{u,l}} = \sum_{f,m}\frac{\partial p(f,m|\boldsymbol{\theta})}{\partial \theta_{u,l}}\log_2(p_F(f)\cdot p_M(m|\boldsymbol{\theta})) + \mu\sum_{f,m}\frac{p(f,m|\boldsymbol{\theta})}{p_M(m|\boldsymbol{\theta})}\frac{\partial p_M(m|\boldsymbol{\theta})}{\partial \theta_{u,l}} =$$

$$= \sum_{f}\log_2(p_F(f))\frac{\partial \sum_m p(f,m|\boldsymbol{\theta})}{\partial \theta_{u,l}}\ (=0) + \sum_{f,m}\frac{\partial p(f,m|\boldsymbol{\theta})}{\partial \theta_{u,l}}\log_2(p_M(m|\boldsymbol{\theta}))$$

$$+\mu\frac{\partial \sum_m p_M(m|\boldsymbol{\theta})}{\partial \theta_{u,l}}\ (=0)$$

$$\frac{\partial \mathcal{E}}{\partial \theta_{u,l}} = \sum_{f,m}\frac{\partial p(f,m|\boldsymbol{\theta})}{\partial \theta_{u,l}}(\log_2(p(f,m|\boldsymbol{\theta})) + \sum_{f}\mu\frac{\partial \sum_m p(f,m|\boldsymbol{\theta})}{\partial \theta_{u,l}}\ (=0))$$

Then, the gradient expression can be simplified as follows

$$\frac{\partial \mathrm{NMI}}{\partial \theta_{u,l}} = \sum_{f,m}\underbrace{\left[\frac{1}{\mathcal{E}}\log_2(p_M(m|\boldsymbol{\theta})) - \frac{\mathcal{D}}{\mathcal{E}^2}\log_2(p(f,m|\boldsymbol{\theta})\right]}_{\mathcal{J}}\frac{\partial p(f,m|\boldsymbol{\theta})}{\partial \theta_{u,l}} \tag{A.9}$$

Therefore, by substituting the expression of the $p(f,m|\boldsymbol{\theta})$ derivative (Eq. (A.5)), the NMI gradient also takes the form

$$\frac{\partial \mathrm{NMI}}{\partial \theta_{u,l}} = \sum_{\mathbf{x}\in\mathcal{X}} \boldsymbol{\Lambda}_{\mathrm{NMI}}(I_F(\mathbf{x}), I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})), \mathcal{J}, \sigma_F, \sigma_M)\frac{\partial I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}))}{\partial T_{\theta_l}(\mathbf{x})}\frac{\partial T_{\theta_l}(\mathbf{x})}{\partial \theta_{u,l}} \tag{A.10}$$

with

$$\boldsymbol{\Lambda}_{\mathrm{NMI}}(I_F(\mathbf{x}), I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x})), \mathcal{J}, \sigma_F, \sigma_M) =$$

$$= \alpha\sum_{f\in\mathcal{L}_F}\sum_{m\in\mathcal{L}_M}\mathcal{J}\cdot K\left(\frac{f - I_F(\mathbf{x})}{\sigma_F}\right)\cdot K'\left(\frac{m - I_M(\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}))}{\sigma_M}\right) \tag{A.11}$$

References

[1] D. Rueckert, L.I. Sonoda, C. Hayes, D.L.G. Hill, M.O. Leach, D.J. Hawkes, Non-rigid registration using free-form deformations: application to breast MR images, IEEE Trans Med Imaging 18 (8) (1999) 712–721, doi:10.1109/42.796284.

[2] D. Rueckert, P. Aljabar, Non-rigid Registration Using Free-form Deformations, Springer US, Boston, MA, 2015, pp. 277–294.

[3] T.W. Sederberg, S.R. Parry, Fee-form deformation of solid geometric models, SIGGRAPH Computer Graphics 20 (4) (1986), doi:10.1145/15886.15903.

[4] R. Hua, J.M. Pozo, Z.A. Taylor, A.F. Frangi, Multiresolution extended free-form deformations (XFFD) for non-rigid registration with discontinuous transforms, Med Image Anal 36 (2017) 113–122, doi:10.1016/j.media.2016.10.008.
[5] R.-M. Menchón-Lara, J. Royuela-del Val, F. Simmross-Wattenberg, P. Casaseca-de-la Higuera, M. Martín-Fernández, C. Alberola-López, Fast 4D elastic group-wise image registration. convolutional interpolation revisited, Comput Methods Programs Biomed 200 (2021) 105812, doi:10.1016/j.cmpb.2020.105812.
[6] H.S. Hou, H. Andrews, Cubic splines for image interpolation and digital filtering, IEEE Trans Acoust 26 (6) (1978) 508–517, doi:10.1109/TASSP.1978.1163154.
[7] R. Keys, Cubic convolution interpolation for digital image processing, IEEE Trans Acoust 29 (6) (1981) 1153–1160.
[8] M. Unser, A. Aldroubi, M. Eden, Fast b-spline transforms for continuous image representation and interpolation, IEEE Trans Pattern Anal Mach Intell 13 (3) (1991) 277–285.
[9] S. Lee, G. Wolberg, S.Y. Shin, Scattered data interpolation with multilevel b-splines, IEEE Trans Vis Comput Graph 3 (3) (1997) 228–244, doi:10.1109/2945.620490.
[10] J.A. Schnabel, D. Rueckert, M. Quist, J.M. Blackall, A.D. Castellano-Smith, T. Hartkens, G.P. Penney, W.A. Hall, H. Liu, C.L. Truwit, F.A. Gerritsen, D.L.G. Hill, D.J. Hawkes, A generic framework for non-rigid registration based on non-uniform multi-level free-form deformations, in: W.J. Niessen, M.A. Viergever (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2001, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 573–581, doi:10.1007/3-540-45468-3_69.
[11] T. Rohlfing, C.R. Maurer, Nonrigid image registration in shared-memory multiprocessor environments with application to brains, breasts, and bees, IEEE Trans. Inf. Technol. Biomed. 7 (1) (2003), doi:10.1109/TITB.2003.808506.
[12] F. Ino, K. Ooyama, K. Hagihara, A data distributed parallel algorithm for non-rigid image registration, Parallel Comput 31 (1) (2005) 19–43, doi:10.1016/j.parco.2004.12.001.
[13] J. Rohrer, L. Gong, G. Székely, Parallel mutual information based 3 d non-rigid registration on a multi-core platform, in: Proceedings of the MICCAI Workshop in High-Performance. HPMICCAI, 2008.
[14] W. Sun, W.J. Niessen, S. Klein, Free-form deformation using lower-order b-spline for nonrigid image registration, in: P. Golland, N. Hata, C. Barillot, J. Hornegger, R. Howe (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014, Springer International Publishing, Cham, 2014, pp. 194–201, doi:10.1007/978-3-319-10404-1_25.
[15] S.Y. Chun, J.A. Fessler, A simple regularizer for b-spline nonrigid image registration that encourages local invertibility, IEEE J Sel Top Signal Process 3 (1) (2009) 159–169, doi:10.1109/JSTSP.2008.2011116.
[16] X. Geng, G.E. Christensen, H. Gu, T.J. Ross, Y. Yang, Implicit reference-based group-wise image registration and its application to structural and functional MRI, Neuroimage 47 (4) (2009) 1341–1351, doi:10.1016/j.neuroimage.2009.04.024.
[17] M. Polfliet, S. Klein, W. Huizinga, M.M. Paulides, W.J. Niessen, J. Vandemeulebroucke, Intrasubject multimodal groupwise registration with the conditional template entropy, Med Image Anal 46 (2018) 15–25, doi:10.1016/j.media.2018.02.003.
[18] M. Modat, G.R. Ridgway, Z.A. Taylor, M. Lehmann, J. Barnes, D.J. Hawkes, N.C. Fox, S. Ourselin, Fast free-form deformation using graphics processing units, Comput Methods Programs Biomed 98 (3) (2010) 278–284, doi:10.1016/j.cmpb.2009.09.002.
[19] D. Ruijters, B.M.t.H. Romeny, P. Suetens, GPU-Accelerated elastic 3D image registration for intra-surgical applications, Comput Methods Programs Biomed 103 (2) (2011) 104–112, doi:10.1016/j.cmpb.2010.08.014.
[20] X. Du, J. Dang, Y. Wang, S. Wang, T. Lei, A parallel nonrigid registration algorithm based on b-spline for medical images, Comput Math Methods Med 2016 (Article ID 7419307) (2016) 14, doi:10.1155/2016/7419307.
[21] N.D. Ellingwood, Y. Yin, M. Smith, C.-L. Lin, Efficient methods for implementation of multi-level nonrigid mass-preserving image registration on GPUs and multi-threaded CPUs, Comput Methods Programs Biomed 127 (2016) 290–300, doi:10.1016/j.cmpb.2015.12.018.
[22] K. Punithakumar, P. Boulanger, M. Noga, A GPU-accelerated deformable image registration algorithm with applications to right ventricular segmentation, IEEE Access 5 (2017) 20374–20382, doi:10.1109/ACCESS.2017.2755863.
[23] M. Brunn, N. Himthani, G. Biros, M. Mehl, A. Mang, Fast GPU 3D diffeomorphic image registration, J Parallel Distrib Comput 149 (2021) 149–162, doi:10.1016/j.jpdc.2020.11.006.
[24] G. Balakrishnan, A. Zhao, M.R. Sabuncu, J. Guttag, A.V. Dalca, Voxelmorph: a learning framework for deformable medical image registration, IEEE Trans Med Imaging 38 (8) (2019) 1788–1800, doi:10.1109/TMI.2019.2897538.
[25] A.V. Dalca, G. Balakrishnan, J. Guttag, M.R. Sabuncu, Unsupervised learning of probabilistic diffeomorphic registration for images and surfaces, Med Image Anal 57 (2019) 226–236, doi:10.1016/j.media.2019.07.006.
[26] T. Che, Y. Zheng, J. Cong, Y. Jiang, Y. Niu, W. Jiao, B. Zhao, Y. Ding, Deep group-wise registration for multi-spectral images from fundus images, IEEE Access 7 (2019) 27650–27661, doi:10.1109/ACCESS.2019.2901580.
[27] C. de Boor, A Practical Guide to Splines, Applied Mathematical Sciences, volume 27, Springer-Verlag, 1978, doi:10.1007/978-1-4612-6333-3.
[28] P. Thevenaz, M. Unser, Optimization of mutual information for multiresolution image registration, IEEE Trans. Image Process. 9 (12) (2000) 2083–2099, doi:10.1109/83.887976.
[29] R. Xu, Y.-W. Chen, S.-Y. Tang, S. Morikawa, Y. Kurumi, Parzen-window based normalized mutual information for medical image registration, IEICE Trans Inf Syst E91-D (1) (2008) 132–144, doi:10.1093/ietisy/e91-d.1.132.
[30] Q. Feng, Y. Zhou, X. Li, et al., Liver DCE-MRI registration in manifold space based on robust principal component analysis, Sci Rep 6 (34461) (2016) 1–16, doi:10.1038/srep34461.
[31] L. Cordero-Grande, S. Merino-Caviedes, S. Aja-Fernández, C. Alberola-López, Groupwise elastic registration by a new sparsity-promoting metric: application to the alignment of cardiac magnetic resonance perfusion images, IEEE Trans Pattern Anal Mach Intell 35 (11) (2013) 2638–2650, doi:10.1109/TPAMI.2013.74.
[32] M.L. Overton, NLCG: Nonlinear conjugate gradient, (https://cs.nyu.edu/faculty/overton/software/) Accessed: 2019-01-20.