The Visual Computer
https://doi.org/10.1007/s00371-021-02372-3

ORIGINAL ARTICLE

SORCNet: robust non-rigid shape correspondence with enhanced descriptors by Shared Optimized Res-CapsuleNet

Yuanfeng Lian[1] · Dingru Gu[1] · Jing Hua[2]

Accepted: 24 November 2021

Abstract
3D non-rigid shape correspondence, as an important research topic in 3D shape analysis, is useful but challenging in computer graphics, computer vision, and pattern recognition. Despite recent success of several deep neural networks for shape correspondence, those networks cannot achieve robust results on non-rigid objects due to their local deformation complexity. This paper presents a novel and efficient shape correspondence network—Shared Optimized Res-CapsuleNet (SORCNet)—that learns point features based on enhanced descriptors to solve dense correspondence between non-rigid 3D shapes. To further improve the iterative efficiency and accuracy of the model, we design an optimized residual network structure, based on the stochastic gradient descent algorithm with momentum and weight decay (SGDW). Moreover, as the convolutional neural network does not perform well when the shape has directional variance, we present a shared capsule network structure with dual routings, which correlates the hierarchical geometric relationships of the semantic parts well to extract more informative point features. We proved that the primary capsule has a greater influence on feature extraction than the routing and decoder parts. The entire network, SORCNet, is integrated and trained/tested by taking the descriptors and Laplacian eigenbases of two shapes as input. The experiments on public datasets, such as FAUST, SCAPE, TOSCA and KIDS, demonstrate the better effectiveness, accuracy, and adaptability of our method than those of the state of the art in 3D shape correspondence.

Keywords Shape correspondence · Shape descriptors · Optimized residual networks · Capsule networks

1 Introduction

Driven by the rapid developments of various 3D sensors [1], 3D shape correspondence has become increasingly important in 3D shape analysis, which is the foundation for shape registration, recognition, retrieval, segmentation, etc. For non-rigid shape correspondence, the main difficulties are that the arbitrary deformation of shape causes additional complexity, and there are so many variables required to define the dense mapping. Even though some progresses [2–4] have been made, the task of finding dense shape correspondence is still very challenging.

✉ Dingru Gu
2605658935@qq.com

Jing Hua
lianyuanfeng@cup.edu.cn

[1] Beijing Key Lab of Petroleum Data Mining, Department of Computer Science and Technology, China University of Petroleum, Beijing, China

[2] Wayne State University, Detroit, MI, USA

The non-rigid shape correspondence problem can be summarized as finding a point-wise matching between the points of two shapes. Traditional approaches minimize certain structure distortion, such as local features [5,6], geodesic [2,7] or diffusion distances [8], to establish the matching. Generally, those methods have a high computational complexity, and may produce a dense correspondence with poor surjectivity. Another kind of matching methods attempt to solve the correspondence in a parameterized domain or functional space with fewer degrees of freedom. For example, some of them restrict the correspondence space to conformal maps [3,9], and some [10,11] choose the eigenfunction of the Laplace–Beltrami operator as the embedding coordinates to compute the matching in the eigenspace. Unlike the point-wise correspondence method, the soft correspondence method maps one point on one shape to more than one point on the other shape. Ovsjanikov et al. [4] introduced a soft correspondence framework, named functional maps, by mapping the correspondence between shapes to linear operators between function spaces, with an effective representation of the Laplacian eigenbases. The pipeline of this framework

Published online: 10 January 2022

can be found in Fig. 1 . This idea has been extended by several subsequent works [12,13]. Litany et al. [14] combined it with deep learning and proposed a deep neural network architecture called FMNet to obtain a good approximation of the functional map. Despite the flexibility the framework has, the highly expressive descriptor functions and a large number of constraints are required to obtain a good solution.

As aforementioned, the functional maps framework requires descriptor functions and Laplacian eigenbases as input. For shape descriptors, they can be roughly divided into two categories: handcrafted shape descriptors [5,15,16] and learning-based shape descriptors [17,18]. Some approaches also use deep neural networks to enhance the representation ability of descriptors [19]. In the deep functional maps framework, extracting features to get high-quality, robust and informative descriptor functions is the key. There have been several deep feature extraction methods proposed. Particularly for objects with position and direction information, the recent capsule network [20] has achieved good results in 2D domains [21,22] and been gradually extended to 3D data [23].

Inspired by the deep functional map [14], we propose a new deep learning method for the shape correspondence, SORCNet, to further improve the convergence rate in the model design and better capture the shape features by using the latent capsule. For most Non-Rigid Shape Correspondence methods, they focused on extracting features using classical descriptors or DNN. On the contrast, CapsNet allows a powerful semantic understanding of the shapes' components and their directions and positions. But, the drawback is that the convolution layer in the standard CapsNet is not capable enough to build capsules that can describe the sophisticated features in the dataset. Based on this observation, we see a great room to explore and create a new, better algorithm. We use the optimized ResNet derived from the SGDW algorithm, which provides efficient feature extraction and description, for the construction of the main capsule and the decoding of the potential capsule. Our approach is general, and we have demonstrated its efficiency in extracting descriptors for Non-Rigid shapes. The SORCNet mainly includes two components. The first one is to propose an optimized residual network to learn the enhanced descriptors and obtain shape correspondence from a pair of shapes. The second one is to use the stacked latent vectors to learn the features for the shape matching/deformation and their probability and hierarchy in the network, which can greatly improve the approximation of the functional maps.

Our main *contributions* can be summarized as follows:

- We propose an optimized network structure inspired by the SGDW algorithm, called OptResBlock, to enhance the expressive ability of the network, which outperforms the original network with a higher accuracy.

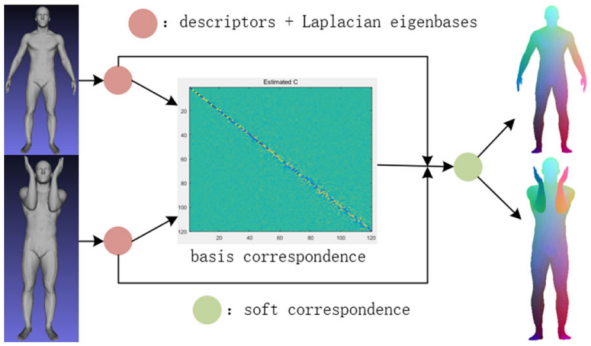

Fig. 1 The pipeline of the functional maps framework. In this framework, functional maps can be concisely represented as a matrix **C**

- Due to the nice property from capsule network structure (e.g., describing the shape size, direction, and deformation informatively, robustness to affine transformation, etc.), a shared capsule network architecture for 3D descriptor learning (i.e., with shared parameters for two shapes), called Desc-CapsNet, is proposed to extract features from two shapes, in order to obtain more powerful descriptor functions with various shape parameters concentrating not spatially but semantically across the shapes. It lead to faster network convergence while significantly improving correspondence accuracy.
- We present a dual routings procedure (a sigmoid dynamic routing and an attention EM routing) for higher classification accuracy of capsules. Besides, we modify the traditional geodesic distance loss to a spectral loss function by defining the ground-truth domain correspondence matrix and considering the orthogonality constraint of the ideal domain mapping matrix, leading to a faster calculation time and better accuracy.
- Experiments on accuracy, generalization and convergence efficiency with different datasets have shown great improvements when integrating the proposed OptResBlock and the shared Desc-CapsNet in the uniquely designed architecture. It demonstrates that the shared Desc-CapsNet can effectively learn the descriptors and outperforms the state-of-the-art deep learning methods as well as traditional methods, especially when adding the optimized block (i.e. Our integrated SORCNet).

2 Related work

The calculation of the correspondence between 3D shapes is a very important research field in computer vision. We review the most closely related methods below, and interested readers can refer to the survey articles [24,25] and go deeper into other shape correspondence methods.

2.1 Non-rigid shape correspondence technology

Various algorithms for shape correspondence have been proposed so far. The most commonly used is the traditional shape matching method based on statistical features, and many scholars have carried out further work on the basis of feature descriptors. Rodolà et al. [26] used random forests to classify shapes based on their wave kernel signature (WKS) features to solve the shape matching problem and achieve good results. With the advent of deep learning technique, it has been used to obtain the correct deformation model from 3D data directly. Monti et al. [27] generalize the CNN architecture to non-Euclidean domains (graphs and manifolds) to learn local, stationary, and compositional task-specific features. Wang et al. [28] parameterize the multi-scale localized neighborhoods of a keypoint into regular 2D grids and use a triplet network to derives discriminative local descriptors of 3D surface for non-rigid shape matching. The SplineCNN [29] advances a novel convolution operator based on B-splines, which makes the computation time independent from the kernel size for irregular structured and geometric input. Specifically, a landmark framework called functional maps [4] represents maps between shapes as small matrices encoding relations between basis functions of the shapes. This framework is described and expanded in [30]. Inspired by that, several learning methods have been proposed for predicting structured maps [12–14,31] rather than labeling each point independently. But the weakness is that the optimality criterion is generally obtained in line with the deviation from the ground-truth functional maps, so that errors in precomputed descriptors will result in mapping deviation. Thus, based on the supervised learning approach, self-supervised and unsupervised learning methods have been proposed [32,33]. However, those methods rely heavily on the selection of descriptors and cannot efficiently represent the local deformation of the shape. Compared with those, our SORCNet can fully extract the information such as the direction and position of the descriptor to the feature space to obtain higher robustness, correspondence accuracy and generalization.

2.2 Capsule network

The CapsNet is introduced by Hinton et al. [20,34] and becomes a popular new neural network architecture due to its improved performance in many aspects of image processing. Because of its general applicability and robustness to affine transformations, etc., capsule network has been widely used in 2D deep learning. Durate et al. [21] proposed the concept of "capsule pool," extending the capsule network to the subdivision and classification of actions. Lin et al. [22] demonstrated that capsule networks can learn more meaningful 2D manifold embeddings than traditional CNNs. For optimization of the original complex dynamic routing, Hinton et al. [35] improved the routing by expectation maximization (EM) algorithm. Chen and Crandall [36] presented "trainable routing" to make capsules better clustered. Since the capsule network has achieved great success in the field of 2D image processing, some researchers consider about extending the capsule networks to 3D domain. Zhao et al. [23] presented 3D point-capsule networks, to extract local 3D features while maintaining the invariance of the spatial arrangement of the input point cloud data. Cheraghian and Petersson [37] extended the concept of capsule in 3D and introduced 3DCapsule for point clouds. It introduces a new module, ComposeCaps, which replaces spatially related feature mapping and learns new mapping. In our work, we extend the capsule network to translate the information from descriptors to latent feature space. From the perspective of learning for two sets of data, it is necessary to share weights in our Desc-CapsNets between two tasks to model the sophisticated attribute relationships, which can lead to higher robustness and learning efficiency of the SORCNet.

2.3 Optimized network structure design

There have been extensive works done on the structural design of neural networks. Early neural network designs were based on generic algorithms to find the architecture and weights [38]. Domhan et al. [39] uses Bayes to optimize the network architecture. Some researchers [40,41] adapt the adaptive strategy to extend the network structure layer by layer from a small network according to some principles. Although impressive results have been achieved, they do not explicitly indicate where the connections should occur in the network architecture. In [42], it is demonstrated that the structure of the neural network can be designed inspired by the optimization algorithm. The optimization algorithm is a type of method that can help us to minimize or maximize the objective function (sometimes called the loss function). The standard feedforward neural networks have been proved that propagation in a neural network is equivalent to using a gradient descent algorithm to minimize certain functions $f(x)$ [42], which means that faster optimization algorithms can inspire better neural network structures. There are many optimization algorithms proposed to solve the general optimization problem, i.e., $min_x f(x)$. The gradient descent (GD) algorithm [43] is one of the most commonly used optimization methods and is the basis of many other optimization algorithms. In order to increase the iteration speed and improve the accuracy of the model, our OptResNet uses the SGDW [44] algorithm to optimize the propagation structure of the residual network to enhance the input descriptors of shapes.

3 Correspondence with functional maps

Our work is based on a manifold functional maps framework and the Laplace–Beltrami operator describing manifolds. For the basic concepts and processes of specific functional maps, we refer readers to [30].

The Laplace–Beltrami operator [45] is a second-order differential operator defined on a Riemannian manifold, which is a generalization of the Laplacian operator in the Euclidean space. It essentially describes the coordinate value of a point in space and the mean value of its neighborhood coordinates. For a compact closed manifold A, an indicator function $f : A \to \mathbb{R}$ is constructed and can be represented as a linear combination of basis functions:

$$f = \sum_{i \geq 1} a_i \phi_i^A \tag{1}$$

where $\phi_i, i = 1, 2, \ldots$ denotes the eigenfunctions which form an orthonormal basis for f. $a_i = \langle \mathrm{f}, \mathrm{Œ_i} \rangle_{\mathrm{A}}$ denotes the coefficient corresponding to the i_{th} eigenfunction. f is the corresponding discretization of the smooth function f.

As described in [30], we consider a correspondence between points of two shapes A and B as a linear operator $T : f(A) \to f(B)$, which means mapping functions on A to functions on B. The map T can be expressed as a matrix $\mathbf{C}$ with coefficients $\{c_{ji}\}$. When A and B are equipped with a set of basis functions $\{\phi_i\}$ and $\{\psi_j\}$ calculated by the Laplacian eigenfunctions, respectively, we have:

$$T(f) = \sum_{j \geq 1} \sum_{i \geq 1} a_i c_{ji} \psi_j \tag{2}$$

Note that for $\{c_{ji}\}$ in matrix $\mathbf{C}$, if $\{\phi_i\}$ and $\{\psi_j\}$ are orthogonal to some inner products $\langle \cdot, \cdot \rangle$, $\{c_{ji}\} = \langle T(\phi_i), \psi_j \rangle$. The basis functions truncate the series in Eq. (2) after the first k coefficients to calculate domain mapping matrix $\mathbf{C}$, which is approximate to the original map. Supposing there exists an operator applied on a pair of shapes A and B, it will produce a set of pairs of corresponding descriptor functions. That means, given a pair of coefficients $a_i = \langle \mathrm{f}, \mathrm{Œ_i} \rangle$ and $b_i = \langle \mathrm{g}, \mathrm{_j} \rangle$ in the bases $\{\phi_i\}$ and $\{\psi_j\}$ respectively, and stack them into the columns of matrices $\mathbf{A}_1$ and $\mathbf{B}_1$, then, the optimization problem of the networks can be defined as computing the optimal functional map $\mathbf{C}$:

$$\mathbf{C}_{opt} = \arg\min_{\mathbf{C}} \|\mathbf{C}\mathbf{A}_1 - \mathbf{B}_1\|^2 \tag{3}$$

4 SORCNet design

In this paper, we build an end-to-end deep neural network named "Shared OptRes-CapsNet" (SORCNet) upon the functional maps framework. We believe that CapsNet allows a powerful understanding of the shapes' components and their directions and positions. Since the convolution layer in the standard CapsNet is not perfect enough to build capsules that can describe the sophisticated features in the dataset, we use the optimized ResNet derived from the SGDW algorithm, which provides efficient feature extraction and description, for the construction of the main capsule and the decoding of the potential capsule.

The basic pipeline can be described by the following steps: first to calculate the first k eigenfunctions of the Laplace–Beltrami operators on a pair of shapes; second, to learn the descriptor functions of shapes, and express it upon the basis of the corresponding shape from the equations $\mathbf{A_1}$ and $\mathbf{B_1}$; third, to calculate the $k \times k$ correspondence matrix $\mathbf{C}$ by solving an least squares problem according to Eq. (3); finally, to compute the spectral error loss according to the ground-truth point-wise mapping, and obtain the soft correspondence between shapes A and B. The entire network architecture is given in the right box in Fig. 2.

Our network takes two sets of descriptors as inputs and outputs the enhanced descriptors using same parameters. Generally, the capsule is a set of vectors. The length of a capsule represents the probability of the existence of an entity, and the direction represents the instantiated parameters, such as hand position, size, direction and shape. In the propagation, each lower-level capsule delivers learned and predicted data to the higher-level capsule. If multiple predictions agree, the higher-level capsule becomes active. The SORCNet firstly learn descriptors with two optimization residual networks. Then, the extracted features are max-pooled and concatenated to form the primary capsules.

4.1 Enhanced primary capsules

According to a study by Li et al. [42], the propagation in the neural network is equivalent to using the gradient descent algorithm to minimize some function $F(x)$. For the feedforward neural network, at each layer, a linear transformation is applied to the input x_k and then a nonlinear transformation follows, which can be described as $x_{k+1} = \Phi(W_k x_k)$. Different from the linear transformation A of traditional optimization, W_k is learnable so that each layer has a different linear transformation matrix. Drawing inspiration from that, we derive a new network structure formula $\theta_{i+1} = \Phi(\mathbf{U}\theta_i) + (\alpha - w_i)\theta_i - \alpha\theta_{i-1}$ from the SGDW algorithm to enhance the representation ability of capsules. From the derivation results, to a certain extent, ResNet and DenseNet can also be seen as special cases of neural network structures inspired by optimization algorithms. In this subsection, we give a proof of the relation between the neural network structure and the GD optimization algorithm and describe the derivation process of the new network structure.

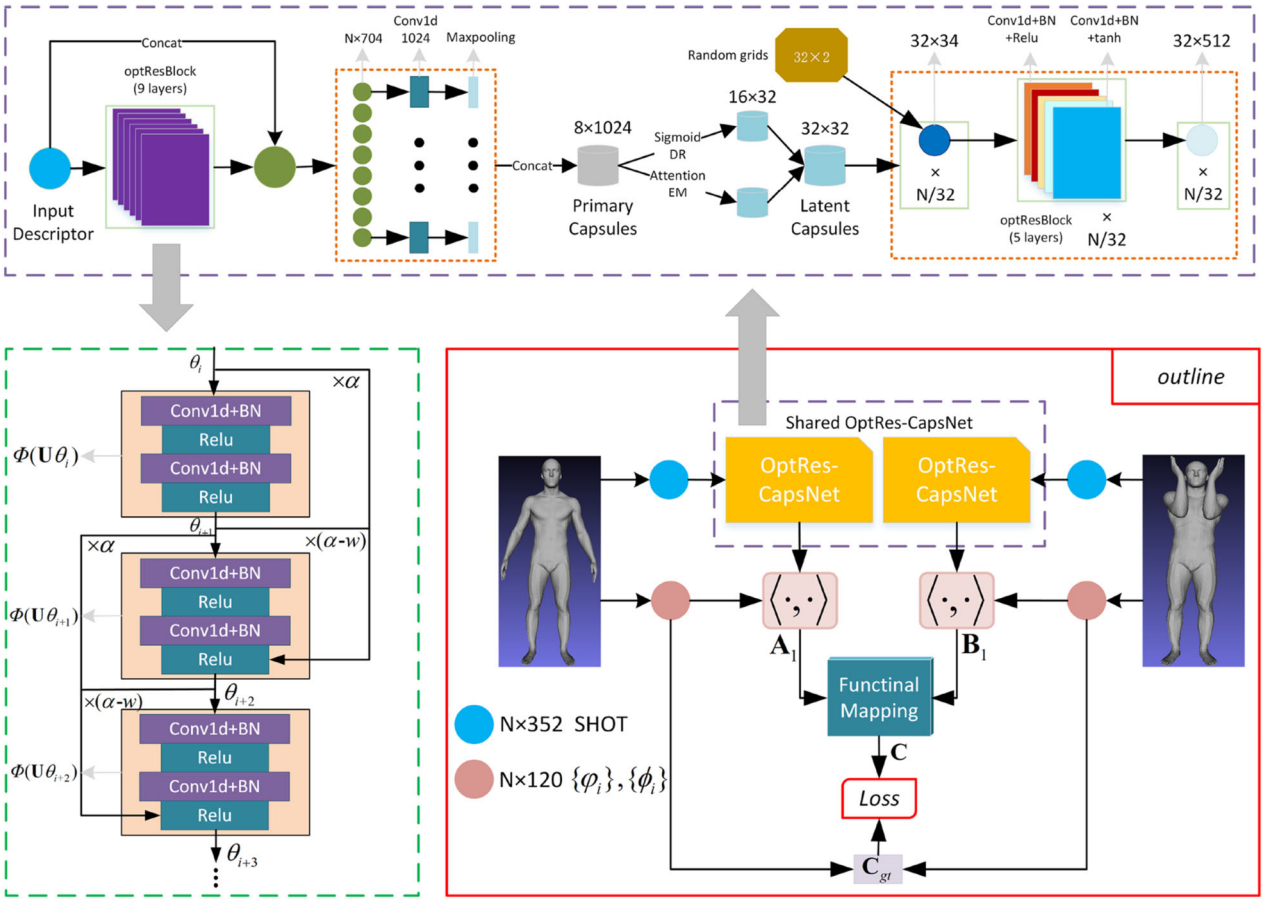

Fig. 2 The framework of our proposed network SORCNet (Shared OptRes-CapsNet). We input the 352-d SHOT descriptors and the Laplacian eigenfunctions from a pair of shapes and then obtain refined descriptors through the optimization blocks and splice with the SHOT operators, which are projected onto the Laplacian basis eigenfunctions $\{\phi_i\}$ and $\{\psi_j\}$ to produce the spectral representations $\mathbf{A}_1$ and $\mathbf{B}_1$. The functional map is represented by Eq. (3), and the loss is derived from Eq. (20). The two OptRes-CapsNets, respectively, share weights for the two sets of data. The purple box above is the structure of OptRes-CapsNet. The green box at the bottom left shows the structure of OptResBlock for the primary capsule construction

In the standard feedforward neural network, the propagation of each layer can be expressed as:

$$\theta_{i+1} = \Phi\left(\mathbf{U}\theta_i\right) \tag{4}$$

where θ_i is the output of the i-th layer, $\Phi(\cdot)$ is an activation function, and $\mathbf{U}$ is a linear transformation. In the structural design stage, the weight matrix $\mathbf{U}$ is considered to be fixed for analysis. In order to relate Eq. (4) to the gradient descent process $\xi_{k+1} = \xi_k - \nabla(\xi_k)$, it is necessary to find an objective function $F(\xi)$ to optimize.

Lemma 1 *Assume that* $\mathbf{U}$ *is a symmetric positive definite matrix. Set* $\mathbf{V} = \sqrt{\mathbf{U}}$. *Then, there is a function* $f(\xi)$ *that makes Eq. (4) be equivalent to optimize* $F(\xi) = f(\mathbf{V}\xi)$*:*

(1) Define a new variable $\Psi'(\boldsymbol{\xi}) = \Phi(\boldsymbol{\xi})$.

(2) Use gradient descent algorithm to optimize $f(\xi)$.

(3) Obtain $\theta_0, \theta_1, \ldots, \theta_k$ *from* $\xi_0, \xi_1, \ldots, \xi_k$ *through* $\theta = \mathbf{V}^{-1}\xi$.

For the commonly used activation function $\Phi(\xi)$*, define function* $\Psi'(\boldsymbol{\xi})$ *such that* $\Psi'(\boldsymbol{\xi}) = \Phi(\boldsymbol{\xi})$. *Then, an expression can be obtained:* $\nabla \sum_j \Psi\left(\mathbf{V}_j^T\xi\right) = \mathbf{V}\Phi(\mathbf{V}^T\xi_i) = \mathbf{V}\Phi(\mathbf{V}\xi_i)$.

Proof Let $f(\xi)$ be defined as:

$$f(\xi) = \frac{\|\xi\|^2}{2} - \sum_j \Psi\left(\mathbf{V}_j^T\xi\right), \quad \mathbf{V}_j^T\xi > 0 \tag{5}$$

where $\mathbf{V}_j$ is the j-th column of the matrix $\mathbf{V}$. Then, we have:

$$\nabla f(\xi_i) = \xi_i - \mathbf{V}\Phi(\mathbf{V}\xi_i) \tag{6}$$

using the iteration of the GD algorithm to minimize Eq. 5. We have:

$$\xi_{i+1} = \mathbf{V}\Phi(\mathbf{V}\xi_i) \tag{7}$$

Recover θ by $\theta = \mathbf{V}^{-1}\xi$, which leads to:

$$\theta_{i+1} = V^{-1}\xi_{i+1} = \Phi(V\xi_i) = \Phi(V^2\theta_i) = \Phi(\mathbf{U}\theta_i) \tag{8}$$

which is the same as Eq. (4), and the proof is complete. □

Then, we replace the GD algorithm with the SGDW algorithm. The SGDW algorithm is an improved algorithm of gradient descent. It adds a momentum after the gradient descent step. The original formula of SGDW is expressed by:

$$x_{t+1} = x_t - v_t - \eta w_t x_t, \; v_t = \gamma v_{t-1} + \eta g_t \tag{9}$$

And it can be simplified as an equivalent form of:

$$\xi_{i+1} = \xi_i - \nabla f(\xi_i) + \alpha(\xi_i - \xi_{i-1}) + w_i\xi_i \tag{10}$$

Substitute Eq. (6) in Eq. (10). Then, it becomes:

$$\xi_{i+1} = \mathbf{V}\Phi(\mathbf{V}\xi_i) + \alpha(\xi_i - \xi_{i-1}) - w_i\xi_i \tag{11}$$

Identically, recover θ by $\theta = \mathbf{V}^{-1}\xi$, we have:

$$\theta_{i+1} = \Phi(\mathbf{U}\theta_i) + (\alpha - w_i)\theta_i - \alpha\theta_{i-1} \tag{12}$$

In the previous derivation, $\mathbf{U}\theta_i$ appears as a fully connected linear transformation. It is the product of the matrix $\mathbf{U}$ and the vector θ_i. Here, it is extended to a convolution operation. In addition, different layers of the network have different weight matrices $\mathbf{U}$, and the form of $\mathbf{U}$ is not limited to square matrices, so the input and output sizes can be different. Φ is a nonlinear transformation defined by an activation function, which can be extended to pooling and batch normalization (BN). $\Phi(\cdot)$ can be a nonlinear activation, which can be expressed as a combination of pooling, batch normalization, convolution, or fully connected linear transformation. Through these different combinations, the network structure Eq. (4) can evolve into different networks.

According to Eq. (12), a neural network structure can be designed, as an optimized residual network structure with two shortcuts. When deriving and designing the neural network structure, the coefficients in the formula can be statically fixed values, or dynamic learning generated. That is, α and w_i can be set to any constants. The network structure inspired by the structure corresponding to this formula is shown in the green box on the bottom left in Fig 2.

In our work, the optimization residual block is used to enhance descriptors for the representation ability of capsules. According to the above formula, we build the network by stacking 9 layers of one-dimensional convolution: Conv1d352+BN+RelU+Conv1d352+BN+RelU. Conv1dx denotes a 1×1 convolution layer with that outputs a vector with x-dimension. Then, we concat the output features and the input descriptors. After extracting features, for the diversity of learning, we first replicate the reconstituted descriptor 8 times, and then use independent conv1d (1024) with different weights to extract feature maps with different attentions. The 8 multiple independent convolutional layers make sure of the diversity of posture feature learning. Then, we do the max-pooling operation on them to obtain a global latent representation, which are the primary capsules (1024×8). Then, the squash function, a special nonlinear activation function, is adopted to ensure the length of the output vector representing the probability of the shape feature, which is denoted as:

$$Squash(s_j) = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \tag{13}$$

The function shrinks short vectors to almost 0 and long vectors to a length below 1. After that, according to the mechanism of capsule network, we use the dynamic routing procedure to send the output of the primary capsules to the higher level latent capsules (32×32).

4.2 Dual routing module

The basis of the capsule network is a routing procedure, which transfers information from the previous capsule layer through a protocol [20]. If the consistency between the lower layer and the higher layer capsules is higher, it means that the coupling coefficient between these capsules is high, and the input of these lower capsules will be sent to the corresponding higher capsules. For the procedure, the prediction $u_{j|i}$ of the next layer is obtained by the dot product of the capsule of the current layer v_i and the weight matrix W_{ij}. The coupling coefficient c_{ij} is a logarithmic probability and is updated iteratively through the agreement between successive capsule layers. These coefficients can be considered as weights to suppress or encourage the contribution of lower-level capsules to certain higher-level capsules. Calculation of the weighted sum of the predicted $u_{j|i}$ is expressed in Eq. (14); the compression is according to the definition of Eq. (13), that is, the capsule v_j of the next layer.

$$s_j = \sum_i c_{ij} u_{j|i}, \; u_{j|i} = W_{ij} v_i \tag{14}$$

In the dynamic routing mechanism module, after the output of the processed low-level capsules is obtained, the activated high-level capsules should be determined, that is, the clustering process of features. This paper mixes two routing methods (i.e., sigmoid dynamic routing and attention EM routing) to cluster features. Fig 3 illustrates dual routing operation for the input capsule.

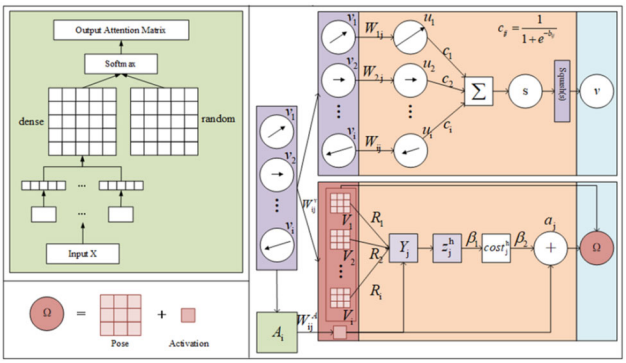

Fig. 3 Dual routing procedure from the first capsule layer to the capsule v_j in the second capsule layer. The green box shows details of the self-attention mechanism in EM routing

Sigmoid dynamic routing In [46], it demonstrates that the probabilities of the features sent to latent capsules are nearly equal, which may lead to a misclassification. Thus, we calculate the coupling coefficients c_{ij} by the sigmoid function:

$$c_{ij} = \frac{1}{1 + e^{-b_{ij}}} \tag{15}$$

It means that c_{ij} no longer represents the distribution probability of the final capsule, but represents the correlation strength between the main capsule and the potential capsule. Important prediction vectors are multiplied by larger coupling coefficients to make important features more decisive, while irrelevant features have less influence. The routing algorithm for the entire iteration process is as in Algorithm 1.

Algorithm 1 Sigmoid Dynamic Routing Algorithm

Require: The primary capsule v_i, parameter W_{ij}
Ensure: the output capsule v_j
1: Initialize weights matrices W_{ij}
2: $u_{j|i} = W_{ij} v_i$
3: $b_{ij} \leftarrow 0$
4: **for** i in N iterations **do**
5: $\quad c_{ij} = \frac{1}{1+e^{-b_{ij}}}$
6: $\quad s_j \leftarrow \sum_i c_{ij} u_{j|i}$
7: $\quad v_j \leftarrow Squash(s_j)$
8: $\quad b_{ij} \leftarrow b_{ij} + u_{j|i} v_j$
9: **end for**
10: **return** v_j;

Attention EM routing In the EM routing algorithm [35], the high level capsules are regarded as Gaussian mixture distribution, and the mean value and variance of the output capsules are updated iteratively, as well as the distribution probability R_{ij} of the capsules. Inspired by that, we propose a routing algorithm with iterative attention layers. As shown in the left green box of Fig. 3, a self-attention mechanism

Algorithm 2 Attention EM Routing Algorithm

Require: The activated primary capsules v_i; the activation values a_i; attention matrix A_{att}
Ensure: the activation a_j and pose Y_j of capsules in layer N
1: Initialize weights matrixes W_{ij}^v, W_{ij}^A; costs β_1 and β_2; hyperparameters λ
2: $R_{ij} \leftarrow 1/(number\ of\ high\text{-}level\ capsules)$
3: $V_{ij} = W_{ij}^v v_i$
4: $v_{ij}^h := h\text{-}th\ component\ of\ V_{ij}$
5: $a_j = W_{ij}^A A_{att}$
6: **for** i in N iterations **do**
7: $\quad$ **procedure** M for high-level capsule j:
8: $\quad R_{ij} = R_{ij} a_i$
9: $\quad y_j^h \leftarrow \frac{\sum_i R_{ij} v_{ij}^h}{\sum_i R_{ij}}$
10: $\quad (z_j^h)^2 \leftarrow \frac{\sum_i R_{ij} (v_{ij}^h - y_j^h)^2}{\sum_i R_{ij}}$
11: $\quad cost^h \leftarrow (\beta_1 + log(z_j^h)) \sum_i R_{ij}$
12: $\quad a_j \leftarrow a_j + logistic(\lambda(\beta_2 - \sum_h cost^h))$
13: $\quad$ **procedure** E for high-level capsule i:
14: $\quad P_{ij} \leftarrow (\prod_h 2\pi (z_j^h)^2)^{-1/2} exp(-\sum_h \frac{(v_{ij}^h - y_j^h)^2}{2(z_j^h)^2})$
15: $\quad R_{ij} \leftarrow \frac{v_j' P_{ij}}{\sum_j v_j' P_{ij}}$
16: **end for**
17: **return** Y_j, a_j;

[47] is added to facilitate the routing algorithm to capture more relevant features more accurately, which is generated by two linear functions and an activation function. For the input $X \in \mathbb{R}^{n \times m}$, a parameterized function projecting input X_i from m dimensions to l dimensions is defined as:

$$F_l(X) = \mathrm{ReLU}(XW_1 + b_1)W_2 + b_2 \tag{16}$$

where $W_1 \in \mathbb{R}^{m \times m}$, $W_2 \in \mathbb{R}^{m \times l}$. In practice, we use two Conv1d layers with ReLU activations to obtain the dense attention matrix. Then, the formula for finding the dense attention matrix D is as follows:

$$D = rep_B(B) * rep_C(C) \tag{17}$$

where $B, C = F_b(X), F_c(X)$. $F_b(.)$ and $F_c(.)$, respectively, project X onto b and c dimensions with $b \times c = n$. $rep_B(.)$ and $rep_C(.)$ represent duplicate the content c and b times respectively. In addition, another two random attention matrices $R_1, R_2 \in \mathbb{R}^{n \times m}$ are defined as a randomly initialized matrix. Then, the mixed attention matrix can be written as:

$$A_{att} = \mathrm{Softmax}(\alpha_1 D + \alpha_2 (R_1 R_2^T)) \tag{18}$$

where α_i are learnable parameters. The iterative attention EM algorithm is as in Algorithm 2:

Mixed capsule routing In the dynamic routing procedure, in order to allow the cluster centers to retain the main information of the features and show the importance of the features better, the results are mixed by concatenate the latent capsules

obtained by the individual two routings. The self-attention mechanism is embedded in the EM routing algorithm to obtain more active regions of interest, and further help to improve the significant part of its familiar with the relationship between the significant objects through a dual routing mechanism.

4.3 Latent capsule auto-encoder

As shown in the upper box in Fig. 2, the whole OptRes-CapsNet can be regarded as an auto-encoder procedure to enhance the latent capsule for the pose regression. The latent capsule is clustered by the routing mechanism, which can be seen as an encoding process of the shape features. Correspondingly, a decoder is designed symmetrically. It firstly replicates the latent capsules N/32 times and attaches a unique random 2D grid to each replica to promote the diversity. Then, we use (multilayer perceptrons) MLPs (18-32-64-128-256-512) for each patch to further extract the features and paste the output patches together, in order to obtain the final feature vectors with the same number of groups as the input descriptors.

In fact, by assigning capsules to individual parts of the object, our SORCNet treats this learning task as a classification problem for each capsule, which greatly improves the convergence rate and accuracy of the model. In order to extract point features from more expressive descriptors, our SORCNet integrates the OptResBlock and shared Desc-CapsNet, and shows better performance and generalization on shape correspondence task.

4.4 Spectral loss function

Different from the most correspondence losses that calculating a pairwise distortion relying on expensive geodesic distance matrix and point-wise map computation, given the domain mapping matrix $\mathbf{C}$ as the functional map, we define the $\mathbf{C}_{gt}$ based on the ground-truth point-wise correspondence:

$$\mathbf{C}_{gt} = (\boldsymbol{\Phi}^{\dagger}\mathbf{P}_{gt}\boldsymbol{\Psi})^{\top} \tag{19}$$

where $\mathbf{P}_{gt} \in \{0, 1\}$ is the ground-truth point-to-point correspondence between the shape pair, which expressed as an $N \times N$ diagonal matrix with elements of 1. $\boldsymbol{\Phi}^{\dagger} = \boldsymbol{\Phi}^{\top}\mathbf{W}$, $\mathbf{W}$ is a diagonal matrix of vertex area elements that has $\boldsymbol{\Phi}^{\top}\mathbf{W}\boldsymbol{\Phi} = \mathbf{I}$. $\boldsymbol{\Phi}$ is the matrix representation of $\{\phi_i\}$, while $\boldsymbol{\Psi}$ is the matrix representation of $\{\psi_i\}$.

According to the works in [4], a point-to-point map retains the local area if and only if the functional map $\mathbf{C}$ is orthonormal. Thus, the spectral loss function $\mathcal{L}$ is defined as:

$$\mathcal{L} = \|\mathbf{C}_{gt} - \mathbf{C}\|^2 + \|\mathbf{C}^{\top}\mathbf{C} - \mathbf{I}\|^2 \tag{20}$$

Compared with calculating the geodesic distance loss function from the point-by-point ground-truth correspondence, spectral loss can be calculated with less effort by restoring a sufficiently accurate functional mapping and making the loss effective. Considering the orthogonality of the ideal domain mapping matrix $\mathbf{C}$, the addition of the regular term increases the constraint on the functional correspondence.

5 Experiments and implementation

A wide range of experiments is performed on different datasets to demonstrate the efficacy of our method. We evaluate our approach with several evaluation criteria by comparing to other methods with the same training dataset on four different shape datasets. We use training set of FAUST [48] (the first 80 meshes) with the ground-truth correspondences to train our models and other compared methods. In our experiments, for designing the OptResBlock by formula 12, we let $\alpha = 1.5$, $w_k = 0.5$. Our network is implemented in TensorFlow [49] using the ADAM [50] stochastic optimization algorithm with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. It sets the adjustment of learning rate using a polynomial decay with the start learning rate $lr = 10^{-3}$ and the end learning rate $lr = 10^{-5}$. We train our model on a GeForce GTX 1080Ti GPU. When batch size becomes 2, it takes about 1.2s around to complete each iteration for our SORCNet. In this section, we provide qualitative and quantitative results of our experiments.

5.1 Datasets

We use the 352-d SHOT descriptor [15] (i.e., 352-d vector per point), which calculates the features of all vertices of two shapes, and $k = 120$ eigenfunctions of the shapes as the input of the network. Our experiments are carried out on FAUST [48], SCAPE [51], TOSCA [52] and KIDS [26] non-rigid shape datasets. In our work, we conduct experiments on FAUST testing dataset (the last 20 meshes) with both 4096 points and 6890 points, while for other datasets we sample them all to 4096 points to make the experiments more convenient.

FAUST dataset This dataset contains 100 meshes of 10 scanned subjects, each with 10 different poses. The prior methods have evaluated on pairs of scans of the same subject in different poses (intra-subject pairs) and on pairs of scans of different subjects (inter-subject pairs). The shapes in the dataset have strong non-isometric deformations, and vertex-wise ground-truth correspondence is known between all the shapes.

SCAPE dataset It is a digitally generated artificial human shape dataset, which has completely different properties from

the FAUST dataset in terms of geometric entities, proportions, ratios, etc.

TOSCA dataset This dataset consists of objects from different domains as animals and humans with different poses. In our work, we choose eight object categories from it to conduct the evaluation experiments.

KIDS dataset We also test our method on the highly non-isometric KIDS dataset. It is a small dataset containing two child subjects with 15 poses, respectively.

5.2 Iterative efficiency

We first conduct experiments on the FAUST testing dataset using our SORCNet. And then, the SORCNet without our proposed OptResBlock is trained to evaluate the optimized network structure. Thus, the iterative efficiency of our networks can be revealed compared with FMNet [14] using our loss function. As shown in Fig. 4, it demonstrates that our proposed shared Des-CapsNet has a obviously faster iterative decline rate than FMNet, while FMNet has not yet completed convergence after iterating 10,000 times. And the SORCNet only needs about 2000 iterations to converge, which means the OptResBlock also has the effect of improving accuracy, and stabilize the convergence faster.

5.3 Evaluation metric

In our experiments, we measure correspondence quality according to the Princeton benchmark protocol [3]. To evaluate our method more intuitively and comprehensively, we calculate the average error and the cumulative geodesic error to estimate the accuracy of the shape correspondence. Given the registration mapping $T : A \rightarrow B$ and the ground-truth match $T_{true} : A \rightarrow B$ between two 3D shapes A and B, for each point $a \in A$ on shape A, the matching error is geodesic distance $d_B\,(T(a), T_{true}(a))$ between the predicted value $T(a)$ of point a on shape B and the ground-truth value $T_{true}(a)$.

Average error computes the average per-vertex error, which evaluates the geodesic error between a predicted correspondence vertex and its ground-truth. It can be calculated by:

$$ae = \frac{1}{N}\sum d_B(T(a), T_{true}(a)) \tag{21}$$

where N is the number of vertices.

Cumulative geodesic error measures matching quality plotting the percent of matches that have error smaller than a variable threshold. The geodesic error can be expressed as:

$$err(a) = \frac{d_B(T(a), T_{true}(a))}{Area(B)^{1/2}} \tag{22}$$

where $Area(B)$ is approximate to the sum of the triangle areas of shape B, used to normalize geodesic errors to eliminate the effects of shape scale transformations. The accuracy of the shape correspondence is defined as the proportion of points of which the matching geodetic distance error is below $err(a)$.

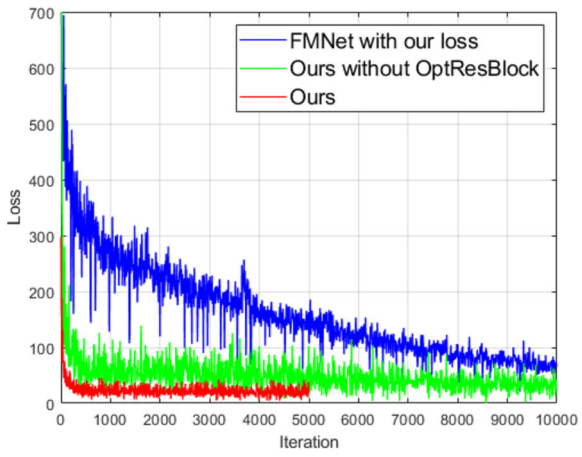

Fig. 4 The loss comparison of FMNet using our loss function, our SORCNet and SORCNet without OptResBlock in 5000 iterations when batch size is set to 2

5.4 Accuracy

For the input pair of shapes, the output of the network expresses the soft correspondence as an N × N matrix that represents the point correspondence probability. Then, by taking the maximum value for each column of it, we can get an N-d vector which indicates the registration information between the two shapes. To evaluate our method qualitatively, the different torso parts of a person are represented by different colors on the models. Figure 5 visualizes some typ-

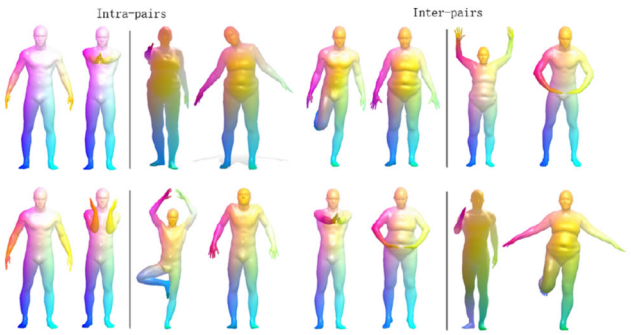

Fig. 5 Some examples of shape matching using our SORCNet method on FAUST dataset. Left four columns: intra-subject pairs; right four columns: inter-subject pairs. In all pairs: left mesh is colored using a predefined color map; right mesh is colored according to our computed correspondence

Table 1 3D shape correspondence error results

Method	intra AE	inter AE
RF [26]	15.04	17.05
3D-CODED [53]	1.98	2.88
FARM [54]	2.81	4.12
FMNet [14]	2.44	4.83
Halimi et al. [55]	2.82	3.40
Cyclic-FM [13]	2.12	4.07
SP [56]	**1.57**	3.13
Chen et al. [57]	4.86	8.30
SURFMNet [33]	1.73	3.63
Ours without OptResBlock	1.96	2.47
Ours	1.89	**2.28**

Comparison of the several existing methods and our proposed method using the average error as an indicator on the intra-FAUST and inter-FAUST datasets with 6890 vertices. Error is measured as the distance between mapped points and the ground-truth (cm)

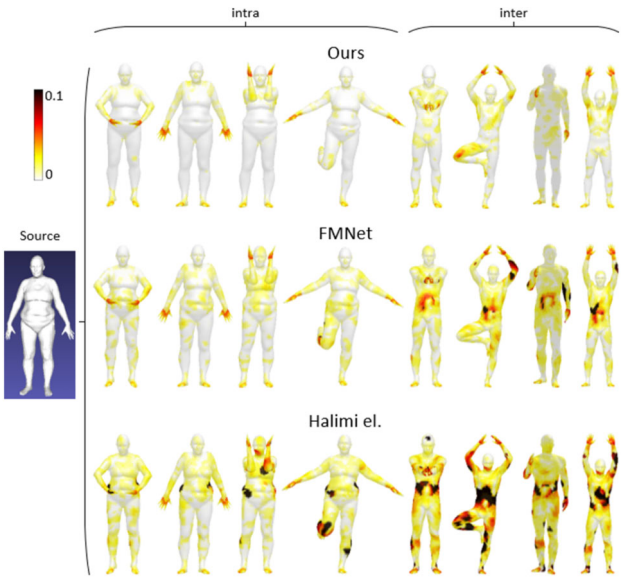

Fig. 7 Visualization of geodesic errors of different methods tested on FAUST dataset. Hot colors correspond to large errors

ical correspondence results of our SORCNet between shape pairs on intra-FAUST and inter-FAUST.

To evaluate the accuracy of our method in a quantitative way, Table 1 shows the average error on the state-of-the-art methods and our approach with intra-FAUST and inter-FAUST datasets. The bold represents the minimum value in each column. It shows that our method has the best results on inter-FAUST, and also performs well on intra-FAUST.

According to Eq. (22), we plot the cumulative curve to demonstrate the percentage of matches, which has the geodesic error smaller than a variable threshold. Figure 6 shows performance of our proposed networks compared with other shape correspondence methods on intra-FAUST and inter-FAUST datasets with 4096 vertices and 6890 vertices, respectively. As the threshold of geodesic distance error increases, the number of successfully matched points increases. It means that the faster the curve rises, the better the corresponding effect. It can be seen that when testing models on the remeshed FAUST with 4096 vertices, our method shows obviously better performance than other methods, which denotes that our model is very robust to different discretizations. But in the original test dataset, our method improves less compared with other methods, and the effect is even not the best on intra-FAUST. This indicates that the descriptor we learned is more robust and has a significant effect on different grid resolutions, while the accuracy of the state-of-the-art methods have achieved good results on the original dataset. The OptResBlock has a slight improvement on the basis of our shared Desc-CapsNet on the remeshed FAUST with 4096 vertices, while even minimal promotion on the original FAUST.

Figure 7 visualizes the geodesic error results of some methods on the original FAUST testing dataset. The geodetic error corresponding to the target shape is displayed with a

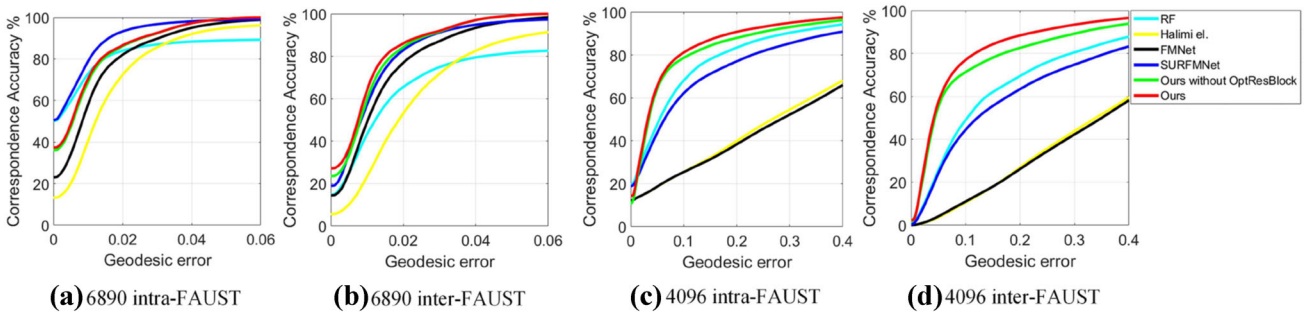

Fig. 6 Quantitative performance of point-to-point correspondences of different methods on the FAUST dataset. **a** and **b** The evaluation on the resampled intra-FAUST dataset with 4096 vertices and 6890 vertices, respectively. **c** and **d** The evaluation on the resampled inter-FAUST dataset with 4096 vertices and 6890 vertices, respectively

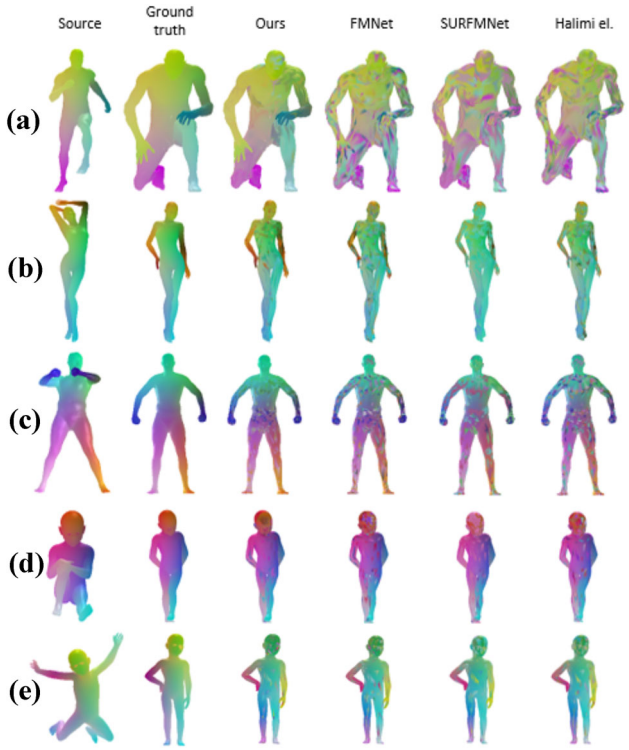

Fig. 8 Visualization of shapes matched to the reference shown on the left of pairs using several networks. Corresponding points should have the same color. **a** and **b**-TOSCA. **c**-SCAPE. **d** and **e**-KIDS are results on different datasets

color scale of 0 to 0.1. It shows the superiority of our method more intuitively.

5.5 Generalization

In order to evaluate the generalization of SORCNet, we test our network on the other non-rigid shape datasets using the model trained on the FAUST dataset. Figure 8 visualizes the point-to-point correspondence results on the remeshed TOSCA, SCAPE and KIDS datasets obtained by different networks trained on FAUST dataset. It shows that our model achieves better performance on these examples than other approaches.

Considering the shape correspondence accuracy of our method on other datasets, Fig. 9 shows the cumulative geodesic errors of our method experimented on the remeshed TOSCA, SCAPE and KIDS datasets compared with other approaches. In Table 2, we show the comparison of the mean geodesic errors experimentally obtained from TOSCA, FAUST and SCAPE datasets by different methods. The bold means the minimum value in each column. Figure 10 visualizes point-wise geodesic error on TOSCA, SCAPE, and KIDS datasets with a color scale of 0 to 0.4 in a more intuitive way. The experimental results show that our method has a good generalization on these datasets, and outperforms other methods using functional maps framework.

5.6 Effectiveness of the OptResBlock

In previous sections, we show that the addition of OptResBlock can make the network performs better. There is a question of where to put the optimization block in the party which has the best effect. In this section, we design experiments to explore the role of OptResBlock in the structure of the capsule network and have the higher influence on which parts. As the OptResBlock position shown in the upper box in Fig. 2 , We use the OptResBlock to modify the network structure before the primary capsule and the MLP after the latent capsule. Table 3 shows the mean geodesic error comparison of our networks on the remeshed test datasets when the OptResBlock is used in different parts of the Desc-CapsNet. The bold represents the minimum error in each row. The error is smaller when it is used before the primary capsule than that after the latent capsule, and the results were optimized by 5.5% and 12%, respectively, on average in both cases. It demonstrates that the optimization effect of the optimization building blocks on the primary capsules layer is better than after the latent capsules. Besides, on the TOSCA dataset, the effect of the network with OptResBlock acting after the latent capsules is even worse. It illustrates that regarding the

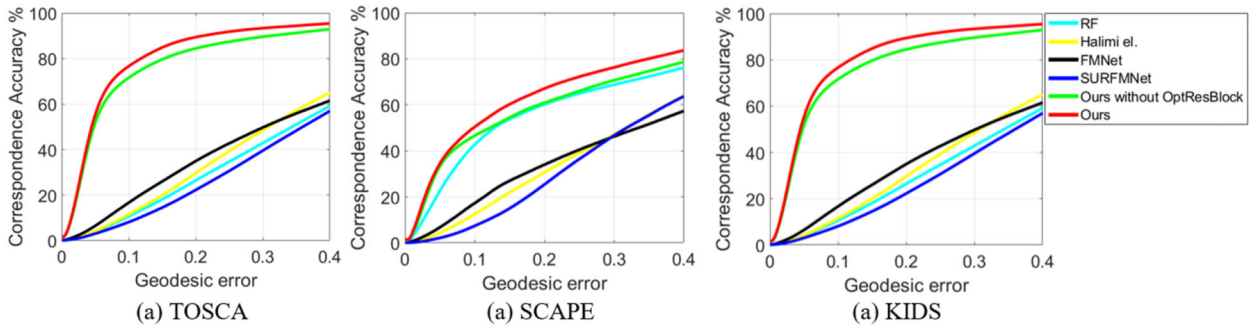

Fig. 9 Quantitative performance of point-wise correspondences comparing our method and other algorithms on the TOSCA, SCAPE and KIDS non-rigid shape dataset

Table 2 The comparison with some methods on FAUST, SCAPE, KIDS and TOSCA datasets with eight object categories

Method	TOSCA								SCAPE	KIDS	FAUST
	cat	centaur	david	dog	horse	michael	victoria	wolf			
FMNet [14]	0.3100	0.3752	0.3574	0.3745	0.3522	0.3709	0.3402	0.0360	0.3721	0.3601	0.3413
Halimi el. [55]	0.3126	0.3319	0.3466	0.3735	0.3544	0.3482	0.3439	0.0357	0.3415	0.3650	0.3362
RF [26]	0.3156	0.3800	0.3716	0.4157	0.3705	0.3713	0.3629	0.3930	0.2321	0.4106	0.1421
SURFMNet [33]	0.3572	0.4084	0.3659	0.3792	0.3496	0.3895	0.3727	0.0545	0.3485	0.3952	0.1673
Ours without OptResBlock	0.1124	0.1362	0.0982	0.1457	0.1228	0.0971	0.0764	0.0197	0.2158	0.1470	0.0982
Ours	**0.0958**	**0.1102**	**0.0845**	**0.1269**	**0.1090**	**0.0842**	**0.0675**	**0.0187**	**0.1857**	**0.1124**	**0.0768**

The datasets are resampled to 4096 vertices. Metric is the mean geodesic error (cm)

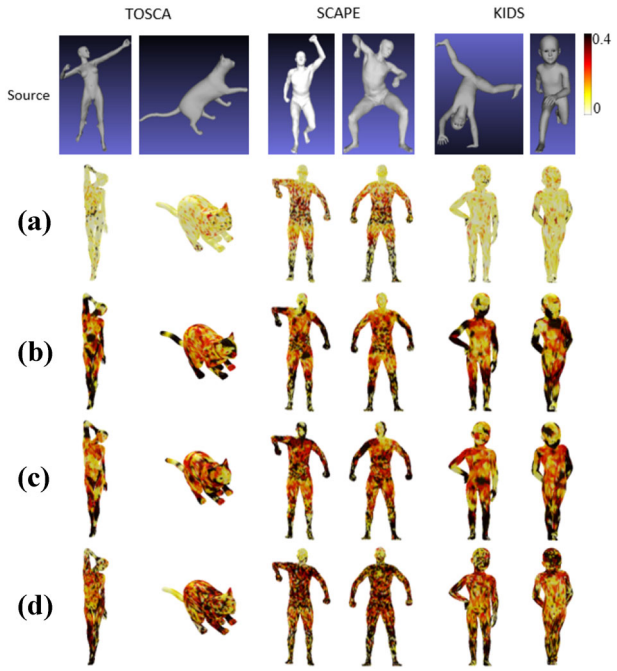

Fig. 10 Visualization of geodesic errors of several methods: **a**-Ours, **b**-FMNet, **c**-Halimi el., **d**-SURFMNet tested on remeshed shape pairs from TOSCA, SCAPE, and KIDS datasets

Table 3 Comparison of performance in terms of the mean geodesic errors (cm) on several remeshed datasets when the OptResBlock is used in different parts of the Desc-CapsNet

Datasets	Primary	Latent	Primary+Latent
FAUST	0.0886	0.0918	**0.0768**
TOSCA	**0.0765**	0.0915	0.0889
KIDS	0.1264	0.1368	**0.1124**
SCAPE	0.1995	0.2071	**0.1857**

network part before the potential capsule as an encoder and the subsequent network as a decoder, the optimization effect on the encoder performs better than that on the decoder.

Table 4 Comparison of performance in terms of the mean geodesic errors (cm) on remeshed TOSCA dataset using different routing procedures in capsule networks

Datasets	Ours	DR [20]	OptimCaps [58]	Xi el. [59]
wolf	**0.0187**	0.0194	0.0199	0.0190
victoria	0.0675	0.0678	**0.0666**	0.0694
michael	**0.0842**	0.0854	0.0856	0.0872
horse	**0.1090**	0.1121	0.1108	0.1118
dog	**0.1269**	0.1302	0.1289	0.1295
david	**0.0845**	0.0893	0.0893	0.0865
centaur	**0.1102**	0.1145	0.1186	0.1292
cat	**0.0958**	0.0977	0.0969	0.1003

5.7 Dynamic routing effectiveness

In Sect. 4, we modified the activation function for calculating coupling coefficients in the routing mechanism. The experiments in this section aim to investigate whether the advantage of CapsNet learning descriptor is affected in the case of transform routings. We train our network with different routing procedures on FAUST training dataset and test the models on TOSCA dataset. The performance is reported in Table 4. The bold means the minimum error in each row. It can be observed that the results did not fluctuate significantly under all routing procedures. From this experiment, we can conclude that the routing process has little effect on CapsNet's generalization ability. Given the performance variance for each model, the performance between different models is relatively small. The reason behind this is that coupling coefficients can be learned in transformation matrices implicitly, and all the models possess a similar transformation process.

In summary from the last and this sections, our experiments show that the high generalization ability and resolution robustness of Desc-CapsNet cannot be attributed to the routing process, but is more related to the composition of the primary capsules.

6 Conclusions

In this paper, we have introduced a novel and robust network for dense shape correspondence based on the functional maps-based DNN framework, which can effectively deal with the non-rigid deformations. Our approach is demonstrated to be superior through the evaluations on several challenging datasets, and can be adapted to different shape categories. We believe that the novel fusion of descriptor functions and deep learning is a promising direction since it is possible to meet the requirement of relevant 3D shape applications.

At present, local scale variation and topological changes still affect adaptability of the model. For many functional maps-based works, recovering the matrix **C** from the spectral representation of the descriptor has a relatively large challenge. So in the future study, we will extend function maps to the partial setting and search for additional descriptors with enhanced attributes. Another inspiration for future work is to suggest optimizing the derivation of the routing in the capsule network. The aggregation of the dynamic routing guided by attention blocks with different optimization methods can be derived according to the deduction formula, rather than a simple combination of CapsNet and residual blocks.

Acknowledgements This work was partially supported by the grants: NSFC 61972353, NSF IIS-1816511 and OAC-1910469.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Sansoni, G., Trebeschi, M., Docchio, F.: State-of-the-art and applications of 3D imaging sensors in industry, cultural heritage, medicine, and criminal investigation. Sensors **9**(1), 568–601 (2009)
2. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. Proc. Natl. Acad. Sci. **103**(5), 1168–1172 (2006)
3. Kim, V.G., Lipman, Y., Funkhouser, T.: Blended intrinsic maps. ACM Trans. Graphics **30**(4), 1–12 (2011)
4. Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L.: Functional maps: a flexible representation of maps between shapes. ACM Trans. Graphics **31**(4), 1–11 (2012)
5. Aubry, M., Schlickewei, U., Cremers, D.: The wave kernel signature: a quantum mechanical approach to shape analysis. In: Proceedings of IEEE International Conference on Computer Vision Workshops, pp. 1626–1633 (2011)
6. Li, P., Ma, H., Ming, A.: A non-rigid 3D model retrieval method based on scale-invariant heat kernel signature features. Multimedia Tools Appl. **76**(7), 10207–10230 (2017)
7. Bronstein, M.M., Bronstein, A.M., Kimmel, R., Yavneh, I.: Multigrid multidimensional scaling. Numer. Linear Algebra Appl. **13**(2–3), 149–171 (2006)
8. Coifman, R.R., Lafon, S., Lee, A.B., Maggioni, M., Nadler, B., Warner, F., Zucker, S.W.: Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps. Proc. Natl. Acad. Sci. **102**(21), 7426–7431 (2005)
9. Lipman, Y., Daubechies, I.: Conformal wasserstein distances: comparing surfaces in polynomial time. Adv. Math. **227**(3), 1047–1077 (2011)
10. Mateus, D., Horaud, R., Knossow, D., Cuzzolin, F., Boyer, E.: Articulated shape matching using Laplacian eigenfunctions and unsupervised point registration. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
11. Shtern, A., Kimmel, R.: Matching the LBO eigenspace of non-rigid shapes via high order statistics. Axioms **3**(3), 300–319 (2014)
12. Corman, É., Ovsjanikov, M., Chambolle, A.: Supervised descriptor learning for non-rigid shape matching. In: European Conference on Computer Vision, pp. 283–298. Springer (2014)
13. Ginzburg, D., Raviv, D.: Cyclic functional mapping: self-supervised correspondence between non-isometric deformable shapes. arXiv preprint arXiv:1912.01249 (2019)
14. Litany, O., Remez, T., Rodolà, E., Bronstein, A., Bronstein, M.: Deep functional maps: structured prediction for dense shape correspondence. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5659–5667 (2017)
15. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: Proceedings of European conference on computer vision, pp. 356–369. Springer (2010)
16. Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion. In: Computer Graphics Forum, vol. 28, pp. 1383–1392 (2009)
17. Litman, R., Bronstein, A.M.: Learning spectral descriptors for deformable shape correspondence. IEEE Trans. Pattern Anal. Mach. Intell. **36**(1), 171–180 (2013)
18. Dai, G., Xie, J., Zhu, F., Fang, Y.: Learning a discriminative deformation-invariant 3d shape descriptor via many-to-one encoder. Pattern Recognit. Lett. **83**, 330–338 (2016)
19. Papadakis, P., Pratikakis, I., Theoharis, T., Perantonis, S.: Panorama: a 3D shape descriptor based on panoramic views for unsupervised 3d object retrieval. Int. J. Comput. Vision **89**(2–3), 177–192 (2010)
20. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Proceedings of Advances in Neural Information Processing Systems, pp. 3856–3866 (2017)
21. Duarte, K., Rawat, Y., Shah, M.: Videocapsulenet: A simplified network for action detection. In: Proceedings of Advances in Neural Information Processing Systems, pp. 7610–7619 (2018)
22. Lin, A., Li, J., Ma, Z.: On learning and learned representation with dynamic routing in capsule networks. arXiv preprint arXiv:1810.04041 **2**(7) (2018)
23. Zhao, Y., Birdal, T., Deng, H., Tombari, F.: 3D point capsule networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1009–1018 (2019)
24. Biasotti, S., Cerri, A., Bronstein, A., Bronstein, M.: Recent trends, applications, and perspectives in 3d shape similarity assessment. In: Computer Graphics Forum, vol. 35, pp. 87–119. Wiley Online Library (2016)
25. Tam, G.K., Cheng, Z.Q., Lai, Y.K., Langbein, F.C., Liu, Y., Marshall, D., Martin, R.R., Sun, X.F., Rosin, P.L.: Registration of 3D point clouds and meshes: a survey from rigid to nonrigid. IEEE Trans. Visual Comput. Graphics **19**(7), 1199–1217 (2012)
26. Rodolà, E., Rota Bulò, S., Windheuser, T., Vestner, M., Cremers, D.: Dense non-rigid shape correspondence using random forests. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4177–4184 (2014)
27. Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M.: Geometric deep learning on graphs and manifolds using mixture model cnns. In: Proceedings of the IEEE Confer-

ence on Computer Vision and Pattern Recognition, pp. 5115–5124 (2017)
28. Wang, H., Guo, J., Yan, D.M., Quan, W., Zhang, X.: Learning 3d keypoint descriptors for non-rigid shape matching. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19 (2018)
29. Fey, M., Lenssen, J.E., Weichert, F., Müller, H.: Splinecnn: fast geometric deep learning with continuous b-spline kernels. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 869–877 (2018)
30. Ovsjanikov, M., Corman, E., Bronstein, M., Rodolà, E., Ben-Chen, M., Guibas, L., Chazal, F., Bronstein, A.: Computing and processing correspondences with functional maps. In: SIGGRAPH ASIA 2016 Courses, pp. 1–60 (2016)
31. Maron, H., Dym, N., Kezurer, I., Kovalsky, S., Lipman, Y.: Point registration via efficient convex relaxation. ACM Trans. Graphics **35**(4), 1–12 (2016)
32. Halimi, O., Litany, O., Rodolà, E., Bronstein, A., Kimmel, R.: Self-supervised learning of dense shape correspondence. arXiv preprint arXiv:1812.02415 (2018)
33. Roufosse, J.M., Sharma, A., Ovsjanikov, M.: Unsupervised deep learning for structured shape matching. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1617–1627 (2019)
34. Hinton, G.E., Krizhevsky, A., Wang, S.D.: Transforming auto-encoders. In: Proceedings of International Conference on Artificial Neural Networks, pp. 44–51. Springer (2011)
35. Hinton, G.E., Sabour, S., Frosst, N.: Matrix capsules with em routing. In: International Conference on Learning Representations (2018)
36. Chen, Z., Crandall, D.: Generalized capsule networks with trainable routing procedure. arXiv preprint arXiv:1808.08692 (2018)
37. Cheraghian, A., Petersson, L.: 3DCapsule: extending the capsule architecture to classify 3D point clouds. In: Proceedings of IEEE Winter Conference on Applications of Computer Vision, pp. 1194–1202 (2019)
38. Leung, F.H.F., Lam, H.K., Ling, S.H., Tam, P.K.S.: Tuning of the structure and parameters of a neural network using an improved genetic algorithm. IEEE Trans. Neural Netw. **14**(1), 79–88 (2003)
39. Domhan, T., Springenberg, J.T., Hutter, F.: Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)
40. Ma, L., Khorasani, K.: A new strategy for adaptively constructing multilayer feedforward neural networks. Neurocomputing **51**, 361–385 (2003)
41. Cortes, C., Gonzalvo, X., Kuznetsov, V., Mohri, M., Yang, S.: Adanet: adaptive structural learning of artificial neural networks. In: International Conference on Machine Learning, pp. 874–883 (2017)
42. Li, H., Yang, Y., Chen, D., Lin, Z.: Optimization algorithm inspired deep neural network structure design. arXiv preprint arXiv:1810.01638 (2018)
43. Mangasarian, O.L.: Nonlinear Programming. SIAM (1994)
44. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
45. Umehara, M., Yamada, K.: Differential geometry of curves and surfaces (2017)
46. Jia, B., Huang, Q.: De-capsnet: a diverse enhanced capsule network with disperse dynamic routing. Appl. Sci. **10**(3), 884 (2020)
47. Tay, Y., Bahri, D., Metzler, D., Juan, D.C., Zhao, Z., Zheng, C.: Synthesizer: Rethinking self-attention in transformer models. arXiv preprint arXiv:2005.00743 (2020)
48. Bogo, F., Romero, J., Loper, M., Black, M.J.: FAUST: Dataset and evaluation for 3D mesh registration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3794–3801 (2014)
49. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: Tensorflow: large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)
50. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
51. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: SCAPE: shape completion and animation of people. In: ACM SIGGRAPH, pp. 408–416 (2005)
52. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Numerical Geometry of Non-rigid Shapes. Springer (2008)
53. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: 3D-CODED: 3D correspondences by deep deformation. In: Proceedings of the European Conference on Computer Vision, pp. 230–246 (2018)
54. Marin, R., Melzi, S., Rodolà, E., Castellani, U.: Farm: Functional automatic registration method for 3d human bodies. In: Computer Graphics Forum, vol. 39, pp. 160–173. Wiley Online Library (2020)
55. Halimi, O., Litany, O., Rodola, E., Bronstein, A.M., Kimmel, R.: Unsupervised learning of dense shape correspondence. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4370–4379 (2019)
56. Zuffi, S., Black, M.J.: The stitched puppet: a graphical model of 3D human shape and pose. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3537–3546 (2015)
57. Chen, Q., Koltun, V.: Robust nonrigid registration by convex optimization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2039–2047 (2015)
58. Wang, D., Liu, Q.: An optimization view on dynamic routing between capsules (2018)
59. Xi, E., Bing, S., Jin, Y.: Capsule network performance on complex data. arXiv preprint arXiv:1712.03480 (2017)

Yuanfeng Lian is an associate professor of computer science in Beijing Key Laboratory of Petroleum Data Mining and College of Information Science and Engineering at China University of Petroleum, Beijing, China. He received his Ph.D. degree from Beihang University, China, in 2012 and M.S. degree from Changchun University of Technology, China, in 2003. His current research interests include computer graphics and image processing.

Dingru Gu is a M.Eng. candidate in the College of Information Science and Engineering in China University of Petroleum. She received her B.Eng. degree in the Computer Science and Technology from China University of Petroleum, China, in 2018. Her current research interests include computer graphics.

Jing Hua is a Professor of Computer Science and the founding director of Computer Graphics and Imaging Lab (GIL) and Vision Lab (VIS) at Computer Science at Wayne State University (WSU). He received his Ph.D. degree (2004) in Computer Science from the State University of New York at Stony Brook. His research interests include computer graphics, visualization, image analysis and informatics, computer vision, etc. He has authored over 100 papers in the above research fields. He received the Gaheon Award for the Best Paper of International Journal of CAD/CAM in 2009, the Best Paper Award at ACM Solid Modeling 2004 and the Best Demo Awards at GENI Engineering Conference 21 (2014) and 23 (2015), respectively. His research is funded by the National Science Foundation, National Institutes of Health, Michigan Technology Tri-Corridor, Michigan Economic Development Corporation and Ford Motor Company.