2021 IEEE/CVF International Conference on Computer Vision (ICCV) | 978-1-6654-2812-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCV48922.2021.01238

Geometric Deep Neural Network using Rigid and Non-Rigid Transformations for Human Action Recognition

Rasha Friji[1,2] Hassen Drira[3] Faten Chaieb[1,4] Hamza Kchok [5,3] Sebastian Kurtek [6]

[1] CRISTAL Lab - National University of Computer Science ENSI, Manouba University Campus, Manouba, Tunisia
[2] Talan Innovation Factory, Talan Tunisia, 10 Rue de l'énergie solaire Impasse N°1 Charguia 1, Tunis 2035, Tunisia
[3] IMT Lille Douai, CRIStAL UMR 9189, University Lille, F-59000 Lille, France
[4] AllianSTIC Lab - EFREI Paris, Villejuif, France
[5] INSAT, Tunisia
[6] Department of Statistics, the Ohio State University Columbus, OH, USA
racha.friji@talan.com hassen.drira@imt-lille-douai.fr faten.chakchouk@efrei.fr
hamza.kchok@insat.u-carthage.tn kurtek.1@stat.osu.edu

Abstract

Deep Learning architectures, albeit successful in most computer vision tasks, were designed for data with an underlying Euclidean structure, which is not usually fulfilled since pre-processed data may lie on a non-linear space. In this paper, we propose a geometry aware deep learning approach using rigid and non rigid transformation optimization for skeleton-based action recognition. Skeleton sequences are first modeled as trajectories on Kendall's shape space and then mapped to the linear tangent space. The resulting structured data are then fed to a deep learning architecture, which includes a layer that optimizes over rigid and non rigid transformations of the 3D skeletons, followed by a CNN-LSTM network. The assessment on two large scale skeleton datasets, namely NTU-RGB+D and NTU-RGB+D 120, has proven that the proposed approach outperforms existing geometric deep learning methods and exceeds recently published approaches with respect to the majority of configurations.

1. Introduction

Human behavior analysis via diverse data types has emerged as an active research issue in computer vision due to 1) the wide spectrum of not yet fully explored application domains, e.g., human-computer interaction, intelligent surveillance security, virtual reality, etc., and 2) the development of advanced sensors such as Intel RealSense, Asus Xtion and the Microsoft Kinect [49], which yield various data modalities, e.g., RGB and depth image sequences, and videos. Conventionally, these modalities have been utilized, solely [23, 37], or merged (e.g., RGB + optical flow), for action recognition tasks [35, 9] using multiple classification techniques, and resulted in excellent results. With the development of human pose estimation algorithms [8, 6], the problem of human joint (i.e., key-points) localization was solved and reliable acquisition of accurate 3D skeleton data became possible. In comparison with former modalities, skeleton data, a topological representation of the human body using joints and bones, appears to be less computationally expensive, and more robust in front of intricate backgrounds and with respect to variable conditions including viewpoints, scales and motion speeds. An efficient way to analyze 3D skeleton motions is to consider their shapes independently of undesirable transformations; the resulting representation space of skeleton data is then non linear.

Accordingly, we represent 3D skeleton landmarks in the Kendall shape space [16] that defines shape as the geometric information that remains when location, scaling and rotational effects are filtered out. A sequence of skeletons is then modeled as a trajectory on this space. Thus, to analyze and classify such data, it is more suitable to consider the geometry of the underlying space. This remains a challenging problem since most commonly used techniques were designed for linear data. Deep learning architectures, despite their efficiency in many computer vision applications, usually ignore the geometry of the underlying data space. Therefore, geometric deep learning architectures have been introduced to remedy this issue. To the best of our knowledge, the main previous geometric deep learning approaches were designed on feature spaces (e.g., SPD matrices, Grassmann manifold, Lie groups [14, 13]) or on the 3D human body manifold [2, 28]. The literature that considers this problem on shape spaces is scarce. Actually, an extension of a conventional deep architecture on a pre-

978-1-6654-2812-5/21/$31.00 ©2021 IEEE
DOI 10.1109/ICCV48922.2021.01238

shape space has been recently proposed in [10], and an auto encoder-decoder has been extended to a shape space for gait analysis in [11].

In this work, we propose a novel geometric deep learning approach on Kendall's shape space, denoted KShapeNet, for skeleton-based action recognition. Skeleton sequences are first modeled as trajectories on Kendall's shape space by filtering out scale and rigid transformations. Then, the sequences are mapped to a linear tangent space and the resulting structured data are fed into a deep learning architecture. The latter includes a novel layer that learns the best rigid or non rigid transformation to be applied to the 3D skeletons to accurately recognize the actions.

Contributions: The main contributions of this paper are:

1. We introduce a novel deep architecture on Kendall's shape space that deeply learns transformations of the skeletons for action recognition tasks.

2. The proposed deep network includes a novel transformation layer that optimizes over rigid and non rigid transformations of skeletons to increase action recognition accuracy.

Organization of the paper The rest of the paper is organized as follows. In Section 2, we briefly review existing solutions for action recognition and geometric deep learning. Section 3 describes geometric modeling of skeleton trajectories on Kendall's shape space. In Section 4, we introduce the proposed geometric deep architecture, KShapeNet. Experimental settings, results and discussions are reported in Section 5. Section 6 concludes the paper and summarizes a few directions for future work.

2. Related work

Geometric deep learning for action recognition has recently attracted a lot of attention from the research community. This has resulted in a variety of related approaches. Accordingly, we focus on highlighting the main categories in the areas of 3D action recognition and geometric deep learning. Interested readers can find exhaustive details in the associated recent surveys [29, 5].

2.1. Action recognition

Presently, deep learning methods for human action recognition are preferred over traditional skeleton-based ones, which tend to focus on extracting hand crafted features [15, 39]. The former methods can be categorized into three major sets: methods based on Recurrent Neural Network (RNN) [19], methods based on Convolutional Neural Network (CNN) [7], and methods based on Graph Convolutional Network (GCN) [17].

Since RNNs are convenient for time series data processing, RNN-based methods consider skeleton sequences as time series of coordinates of the joints. For the purpose of improving the capability of learning the temporal context of skeleton sequences, Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) have been introduced as efficient alternatives for skeleton-based action recognition. Zhu et al. [51] used an LSTM network and characterized joints through the co-occurrence between actions. In [48], geometric joint features are applied to a multi layered LSTM network instead of directly passing in the joint positions. The pitfall of some of these methods [22, 50] is their weak ability of spatial modeling, resulting in non competitive results. A novel two-stream RNN architecture was recently proposed by Hong and Liang [41]. This architecture models both the temporal dynamics and spatial configurations of skeleton data by applying an exchange of the skeleton axes at data level pre-processing. Relatedly, Jun and Amir [25] focused on extracting the hidden relationship between the two domains (spatial and temporal) using a traversal approach on a given skeleton sequence. Unlike the general method where joints are arranged in a simple chain ignoring kinectic dependency relations between adjacent joints, this tree-structure based traversal does not add false connections between body joints when their relation is not strong enough. Another pitfall of RNN based methods are the gradient exploding and vanishing problems over layers. Some new RNN architectures [21, 46] were proposed to address this particular limitation.

CNN models have excellent capability to extract high level information and semantic cues. Multiple works [43, 42, 47] have exploited CNN models for action recognition by encoding the skeleton joints as images or pseudo-images prior to feeding them to the network. In [47], Zhang et al. map a skeleton sequence to an image, referred to as the skeleton map, to facilitate spatio temporal dynamics modeling via the ConvNet. The challenge with CNN based methods is the extraction and utilization of spatial as well as temporal information from 3D skeleton sequences. Several other problems hinder these techniques including model size and speed [45], occlusions, CNN architecture definition [30], and viewpoint variations [47]. Skeleton based action recognition using CNNs thus remains a not completely solved research question.

Recently, the GCN has been adapted to action recognition. This network represents human 3D skeleton data as a graph. There are two main types of graph related neural networks: the graph recurrent neural network, and the graph convolutional neural network [44, 20].

2.2. Geometric deep learning

Compared to previous techniques, geometric deep learning is a nascent research area. As mentioned earlier, it studies the extension of existing deep learning frameworks and algorithms to effectively process graph and manifold data.

Some manifold based techniques have proven their success in 3D human action recognition due to view invariance of the manifold based representation of skeletal data. As examples, we cite the projection on Riemanian manifold [10], shape silhouettes in Kendall's shape space [1], and linear dynamical systems on the Grassmann manifold [38]. Geometric deep learning approaches can be categorized into two main classes: approaches on manifolds and approaches on graphs. This paper is related to deep approaches on manifolds, and thus, we give a quick review of the state-of-the-art in this category.

Manifold-based geometric deep learning approaches extend deep architectures to Riemannian manifolds, interpreted either as feature spaces [14, 12, 13] or the human body shape (i.e., the human body is viewed as a manifold) [2, 28]. Huang et al. proposed several networks on non linear manifolds. In [14], they introduced the first network architecture to perform deep learning on the Grassmann manifold. They presented competitive results on three datasets of emotion recognition, action recognition and face verification, respectively. Along similar lines, an architecture on the manifold of SPD matrices was proposed in [12], and similar experimental evaluation proved the effectiveness of this approach. Recently, the same authors proposed an architecture on Lie groups with application to skeleton-based action recognition [13]. These approaches investigate the non-linearity of various feature spaces, but did not consider shape spaces. Limited efforts have recently been made to design deep architectures on some shape-preshape spaces. Friji et al. [10] proposed a deep architecture on the sphere for modeling unit-norm skeletons with application to action recognition. Along similar lines, Hosni et al. [11] extended the auto-encoder to a shape space with application to gait recognition.

3. Modeling of shape space trajectories

We use the landmark shape based representation of the human skeleton, and geometric tools from Kendall's shape analysis [18] to model skeleton shapes and their temporal evolution. Every point in the shape space represents a single static action shape, and the distance between two such points illustrates the magnitude of shape discrepancies between the respective shapes.

Each skeleton X in an action sequence is represented as a set of n landmarks in $\mathbb{R}^3$, i.e., $X \in \mathbb{R}^{n\times 3}$. In our framework, we model skeletal shape sequences and use Kendall's shape representation to achieve the required invariances with respect to translation, scale and rotation. First, we perform data interpolation via cubic splines, to have the same number of frames for each sequence, rather than the commonly used zero-padding technique.

Translation and scale variabilities can be removed from the representation space via normalization as follows. Let H denote the $(n-1)\times n$ sub-matrix of a Helmert matrix, as detailed in [18], where the first row is removed. In order to center a skeleton X, we pre-multiply it by H, $HX \in \mathbb{R}^{(n-1)\times 3}$; then, HX contains the centered Euclidean coordinates of X. Let $C_0 = \{HX \in \mathbb{R}^{(n-1)\times 3} | X \in \mathbb{R}^{n\times 3}\}$, which is a $3(n-1)$ dimensional vector space, which can be identified with $\mathbb{R}^{3(n-1)}$. Using the standard Euclidean inner product (norm) on C_0, we scale all centered skeletons to have unit norm. As a result, we define the pre-shape space as $C = \{HX \in C_0 | \|HX\|^2 = (HX)^T(HX) = 1\}$; due to the unit norm constraint, C is a $(3n-4)$-dimensional unit sphere in $\mathbb{R}^{3(n-1)}$. Henceforth, we will refer to an element of C as $\tilde{X}$, i.e., a centered and unit norm skeleton. The tangent space at any pre-shape $\tilde{X}$ is given by $T_{\tilde{X}}(C) = \{V \in \mathbb{R}^{3n-4} | \langle V, X\rangle = V^T X = 0\}$.

In subsequent analyses, our representation of skeleton sequences further passes to the tangent space. Thus, it is useful to define three Riemannian geometric tools that allow one to map points 1) from the pre-shape space to a tangent space, 2) from a tangent space to the pre-shape space, and 3) between different tangent spaces. Task 1) can be achieved via the logarithmic map, $log_{\tilde{X}} : C \to T_{\tilde{X}}(C)$, defined as (for $\tilde{X},\ \tilde{Y} \in C$):

$$log_{\tilde{X}}(\tilde{Y}) = \frac{\theta}{\sin(\theta)}(\tilde{Y} - \cos(\theta)\tilde{X}), \tag{1}$$

where $\theta = \cos^{-1}\left(\langle \tilde{X}, \tilde{Y}\rangle\right)$ is the arc-length distance between $\tilde{X}$ and $\tilde{Y}$ on C. Task 2) is carried out via the exponential map, $exp_{\tilde{X}} : T_{\tilde{X}}(C) \to C$, defined as (for $\tilde{X} \in C$ and $V \in T_{\tilde{X}}(C)$):

$$\tilde{Y} = \cos(\|V\|)\tilde{X} + \sin(\|V\|)\frac{V}{\|V\|}, \tag{2}$$

where $\|V\| = \sqrt{V^T V}$ as before. Finally, for task 3), we use parallel transport, which, in short, defines an isometric mapping between tangent spaces. The parallel transport, $PT_{\tilde{X}\to\tilde{Y}} : T_{\tilde{X}}(C) \to T_{\tilde{Y}}(C)$ is defined as (for $\tilde{X},\ \tilde{Y} \in C$ and $U \in T_{\tilde{X}}(C)$):

$$PT_{\tilde{X}\to\tilde{Y}}(U) = U - \frac{\langle log_{\tilde{X}}(\tilde{Y}), U\rangle}{\theta}\left(log_{\tilde{Y}}(\tilde{X}) + log_{\tilde{X}}(\tilde{Y})\right), \tag{3}$$

where $\langle\cdot,\cdot\rangle$ and θ are the standard Euclidean inner product and the distance between $\tilde{X}$ and $\tilde{Y}$ on C, respectively, as before.

While translation and scale can be dealt with through normalization, rotation variability in Kendall's framework is removed algebraically using the notion of equivalence classes. The rotation group in $\mathbb{R}^3$ is given by $SO(3) = \{O \in \mathbb{R}^{3\times 3} | O^T O = I,\ det(O) = 1\}$. For $O \in SO(3)$ and $\tilde{X} \in C$, the action of the rotation group is given by matrix multiplication, i.e., $O\tilde{X}$ is a rotation of $\tilde{X}$. Let

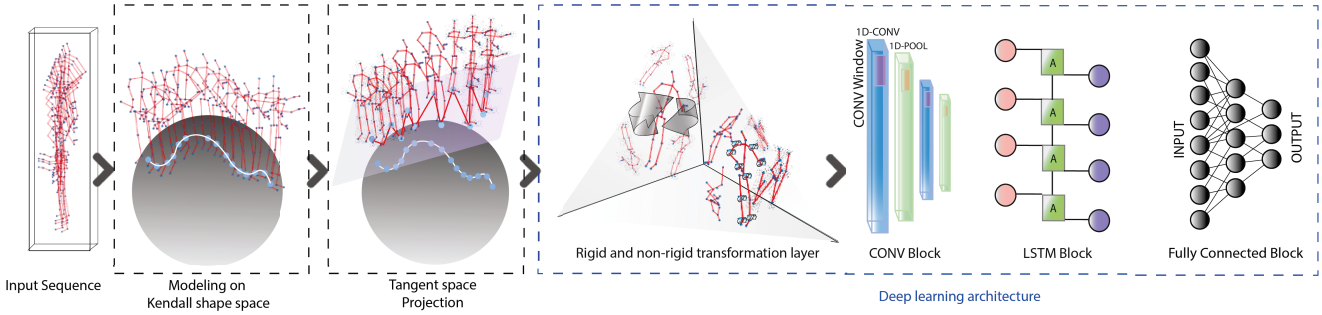

Figure 1. Illustration of KShapeNet full architecture and different blocks: 1- Modeling of the input sequence as trajectories on Kendall shape space. 2- Projection on the tangent space 3-Deep learning architecture embedding the rigid and non-rigid transformation layer.

$[\tilde{X}] = \{O\tilde{X} | O \in SO(3),\ \tilde{X} \in C\}$ denote an equivalence class of a pre-shape $\tilde{X}$. Then, Kendall's shape space is the quotient space $C/SO(3)$. Rotation variability is removed in a pairwise manner (or with respect to a given template), by optimally aligning two configurations $\tilde{X}$ and $\tilde{Y}$ via Procrustes analysis [18]; we omit the details of this process here for brevity. After optimal rotation, one can use the same Riemannian geometric tools as on the pre-shape space C, e.g., Equations 1-3, to model shapes of skeleton landmark configurations.

4. Shape space deep architecture

The proposed deep learning architecture of Kendall's Shape Space Network, KShapeNet, is illustrated in Fig. 1.

Input skeleton sequences are first modeled as trajectories on C, after which each skeleton $\tilde{X}$ is mapped to a common tangent space $T_{\tilde{X}_0}(C)$ at a reference shape $\tilde{X}_0$.The reference shape $\tilde{X}_0$ is defined as a pre-selected skeleton representing the neutral pose. Then, a transformation layer is built in this tangent space to increase global or local dissimilarities between class actions. This layer is followed by a CONV Block and a one-layer LSTM network, which learns the temporal dynamics of the sequences. As output, a fully connected block yields the corresponding action class. The CONV block consists of two 1D convolution layers followed by a pooling layer. For end-to-end network training, we use the cross-entropy loss as the training loss.

4.1. Optimization over rigid transformations

To optimize over rigid transformations, 3D rotations are applied to individual skeletons across sequences within this layer, and are updated during the training step.

Let $\tilde{Y}_i$ denote the i^{th} centered, unit norm skeleton in a sequence S, and $\hat{Y}_i$ its representative in the tangent space (reshaped from a $3(n-3)$ vector into a $3 \times (n-1)$ matrix represented in the ambient coordinates). The transformation layer is performed on each sequence resulting in a hidden output h, given by:

$$h_i = O_i \hat{Y}_i \tag{4}$$

where $O_i \in SO(3)$. In the back-propagation phase, the gradient descent adapts the kernels O_i directly so that they may not lie in $SO(3)$. To ensure that the updated kernels lie in $SO(3)$, we propose a second variant of this layer, denoted angle-based, where the optimization is performed over the rotation angles. Rotation matrices are then generated in the feed-forward pass.

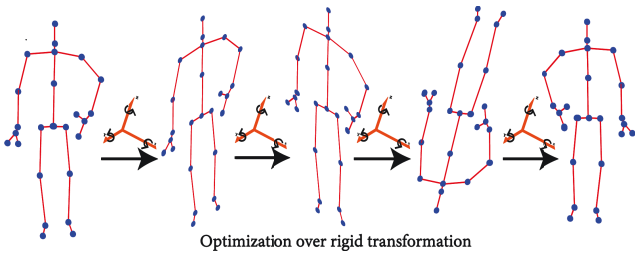

Figure 2. Optimization over rigid transformation: 3D rotations of the entire skeleton are applied during the training step.

Fig. 2 illustrates how this first category of optimization deals with the skeleton as one rigid entity, i.e., each transformation used for this optimization is applied to the whole skeleton.

4.2. Optimization over non rigid transformations

The optimization over local transformations is performed by finding the best rotations of 3D skeleton joints, with respect to the x, y and z axes, that improve performance on the action recognition task.

Let $\tilde{Y}_i$ denote the i^{th} centered, unit norm skeleton in a sequence S, $\hat{Y}_i$ its representative in the tangent space (reshaped from a $3(n-3)$ vector into a $3 \times (n-1)$ matrix represented in the ambient coordinates), and $q_i^j \in \mathbb{R}^3$ the j^{th} joint of $\hat{Y}_i$. The transformation layer is performed on

each sequence resulting in a hidden output h, given by:

$$h_i = \{O_{i,j} q_i^j\}_{j=1}^n, \tag{5}$$

where $O_{i,j} \in SO(3)$.

Similarly to the rigid transformation case, an angle-based optimization variant is proposed to ensure that each $O_{i,j}$ is a rotation matrix. In Section 5, we perform a study that compares the two variants for optimization over rigid and non rigid transformations: 1) the variant that allows the network to use general kernels as 3×3 matrices (not necessarily rotation matrices), and 2) the angle-based approach that constrains the network to allow rotation matrices only.

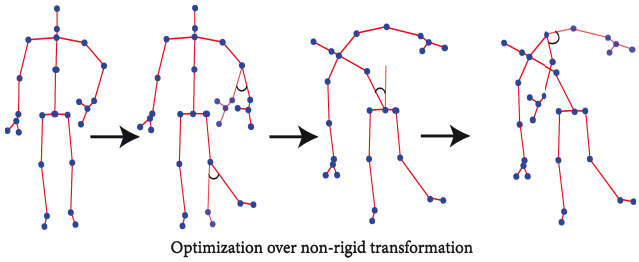

Figure 3. Optimization over non rigid transformation: 3D rotations are applied to joints during training step.

Fig. 3 depicts that, in contrast to optimization over rigid transformations, this second category of optimization deals with the skeleton as a non rigid entity, i.e., each transformation used for this optimization is applied on each joint individually.

5. Experimental results

First, in Section 5.1, we first describe the datasets used to validate our architecture and experimental settings. For the demonstration of KShapeNet efficiency, an ablation study is presented in Section 5.2 with the discussion of the impact of intermediate layers, i.e., the transformation layer and logarithmic map layer. Then, in Section 5.3, we compare the performance of action recognition of the KShapeNet architecture to state-of-the-art approaches on the same datasets. We conclude in Section 5.4 with the comparison and discussion of the different variants of the transformation as well as the projection on tangent space layers. The implementation code will be publicly released.

5.1. Datasets and settings

We evaluate the effectiveness of our KShapeNet framework on two large scale state-of-the-art datasets, NTU-RGB+D and NTU-RGB+D120.

NTU-RGB+D [31] is one of the largest 3D human action recognition datasets. It consists of 56,000 action clips of 60 classes. 40 participants have been asked to perform these actions in a constrained lab environment, with three camera views recorded simultaneously. The Kinect sensors estimate and record the 3D coordinates of 25 joints in the 3D camera's coordinate system. For standard assessment, we utilize two state-of-the-art protocols: cross-subject (CS) and cross-view (CV). In the cross-subject protocol, the 40 subjects are split into training and testing sets (20 subjects each) made up of 40,320 and 16,560 samples, respectively. In the cross-view protocol, we select the samples from cameras 2 and 3 for training, and the samples from camera 1 for testing. The training set then consists of the front and two side views of the actions, while the testing set incorporates left and right 45 degree views of the actions. For this assessment, the training and testing sets have 37,920 and 18,960 samples, respectively.

NTU120 RGB+D (NTU120) [24] is an extension of NTU60. It is the largest RGB+D dataset for 3D action recognition with 114,480 skeleton sequences. It contains 120 action classes performed by 106 distinct human subjects. For this dataset, the two protocols used for evaluation are cross-subject (CS) and cross-setup (Cset). For the cross-subject setting, half of the 106 subjects are used for training and the rest for testing. For the cross-setup setting, half of the setups are used for training and the rest for testing.

For the KShapeNet implementation, we set the number of frames to 100, and the batch size to 64 for the NTU dataset and 32 for the NTU 120 dataset. To estimate the model's parameters, we use the cross-entropy loss function and set the number of epochs to 30. The Adam optimizer is adapted to train the network, and the initial learning rate is fixed to 1×10^{-4} for both datasets. For training the network, we used a machine with a processor speed of 3.40 GHz, memory of 32 GB and an NVIDIA GTX 1070 Ti GPU.

5.2. Ablation study

In order to validate the effectiveness of the proposed framework and highlight the impact of each processing block, we performed an ablation study by gradually adding 1) the projection to tangent space block, and 2) the transformation layer. Table 1 reports the results of this study on the NTU and NTU120 datasets. In the first row, labeled "Baseline", we illustrate the results of the deep network (CNN-LSTM used in KShapeNet) obtained with input data represented on the pre-shape space (without moving to the linear tangent space) and without the optimization over rigid or non rigid transformations. The baseline architecture actually presents fairly satisfactory results. However, they are not competitive to those produced by state-of-the-art approaches.

The second row of Table 1 depicts the results achieved by adding the transformation layer to the baseline architecture.

Dataset	NTU-RGB+D		NTU-RGB+D120	
Protocol	CS	CV	CS	Cset
Baseline	85.1	91.2	56.0	63.5
Transformation layer only	89.6	91.5	57.2	63.8
Projection to tangent space only	94.1	95.5	63.9	65.3
Ours (KShapeNet)	**97.0**	**98.5**	**90.6**	**86.7**

Table 1. Ablation study results on the NTU and NTU120 datasets (% accuracy).

The transformation layer adopted here considers optimization over non rigid transformations using the angle-based variant. Further discussion about the choice of this configuration is presented in Section 5.4.2. With reference to the "Baseline" results, the transformation layer improves the recognition performance by 4.5% for CS and by 0.3% for CV on the NTU dataset, and by 1.2% for CS and 0.3% for Cset on the NTU120. As explained in Section 4, this configuration of the transformation layer optimizes over non rigid transformations, hence urging the network to find the best local rotations that are applied to the the skeleton sequences; this justifies the improvement of action recognition accuracy.

In the third row of Table 1, we present the results obtained by only adding the projection to tangent space block to the baseline model. The tangent space projection provides significant improvements in recognition performance, jumping from 85.1% to 94.1% for the CS protocol on the NTU dataset. The increase in accuracy due to the projection of skeleton sequences to the tangent space is the result of a new skeleton representation in this Euclidean space, allowing for the definition of a linear distance metric between skeleton shapes.

In the fourth row of Table 1, we report the final results produced by the KShapeNet framework, embedding both the projection on tangent space block and the transformation layer. KShapeNet results in a significant improvement over the baseline model, and most importantly, further increases recognition accuracy over the two models with individually added components (the projection on tangent space block or transformation layer). It is worth to point out that the combination of both components empowers the network to properly discriminate action classes. For instance, for the CS protocol on the NTU dataset, the accuracy increase due to the additional transformation layer was only 4.5% and the increase due to the projection to tangent space block was only 9%. The addition of both the transformation layer and the projection to tangent space block (i.e., KShapeNet) increased recognition accuracy by more than 11%. Accordingly, we conclude that the efficiency of KShapeNet is not only due to the advanced feature extraction capacity of the CNN-LSTM network, but equally due to the convenient data representation of skeleton shapes in the linear tangent space and the optimization over local rotational transformations.

5.3. Comparison to state-of-the-art approaches

In this section, we compare the performance of the proposed framework to state-of-the-art approaches on the two datasets, NTU and NTU 120.

Table 2 shows the results of the top performing state-of-the-art approaches on the NTU dataset, and compares them to the results of KShapeNet. In this table, we distinguish between three classes of action recognition methods: deep learning methods, Riemannian methods and hybrid (deep Riemannian) methods; our framework, KShapeNet, falls into the third category. The results demonstrate that KShapeNet consistently outperforms deep learning (leveraging CNNs and RNNs), Riemannian, and even hybrid approaches. Indeed, our method outperforms the best of these state-of-the-art approaches by 7.3% and 0.1% on the CS and CV settings, respectively. Comparing to the hybrid method of [13], in which the authors incorporate the Lie group structure into a deep network architecture using rotation mapping layers, our approach increases recognition accuracy by more than 35%.

Table 3 compares recognition accuracies between the most effective state-of-the-art approaches and KShapeNet on the NTU 120 dataset [27]. KShapeNet achieves competitive recognition results under the Cset protocol and outperforms the competitors under the CS protocol by 3.7%.

5.4. Additional studies

Next, we present intermediate experiments that were performed during the design of KShapeNet. In particular, we discuss the different configurations that were tested in terms of the variants of the transformation layer and the projection onto tangent space block.

5.4.1 Comparison of preprocessing techniques

It is worth-noting that we used the code of Maosen et al. [20] to generate input data for our algorithm: (1) to extract skeleton bodies and frames, (2) to extract joint coordinates, and (3) to split sequences into training and test sets for the different protocols. As an additional important data processing step, we interpolated the data using cubic splines to estimate equally-spaced skeleton trajectories, with constant change between frames. For comparison, we tested the network by zero padding the missing frames instead of interpolation. Since this operation results in frames that contain "wrong" data, the network is misled during the learning stage and the performance deteriorates. Table 4 reports the recognition results on the NTU RGB+D dataset with zero padding and with interpolation.

NTU-RGB+D Dataset		
Deep learning methods	Cross Subject	Cross View
Directed Graph Neural Networks[32]	89.9%	96.1%
Two stream adaptive GCN[33]	88.5%	95.1%
LSTM based RNN[47]	89.2%	95.0%
AGC-LSTM(Joints&Part)[34]	89.2%	95.0%
Riemannian methods	Cross Subject	Cross View
Lie Group [40]	50.1%	52.8%
Intrinsic SCDL [36]	73.89%	82.95%
Deep Riemannian methods	Cross Subject	Cross View
Deep learning on $SO(3)^n$ [13]	61.37%	66.95%
Ours (KShapeNet)	**97.0%**	**98.5%**

Table 2. Comparison to state-of-the-art top performing approaches on the NTU dataset.

NTU-RGB+D Dataset120		
Method	Cross Subject	Cross Setup
Tree Structure + CNN[3]	67.9%	62.8%
SkeleMotion[4]	67.7%	66.9%
Body Pose Evolution Map[26]	64.6%	66.9%
MS-G3D Net[27]	86.9%	88.4%
Ours (KShapeNet)	**90.6%**	**86.7%**

Table 3. Comparison to state-of-the-art top performing approaches on the NTU 120 dataset.

Protocol	CS	CV
Zero padding	81.3%	85.1%
Interpolation	97.0%	98.5%

Table 4. Comparison of results with zero padding and with interpolation on NTU RGB+D.

Dataset	NTU-RGB+D		NTU-RGB+D120	
Protocol	CS	CV	CS	Cset
Rigid Matrix based	97.0	97.1	90.2	85.9
Rigid Angle based	96.9	96.3	89.1	84.9
NonRigid Matrix based	96.8	96.9	90.6	84.3
NonRigid Angle based	**97.0**	**98.5**	**90.6**	**86.7**

Table 5. Comparison of different variants of the transformation layer (% accuracy).

5.4.2 Comparison of transformation layer variants

Table 5 presents a comparison of the four different variants of the transformation layer, based on the recognition results of KShapeNet, for the NTU and NTU120 datasets. Each row in Table 5 refers to one of the four tested settings of the transformation layer: 1) optimization over rigid transformations using the rotation matrix based variant (Rigid Matrix), 2) optimization over rigid transformations using the angle-based variant (Rigid Angle), 3) optimization over non rigid transformations using the rotation matrix based variant (NonRigid Matrix), and 4) optimization over non rigid transformations using the angle-based variant (NonRigid).

At a global level, we notice that the transformation layer (with only one exception) preserves state-of-the-art results on the NTU and NTU120 datasets; this means that performance is better on the CV protocol than the CS protocol for NTU, and it is better on the CS protocol than the Cset protocol for NTU120. At a granular level, we highlight two different behaviors of the optimization over rigid transformations and the optimization over non rigid transformations, with regards to the two different variants: rotation matrix-based and angle-based. On the one hand, the rotation matrix-based variant, which gives the network the liberty to optimize matrix coefficients without any constraints (updated matrices may not be in $SO(3)$), yields better results for the optimization over rigid transformations than for the optimization over non rigid ones. On the other hand, the angle-based variant, which only updates the angles resulting in elements of $SO(3)$, performs worse for rigid transformations than non rigid ones.

Rigid transformations, i.e., rotations of the entire skeleton, are characterized by preserving the skeleton's shape, distance and angle properties (i.e., all joints move in the same direction by the same amount). We argue that, for this reason, the rotation matrix-based variant is more ad-

equate for the optimization over such transformations. In other words, the rigid transformation is not subject to shape and angle variations, and the network tends to perceive the transformations applied to the skeleton as a one entity operation. Therefore, it is more efficient to allow the network to freely optimize over matrices during the back forward phase without the orthogonality constraint. As a result of the non rigid transformations, i.e., different rotations applied to all of the joints, the shape and angle properties of the skeletons are not preserved at each pass. Beyond the first feed forward pass, the network will alter the representation of each sequence. Thus, for the optimization over non rigid transformations, it is more convenient to constrain the network to allow rotations only. The rotation matrices are generated based on updated rotation angles, always resulting in elements of $SO(3)$.

For the final configuration of KShapeNet, we chose to optimize over non rigid transformations using the angle-based variant, allowing flexible modeling of inter-joint transformations; the corresponding recognition results are highlighted in bold in Table 5.

5.4.3 Comparison of projection to tangent space methods

As another intermediate experiment, we tested two variants of the projection to tangent space block. The first variant uses the logarithmic map to project all skeleton sequences to a single tangent space defined at a neutral reference skeleton. In this variant, the distances between skeleton shapes computed in the tangent space are different than those computed directly on Kendall's shape space, which introduces distortion (the only distances that are preserved after the projection are those from the reference to each projected shape). The issue is exacerbated when projecting skeleton shapes that are far away from the reference skeleton. Since all first frames of all skeleton sequences in the two datasets are neutral, i.e., they are very close to each other on Kendall's shape space, we considered that we can alternatively map each sequence to the tangent space defined at the skeleton shape corresponding to its first frame; we again use the logarithmic map for this projection. The pitfalls of this second variant are twofold: 1) distance computations are no longer executed between points in the same Euclidean space, but between points in a set of "nearby" planes, and 2) the tangent spaces generally have different coordinate systems.

To push the capabilities of our model, we next tried to incorporate parallel transport (PT) (refer to Section 3) as an alternative approach to map the skeleton sequences from the preshape space to the tangent space. In this approach, we first compute the shooting vectors between each consecutive frame within each sequence (using the logarithmic map). We then use PT to map these shooting vectors to the tangent space at the reference skeleton shape. Table 6 presents the results of applying the one-shot logarithmic map and the PT approach on the NTU dataset.

Dataset	NTU-RGB+D	
Protocol	CS	CV
Log map	97.0	98.5
Parallel Transport	96.8	96.7

Table 6. Comparison of performance when projecting to tangent space at the same reference skeleton using the logarithmic map and when using parallel transport (% accuracy).

Theoretically, PT should perform better than the direct projection to a tangent space at the reference skeleton shape since it remedies the distortion issues mentioned earlier. Nevertheless, as shown in Table 6, the simpler approach, paradoxically, tends to outperform the PT approach based on overall accuracy. In our implementation, the mapping to the tangent space iterations were not performed along the whole geodesic path, because this would have been computationally expensive. In fact, considering the computation-accuracy improvement trade-off, we decided that it was not worth to iterate the PT mapping along the entire geodesic path. This in part justifies the better performance of the simple logarithmic map to a common reference point over the more complicated PT approach.

At the end of the various experiments, we decided to adopt the following configurations for KShapeNet: projection on the tangent space using the logarithmic map with reference to the first frame, and optimization over non rigid transformations using the angle-based variant (which corresponding results are cited in Table 2 and Table 3).

6. Conclusion

In this paper, we proposed a geometric deep architecture, KShapeNet, for action recognition based on modeling human actions on Kendall's shape space. As part of our framework, we introduced a novel transformation layer to increase global or local dissimilarities between different types of actions. In the transformation layer, we optimize over rigid and non rigid transformations. In addition, we explored the use of two optimization variants: 1) rotation matrix-based, and 2) angle-based. We showed that the first variant "rotation matrix-based" is better suited for optimizing rigid transformations, while the second variant "angle-based" is more efficient for optimizing non rigid transformations. Extensive experiments, conducted on two challenging large benchmark datasets for action recognition, demonstrate that the proposed framework, KShapeNet, is exceeding state-of-the-art approaches recognition rates for the majority of configurations.

References

[1] Rushil Anirudh, Pavan K. Turaga, Jingyong Su, and Anuj Srivastava. Elastic functional coding of riemannian trajectories. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(5):922–936, 2017. 3

[2] Michael. M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 1, 3

[3] Carlos Caetano, François Brémond, and William Robson Schwartz. Skeleton image representation for 3d action recognition based on tree structure and reference joints. In *32nd SIBGRAPI Conference on Graphics, Patterns and Images*, pages 16–23, 2019. 7

[4] Carlos Caetano, Jessica Sena de Souza, François Brémond, Jefersson A. dos Santos, and William Robson Schwartz. Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition. pages 1–8, 2019. 7

[5] Wenming Cao, Zhiyue Yan, Zhiquan He, and Zhihai He. A comprehensive survey on geometric deep learning. *IEEE Access*, 8:35929–35949, 2020. 2

[6] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *CoRR*, abs/1812.08008, 2018. 1

[7] Guilhem Chéron, Ivan Laptev, and Cordelia Schmid. P-CNN: pose-based CNN features for action recognition. In *2015 IEEE International Conference on Computer Vision, ICCV 2015*, pages 3218–3226, 2015. 2

[8] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L. Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. pages 5669–5678, 2017. 1

[9] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016. 1

[10] Rasha Friji, Hassen Drira, and Faten Chaieb. Geometric deep learning on skeleton sequences for 2d/3d action recognition. In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2020, Volume 5: VISAPP*, pages 196–204, 2020. 2, 3

[11] Nadia Hosni and Boulbaba Ben Amor. A geometric convnet on 3d shape manifold for gait recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020*, pages 3725–3734, 2020. 2, 3

[12] Zhiwu Huang and Luc Van Gool. A riemannian network for SPD matrix learning. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. 3

[13] Zhiwu Huang, Chengde Wan, Thomas Probst, and Luc Van Gool. Deep learning on lie groups for skeleton-based action recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pages 1243–1252, 2017. 1, 3, 6, 7

[14] Zhiwu Huang, Jiqing Wu, and Luc Van Gool. Building deep networks on grassmann manifolds. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, pages 3279–3286, 2018. 1, 3

[15] Mohamed E. Hussein, Marwan Torki, Mohammad Abdelaziz Gowayyed, and Motaz El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In Francesca Rossi, editor, *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 2466–2472. IJCAI/AAAI, 2013. 2

[16] David. G. Kendall. Shape manifolds, procrustean metrics, and complex projective spaces. *Bulletin of the London mathematical society*, 16(2):81–121, 1984. 1

[17] Yinghui Kong, Li Li, Ke Zhang, Qiang Ni, and Jungong Han. Attention module-based spatial-temporal graph convolutional networks for skeleton-based action recognition. *J. Electronic Imaging*, 28(04):043032, 2019. 2

[18] Ian. L.Dryden and Kanti. V. Mardia. *Statistical Shape Analysis*. Wiley, 1998. 3, 4

[19] Guy Lev, Gil Sadeh, Benjamin Klein, and Lior Wolf. RNN fisher vectors for action recognition and image annotation. In *European Conference on Computer Vision - ECCV*, volume 9910, pages 833–850, 2016. 2

[20] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, pages 3595–3603, 2019. 2, 6

[21] Shuai Li, Wanqing Li, Chris Cook, Ce Zhu, and Yanbo Gao. Independently recurrent neural network (indrnn): Building a longer and deeper RNN. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5457–5466, 2018. 2

[22] Wenbo Li, Longyin Wen, Ming-Ching Chang, Ser-Nam Lim, and Siwei Lyu. Adaptive RNN tree for large-scale human action recognition. In *IEEE International Conference on Computer Vision, ICCV 2017*, pages 1453–1461, 2017. 2

[23] Ji Lin, Chuang Gan, and Song Han. Temporal shift module for efficient video understanding. *CoRR*, abs/1811.08383, 2018. 1

[24] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C. Kot. NTU RGB+D 120: A large-scale benchmark for 3d human activity understanding. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(10):2684–2701, 2020. 5

[25] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal LSTM with trust gates for 3d human action recognition. In *14th European Conference on Computer Vision - ECCV 2016*, volume 9907, pages 816–833, 2016. 2

[26] Mengyuan Liu and Junsong Yuan. Recognizing human actions as the evolution of pose estimation maps. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1159–1168, 2018. 7

[27] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Conference*

on Computer Vision and Pattern Recognition, CVPR, 2020. 6, 7

[28] Jonathan Masci, Davide Boscaini, Michael M. Bronstein, and Pierre Vandergheynst. Shapenet: Convolutional neural networks on non-euclidean manifolds. Technical report, 2015. 1, 3

[29] Bin Ren, Mengyuan Liu, Runwei Ding, and Hong Liu. A survey on 3d skeleton-based action recognition using learning method. *CoRR*, abs/2002.05907, 2020. 2

[30] Alejandro Hernandez Ruiz, Lorenzo Porzi, Samuel Rota Bulò, and Francesc Moreno-Noguer. 3d cnns on distance matrices for human action recognition. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 1087–1095, 2017. 2

[31] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A large scale dataset for 3d human activity analysis. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pages 1010–1019, 2016. 5

[32] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, pages 7912–7921, 2019. 7

[33] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, pages 12026–12035, 2019. 7

[34] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional LSTM network for skeleton-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1227–1236, 2019. 7

[35] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 568–576, 2014. 1

[36] Amor Ben Tanfous, Hassen Drira, and Boulbaba Ben Amor. Sparse coding of shape trajectories for facial expression and action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(10):2594–2607, 2020. 7

[37] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 1

[38] Pavan. K. Turaga and Rama Chellappa. Locally time-invariant models of human activities using trajectories on the grassmannian. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 2435–2441, 2009. 3

[39] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014*, pages 588–595, 2014. 2

[40] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595, 2014. 7

[41] Hongsong Wang and Liang Wang. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3633–3642, 2017. 2

[42] Pichao Wang, Zhaoyang Li, Yonghong Hou, and Wanqing Li. Action recognition based on joint trajectory maps with convolutional neural networks. *Knowl. Based Syst.*, 158:43–53, 2018. 2

[43] Yangyang Xu, Jun Cheng, Lei Wang, Haiying Xia, Feng Liu, and Dapeng Tao. Ensemble one-dimensional convolution neural networks for skeleton-based action recognition. *IEEE Signal Process. Lett.*, 25(7):1044–1048, 2018. 2

[44] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, pages 7444–7452, 2018. 2

[45] Fan Yang, Yang Wu, Sakriani Sakti, and Satoshi Nakamura. Make skeleton-based action recognition model smaller, faster and better. In *MMAsia'19: ACM Multimedia Asia*, pages 31:1–31:6, 2019. 2

[46] Shuai Yang, Juan Yu, Cheng Hu, and Haijun Jiang. Quasi-projective synchronization of fractional-order complex-valued recurrent neural networks. *Neural Networks*, 104:104–113, 2018. 2

[47] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive neural networks for high performance skeleton-based human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):1963–1978, 2019. 2, 7

[48] Songyang Zhang, Xiaoming Liu, and Jun Xiao. On geometric features for skeleton-based action recognition using multilayer LSTM networks. In *2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017*, pages 148–157, 2017. 2

[49] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE Multim.*, 19(2):4–10, 2012. 1

[50] Rui Zhao, Haider Ali, and Patrick van der Smagt. Two-stream RNN/CNN for action recognition in 3d videos. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages = 4260–4267, year = 2017.* 2

[51] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3697–3704, 2016. 2