Simplex-Based 3D Spatio-Temporal Feature Description for Action Recognition

Hao Zhang, Wenjun Zhou, Christopher Reardon, Lynne E. Parker
University of Tennessee, Knoxville, TN 37996, USA
{haozhang,wzhou4,creardon,leparker}@utk.edu

Abstract

We present a novel feature description algorithm to describe 3D local spatio-temporal features for human action recognition. Our descriptor avoids the singularity and limited discrimination power issues of traditional 3D descriptors by quantizing and describing visual features in the simplex topological vector space. Specifically, given a feature's support region containing a set of 3D visual cues, we decompose the cues' orientation into three angles, transform the decomposed angles into the simplex space, and describe them in such a space. Then, quadrant decomposition is performed to improve discrimination, and a final feature vector is composed from the resulting histograms. We develop intuitive visualization tools for analyzing feature characteristics in the simplex topological vector space. Experimental results demonstrate that our novel simplex-based orientation decomposition (SOD) descriptor substantially outperforms traditional 3D descriptors for the KTH, UCF Sport, and Hollywood-2 benchmark action datasets. In addition, the results show that our SOD descriptor is a superior individual descriptor for action recognition.

1. Introduction

Local spatio-temporal features have shown promising performance for human action recognition in unconstrained scenarios [5, 7, 13, 17, 23, 27, 30]. These features characterize local shape and motion variations, in space and time dimensions, and can provide robust representation of human actions against disturbing effects such as background clusters, occlusions, illumination, view variations, etc. Typically, local features are directly extracted from videos and thus avoid potential failures resulting from pre-processing steps, such as human segmentation. These desirable properties make local spatio-temporal features the most popular method to recognize actions, and continue to attract increasing attention from the computer vision community [7, 28].

Feature description is a fundamental research problem in local feature extraction [3, 13, 16, 23] aimed at construction of compact, descriptive representations of visual cues, including gradients and normals, computed within a feature's support region of a detected interest point. For example, the well-known scale-invariant feature transform (SIFT) [16] and histograms of oriented gradients (HOG) [3] descriptors quantize 2D gradients in a support region by computing a histogram from their orientations. Since the orientation of a visual cue is independent of its magnitude, which is usually affected by image noise and illumination changes, orientation quantization has proven to be a powerful, robust approach for feature description [3, 16, 27].

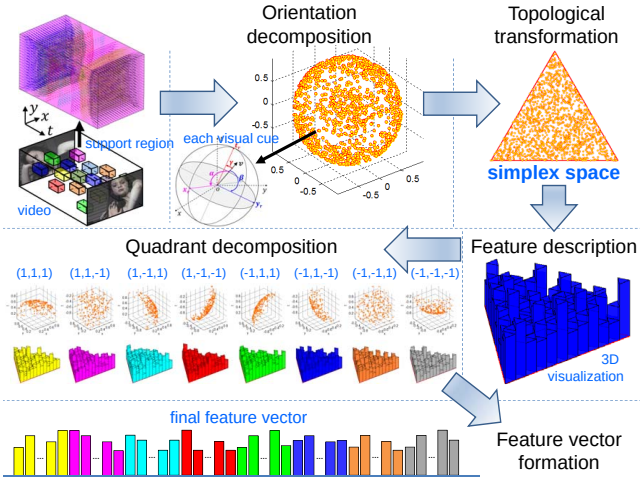
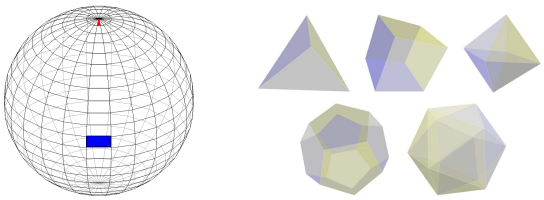


Figure 1: Overview of our novel *simplex-based orientation decomposition* feature descriptor to quantize and represent visual features in 3D space. Given a feature's support region containing a set of visual cues, our descriptor decomposes each cue's orientation into three angles. Then, the decomposed orientation vectors are transformed into the *simplex topological vector space*, and features are described in this space. After performing quadrant decomposition to further increase discrimination power, our SOD descriptor concatenates the histograms from all decomposed quadrants into a final feature vector.

To recognize unconstrained human actions, a large number of 3D local spatio-temporal features have been recently introduced that are computed in xyt (i.e., 2D spatial and 1D temporal) space [1, 4, 7, 13, 23]. Although orientation



(a) Spherical coordinate based (b) Regular polyhedron based

Figure 2: Issues of previous 3D feature description methodologies: Spherical coordinate based approaches suffer from the singularity issue (Figure 2a): bins at the poles (red triangle) are significantly smaller than bins around the equator (blue rectangle). Regular polyhedron based approaches have limited discrimination power (Figure 2b), since only five regular polyhedrons exist.

description in 2D space is intuitive and well defined, description of 3D features is much more challenging. Previous methods to describe 3D feature orientations can be generally categorized into two groups: spherical coordinate-based description and regular polyhedron-based description. As shown in Figure 2, spherical coordinate description of 3D features suffers from the singularity issue at the poles, while regular polyhedron descriptors have limited discrimination power due to the limited number of regular polyhedrons (discussed further in Section 2.1).

In this paper, we introduce a novel algorithm to describe visual features in 3D space, which addresses the singularity issue and provides a powerful description capability. The overview of our feature description algorithm is illustrated in Figure 1. Given the support region of a visual feature in 3D space (e.g., xyt spatio-temporal space), our description algorithm decomposes each 3D visual cue (e.g., gradients) into three dependent orientations. Then, all orientations are transformed into the standard 2-simplex topological vector space to deal with orientation dependency, and description is performed in the simplex topological vector space. Finally, to increase descriptive power, quadrant decomposition is performed to refine the quantization results. The final descriptor is a concatenated vector of the decomposed quantization results. Since our algorithm describes 3D features in the simplex topological vector space, we name it *Simplex-based Orientation Decomposition* (SOD) descriptor.

Our contributions are threefold. First, we introduce the novel simplex-based feature description algorithm to quantize and describe orientations of 3D visual features, which is an efficient, powerful, general algorithm to represent spatio-temporal (xyt) visual features in 3D space. Second, we develop visualization tools that can be applied to intuitively analyze feature characteristics in the abstract simplex topological space. Third, we empirically validate that visual features in 3D space, e.g., 3D local spatio-temporal features in xyt space, can greatly benefit from our descriptor, through demonstrating their state-of-the-art performance on unconstrained action recognition. The code of our SOD descriptor and its visualization tools are made available at: `http://dilab.eecs.utk.edu/SOD`.

The remainder of the paper is structured as follows. Section 2 discusses related studies. Then, Section 3 introduces our novel SOD algorithm for 3D feature description. Additional characteristics of our algorithm are discussed in Section 4. Experimental results are presented in Section 5. Finally, the paper is concluded in Section 6.

2. Related Work

In this section, we discuss previous 3D visual feature description methods and briefly review existing 3D features with the focus on human action recognition applications.

2.1. Description of 3D Features

A naive method to describe visual features in 3D space is to directly concatenate 3D visual cues, such as 3D gradients or normals, into a single vector [30]. However, this method is not robust [16, 27], since the magnitude of a visual cue is usually affected by image noise, illumination variations, etc. Because a visual cue's orientation is independent of its magnitude and is not similarly affected, orientation-based methodology dominates 3D feature description approaches.

A large number of 3D feature description methods are based on spherical coordinate systems [8, 10, 18, 23, 24, 29]. This description method applies polar angle θ and azimuthal angle ϕ in spherical coordinate systems to encode orientations and build orientation histograms. Then, θ and ϕ are divided into a set of bins, as illustrated in Figure 2a, which are used to construct a histogram of orientations of visual cues in a 3D feature's support region. However, as observed in [8, 13], spherical coordinate based descriptors suffer from the *singularity issue* at the poles, as in Figure 2a, where the blue bin near the equator is significantly larger than the red bin at the north pole.

Another popular 3D feature description methodology is based on regular polyhedrons [1, 7, 11, 13, 25]. This technique approximates the orientation space by a regular polyhedron with congruent faces that are regular polygons, each of which serves as a bin. Tracing each 3D vector along its direction up to the intersection with a polyhedron face identifies the bin. Then, a feature is described using a histogram of visual cues' orientations. Since only five regular polyhedrons exist that support a maximum of 20 bins, as depicted in Figure 2b, this methodology has *limited discrimination power* when quantizing a large number of distinct features.

Because our SOD descriptor transforms 3D visual cues to the simplex topological vector space instead of describing them in original Euclidian space, we are able to appropriately subdivide the transformed feature space and avoid the singularity and limited discrimination power issues.

2.2. 3D Features for Action Recognition

The large quantity of 3D spatio-temporal features proposed in recent years can be generally grouped based upon their information sources as follows:

- 3D spatio-temporal features computed in xyt spatio-temporal space using a temporal sequence of images, including 3D SIFT [18, 23], ST-SIFT [1], HOG3D [13], CHOG3D [11], 3D optical flow [10], etc.
- Multi-channel 3D features, typically computed in xyt spatio-temporal space and from multiple information channels, such as RGB and depth channels, including Color-SIFT [7], 4D-LST [30], etc.

The research problem we discuss in this paper, i.e., 3D feature description, is an integral part of the methods to extract the above-mentioned features. Our SOD descriptor is mathematically proven to work with any 3D vector and can be directly applied to each of these 3D features. The universal applicability to a large number of 3D features highlights the significance of our SOD descriptor.

It is also worth noting that, unlike feature encoding approaches such as unsupervised k-means and supervised entropy optimization [14], which aim to build a vocabulary of quantized features [2], our objective is to provide a description of each individual 3D visual feature.

3. The SOD Descriptor

In this section, we discuss our simplex-based orientation description algorithm. The goal is to construct a compact, representative description of 3D visual features. In particular, we describe 3D features in the simplex topological vector space to allow for appropriate subdivision of the 3D feature space. An overview of our SOD descriptor is depicted in Figure 1, and its algorithmic description is presented in Algorithm 1. Without loss of generality, we focus our discussion on describing 3D local spatio-temporal features that are extracted in xyt space.

3.1. Orientation Decomposition

The input to our SOD descriptor is the support region of a visual feature centered at a detected interest point in 3D space, which contains a set of 3D visual cues. An example of such a region containing 3D gradient cues in xyt space is visualized in Figure 3a. Given a support region, the goal of orientation decomposition is to decompose the orientation of each 3D visual cue into three angles.

Let $\boldsymbol{S} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\}$ denote a visual feature's support region that contains a set of 3D cues $\boldsymbol{v}_i = (x_i, y_i, t_i) \in \mathbb{R}^3$, $i = 1, \ldots, N$. Given a user-defined reference Cartesian coordinate system $\mathcal{C}$ defined by the unit vectors $\boldsymbol{v}_x^r$, $\boldsymbol{v}_y^r$ and $\boldsymbol{v}_t^r$ in the direction of x_r-axis, y_r-axis and t_r-axis, respectively, the orientation of $\boldsymbol{v}$ can be decomposed into three angles α, β, and γ with respect to the reference coordinates, which can be computed in constant time by:

$$\cos\alpha = \frac{\boldsymbol{v}\cdot\boldsymbol{v}_x^r}{\|\boldsymbol{v}\|}, \quad \cos\beta = \frac{\boldsymbol{v}\cdot\boldsymbol{v}_y^r}{\|\boldsymbol{v}\|}, \quad \cos\gamma = \frac{\boldsymbol{v}\cdot\boldsymbol{v}_t^r}{\|\boldsymbol{v}\|} \qquad (1)$$

The definitions of the decomposed angles are illustrated in Figure 3b. To allow for flexible orientation decomposition, the reference coordinate system does not necessarily overlap the standard Cartesian coordinate system that is represented by the standard basis $\boldsymbol{i} = (1, 0, 0)$, $\boldsymbol{j} = (0, 1, 0)$, and $\boldsymbol{k} = (0, 0, 1)$ in the directions of x-axis, y-axis and t-axis, respectively, as shown in Figure 3b.

It is noteworthy that description through independently dividing α, β and γ into equally sized cells in 3D space is problematic, because the decomposed angles α, β and γ are not independent, as will be demonstrated by Eq. (3). For example, when α, β and γ are equally divided into six bins (a total number of 6^3 cells in 3D space), the 3D cell representing the angle range α, β, $\gamma \in [5\pi/6, \pi)$ can never be assigned by any cues, due to the constraints of the decomposed angles. We name this problem *constrained orientation quantization*, and for this reason it is not appropriate to independently discretize the angles into bins in 3D space.

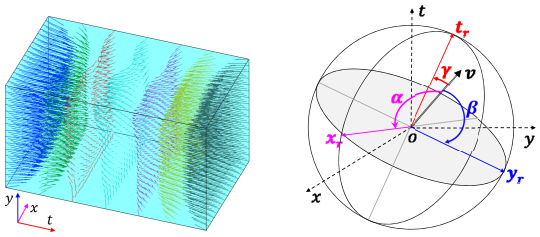
3.2. Transformation to Simplex Space

We provide an elegant solution to the constrained orientation quantization problem to describe 3D visual features. Our novel visual feature description algorithm is based on the topological concept of simplex [6, 19, 21], which is a generalization of a tetrahedral region of space to arbitrary dimensions. Specifically, an n-simplex is the smallest closed convex set that contains $n+1$ vertices. For example,

Algorithm 1: Simplex-based 3D feature description

Input : $\boldsymbol{S} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\}$ (3D support region), $\mathcal{C} = \{\boldsymbol{v}_x^r, \boldsymbol{v}_y^r, \boldsymbol{v}_z^r\}$ (reference Cartesian coordinate), k (parameter of edgewise simplex subdivision)
Output : $\boldsymbol{f}(\boldsymbol{S})$ (feature vector)

1: **for** $i \leftarrow 1$ **to** N **do**
2: Decompose the orientation of $\boldsymbol{v}_i$ by computing $\cos\alpha$, $\cos\beta$, and $\cos\gamma$ with respect to $\mathcal{C}$ acc. to Eq. (1);
3: Transform $\boldsymbol{v}_i$ into the standard 2-simplex topological vector space Δ^2: $\boldsymbol{\delta}_i = \{\cos^2\alpha, \cos^2\beta, \cos^2\gamma\}$;
4: Compute indices $\boldsymbol{i}(\boldsymbol{\delta_i}) = (r(\boldsymbol{\delta_i}), c(\boldsymbol{\delta_i}), l(\boldsymbol{\delta_i}))$, acc. to Eq.(5–7), of the sub-simplex in k edgewise subdivision;
5: Compute decomposed orientation quadrant assignment $\boldsymbol{q}(\boldsymbol{\delta_i})$ acc. to Eq. (8);
6: Increase the count of the sub-simplex indexed by $\boldsymbol{i}(\boldsymbol{\delta}_i)$ in quadrant $\boldsymbol{q}(\boldsymbol{\delta}_i)$ by one;
7: **end**
8: Form $\boldsymbol{f}(\boldsymbol{S})$ by concatenating counts of the sub-simplices in all eight quadrants;
9: **return** $\boldsymbol{f}(\boldsymbol{S})$



(a) Input visual cues (b) Angle decomposition

Figure 3: Orientation decomposition: given a feature's support region, shown in Figure 3a computed from seven temporally adjacent frames, each 3D visual cue's orientation is decomposed into three angles (α, β and γ) with respect to a user-defined reference Cartesian coordinate system defined by axes x_r, y_r, and t_r (Figure 3b).

a 1-simplex is a line segment that contains two vertices, and a 2-simplex is a triangle that is specified by three vertices.

We start discussion of our novel *simplex-based orientation decomposition* descriptor by showing that each 3D cue can be transformed into a standard simplex topological vector space, where a standard n-simplex is a simplex whose edges have the same length. This is mathematically defined, in the context of a topological vector space, as follows:

Definition 1 (Standard n-simplex)**.** *The standard n-simplex is defined as a topological vector space that is the subspace of $\mathbb{R}^{n+1}$ satisfying:*

$$\Delta^n = \left\{ (\delta_0, \cdots, \delta_n) \in \mathbb{R}^{n+1} \mid \sum_{i=0}^{n} \delta_i = 1, \delta_i \geq 0, \forall i \right\} \quad (2)$$

Since we aim at describing visual features in 3D space, we are interested in the *standard 2-simplex* that is defined by three vertices $\Delta^2 = \{\boldsymbol{\delta}^r_\alpha, \boldsymbol{\delta}^r_\beta, \boldsymbol{\delta}^r_\gamma\}$, which can be used to represent feature vectors that take values in the space $\mathbb{R}^3$.

Given a feature's support region that contains a set of 3D visual cues (e.g., gradients), i.e., $S = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\}$, each 3D visual cue $\boldsymbol{v} \in S$ satisfies the following theorem:

Theorem 1. *Any visual cue in a 3D Cartesian space can be transformed into the standard 2-simplex topological vector space.*

Proof. For a given 3D visual cue $\boldsymbol{v} \in \mathbb{R}^3$, its orientation in 3D space can be decomposed into α, β and γ with respect to a given reference Cartesian space defined by the unit vectors $\boldsymbol{v}^r_x$, $\boldsymbol{v}^r_y$ and $\boldsymbol{v}^r_t$ (as shown in Eq. (1)). Assuming $\delta_\alpha = \cos^2\alpha$, $\delta_\beta = \cos^2\beta$, and $\delta_\gamma = \cos^2\gamma$, the vector representing the cue belongs to a standard simplex topological vector space, i.e., $\boldsymbol{\delta} = (\delta_\alpha, \delta_\beta, \delta_\gamma) \in \Delta^2$, because $\delta_\alpha \geq 0$, $\delta_\beta \geq 0$, $\delta_\gamma \geq 0$, and:

$$\begin{aligned} \delta_\alpha + \delta_\beta + \delta_\gamma &= \cos^2\alpha + \cos^2\beta + \cos^2\gamma \\ &= \frac{(\boldsymbol{v}\cdot\boldsymbol{v}^r_x)^2 + (\boldsymbol{v}\cdot\boldsymbol{v}^r_y)^2 + (\boldsymbol{v}\cdot\boldsymbol{v}^r_t)^2}{\|\boldsymbol{v}\|^2} = 1 \end{aligned} \quad (3)$$

Thus, the 3D visual cue encoded by $\boldsymbol{\delta} = (\delta_\alpha, \delta_\beta, \delta_\gamma)$ takes values in the standard 2-simplex vector space. □

The concept of simplex is rather abstract. To address this issue, we developed visualization tools for intuitive analysis of the visual cues' characteristics in the standard 2-simplex topological vector space. In the paper, we also adopt these tools to intuitively explain the idea of our descriptor.

As shown in Figure 4a, the standard 2-simplex topological vector space can be graphically represented as an equilateral triangle on a plane. Using this representation, the element of a transformed visual cue vector $\boldsymbol{\delta} = (\delta_\alpha, \delta_\beta, \delta_\gamma)$ represents the distance ratio of the projected point on the 2-simplex to its respective edge; that is:

$$\boldsymbol{\delta} = (\delta_\alpha, \delta_\beta, \delta_\gamma) = \frac{1}{d_\alpha + d_\beta + d_\gamma}(d_\alpha, d_\beta, d_\gamma) \quad (4)$$

where $d_\alpha + d_\beta + d_\gamma = h$, and h is the height of the standard 2-simplex triangle that is computed by $h = \sqrt{3}b/2$, given the edge length b. For example, given the transformed vector $\boldsymbol{\delta} = (0.5, 0.3, 0.2)$ of the visual cue in Figure 3b, its projected data point on the simplex satisfies that $d_\alpha = 0.5h$, $d_\beta = 0.3h$, and $d_\gamma = 0.2h$, as illustrated in Figure 4a.

3.3. Description in Simplex Space

After projecting the 3D visual cues onto the standard 2-simplex, we discuss how to describe the transformed visual cue vectors in the standard 2-simplex topological space. In particular, we prove that the standard 2-simplex topological vector space can be subdivided into a large number of equally-sized cells, as stated by the following theorem:

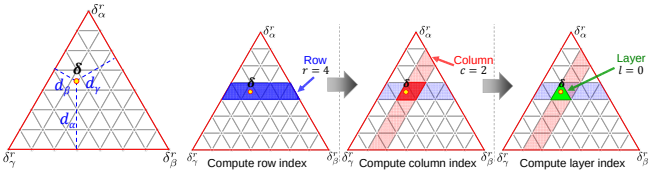
Theorem 2. *For every integer $k \geq 1$, there exists a subdivision of the standard 2-simplex topological vector space into k^2 standard sub-simplices that have the same size.*

Proof. Given a standard 2-simplex Δ^2 with edge length b and height h, we apply edgewise subdivision to divide Δ^2, which equally divides each edge into k segments and connects any pair of endpoints if the line segment represented by the endpoints is parallel to an edge. Then, the total number of sub-simplices is: $1 + 3 + \cdots + (2k-1) = k^2$. Since all sub-simplices have the same edge length b/k and height h/k, they are thus standard and have the same size. □

From Theorem 2 arises the description power of our algorithm, which can scale without bound and therefore avoid the limited discrimination power issue of the regular polyhedron based approach. Theorem 2 also demonstrates that all bins (i.e., sub-simplices) have the same size, which addresses the singularity issue of the spheral coordinate based descriptor. Figure 4a depicts an example that subdivides the standard 2-simplex topological vector space into $k^2 = 49$ equally-sized sub-simplices.

To efficiently identify each individual sub-simplex in the standard 2-simplex topological vector space, we propose a new sub-simplex indexing method using three indices, i.e., *row*, *column*, and *layer*, which are defined as follows:

Definition 2 (Indices of sub-simplices)**.** *Given k edgewise subdivision of the standard 2-simplex $\Delta^2 = \{\boldsymbol{\delta}^r_\alpha, \boldsymbol{\delta}^r_\beta, \boldsymbol{\delta}^r_\gamma\}$, each height is divided into k intervals indexed by $1, \ldots, k$. Then,* row *and* column *are defined as the interval indices of the heights with respect to the edges opposite to $\boldsymbol{\delta}^r_\alpha$ and $\boldsymbol{\delta}^r_\beta$, respectively.* Layer *is a binary value that indicates whether a sub-simplex has a down-pointing triangular shape with respect to an edge.*



(a) Feature projection (b) Computation of subdivision indices

Figure 4: An illustrative example of our topological transformation and sub-simplex index computation in the standard 2-simplex topological vector space, when $k = 7$.

Using the row, column and layer definitions, we are able to efficiently assign each transformed visual cue vector to a sub-simplex in constant time. Given a transformed visual cue $\boldsymbol{\delta} = (\delta_\alpha, \delta_\beta, \delta_\gamma) \in \Delta^2$, our SOD algorithm computes its row r, column c and layer l indices as follows:

$$r(\boldsymbol{\delta}) = \lceil k\delta_\alpha \rceil + \mathbb{1}(\delta_\alpha = 0), \qquad r \in \{1, \ldots, k\} \quad (5)$$

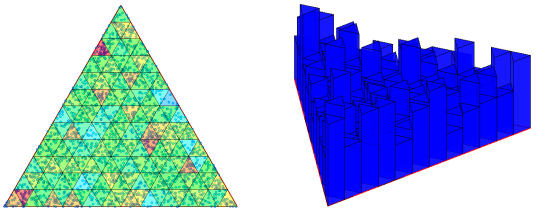
$$c(\boldsymbol{\delta}) = \lceil k\delta_\beta \rceil + \mathbb{1}(\delta_\beta = 0), \qquad c \in \{1, \ldots, k\} \quad (6)$$

$$l(\boldsymbol{\delta}) = (r(\boldsymbol{\delta}) + c(\boldsymbol{\delta}) + \lfloor k\delta_\gamma \rfloor + \mathbb{1}(\delta_\beta \neq 1) + k) \mod 2, \qquad l \in \{0, 1\} \quad (7)$$

where $\mathbb{1}(\cdot)$ is the indicator function that is used to deal with the special cases when $\boldsymbol{\delta}$ is projected onto the edges of the sub-simplices in the standard 2-simplex vector space.

Then, we can directly assign $\boldsymbol{\delta}$ to a sub-simplex indexed by r, c and l. An illustrative example is provided in Figure 4b to explain our index computation method. For the transformed 3D visual cue $\boldsymbol{\delta} = (0.5, 0.3, 0.2)$, after computing its row and column indices, i.e., $r(\boldsymbol{\delta}) = 4$ and $c(\boldsymbol{\delta}) = 2$, a diamond that contains a pair of sub-simplices is located. Then, the layer index is computed, i.e., $l(\boldsymbol{\delta}) = 0$ indicating that the sub-simplex is not upside-down, which determines the final sub-simplex assignment to the 3D visual cue.

After assigning all 3D visual cues in a feature's support region into their respective sub-simplices, each sub-simplex counts the number of cues assigned to it, and a histogram using these sub-simplices as bins is formed to describe the visual feature. An intuitive visualization tool is provided to investigate the histogram in the simplex topological vector space, as depicted in Figure 5. In particular, Figure 5a also visualizes the 3D visual cue's orientation distribution in the transformed simplex vector space.



(a) 2D view (b) 3D view

Figure 5: Visualization of the histogram of the visual cues contained in the support region of a 3D feature when $k = 12$. Figure 5a shows a 2D view with projection distribution of the cues, where a warmer color denotes a larger number of cues falling in the sub-simplex. A more intuitive 3D view is depicted Figure 5b.

3.4. Quadrant Decomposition

When the histogram of 3D visual cues is obtained in the simplex space, quadrant decomposition is performed to further improve the discriminative power of our SOD descriptor. Since the cosine-squared function maps all visual cues to the first quadrant and removes the signs of their orientations, the objective of quadrant decomposition is to describe the orientation signs of visual cues from different quadrants in the reference Cartesian coordinate system. There exist eight quadrants in a 3D Cartesian space that are represented by their signs $(\pm 1, \pm 1, \pm 1)$. Given the orientation of a 3D cue, its quadrant assignment is efficiently computed by:

$$\boldsymbol{q}(\boldsymbol{\delta}) = \left(\frac{\cos\alpha}{|\cos\alpha|}, \frac{\cos\beta}{|\cos\beta|}, \frac{\cos\gamma}{|\cos\gamma|} \right) \quad (8)$$

As a result, the orientation histogram obtained in the simplex vector space is decomposed into eight parts according to different orientation quadrants. It is noteworthy that quadrant assignments are computed with respect to a user-defined coordinate system, which provides additional flexibility to our SOD descriptor. An example of quadrant decomposition is shown in Figure 1.

In order to construct a final vector to describe a 3D visual feature $\boldsymbol{S} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\}$, all decomposed histograms in different quadrants are concatenated into a single vector $\boldsymbol{f}(\boldsymbol{S})$ that is of size $8k^2$, i.e., each of the eight orientation quadrants has a histogram formed by k^2 sub-simplices.

4. Discussion

Efficiency and Runtime Our SOD descriptor employs cosine values to quantize feature orientations in 3D space,

which are efficiently computed using the dot product. For each single 3D visual cue, orientation decomposition (Eq. (1)), topological space transformation (in Theorem 1), sub-simplex index computation (Eq.(5, 6, 7)), and quadrant decomposition (Eq. (8)) take constant time $O(1)$ to perform. Concatenation to form a feature vector takes $O(k^2)$ runtime, where k is the edgewise simplex subdivision parameter. Because typically $N \gg k^2$, i.e., the number of visual cues is much greater than the number of bins in a histogram, our SOD algorithm only takes $O(N)$ time to describe a 3D visual feature that contains N visual cues.

Multi-Channel 3D Features Our SOD algorithm can be directly applied on visual features extracted from multiple channels in 3D space, which include color-depth spatio-temporal features [30] that typically apply descriptors to intensity and depth image sequences in xyt space, and multi-color spatio-temporal features [7] that apply descriptors to multiple color channels of color image sequences. Following [7, 30], one can apply our descriptor over each channel to obtain a vector that describes 3D visual cues in that channel, and combine them together to form a final feature vector. In this scenario, the 3D visualization of our descriptor is a stacked bar plot on the standard 2-simplex.

High Dimensional Features The SOD descriptor is not limited to describing features in 3D space; our methodology can be extended to quantize and describe high dimensional features. Given a d-dimensional visual cue $\boldsymbol{v} \in \mathbb{R}^d$ and a reference coordinate $\mathcal{C} = \{\boldsymbol{v}_1^r, \ldots, \boldsymbol{v}_d^r\}$, its orientation can be decomposed into d angles $(\alpha_1, \ldots, \alpha_d)$, in a manner similar to Eq. (1), which satisfies $\sum_{i=1}^{d} \cos^2 \alpha_i = 1$. Thus, $\boldsymbol{v}$ can be projected onto the standard $(d-1)$-simplex (i.e., an extension of Theorem 1). In addition, [6] showed that a $(d-1)$-simplex can be subdivided into k^{d-1} sub-simplices with the same $(d-1)$-dimensional volume using k edgewise subdivision (i.e., an extension of Theorem 2). Thus, our fundamental theorems still hold, meaning the SOD descriptor can be applied to features in high dimensional space.

5. Empirical Study

Here we detail the experiments conducted to evaluate the performance of our SOD descriptor on action recognition. We would like to highlight that we are not constructing new classifiers and detectors; rather, we intentionally use existing benchmark classifiers and detectors in combination with our novel descriptor to emphasize the performance gain resulting specifically from our SOD descriptor.

5.1. Implementation and Experiment Setup

Detectors Three detectors are adopted to detect spatio-temporal interest points from videos in xyt space. (1) **Harris3D** detector [15] is a spatio-temporal extension of the Harris cornerness criterion that is based on the eigenvalues of a spatio-temporal second-moment matrix. We apply the original implementation [15] and standard parameter setups $\sigma = \sqrt{2^i}$, $i = 2, \ldots, 7$ and $\tau = \{\sqrt{2}, \sqrt{4}\}$. (2) **Gabor** detector [5] applies separable filters on spatial and temporal dimensions to select interest points in xyt space. We adopt the original implementation [5] and standard parameter setups $\sigma = 2$, $\tau = 4$ in our experiments. (3) **Multi-channel Gabor** detector [7] detects spatial-temporal interest points using Gabor detectors to compute image responses based on intensity and normalized chromatic channels. We apply $\sigma = 2$, $\tau = 4$ as in the original work [7].

Descriptors The size of support regions is set to $\Delta_x = \Delta_y = 8\sigma$, and $\Delta_t = 6\tau$, as in [13, 27]. The support region's size and cell layout may be optimized over a specific dataset [13]. To maintain focus on the descriptors themselves, we refrain from such an optimization, following [7, 27]. We use the standard Cartesian space as our reference coordinates. When using multi-channel detectors, the multi-channel description mechanism (discussed in Section 4) is applied.

Two 3D description methodologies based on **spherical coordinates**, such as 3D SIFT [23], and **regular polyhedrons**, such as HOG3D [13] are used as our 3D description baselines (discussed in Section 2.1). Feature descriptors in previous works are also adopted as baselines to compare the feature discrimination's ability to recognize human actions.

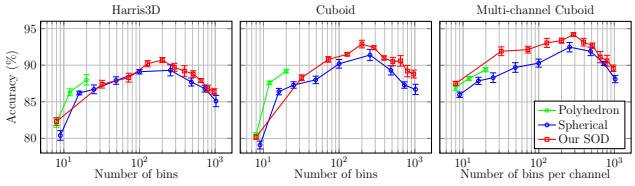
Recognition Following [7, 13, 27], action recognition is performed in a standard bag-of-features learning framework and a codebook is created through clustering 200,000 randomly sampled features using k-means into 4000 codewords. For classification, we use non-linear SVMs with χ^2-kernels and the one-against-all approach [7, 13, 27].

5.2. Datasets

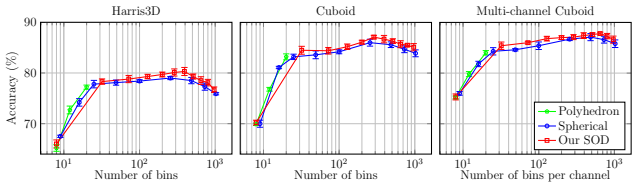
We perform experiments using three action datasets. The **KTH** dataset [22] contains six actions performed by 25 subjects in four scenarios. Following [22], we apply the all-in-one experimental settings and the accuracy metric as the performance measure. The **UCF Sport** dataset [20] contains ten sporting actions in 150 videos that exhibits a large intra-class variability. Following the standard settings [20], performance is evaluated using accuracy in a leave-one-out cross validation framework. The **Hollywood-2** dataset [17] contains 12 complex human actions that are collected from 69 different Hollywood movies. The actions are performed in unconstrained, realistic scenarios, and viewed from different camera angles. Following the standard setup [17], the dataset is divided into 823 training and 884 testing examples; performance is evaluated using the precision measure.

5.3. Descriptor Evaluation

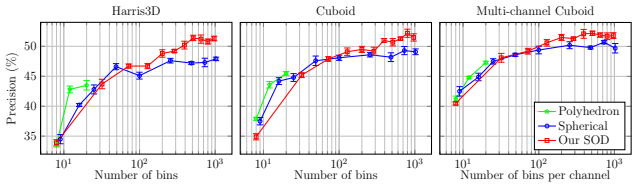
We show our SOD descriptor's superior performance by comparing it with the 3D baseline descriptors. We also investigate our descriptor's sensitivity with respect to the size of the final feature vector, which turns out to be very impor-



(a) KTH



(b) UCF Sport



(c) Hollywood-2

Figure 6: Sensitivity of our SOD descriptor and comparison with baseline 3D descriptors based on spherical coordinates or regular polyhedrons. Error bars are standard deviations.

tant but is rarely studied in previous descriptors. Sensitivity is empirically analyzed using five fold cross-validation over training sets. To focus on investigating characteristics of the descriptors themselves, no additional feature aggregation is applied, i.e., the support region is not divided into cells. Experimental results over three datasets are graphically shown in Figure 6. Because the baseline descriptor based on regular polyhedrons with four and six faces (i.e., bins) performs poorly, we only present the results using polyhedrons with 8, 12 and 20 faces. It is worth recalling that 20 is the maximum number of bins supported by this descriptor as it suffers from the limited discrimination power issue.

For all tested spatio-temporal feature detectors, our SOD descriptor significantly outperforms the 3D baseline descriptors, in general. The discrimination ability provided by the polyhedron baseline is not sufficient to represent complex actions in real-world scenarios. The performance improvement provided by our SOD descriptor over the spherical baseline highlights *the advantages of quantizing and describing spatio-temporal features in the simplex topological space* that can be equally subdivided into any large number of sub-simplices, thus addressing the singularity issue.

In addition, as Figure 6 illustrates, the descriptor's representation ability is greatly affected by the number of bins used to form the final feature vector. All descriptors generally produce poor recognition results when a small number of bins (e.g., less than 15) is used; in this case, the descriptors are not sufficiently discriminative. On the other hand, a very large number of bins (e.g., greater than 1000) also hurts recognition performance. This occurs because although the descriptors discriminate well between visual features, not enough cues fall into each bin. Another important observation is that the ideal number of bins depends on the dataset complexity; a more complex dataset usually requires a larger number of bins. For example, using around 300 bins for the KTH and UCF Sport datasets and around 600 bins for the more complex Hollywood-2 dataset generally leads to satisfactory recognition performance. In summary, our sensitivity analysis results demonstrate *the importance of carefully selecting the number of bins*, by considering both descriptor's discrimination ability and dataset complexity.

5.4. Comparison with the State of the Art

We compare our SOD descriptor with the state-of-the-art feature description methods, in terms of their performance on human action recognition. The compared methods generally follow similar experimental setups that are based on feature pooling, bag-of-features encoding and SVM-based classification. Following [1, 4, 7, 11, 27], we adopt a spatio-temporal pooling scheme that divides each support region into $4\times4\times3$ cells to construct bag-of-features models.

Different descriptors are compared in Tables 1, 2 and 3, which show human action recognition performance over the KTH, UCF Sport and Hollywood-2 datasets, respectively. Our SOD descriptor achieves a 94.8% accuracy on KTH, a 87.5% accuracy on UCF Sport, and a 50.9% overall precision on Hollywood-2. Comparison shows that our SOD descriptor is the best-performing individual descriptor (i.e., *without* combining multiple descriptors, as in [26]), which again shows the effectiveness of our SOD algorithm to describe local spatio-temporal features in xyt space.

Table 1: Comparison of accuracy (%) on the KTH dataset.

2D description methods	Acc.	3D description methods	Acc.
Harris3D + HOG [27]	80.9	Harris3D + 3D SIFT [18]	82.7
Gabor + HOG [12]	82.3	Gabor + Cuboid [5]	89.1
Gabor + HOF [12]	88.2	ST-SIFT + HOG3D [1]	90.7
Gabor + HOF/HOF [12]	88.7	Gabor + HOG3D [13]	91.4
Hessian3D + HOG/HOF [27]	88.7	Harris3D + HOG3D [12]	92.4
Harris3D + HOG/HOF [27]	91.8	FAST + CHOG3D [11]	93.1
Harris3D + HOF [27]	92.1	Multi-ch. Gabor + Poly.	92.9
Oriented energy desc. [4]	93.2	Multi-ch. Gabor + Sphe.	93.8
Context + HOG/HOF [9]	94.1	Multi-ch. Gabor + **SOD**	**94.8**

Table 2: Comparison of accuracy (%) with state-of-the-art descriptors on the UCF Sport datset.

2D description methods	Acc.	3D description methods	Acc.
Harris3D + HOG [27]	71.4	Gabor + Cuboids [12]	76.6
Gabor + HOG [12]	72.7	Harris3D + HOG3D [27]	79.7
Harris3D + HOF [27]	75.4	ST-SIFT + HOG3D [1]	80.5
Gabor + HOF [12]	76.7	Gabor + HOG3D [13]	82.9
Gabor + HOG/HOF [12]	77.7	Multi-ch. G. + HOG3D [7]	85.6
Harris3D + HOG/HOF [27]	78.1	Multi-ch. Gabor + Poly.	85.2
Hessian3D + HOG/HOF [27]	79.3	Multi-ch. Gabor + Sphe.	86.3
Oriented energy desc. [4]	81.5	Multi-ch. Gabor + **SOD**	**87.5**

Table 3: Descriptor comparison on Hollywood-2 using precision (%). '& f' denotes 'HOG/HOF combined with f features'.

Actions	Multi-ch. Cuboid + 3D descriptors			Harris3D +	Harris3D + 2D descriptors					
	Our SOD	Polyhedron	Spherical	HOG3D [13]	HOG [12]	HOF [12]	HOG/HOF [12]	& SIFT [17]	& context [9]	& global [26]
AnswerPhone	18.1	15.9	17.1	16.3	11.8	11.6	15.3	13.1	15.57	**25.9**
DriveCar	**88.1**	85.8	87.2	86.3	79.0	84.8	85.8	81.0	87.0	85.9
Eat	61.6	57.8	60.7	55.8	43.4	58.6	**63.1**	30.6	50.9	56.4
FightPerson	76.2	74.5	75.8	**77.2**	60.4	72.1	71.3	62.5	73.1	74.9
GetOutCar	36.3	33.5	34.3	35.7	24.9	19.6	32.3	8.6	27.2	**44.0**
HandShake	**55.9**	51.3	53.5	55.7	36.3	50.2	49.5	19.1	17.2	29.7
HugPerson	**48.3**	46.5	47.2	47.9	29.6	30.9	38.6	17.0	27.2	46.1
Kiss	**58.4**	54.2	55.3	51.1	43.5	45.1	49.3	57.6	42.9	55.0
Run	**72.1**	67.3	69.7	71.7	62.1	68.5	67.2	55.5	66.9	69.4
SitDown	51.9	48.2	49.3	47.6	30.3	56.4	57.3	30.0	41.6	**58.9**
SitUp	22.4	18.5	20.3	22.2	16.1	8.5	**22.5**	17.8	7.2	18.4
StandUp	21.6	19.6	20.8	15.6	20.9	18.9	20.4	33.5	48.6	**57.4**
Overall	50.9	47.8	49.3	48.6	38.2	43.8	47.7	35.5	42.1	51.8

6. Conclusion

We introduce a novel *simplex-based orientation decomposition* descriptor to quantize and represent 3D visual features including local spatio-temporal features in xyt space. Our technique decomposes each 3D visual cue in a feature's support region into three angles and transforms the decomposed angles into the simplex topological vector space. Feature description is performed in the simplex space, which is able to address the singularity and limited discrimination power issues. Then, quadrant decomposition is performed to improve our SOD descriptor's discrimination capability, and a final feature vector is formed by combining decomposed histograms from all quadrants. Extensive empirical study using three benchmark action datasets has been conducted, which shows that our descriptor significantly outperforms previous 3D feature descriptors based on spherical coordinates or regular polyhedrons and achieves state-of-the-art description power for recognition of human actions.

References

[1] M. Al Ghamdi, L. Zhang, and Y. Gotoh. Spatio-temporal SIFT and its application to human action classification. In *ECCV*, 2012.

[2] K. Chatfield, V. S. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[4] K. G. Derpanis, M. Sizintsev, K. J. Cannons, and R. P. Wildes. Action spotting and recognition based on a spatiotemporal orientation analysis. *PAMI*, 35(3):527–540, Mar. 2013.

[5] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VSPETS*, 2005.

[6] H. Edelsbrunner and D. R. Grayson. Edgewise subdivision of a simplex. In *SoCG*, 1999.

[7] I. Everts, J. C. van Gemert, and T. Gevers. Evaluation of color STIPs for human action recognition. In *CVPR*, 2013.

[8] G. Flitton, T. P. Breckon, and N. Megherbi. A comparison of 3D interest point descriptors with application to airport baggage object detection in complex CT imagery. *PR*, 46(9):2420–2436, Sept. 2013.

[9] D. Han, L. Bo, and C. Sminchisescu. Selection and context for action recognition. In *ICCV*, 2009.

[10] M. Holte, T. Moeslund, and P. Fihl. View-invariant gesture recognition using 3D optical flow and harmonic motion context. *CVIU*, 114(12):1353–1361, 2010.

[11] Y. Ji, A. Shimada, H. Nagahara, and R. ichiro Taniguchi. A compact descriptor CHOG3D and its application in human action recognition. *IEEJ Trans. Electr. and Electron. Eng.*, 8(1):69–77, 2013.

[12] A. Kläser. *Learning human actions in video*. PhD thesis, Université de Grenoble, 2010.

[13] A. Kläser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In *BMVC*, 2008.

[14] Y. Kuang, M. Byrod, and K. Astrom. Supervised feature quantization with entropy optimization. In *ICCVW*, 2011.

[15] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, Nov. 2004.

[17] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009.

[18] R. Mattivi and L. Shao. Robust spatio-temporal features for human action recognition. In *MAPC*, 2011.

[19] J. Munkres. *Elements of Algebraic Topology*. Advanced book classics. Perseus Books, 1984.

[20] M. D. Rodriguez, J. Ahmed, and M. Shah. Action MACH a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008.

[21] W. Rudin. *Principles of mathematical analysis*. McGraw-Hill, 1964.

[22] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *CVPR*, 2004.

[23] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional SIFT descriptor and its application to action recognition. In *ICME*, 2007.

[24] S. Tang, X. Wang, X. Lv, T. X. Han, J. Keller, Z. He, M. Skubic, and S. Lao. Histogram of oriented normal vectors for object recognition with a depth sensor. In *ACCV*, 2012.

[25] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *CVPR*, 2013.

[26] M. M. Ullah, S. N. Parizi, and I. Laptev. Improving bag-of-features action recognition with non-local cues. In *BMVC*, 2010.

[27] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[28] L. Xia and J. K. Aggarwal. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *CPVR*, 2013.

[29] L. Xia, C.-C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3D joints. In *CVPRW*, 2012.

[30] H. Zhang and L. Parker. 4-dimensional local spatio-temporal features for human activity recognition. In *IROS*, 2011.