Spectral Graph Skeletons for 3D Action Recognition

Tommi Kerola, Nakamasa Inoue, Koichi Shinoda

Dept. of Computer Science, Tokyo Institute of Technology, Japan
{kerola,inoue}@ks.cs.titech.ac.jp, shinoda@cs.titech.ac.jp

Abstract. We present spectral graph skeletons (SGS), a novel graph-based method for action recognition from depth cameras. The contribution of this paper is to leverage a spectral graph wavelet transform (SGWT) for creating an overcomplete representation of an action signal lying on a 3D skeleton graph. The resulting SGS descriptor is efficiently computable in time linear in the action sequence length. We investigate the suitability of our method by experiments on three publicly available datasets, resulting in performance comparable to state-of-the-art action recognition approaches. Namely, our method achieves 91.4% accuracy on the challenging MSRAction3D dataset in the cross-subject setting. SGS also achieves 96.0% and 98.8% accuracy on the MSRActionPairs3D and UCF-Kinect datasets, respectively. While this study focuses on action recognition, the proposed framework can in general be applied to any time series of graphs.

1 Introduction

We live in a world where machines are able to either aid or completely replace humans in a large variety of tasks. Most such tasks are quite trivial and monotonic, but thanks to the advent of machine learning, we are at the verge of being able to demand satisfying performance even for more complex tasks. One such task is action recognition. If machines could robustly recognize and interpret human actions and gestures, the benefits would be vast for a number of areas, including games, health care and the security industry.

Classic approaches to action recognition based on simple color images face numerous difficulties due to intra-class variations of actions, background clutter and illumination variations. However, thanks to the emergence of cheap and affordable depth maps with devices such as the Microsoft Kinect, there has been a recent increase in research using 3D features [13]. Leveraging 3D cameras solves the problem of separating the action subject from the video background, and also eliminates irrelevant information such as illumination variance. Recently, due to the work of Shotton *et al.* [25], we have access to low-dimensional skeletons mapped to the human body. Out of the box, these skeletons are much more discriminative than the raw high-dimensional RGB-D data and allow the development of efficient methods for action recognition. However, while the 3D

skeletons provide means of alleviating the action recognition task, they also provide new challenges due to unstable joint positions resulting from tracking errors in the noisy depth maps.

A recurring question in machine learning is the one of how to best represent objects for handling the pattern learning task. Generally, the approaches to this problem can be divided into two: statistical and structural [2]. While statistical methods have received a great deal of attention in the past years, we ask ourselves if objects are not better represented by an explicit structure suitable to the task at hand. Considering that the human skeleton may be viewed as a graph in 3D space (see Fig. 2), is it feasible to believe that patterns such as actions may be well represented by a time series of such graphs? This question is our motivation for exploring the usage of graphs for action recognition.

In real life problems, graphs can be found everywhere. They occur in forms of *e.g.* social- and transportation networks, finite state machines, and also in domains such as brain fMRI and computer graphics [7]. Recent approaches for using graphs in machine learning include graph kernels [1,14,38], generalizations of signal processing frameworks to the graph domain [7, 24], and also graph wavelets [5,6,12,20,22], such as the spectral graph wavelet transform [12]. While some difficulties and unsolved problems do remain, we believe that the future will hold even more promising new methods for the application of graphs in machine learning [2, 7].

In this paper, we propose to use the spectral graph wavelet transform (SGWT) framework of Hammond *et al.* [12] for the depth map action recognition task. Our method encodes body joint positions from a skeleton tracker [25] and embeds these on a temporal skeleton graph in 3D space. Graph wavelets capture information about a signal at different scales, in four dimensions on the temporal skeleton graph; along both 3D joint positions and time. Further, spectral graph wavelets offer more flexibility than classical wavelets due to the freedom of graph design and selection of spectral kernels. To capture the sequential behavior of actions, we utilize a temporal pyramid pooling scheme [11, 18, 29] on the wavelet coefficients. This improves over approaches that consider only global information [17, 34], since it allows us to capture differently segmented levels of temporal dependencies. Classification is finally performed using an off-the-shelf support vector machine (SVM). We name our action descriptor *spectral graph skeletons* (SGS), as it encodes the spectral content of a 3D skeleton sequence. Our proposed SGS descriptor has the following advantages:

- It is efficiently computable in $\mathcal{O}(T)$ time, where T is the number of frames in the action sequence. This makes it more computationally efficient than approaches that rely on solving heavy optimization problems [18, 30].
- Its underlying spectral basis is mathematically well defined [12], enabling analysis about each part of the descriptor. On the contrary, methods such as sparse coding [18] produce bases that are not easily analyzable.

To the best of our knowledge, this is the first application of graph signal processing to the action recognition task in computer vision. While this paper

focuses on recognition of actions, the framework can in general be applied to any time series of graphs.

The paper is organized as follows. Section 2 reviews related research in action recognition and graph signal processing. Section 3 provides a brief introduction to spectral graph wavelets. Our proposed method is then shown in Sec. 4, with related experiments in Sec. 5. Section 6 finally concludes the paper.

2 Related Work

2.1 3D Action Recognition

The advent of cheap 3D cameras such as the Kinect has enabled a great performance increase for action recognition tasks [17]. The availability of RGB-D data has considerably eased the task of segmenting an actor from its background; something that is normally quite challenging when using only RGB data. Related research in this field can be roughly divided into three categories: depth map-based, skeleton-based, and methods that utilize both.

Methods that make use of the raw depth map voxel data include Li *et al.* [17], who present a method where a bag of 3D points is sampled from 2D projections of salient depth map poses. Their results show that 3D action recognition clearly outperforms 2D approaches while additionally providing robustness against occlusions. Viera *et al.* [28] introduced space-time occupancy patterns (STOP), where the 3D points of the depth map are represented by a modified 4D histogram. Oreifej and Liu [21] learn a non-uniformly quantized 4D space, in which histograms of oriented 4D normals (HON4D) of the depth map are used for classification. Yang *et al.* [34] create DMM-HOG, which stacks orthogonally projected depth maps that are then applied to histograms of oriented gradients. Although depth map-based methods are able to capture information about shapes in great detail, they do however suffer from not knowing the correspondence between regions in the RGB-D data and the human body.

Other approaches rely only on the provided 3D skeletons. This includes DL-GSGC by Luo *et al.* [18], which uses sparse coding with constraints for group sparsity and feature geometry to increase the discriminative power. Together with max pooling and a temporal pyramid pooling scheme, their method also achieves an enhanced sequential representation structure. Zhao *et al.* [36] create SSS, which employs sparse coding and dictionary template learning to learn gestures based on distances between pairwise joints. Another method includes HOJ3D by Xia *et al.* [31], which applies linear discriminant analysis to create a time series of visual words (postures) that are then used as features in a hidden Markov model. Other methods use nearest-neighbor classifiers for classifying derivatives [35] (MP), or dimensionality-reduced relative measurements [33] (Eigenjoints) of 3D joint positions. Gowayyed *et al.* [11] create histograms of oriented displacements (HOD), where quantized angles of skeleton joints are applied to a temporal pyramid for handling temporal dependencies of actions. Ellis *et al.* [10] create a low latency scheme for classifying actions by finding canonical

poses using multiple instance learning. While their method is efficiently computable, it is unsuitable for actions that have a strict temporal structure rather than a characteristic pose, such as the action "*drawing an x*".

Finally, some works utilize both depth data and 3D skeletons simultaneously. Wang *et al.* [29] create an algorithm for selecting discriminative relative joint pairs that reduce ambiguity between action classes (AE). They also utilize a temporal pyramid, and classification is done using multiple kernel learning. Wang and Wu [30] develop MMTW for tackling temporal misalignment of actions by leveraging a discriminatively learned warping matrix for aligning action sequences before the classification step. Warping templates are learned one per class and classification is done using a latent structural SVM.

2.2 Signal Processing on Graphs

Recently, several techniques for generalizing classical signals processing (CSP) techniques to arbitrary graphs have been proposed [7]. Graph signal processing (GSP) provides graph analogs to classical Fourier transform tools, such as filtering, translation, convolution, *etc.*. CSP is restricted to signals in regular grids, but most natural signals do not follow this structure (*e.g.* sensor networks and anthropometric meshes). On the other hand, GSP allows processing signals on graphs that are directly adapted to the signal domain itself. By the increased freedom of graph design, we are able to extend CSP approaches to include additional information along *e.g.* extra added graph edges, ultimately increasing the descriptive power of the signal itself.

Several works have created wavelets on graphs using GSP [5, 6, 12, 20, 22]. One of the earliest works on graph wavelets include a method by Crovella and Kolaczyk [6] for analyzing computer traffic data on unweighted graphs. Hammond *et al.* [12] develop a spectral graph wavelet transform (SGWT), which allows analysis of localized signals on the graph Fourier spectrum of an undirected graph. We note that spectral graph wavelets can been seen related to sparse coding [27, 32]. The spectral graph wavelets are however more efficiently computable, since they are based on a fixed mathematical structure (see Sec. 3).

In addition to these frameworks, applications of graph signal processing include edge-aware image processing [19], depth video coding [15], image compression [23], anomaly detection in wireless sensor networks [9], bridge structure health monitoring [3], brain functional connectivity analysis [16] and mobility pattern prediction [8]. To the best of our knowledge, GSP has not before been applied to action recognition; this paper presents the first such study.

3 Background of Spectral Graph Wavelets

We briefly review some theory of graph signal processing and spectral graph wavelets; see Shuman *et al.* [7] and Hammond *et al.* [12] for details. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$ with $N = |\mathcal{V}|$ vertices. We let $\mathbf{W} \in \mathbb{R}^{N \times N}$ denote the weight matrix associated with $\mathcal{G}$, where $\mathrm{W}(n, m) \in \mathbb{R}^{+}$

is the weight of the edge between vertices n and m, or 0 if there is no edge. Then $\mathbf{Ł} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian matrix, where $\mathbf{D} = \text{diag}\{\mathbf{W1}\}$ is the diagonal degree matrix and $\mathbf{1}$ is the vector of all ones. We let $\{\lambda_\ell, \mathbf{u}_\ell\}_{\ell=0,\ldots,N-1}$ denote the eigenvalue and eigenvector pairs of $\mathbf{Ł}$. The spectrum of $\mathbf{Ł}$ carries a frequency interpretation [37], making it applicable for harmonic analysis on graphs. We will only consider undirected simple graphs, which makes all eigenvalues real and non-negative, since $\mathbf{Ł}$ is a real symmetric matrix [4].

A graph signal is a function $f : \mathcal{V} \to \mathbb{R}$ that assigns a value to each vertex. Such a signal can be represented as a vector $\mathbf{f} \in \mathbb{R}^N$ lying on a graph $\mathcal{G}$. By writing the eigendecomposition $\mathbf{Ł} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, frequency analysis of $\mathbf{f}$ can be performed by taking the graph Fourier transform (GFT) $\hat{\mathbf{f}} = \mathbf{U}^T\mathbf{f}$ [7]. Hammond *et al.* [12] define a spectral graph wavelet transform (SGWT) for graph signals on the eigenspectrum of $\mathbf{Ł}$.[1] Each spectral graph wavelet is realized by taking a kernel function $g : \mathbb{R}^+ \to \mathbb{R}^+$, scaling its domain by a scalar t, and finally localizing the result by convolving it with an impulse $\boldsymbol{\delta}_n \in \mathbb{R}^N$, which has value 1 at vertex n, and 0 everywhere else. A spectral graph wavelet $\boldsymbol{\psi}_{t,n} \in \mathbb{R}^N$ at scale t localized around vertex n can be written explicitly as

$$\psi_{t,n}(m) = \sum_{\ell=0}^{N-1} g(t\lambda_\ell)\mathrm{u}_\ell(n)\mathrm{u}_\ell(m), \; m = 1, \ldots, N \tag{1}$$

Given a graph signal $\mathbf{f}$, an SGWT coefficient is extracted by the inner product $\langle \boldsymbol{\psi}_{t,n}, \mathbf{f} \rangle$. The kernel g is chosen to act as the following band-pass filter [12]

$$g(x) = \begin{cases} x_1^{-\alpha} x^\alpha & \text{for } x < x_1 \\ s(x) & \text{for } x_1 \leq x \leq x_2 \\ x_2^\beta x^{-\beta} & \text{for } x > x_2 \end{cases} \tag{2}$$

where $\alpha = \beta = 2$, $x_1 = 1$, $x_2 = 2$ and $s(x)$ is a unique cubic spline that respects the curvature of g. Then, coefficients for smaller scales (small t) will localize high-frequency information around a vertex, while larger scales (large t) capture low-frequency information. The transform also includes a scaling kernel $h : \mathbb{R}^+ \to \mathbb{R}$, $h(x) = \gamma \exp(-(x/(0.6\epsilon))^4)$, for creating a scaling function $\boldsymbol{\phi}_n$ for stably representing low-frequency content in the graph [12]. Here, γ is set so that $h(0)$ equals the maximum value of g, and the design parameter $\epsilon = \lambda_{\max}/20$, where $\lambda_{\max}$ is an upper bound of the maximum eigenvalue of the graph Laplacian. The vector $\boldsymbol{\phi}_n$ is defined similarly to Eq. (1), with g replaced by h and setting $t = 1$.

Let M denote an integer such that the set of wavelet scales is $\{t_j\}_{j=1,\ldots,M}$. Then, the SGWT provides a transform with $M + 1$ scales; M wavelets and one scaling function. By gathering the wavelet and scaling function vectors in a transformation matrix $\mathbf{T} = [\boldsymbol{\psi}_{t_1,1}, \ldots, \boldsymbol{\psi}_{t_M,N}, \boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N]$, the transform coefficients can be expressed as a $(M + 1)N$-dimensional vector

$$\mathbf{c} = \mathbf{T}^T\mathbf{f} \tag{3}$$

[1] Online source code available at `http://wiki.epfl.ch/sgwt` .

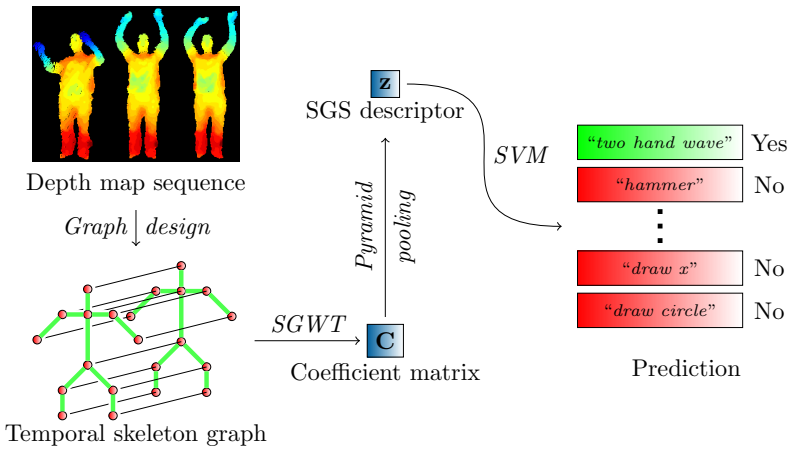

Fig. 1. Overview of the proposed action recognition system.

We also note that the SGWT is an overcomplete transform, as it contains more wavelet coefficients than vertices in the graph. If a signal is representable using only a few wavelet scales, then the SGWT can be viewed as quite similar to sparse coding [32], and each wavelet as an atom in a sparse dictionary [27]. However, since spectral graph wavelets are based on a fixed mathematical structure, they can be computed more efficiently, while sparse coding requires solving a heavy optimization problem [27]. It should be noted that while attempts to embed graph structure into the learned dictionary exists, this does not guarantee an efficient implementation [27]. Another advantage of spectral graph wavelets is that the explicit mathematical structure enables formal analysis of the effects of each wavelet basis.

In order to avoid explicit computation of the eigenspectrum of $\mathbf{Ł}$, which takes $\mathcal{O}(|\mathcal{V}|^3)$ time, the authors of the SGWT also introduce a method based on truncated Chebyshev polynomials for approximating the transform in $\mathcal{O}(|\mathcal{E}| + M|\mathcal{V}|)$ time [12]. Given a spectrum upper bound $\lambda_{\max}$, the approximation accesses $\mathbf{Ł}$ only through matrix-vector multiplications and is fast for sparse graphs. For the *normalized* graph Laplacian matrix $\boldsymbol{\mathcal{L}} = \mathbf{D}^{-1/2}\mathbf{Ł}\mathbf{D}^{-1/2}$, there is a trivial upper bound $\lambda_{\max} = 2$ for the maximum eigenvalue, which is tight when the graph is bipartite [4]. We will use this approximation for all practical purposes in this paper. We further note that the approximation has been shown to be computable in a distributed manner [26].

4 Spectral Graph Skeletons

This section presents our method for 3D action recognition using the SGWT. A method overview can be seen in Fig. 1. We limit our study to the quite elementary graph gained from the tracked skeleton as described below. While it is plausible to believe that better performance might be achieved through combined usage

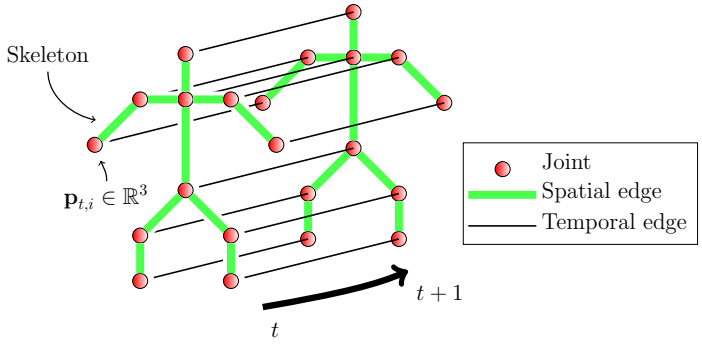

Fig. 2. Temporal skeleton graph, here shown partly by two temporally connected spatial skeleton graphs at frame t and $t+1$. Spatial edges between skeleton joints in the same frame and temporal edges between consequent frames in the action sequence are shown. Each joint i has a position $\mathbf{p}_{t,i} \in \mathbb{R}^3$. Note that the skeleton graph in this example is simplified for the purpose of illustration, and thus has fewer than the 20 joints given by Shotton *et al.* [25].

of both depth maps and skeletons, we are in this study specifically interested in investigating the representative power of the skeleton as a graph.

Joint position feature For action recognition, we first acquire a sequence of T depth images from a depth camera, such as the Microsoft Kinect, with each pixel indicating the z-location of the corresponding area. Then, we obtain a tracked skeleton [25] with $J = 20$ joints of the human body for each frame of the depth image sequence, where the i-th joint at frame t has a 3D position $\mathbf{p}_{t,i} = [\mathrm{x}_i(t), \mathrm{y}_i(t), \mathrm{z}_i(t)]^T$ (see also Fig. 2). As body size differs between different human subjects, we use the limb normalization procedure of Zanfir *et al.* [35] for normalizing skeleton limb length, while still keeping limb angles and positions intact. As noted in previous research [18,29], the relative inter-joint positions give quite discriminative features. As the center hip joint of the tracked 3D skeleton is deemed quite stationary throughout actions, we create a relative position vector $\hat{\mathbf{p}}_{t,i} = \mathbf{p}_{t,i} - \mathbf{p}_{t,\mathrm{center\_hip}}$ for describing the position of joint i.

Temporal skeleton graph A tracked 3D skeleton at time t can be represented by a graph $\mathcal{G}^{(t)}_{\mathrm{skel}} = (\mathcal{V}^{(t)}_{\mathrm{skel}}, \mathcal{E}^{(t)}_{\mathrm{skel}})$ with $|\mathcal{V}^{(t)}_{\mathrm{skel}}| = J$ vertices. Consider the case where we have a sequence of T such skeleton graphs. The GFT on the graph Laplacian of a 1D ring graph produces an eigenbasis equal to the basis of the DFT on the real line [37]. Therefore, we link together each pair of consecutive skeleton graphs, and also the graph from the last frame together with the first frame in order to create a "ring" structure. Explicitly, we can write $\mathcal{E}^{(t)}_{\mathrm{temporal}} = \{(v_{t,i}, v_{t',i}) : v_{t,i} \in \mathcal{V}^{(t)}_{\mathrm{skel}}, v_{t',i} \in \mathcal{V}^{(t')}_{\mathrm{skel}}\}$, $\mathcal{E}^{(t)}_{\mathrm{spatial}} = \mathcal{E}^{(t)}_{\mathrm{skel}}$, $\mathcal{E} = \bigcup_{t=1}^{T} \mathcal{E}^{(t)}_{\mathrm{temporal}} \cup \mathcal{E}^{(t)}_{\mathrm{spatial}}$, where $t' = (t \bmod T) + 1$. Then, using $\mathcal{E}$ and setting $\mathcal{V} = \bigcup_{t=1}^{T} \mathcal{V}^{(t)}_{\mathrm{skel}}$, we can design a temporal skeleton graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = TJ$, $|\mathcal{E}| = (|\mathcal{V}_{\mathrm{skel}}| + |\mathcal{E}_{\mathrm{skel}}|)T$, corresponding to the T frames long skeleton sequence, such that skeletons in consequent frames

have their joints linked together by temporal edges. Spatial edges in $\mathcal{G}$ therefore correspond to directly connected physical limbs of the human body. See Fig. 2 for a visual explanation.

As graph signals are scalars by definition (see Sec. 3), we process each axis of the 3D space separately, defining a graph signal $\mathbf{f}_a$ on $\mathcal{G}$ so that $\mathrm{f}_a(n) = \hat{\mathrm{p}}_{t,i}(a)$ at vertex $n = J(t-1)+i$, where $a \in \{1,2,3\}$ is the coordinate axis of choice. Edge weights in the graph are set as follows. Since a signal along a temporal edge can be assumed to be strongly correlated between vertices, we set temporal edge weights to unity. Spatial edges, on the other hand, cannot be assumed to follow the same phenomenon. We instead assume that a signal along a spatial edge provides relevant information inversely proportional to the distance between a pair of joints. Spatial edge weights are therefore set by a radial basis function $\alpha \exp(-\|\hat{\mathbf{p}}_{t,i} - \hat{\mathbf{p}}_{t,j}\|_2^2/(2\sigma^2))$, $\forall (v_{t,i}, v_{t,j}) \in \mathcal{E}_{\text{skel}}^{(t)}$, which gives spatially closer joints a higher weight. Here, $\alpha = 1$ is a fusion factor for weights between the temporal and spatial domains. We believe this factor is formally necessary since we cannot assume that these spaces should use the same measure of distance. At this stage, we do not however have any theoretical means of determining α, so we set it to unity. Further, since we can assume that σ is not equal for all connected joint pairs in the skeleton, we define a pair-specific set $\Sigma_{\text{spatial}} = \{\frac{1}{3}\sum_a \sigma_{i,j,\text{spatial}}(a) : (v_i, v_j) \in \mathcal{E}_{\text{skel}}\}$, where $\boldsymbol{\sigma}_{i,j,\text{spatial}} \in \mathbb{R}^3$ is a vector describing the axis-wise standard deviation between joints i and j. The set Σ_{spatial} can easily be estimated from training data. Assuming normalized skeleton size, edge weights in $\mathcal{G}$ will thus become time invariant.

Using the SGWT with the normalized Laplacian matrix $\boldsymbol{\mathcal{L}}$ for computational convenience, we extract wavelet coefficients from $\mathcal{G}$ at vertex n and scale t_j by calculating $\boldsymbol{\psi}_{t_j,n}^T \mathbf{F}$ as in Sec. 3, where $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3] \in \mathbb{R}^{N \times 3}$ is the matrix of concatenated axis-wise graph signals embedded on $\mathcal{G}$, and $\boldsymbol{\psi}_{t_j,n}$ is the spectral graph wavelet in Eq. (1). Consequently, each vertex will result in $M' = M + 1$ coefficients per axis, one for each wavelet scale (including the scaling kernel). The coefficients are represented by a coefficient matrix $\mathbf{C}$ similar to Eq. (3), but reshaped so that $\mathbf{C} \in \mathbb{R}^{T \times 3JM'}$. This will store the coefficients for each frame of the action sequence on each row.

Temporal pyramid pooling scheme In order to cope with varying action sequence length, we leverage a vector-valued pooling function $p : \mathbb{R}^{d \times 3JM'} \to \mathbb{R}^{3JM'}$ to create a feature vector $\mathbf{z} = p(\mathbf{C})$, where d is equal to the input matrix row count. The pooling function can for example be chosen as to do either absolute max or mean pooling along the temporal axis as

$$p_{\max}(\mathbf{C}) = \left[\max_t |\mathrm{C}(t,i)|\right]_{i=1,\ldots,3JM'} \tag{4}$$

$$p_{\text{mean}}(\mathbf{C}) = \left[\frac{1}{T}\sum_{t=1}^{T} |\mathrm{C}(t,i)|\right]_{i=1,\ldots,3JM'} \tag{5}$$

In the case of mean pooling, the resulting feature will encode the average acceleration for each axis and joint, windowed by SGWT kernels.

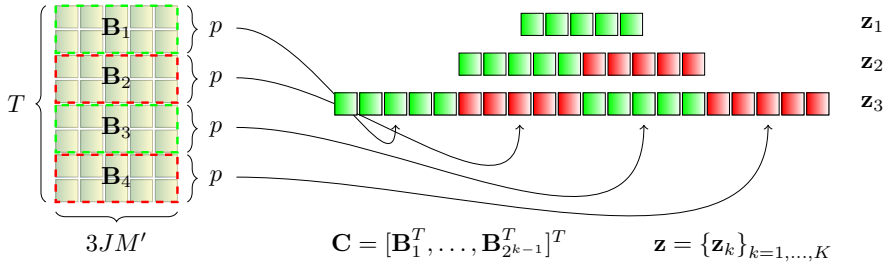

Fig. 3. Temporal pyramid pooling. Coefficient matrix $\mathbf{C}$ from a T frames long action sequence is pooled by a function $p : \mathbb{R}^{d \times 3JM'} \to \mathbb{R}^{3JM'}$ into $K = 3$ pyramid levels. The arrows illustrate the creation of the level 3 pyramid level vector $\mathbf{z}_3$. The final feature vector $\mathbf{z}$ is given by concatenation of the pyramid level vectors $\{\mathbf{z}_k\}_{k=1,\ldots,K}$.

Similar to previous research [11,18,29], we create a temporal pyramid of coefficients for capturing the temporal order of actions. Let K denote the maximum pyramid level. Then, the pooled feature vector at pyramid level $k \leq K$ is defined as $\mathbf{z}_k = [p(\mathbf{B}_1)^T, \ldots, p(\mathbf{B}_{2^{k-1}})^T]^T$, where $\{\mathbf{B}_i\}$ is a set of non-intersecting block matrices dividing $\mathbf{C}$ uniformly so that $\mathbf{C} = [\mathbf{B}_1^T, \ldots, \mathbf{B}_{2^{k-1}}^T]^T$. The final feature vector $\mathbf{z}$ is then a concatenation of the pyramid level vectors $\{\mathbf{z}_k\}_{k=1,\ldots,K}$. A visual explanation of the temporal pyramid pooling scheme applied to $\mathbf{C}$ can be seen in Fig. 3.

If we assume that an action is most often performed using a limited part of the body (*e.g.* just the right hand), then most elements of $\mathbf{z}$ will become close to zero. We therefore reduce the $(2^K - 1)3JM'$-dimensional $\mathbf{z}$ using PCA. After applying PCA to $\mathbf{z}$, we ℓ^2-normalize and finally classify each action using a standard SVM. Our action descriptor encodes the spectral content of a sequence of skeletons. We thus name it *spectral graph skeletons* (SGS). As computing the SGWT approximation [12] in $\mathcal{O}(|\mathcal{E}| + M|\mathcal{V}|)$ time is the most costly part of the descriptor creation process, we have that for one action sequence, the descriptor is computable in $\mathcal{O}(T)$ time, treating parameters J, K, M constant.

Comparison with previous methods While DMM-HOG [34] collapses the temporal variations into one axis, and thus suffers when temporal motion directionality is crucial, SGS is similar to STOP [28] and HON4D [21] in that we divide the space along the temporal axis. However, while STOP and HON4D only use the divided parts of the space separately, we combine them into a temporal pyramid like other works did [11, 18, 29] in order to capture both local and global information. MMTW [30] is able to find a non-uniform partition of the time axis that best captures discriminative parts of an action sequence, while our approach is more efficiently computable using only a uniformly partitioned temporal pyramid. Further, SGS is part of a group of methods that use relative joint positions [18, 29, 33, 35, 36]. While MP [35] and Eigenjoints [33] take relative velocity between joint pairs into account, SGS works well with just using the plain relative 3D positions. Since spectral graph wavelets can be seen related

Table 1. Recognition performance on the MSRAction3D dataset.

Method	Accuracy (%)
DL-GSGC [18]	96.7
MMTW [30]	92.7
MP [35]	91.7
SGS ($p_{\mathbf{mean}}$)	**91.4**
HOD [11]	90.2
HON4D [21]	88.9
AE [29]	88.2
SGS ($p_{\mathbf{max}}$)	**86.3**
SSS [36]	81.7
Canonical poses [10]	65.7
SGS (p_{mean}), no ring graph	87.6
SGS (p_{mean}), no SGWT	74.2

Table 2. Recognition performance on the MSRActionPairs3D dataset.

Method	Accuracy (%)
HON4D [21]	96.7
SGS ($p_{\mathbf{mean}}$)	**96.0**
SGS ($p_{\mathbf{max}}$)	**93.1**
AE [29]	82.2
DMM-HOG [34]	66.1

Table 3. Recognition performance on the UCF-Kinect dataset.

Method	Accuracy (%)
SGS ($p_{\mathbf{mean}}$)	**98.8**
SGS ($p_{\mathbf{max}}$)	**98.8**
MP [35]	98.5
Canonical poses [10]	95.9

to sparse coding, as previously noted in Sec. 3, our approach is also similar to sparse coding methods [18, 36], while being more efficiently computable.

5 Experiments

We test our proposed method on three publicly available datasets: MSRAction3D [17], MSRActionPairs3D [21] and UCF-Kinect [10]. The PCA dimension is set so that 98% of the variance explained by the principal components is retained. For the SVM, we use a radial basis function (RBF) kernel. Both max (Eq. (4)) and mean (Eq. (5)) pooling are tried. Pyramid level K and the number of spectral graph wavelet scales M are decided by stratified cross-validation on the training set of each dataset.[2] We describe our results on the datasets that follow.

5.1 Datasets and Results

MSRAction3D The MSRAction3D dataset [17] contains 10 subjects performing 20 different actions, of out which some are quite similar, such as "*draw x*" and "*draw circle*". Each subject performs each action up to three times; not necessarily in the same manner each time. Due to a large body of related research (see Sec. 2), this dataset has become quite a representative benchmark for 3D action recognition. Despite the availability of discriminative depth maps, this dataset remains quite challenging due to an abundance of visually similar actions as well as noisy joint positions. For fair comparison with previous research, we run

[2] In stratified cross-validation, the folds are selected so that the percentage of samples for each class in the dataset is preserved in each fold.

our experiments in the cross-subject setting, where samples from half of the subjects (*i.e.* subjects $1, 2, 3, 4, 5$) are used for training, and the rest for testing. This dataset contains some frames where the skeleton tracking fails, resulting in the joints to be erroneously located at the origin of the 3D coordinate system. We judge values to be missing only when the coordinates $(x, y, z) = (0, 0, 0)$, which Kinect outputs when the object is closer than 40 cm, or when no depth value could be found. For such missing values, the invalid joint positions are repaired using standard inter-frame linear interpolation.

The best parameters were $K = 4$ and $M = 50$ (decided by 5-fold cross-validation). PCA reduced the feature dimension from 45900 to 152. Results can be seen in Table 1. The confusion matrix is shown in Fig. 5. We see that mean pooling works better than max pooling, although both seem to be quite effective. Our SGS descriptor worked best with $K = 4$, but we note that even with $K = 1$ (no temporal pyramid), we got 83.5% recognition accuracy. Note that $K > 4$ could not be tested due to insufficiently long sequences in the dataset. Our method is able to fully distinguish between visually similar actions such as "*draw x/circle*" (see Fig. 4) and achieves perfect accuracy for most actions. On the other hand, the method repeatedly mistakes the action "*hammer*" for "*draw x*". These two classes are both characterized by similar highly accelerating movements along all axes of the 3D space. While SGS is able to capture different ranges of acceleration, it has trouble capturing the small temporal order of how these accelerations occur, which is an important point of future work. Although our method gains comparable results to most previous researches, it is unable to achieve results comparable to the sparse coding approach DL-GSGC [18]. Note however that our method has the advantage of being computable in time linear in the sequence length, while DL-GSGC requires solving a computationally heavy optimization problem. Our method falls just short of MMTW [30], but it should be noted that while MMTW discriminatively learns a non-uniform warping of the time axis, our method works with a mere uniform division of the action sequence due to our temporal pyramid pooling scheme. Augmenting our temporal pyramid with non-uniform division is a probable point of future work.

To illustrate the significance of using the SGWT, Table 1 includes a result of using SGS without the SGWT, where temporal pyramid pooling is applied directly to the raw 3D coordinates. The table also shows that connecting the last skeleton with the first, creating a "ring graph" provides a slight improvement in performance.

Earlier work has also reported results on three separate action sets of MSRAction3D. The three action sets are defined to group visually similar action classes together [17], in order to test performance on small sets of similar actions. Our experiments follow this setup and results are shown in Table 4. Contrary to the previous experiment, max pooling is here seen slightly superior to mean pooling, indicating that the choice of max or mean pooling might depend on datasets. We can see that in this scenario with fewer action classes, our method achieves performance closer to DL-GSGC while being more efficiently computable.

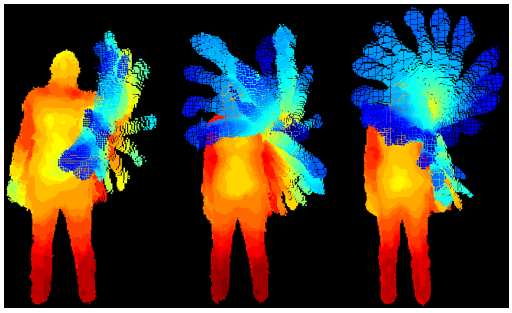

Fig. 4. Frontal view examples of the actions "*hammer*" (left), "*draw x*" (middle) and "*draw circle*" (right) in the MSRAction3D dataset.

Table 4. Recognition performance on the MSRAction3D dataset for the three different subject configurations on the three action sets as in Li *et al.*. [17] Each cell shows accuracy (%). Test 1 uses the first 1/3 samples for training and the rest for testing. Test 2 uses the first 2/3 samples for training and the rest for testing. The cross-subject test follows the same setup as in Table 1.

Method	Test 1				Test 2				Cross-subject test			
	AS1	AS2	AS3	Avg.	AS1	AS2	AS3	Avg.	AS1	AS2	AS3	Avg.
DL-GSGC [18]	100	98.7	100	99.6	100	98.7	100	99.6	97.2	95.5	99.1	97.3
SGS ($p_{\max}$)	**94.5**	**94.8**	**96.6**	**95.3**	**94.6**	**98.7**	**97.3**	**96.9**	**89.3**	**95.0**	**100**	**94.8**
SGS (p_{mean})	**96.6**	**90.8**	**98.0**	**95.1**	**98.6**	**96.0**	**98.6**	**97.7**	**88.4**	**91.6**	**100**	**93.3**
DMM-HOG [34]	97.3	92.2	98.0	95.8	98.7	94.7	98.7	97.4	96.2	84.1	94.6	91.6
STOP [28]	98.2	94.8	97.4	96.8	99.1	97.0	98.7	98.3	84.7	81.3	88.4	84.8
Eigenjoints [33]	94.7	95.4	97.3	95.8	97.3	98.7	97.3	97.8	74.5	76.1	96.4	82.3
HOJ3D [31]	98.5	96.6	93.5	96.2	98.6	97.9	94.9	97.2	88.0	85.5	63.5	79.0
Bag of 3D points [17]	89.5	89.0	96.3	91.6	93.4	92.9	96.3	94.2	72.9	71.9	79.2	74.7

MSRActionPairs3D [21] This dataset was created to test performance for recognizing action pairs that are similar in motion, and differ in motion directionality only. An example of such an action pair is "*pick up box*" and "*put down box*". The dataset contains six action pairs performed by ten subjects, each subject performed each action three times. We run our experiments in the cross-subject setting, where the first five actors are used for training, and the rest for testing. The best parameters were $K = 5$ and $M = 1$ (decided by 5-fold cross-validation). PCA reduced the feature dimension from 3720 to 80. Results on MSRAction-Pairs3D can be seen in Table 2. Our method achieves comparable performance to HON4D [21], despite using only skeleton information. Additionally, HON4D discriminatively learns a non-uniform quantization of the 4D space, while our method works with only a simple uniform quantization along time using the temporal pyramid. We note that our method gets accuracy 56.6% with $K = 1$ and 86.3% with $K = 2$, confirming the importance of the temporal pyramid pooling scheme for recognizing motion directionality.

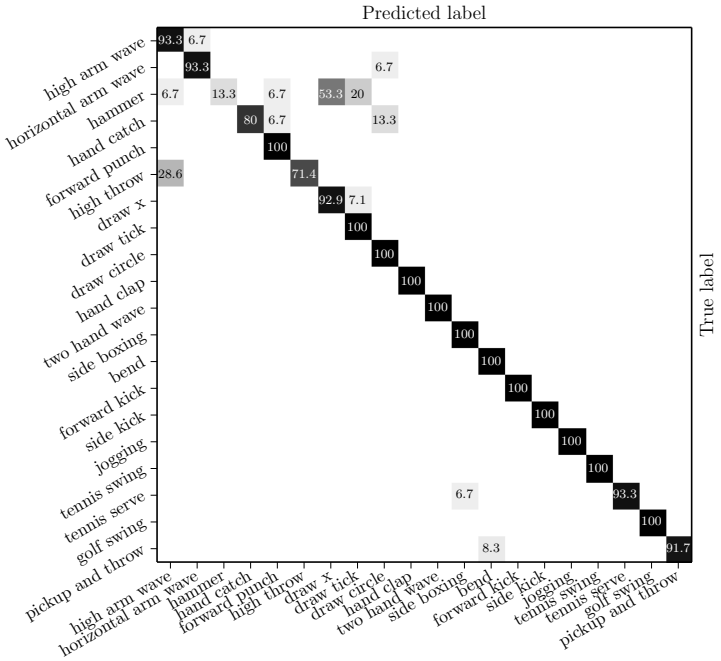

Fig. 5. Confusion matrix for using our method on the MSRAction3D dataset. Each cell shows classification accuracy (%) from white (0) to black (100) in the cross-subject setting. The average accuracy is 91.4%.

UCF-Kinect The UCF-Kinect dataset [10] contains presegmented actions suitable for games, *e.g.* "*climb ladder*", "*leap*" and "*twist left*", with 1280 action sequences in total. 16 actions are performed by 16 subjects, with each subject performing each action five times each. Note that in this dataset the provided skeletons only have 15 joints. As the center hip joint is missing, we approximate it by the average of the left and right hip joint positions. We run our experiments in the same setting as Ellis *et al.* [10], reporting the average accuracy of 4-fold cross-validation. The best parameters were $K = 3$ and $M = 43$ (decided by 4-fold cross-validation). PCA reduced the feature dimension from 18480 to 127. Results on UCF-Kinect can be seen in Table 3. We can see that our method achieves superior performance compared to the original canonical pose approach [10], while performing slightly better than MP [35]. This shows that our proposed framework is suitable for recognition of game-related actions that make use of all tracked parts of the body.

5.2 Discussion

Since the graph Laplacian matrix $\boldsymbol{\mathcal{L}}$ acts as a graph-analog to the classical Laplace operator [7], SGS is able to capture, per each joint and axis, the existence of ranges of acceleration. This range is determined by the SGWT kernel g. The window created by the SGWT kernel h in turn captures aggregated low-frequency information, such as the average position of the action in 3D space. We

believe that SGS is able to distinguish between actions that can be characterized by different acceleration at each joint. On the other hand, this means that SGS potentially has trouble separating sets of actions that have the same such characteristics. This became evident in MSRAction3D, where "*hammer/draw x/draw tick*" exhibit a set of actions that when looked at along each axis, display similar ranges of acceleration around the same spatial location. In its basic form ($K = 1$), SGS is not able to capture the order in which ranges of accelerations occur, something which is important for actions bound by motion directionality, such as the ones in MSRActionPairs3D. While using the temporal pyramid ($K > 1$) effectively helps capturing such temporal order, we believe that a non-uniform partition of the time-axis might be required to fully capture action classes that exhibit a very locally dependent temporal order.

6 Conclusion

We have presented spectral graph skeletons (SGS), a novel graph-based method for action recognition from depth cameras. Our method leverages the SGWT framework [12] for creating an overcomplete representation of an action signal lying on a 3D skeleton graph. The graph wavelet coefficients are applied to a temporal pyramid pooling scheme, which creates a descriptor of an action sequence. For a T frames long action sequence, the SGS descriptor is efficiently computable in $\mathcal{O}(T)$ time. The power of our method was demonstrated by experiments on three publicly available datasets, resulting in performance comparable to state-of-the-art action recognition approaches.

While this early study of using graph wavelets for action recognition has shown some promising results, it is still in its infant stage. Several aspects of the method are subject to further exploration, such as investigating the possibility of constructing graphs directly on the raw depth data, using subgraphs instead of the whole skeleton, or the suitability of the descriptor for real-time recognition. We would like to emphasize that optimized selection of the wavelet kernel g in Sec. 3 could lead to increased performance and is therefore an important point of future work. One strategy could be to learn the kernel discriminatively from training data. The weight settings of temporal and spatial edges should also be looked at. Other frameworks for processing graph signals should also be explored, together with a more detailed analysis of the suitability of graph signals for action recognition. This paper has focused on action recognition, but the proposed framework is in general applicable to any time series of graphs.

Acknowledgement. The first author acknowledges the Japanese Government (Monbukagakusho:MEXT) scholarship support for carrying out this research.

References

1. Bunke, H., Riesen, K.: Recent advances in graph-based pattern recognition with applications in document analysis. Pattern Recognition **44** (2011) 1057–1067

2. Bunke, H., Riesen, K.: Towards the unification of structural and statistical pattern recognition. Pattern Recognition Letters **33** (2012) 811–825
3. Chen, S., Cerda, F., Rizzo, P., Bielak, J., Garrett, J., Kovacevic, J.: Semi-supervised multiresolution classification using adaptive graph filtering with application to indirect bridge structural health monitoring. IEEE Transactions on Signal Processing **62** (2014) 2879–2893
4. Chung, F.R.: Spectral graph theory. American Mathematical Soc. (1997)
5. Coifman, R.R., Maggioni, M.: Diffusion wavelets. Applied and Computational Harmonic Analysis **21** (2006) 53–94
6. Crovella, M., Kolaczyk, E.: Graph wavelets for spatial traffic analysis. In: INFOCOM. (2003)
7. D. I Shuman, Narang, S.K., Frossard, P., Ortega, A., Vandergheynst, P.: The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Processing Magazine **30** (2013) 83–98
8. Dong, X., Ortega, A., Frossard, P., Vandergheynst, P.: Inference of mobility patterns via spectral graph wavelets. In: ICASSP. (2013)
9. Egilmez, H.E., Ortega, A.: Spectral anomaly detection using graph-based filtering for wireless sensor networks. In: ICASSP. (2014)
10. Ellis, C., Masood, S.Z., Tappen, M.F., Laviola Jr, J.J., Sukthankar, R.: Exploring the trade-off between accuracy and observational latency in action recognition. International Journal of Computer Vision **101** (2013) 420–436
11. Gowayyed, M.A., Torki, M., Hussein, M.E., El-Saban, M.: Histogram of oriented displacements (hod): describing trajectories of human joints for action recognition. In: IJCAI. (2013)
12. Hammond, D.K., Vandergheynst, P., Gribonval, R.: Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis **30** (2011) 129–150
13. Han, J., Shao, L., Xu, D., Shotton, J.: Enhanced computer vision with Microsoft Kinect sensor: A review. IEEE Transactions on Cybernetics **43** (2013) 1318–1334
14. Hermansson, L., Kerola, T., Johansson, F., Jethava, V., Dubhashi, D.: Entity disambiguation in anonymized graphs using graph kernels. In: CIKM, ACM (2013)
15. Kim, W.S., Narang, S.K., Ortega, A.: Graph based transforms for depth video coding. In: ICASSP. (2012)
16. Leonardi, N., Van De Ville, D.: Wavelet frames on graphs defined by fmri functional connectivity. In: ISBI. (2011)
17. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d points. In: CVPR Workshops. (2010)
18. Luo, J., Wang, W., Qi, H.: Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. In: ICCV. (2013)
19. Narang, S.K., Chao, Y.H., Ortega, A.: Graph-wavelet filterbanks for edge-aware image processing. In: Statistical Signal Processing Workshop (SSP), IEEE (2012)
20. Narang, S.K., Ortega, A.: Perfect reconstruction two-channel wavelet filter banks for graph structured data. IEEE Transactions on Signal Processing **60** (2012) 2786–2799
21. Oreifej, O., Liu, Z., Redmond, W.: Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In: CVPR. (2013)
22. Ram, I., Elad, M., Cohen, I.: Generalized tree-based wavelet transform. IEEE Transactions on Signal Processing **59** (2011) 4199–4209
23. Sandryhaila, A., Moura, J.M.F.: Nearest-neighbor image model. In: ICIP. (2012)

24. Sandryhaila, A., Moura, J.M.F.: Discrete signal processing on graphs. IEEE Transactions on Signal Processing **61** (2013) 1644–1656
25. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. Communications of the ACM **56** (2013) 116–124
26. Shuman, D.I., Vandergheynst, P., Frossard, P.: Chebyshev polynomial approximation for distributed signal processing. In: DCOSS. (2011)
27. Thanou, D., Shuman, D.I., Frossard, P.: Parametric dictionary learning for graph signals. In: IEEE GlobalSIP. (2013)
28. Vieira, A.W., Nascimento, E.R., Oliveira, G.L., Liu, Z., Campos, M.F.: Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences. In: CIARP. (2012)
29. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining actionlet ensemble for action recognition with depth cameras. In: CVPR. (2012)
30. Wang, J., Wu, Y.: Learning maximum margin temporal warping for action recognition. In: ICCV. (2013)
31. Xia, L., Chen, C.C., Aggarwal, J.: View invariant human action recognition using histograms of 3d joints. In: CVPR Workshops. (2012)
32. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR. (2009)
33. Yang, X., Tian, Y.: Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In: CVPR Workshops. (2012)
34. Yang, X., Zhang, C., Tian, Y.: Recognizing actions using depth motion maps-based histograms of oriented gradients. In: ACM MM. (2012)
35. Zanfir, M., Leordeanu, M., Sminchisescu, C.: The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In: ICCV. (2013)
36. Zhao, X., Li, X., Pang, C., Zhu, X., Sheng, Q.Z.: Online human gesture recognition from motion data streams. In: ACM MM. (2013)
37. Zhu, X., Rabbat, M.: Approximating signals supported on graphs. In: ICASSP. (2012)
38. Zhu, X., Kandola, J., Lafferty, J., Ghahramani, Z.: Graph kernels by spectral transforms. In Chapelle, O., Schoelkopf, B., Zien, A., eds.: Semi-Supervised Learning. MIT Press (2006) 277–291