A Spatio-Temporal Descriptor Based on 3D-Gradients

Alexander Klaser, Marcin Marszalek, Cordelia Schmid

A Spatio-Temporal Descriptor Based on 3D-Gradients

Alexander Kläser Marcin Marszałek Cordelia Schmid
INRIA Grenoble, LEAR, LJK
{alexander.klaser,marcin.marszalek,cordelia.schmid}@inrialpes.fr

Abstract

In this work, we present a novel local descriptor for video sequences. The proposed descriptor is based on histograms of oriented 3D spatio-temporal gradients. Our contribution is four-fold. (i) To compute 3D gradients for arbitrary scales, we develop a memory-efficient algorithm based on integral videos. (ii) We propose a generic 3D orientation quantization which is based on regular polyhedrons. (iii) We perform an in-depth evaluation of all descriptor parameters and optimize them for action recognition. (iv) We apply our descriptor to various action datasets (KTH, Weizmann, Hollywood) and show that we outperform the state-of-the-art.

1 Introduction

Action recognition in videos has been a very active field of research over the past years. It can be considered one of the key prerequisites for video analysis and understanding. Typical applications include video surveillance, video indexing/browsing, recognition of gestures, human-computer interfacing, or analysis of sport-events.

Based on the recent developments of visual recognition in static images, many concepts have been successfully extended to video sequences. Similar to object recognition in images, bag-of-features based methods have recently shown excellent results for action recognition [7, 10, 14]. Despite recent developments, the representation of local regions in videos is still an open field of research.

In this paper, we propose a novel spatio-temporal descriptor. Building on the success of HoG-based descriptors for static images [3, 13], we view videos as spatio-temporal volumes and generalize the key HoG concepts to 3D. We evaluate all the parameters of the new descriptor in the context of action recognition.

The following subsection 1.1 presents related work. In section 2 we introduce our descriptor. The experimental setup and results are discussed in section 3. We conclude the paper in section 4.

1.1 Related work

Local descriptors based on normalized pixel values, brightness gradients, and windowed optical flow were evaluated for action recognition by Dollár et al. [4]. Experiments on three datasets—KTH human actions, facial expressions and mouse behavior—show best results for gradient descriptors. Those descriptors, however, were computed by concatenating all gradient vectors in a region or by building histograms on gradient components. Primarily based on gradient magnitudes, they suffer from sensitivity to illumination

changes. We base our descriptor on gradient orientations, as those are robust to changes in illumination [6]. This also goes along with the works of Lowe [13] and Dalal et al. [3].

Laptev and Lindeberg [9] investigated single- and multi-scale N-jets, histograms of optical flow, and histograms of gradients as local descriptors for video sequences. Best performance has been obtained with optical flow and spatio-temporal gradients. Instead of a direct quantization of the gradient orientations, however, each component of the gradient vector was quantized separately. In later work, Laptev et al. [10, 11] applied a coarse quantization to gradient orientations. However, as only spatial gradients have been used, histogram features based on optical flow were employed in order to capture the temporal information. The computation of optical flow is rather expensive and results depend on the choice of regularization method [1]. Therefore, we base our descriptor on pure spatio-temporal 3D gradients which are robust and cheap to compute. We perform orientation quantization with up to 20 bins by using regular polyhedrons. Furthermore, we propose integral histograms for memory-efficient computation of features at arbitrary spatial and temporal scales.

An extension of the SIFT descriptor [13] to 3D was proposed by Scovanner et al. [17]. For a given cuboid, spatio-temporal gradients are computed for each pixel. All pixels vote into a $N_x \times N_y \times N_t$ grid of histograms of oriented gradients. For orientation quantization, gradients are represented in polar coordinates ϕ, ψ that are divided into a 8×4 histogram by meridians and parallels. This leads to problems due to singularities at the poles since bins get progressively smaller. We avoid the problem of singularities by employing regular polyhedrons and use their homogeneously distributed faces as histogram bins. Efficient histogram computation is then done by projecting gradient vectors onto the axes running through the center of the polyhedron and the face centers.

2 Spatio-temporal descriptor

Local descriptors are used to describe a local fragment in an image or a video. Usually, local regions are determined first by using an interest point detector or by dense sampling

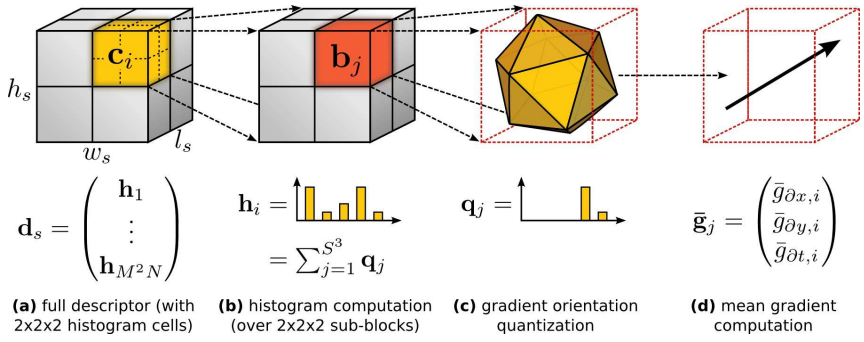

Figure 1: Overview of the descriptor computation; (a) the support region around a point of interest is divided into a grid of gradient orientation histograms; (b) each histogram is computed over a grid of mean gradients; (c) each gradient orientation is quantized using regular polyhedrons; (d) each mean gradient is computed using integral videos.

of the image plane or video volume. The coordinates of sampled points are in general accompanied by characteristic spatial and temporal scales, which determine the extent of the neighborhood. Given an interest region, a descriptor represents the region by a feature vector. The visual content of the whole image or the whole video can be represented as a set of features computed at different scales and positions. In order to successfully perform recognition tasks in such a framework, a descriptor needs to be highly distinctive. At the same time, however, it has to be invariant to changes in illumination, small deformations, etc. In this work, we focus on spatio-temporal descriptors for action classification.

Figure 1 illustrates the different steps for computing our 3D gradient orientation descriptor. Each step is discussed in detail in the following. Subsection 2.1 explains the proposed efficient computation of 3D gradients with arbitrary spatial and temporal scales (fig. 1d). The orientation quantization of 3D gradients is presented in subsection 2.2 (fig. 1c). Subsection 2.3 summarizes the computation of orientation histograms (fig. 1b), and finally the construction of the descriptor itself is explained in subsection 2.4 (fig. 1a).

2.1 Gradient computation

In order to compute a histogram of 3D gradient orientations, gradient vectors need to be computed efficiently for many different regions (cf. fig. 1d). As it is necessary to account for different spatial and temporal scales, these regions will vary not only in their positions, but also in their extents. One strategy to improve computational efficiency is to use spatio-temporal "pyramids", i.e., to precompute the gradients on different temporal and spatial scales [10, 11]. However, for each spatio-temporal scale, the video sequence needs to be rescaled and stored.

Precisely, given N scale steps in total as well as a spatial and a temporal scaling factor σ_{xy}, σ_t, this amounts in a factor $z = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \sigma_{xy}^{-2i} \sigma_t^{-j}$ of additional data that needs to be stored as well as processed. For instance, if we assume a fine spatial and temporal scale grid with $\sigma_{xy} = \sigma_t = \sqrt[4]{2}$ over six octaves in total, i.e., $N = 24$, one will need to compute 24×24 different video scales. This results in a factor $z \approx 21$ of extra data.

As memory-efficient yet still flexible alternative, we propose to use integral videos for computing mean gradient vectors. Effectively, this can be seen as smoothing operation by convolution with a box filter instead of (typically) a Gaussian. With integral videos, mean 3D gradients can be efficiently computed for any arbitrary x-, y-, and t-scale in a video.

The concept of integral images has been popularized by Viola and Jones [18]. They used integral images as an intermediate representation for efficient computation of Haar features. We extend integral images to integral videos on gradient vectors. Given a video sequence $v(x,y,t)$, its partial derivatives along x,y,t are denoted by $v_{\partial x}, v_{\partial y}, v_{\partial t}$, respectively. The integral video for $v_{\partial x}$ can be described as

$$iv_{\partial x}(x,y,t) = \sum_{x' \leq x, y' \leq y, t' \leq t} v_{\partial x}(x',y',t') \tag{1}$$

and for $iv_{\partial y}$ and $iv_{\partial t}$ accordingly. For any 3D cuboid $\mathbf{b} = (x,y,t,w,h,l)^\mathsf{T}$ described by its position $(x,y,t)^\mathsf{T}$ and its width (w), height (h), and length (l), we can compute its mean gradient $\bar{\mathbf{g}}_\mathbf{b} = (\bar{g}_{\mathbf{b}\partial x}, \bar{g}_{\mathbf{b}\partial y}, \bar{g}_{\mathbf{b}\partial t})^\mathsf{T}$. For $\bar{g}_{\mathbf{b}\partial x}$ ($\bar{g}_{\mathbf{b}\partial y}$ and $\bar{g}_{\mathbf{b}\partial t}$ respectively):

$$\begin{aligned}\bar{g}_{\mathbf{b}\partial x} = &[iv_{\partial x}(x+w,y+h,t+l) - iv_{\partial x}(x,y+h,t+l) - iv_{\partial x}(x+w,y,t+l) + iv_{\partial x}(x,y,t+l)] \\ &- [iv_{\partial x}(x+w,y+h,t) - iv_{\partial x}(x,y+h,t) - iv_{\partial x}(x+w,y,t) + iv_{\partial x}(x,y,t)] \quad . \end{aligned} \tag{2}$$

2.2 Orientation quantization

A n-bin histogram of gradient orientations in 2D (i.e., for static images) can in fact be seen as approximation of a circle (i.e., the continuous space of orientations) with a regular n-sided polygon. Each side of the polygon corresponds then to a histogram bin. In 3D, a polygon becomes a polyhedron. Regular polyhedrons with congruent faces are referred to as platonic solids. There are only five of them: the tetrahedron (4-sided), cube (6-sided), octahedron (8-sided), dodecahedron (12-sided), and icosahedron (20-sided). As the octagon (8-sided polygon) is commonly used to quantize 2D gradients, we consider the dodecahedron and the icosahedron for 3D gradient quantization (cf. fig. 1c).

Given a regular n-sided polyhedron, let its center of gravity lie at the origin of a three-dimensional Euclidean coordinate system. In order to quantize a 3D gradient vector $\bar{\mathbf{g}}_\mathbf{b}$ w.r.t. its orientation, we first project $\bar{\mathbf{g}}_\mathbf{b}$ on the axes running through the origin of the coordinate system and the center positions of all faces. This can be done through a matrix multiplication. Let P be the matrix of the center positions $\mathbf{p}_1, \ldots, \mathbf{p}_n$ of all n faces

$$\mathsf{P} = (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n)^\mathsf{T} \qquad \text{with} \qquad \mathbf{p}_i = (x_i, y_i, t_i)^\mathsf{T} \ . \tag{3}$$

For instance, the icosahedron can be described with the following 20 center points:

$$(\pm 1, \pm 1, \pm 1) \quad (0, \pm 1/\varphi, \pm\varphi) \quad (\pm 1/\varphi, \pm\varphi, 0) \quad (\pm\varphi, 0, \pm 1/\varphi) \tag{4}$$

where $\varphi = \frac{1+\sqrt{5}}{2}$ is called the golden ratio. The projection $\hat{\mathbf{q}}_\mathbf{b}$ of $\bar{\mathbf{g}}_\mathbf{b}$ is obtained through:

$$\hat{\mathbf{q}}_\mathbf{b} = (\hat{q}_{\mathbf{b}1}, \ldots, \hat{q}_{\mathbf{b}n})^\mathsf{T} = \frac{\mathsf{P} \cdot \bar{\mathbf{g}}_\mathbf{b}}{\|\bar{\mathbf{g}}_\mathbf{b}\|_2} \ . \tag{5}$$

Thus, each $\hat{q}_{\mathbf{b}i}$ of $\hat{\mathbf{q}}_\mathbf{b}$ holds the normalized projection of the gradient vector $\bar{\mathbf{g}}_\mathbf{b}$ onto the axes through the face center $\mathbf{p}_i$, i.e., $\hat{q}_{\mathbf{b}i} = \|\mathbf{p}_i\|_2 \cdot \cos_\angle(\mathbf{p}_i, \bar{\mathbf{g}}_\mathbf{b}) = \|\bar{\mathbf{g}}_\mathbf{b}\|_2^{-1} \cdot \mathbf{p}_i{}^\mathsf{T} \cdot \bar{\mathbf{g}}_\mathbf{b}$. We can optionally put opposite directions in one bin by halving the set of face centers and taking the absolute value of $\hat{q}_{\mathbf{b}i}$. We then obtain a histogram with half of the orientations.

Next, the resulting vector $\hat{\mathbf{q}}_\mathbf{b}$ of the projection is thresholded. This is done, since $\bar{\mathbf{g}}_\mathbf{b}$ should vote into only one single bin in case it is perfectly aligned with the corresponding axis running through the origin and the face center. By comparing two neighboring axes $\mathbf{p}_i$ and $\mathbf{p}_j$, this threshold value is given by $t = \mathbf{p}_i{}^\mathsf{T} \cdot \mathbf{p}_j$. For the icosahedron given in (4) $t \approx 1.29107$. Threshold t is subtracted from $\hat{\mathbf{q}}_\mathbf{b}$ and all negative elements are set to zero. The gradient magnitude is distributed according to the thresholded histogram $\hat{\mathbf{q}}'_\mathbf{b}$:

$$\mathbf{q}_\mathbf{b} = \frac{||\bar{\mathbf{g}}_\mathbf{b}||_2 \cdot \hat{\mathbf{q}}'_\mathbf{b}}{||\hat{\mathbf{q}}'_\mathbf{b}||_2} \ . \tag{6}$$

In our experiments (see subsection 3.3 for details) we have found the icosahedron ($n = 20$) to be an appropriate regular polyhedron for quantization.

2.3 Histogram computation

A histogram of gradient orientations needs to be computed over a set of gradient vectors (see figure 1b). This is done as follows. Given a cuboid $\mathbf{c} = (x_\mathbf{c}, y_\mathbf{c}, t_\mathbf{c}, w_\mathbf{c}, h_\mathbf{c}, l_\mathbf{c})^\mathsf{T}$, we divide $\mathbf{c}$ into $S \times S \times S$ subblocks $\mathbf{b}_i$. These S^3 subblocks form the set over which the

histogram is computed. For each of the subblocks $\mathbf{b}_i$ the corresponding mean gradient $\bar{\mathbf{g}}_{\mathbf{b}_i}$ is computed using integral videos as defined in equation (2). $\bar{\mathbf{g}}_{\mathbf{b}_i}$ is subsequently quantized as $\mathbf{q}_{\mathbf{b}_i}$ employing a regular polyhedron (see equation (6)). The histogram $\mathbf{h}_\mathbf{c}$ for the region $\mathbf{c}$ is then obtained by summing the quantized mean gradients $\mathbf{q}_{\mathbf{b}_i}$ of all subblocks $\mathbf{b}_i$:

$$\mathbf{h}_\mathbf{c} = \sum_{i=1}^{S^3} \mathbf{q}_{\mathbf{b}_i} \quad . \tag{7}$$

With a fixed number of supporting mean gradient vectors (S^3), and by using integral videos for computing mean gradients of subblocks, a histogram can be computed for any arbitrary scale along x, y, t. At the same time the memory requirements for storage are linear in the number of pixels in the video sequence. They do not depend on the number of predefined spatio-temporal scales as in Laptev et al. [10, 11]. Our experiments show (see subsection 3.3 for details) that $S = 3$, resulting in 27 supporting mean gradient vectors, is a good choice in terms of computational efficiency and recognition accuracy.

2.4 Descriptor computation

A sampling point $\mathbf{s} = (x_\mathbf{s}, y_\mathbf{s}, t_\mathbf{s}, \sigma_\mathbf{s}, \tau_\mathbf{s})^\mathsf{T}$ is located in the video sequence at $(x_\mathbf{s}, y_\mathbf{s}, t_\mathbf{s})^\mathsf{T}$. Its characteristic spatial and temporal scale are given by $\sigma_\mathbf{s}$ and $\tau_\mathbf{s}$, respectively. The final descriptor $\mathbf{d}_\mathbf{s}$ for $\mathbf{s}$ is computed for a local support region $\mathbf{r}_\mathbf{s} = (x_\mathbf{r}, y_\mathbf{r}, t_\mathbf{r}, w_\mathbf{r}, h_\mathbf{r}, l_\mathbf{r})^\mathsf{T}$ around the position $\mathbf{s}$ (see figure 1a) with width ($w_\mathbf{s}$), height ($h_\mathbf{s}$), and length ($l_\mathbf{s}$) given by

$$w_\mathbf{s} = h_\mathbf{s} = \sigma_0 \sigma_\mathbf{s} \quad , \qquad l_\mathbf{s} = \tau_0 \tau_\mathbf{s} \quad . \tag{8}$$

The parameters σ_0 and τ_0 characterize the relative size of the support region around $\mathbf{s}$.

Similar to other approaches (e.g., [4, 8, 9, 10, 17]), the local support region $\mathbf{r}_\mathbf{s}$ is divided into a set of $M \times M \times N$ cells $\mathbf{c}_i$. For each cell, an orientation histogram is computed (see equation (7)). All histograms are finally concatenated to one feature vector $\mathbf{d}_\mathbf{s} = (d_1, \ldots, d_{M^2N})^\mathsf{T}$. Accordingly to Lowe [13], we normalize $\mathbf{d}_\mathbf{s}$ with its $\mathscr{L}_2$-norm, cut values to a predefined value c, and renormalize the vector. Our experiments show that $c = 0.25$ is a reasonable cut value, which is consistent with the findings of Lowe for SIFT. Furthermore, scale parameters $\sigma_0 = 8$ and $\tau_0 = 6$ give good results. Plausible values for the number of cells are $M = 4$ and $N = 4$.

3 Experiments

In order to evaluate the performance of our descriptor, the experimental setup in this work closely follows the setup of Laptev et al. [10]. We summarize the setup in the following. A video sequence is represented as a *bag-of-words* (BoWs) using sparse space-time features. A sparse set of spatio-temporal interest points is obtained by a space-time extension of the Harris operator [8]. Similar to [10], we sample features at multiple spatial and temporal scales. Interest point detections due to artifacts at shot boundaries are removed.

The bag-of-words representation requires creating a visual vocabulary. Here, we randomly sample V training features. This is very fast and in our experiments the results were very close to those obtained using vocabularies built with k-means. To compensate for the randomness, we repeat the random sampling three times and report values for average and

standard deviation. For the representation, all features of a video sequence are assigned to their closest (using Euclidean distance) vocabulary word. This produces histograms of visual word occurrences which are then used for classification.

For classification, we use non-linear support vector machines with χ^2-kernel as in [10]

$$K(H_i, H_j) = exp(-\frac{1}{2A}\sum_{n=1}^{V}\frac{(h_{in}-h_{jn})^2}{h_{in}+h_{jn}}) \ , \tag{9}$$

where $H_i = \{h_{in}\}$ and $H_j = \{h_{jn}\}$ are the histograms of word occurrences and V is the vocabulary size. A is the mean value of distances between all training samples [21]. For multi-class classification, we use the *one-against-rest* approach.

3.1 Parameter learning

In order to obtain an appropriate set of parameters, we perform a simple optimization on the training data of the KTH action set. The value being optimized is the average classification accuracy. We perform leave-one-out cross-validation on a per person basis using only the training/validation set. The following parameters are optimized: code book size (V), spatial and temporal support (σ_0, τ_0), number of histogram cells (M, N), number of supporting mean gradients (S), cut-off value (c), orientation type (full or half orientation), polyhedron type (icosahedron or dodecahedron). For the optimization, we divide the parameter space into a rough grid and start the optimization at a meaningful manually chosen point. The optimization is a gradient ascent method that evaluates for each parameter the two neighboring values of the current optimum on the grid. To account for a sometimes significantly large variance, we perform three runs separately. By caching results of previous runs, the approximation of the true mean becomes more precise with each iteration. For a new iteration, the point with the highest mean accuracy is chosen as the new optimum. The optimization is stopped on convergence, when the maximum remains stable for three consecutive runs.

3.2 Datasets

We compare our work to the state-of-the-art on three different action datasets. The **KTH actions** dataset [16] contains six types of different human action classes: walking, jogging, running, boxing, waving, and clapping (cf. fig. 2, top). Each action class is performed several times by 25 subjects. The sequences were recorded in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes, and indoors. The background is homogeneous and static in most sequences. In total, the data consists of 2391 video samples. We follow the experimental setup of Schüldt et al. [16] and divide the samples into training/validation set (8+8 people) and test set (9 people).

The **Weizmann actions** dataset [2] consists of ten different types of actions classes: bending downwards, running, walking, skipping, jumping-jack, jumping forward, jumping in place, galloping sideways, waving with two hands, and waving with one hand (cf. fig. 2, middle). Each action class is performed once (sometimes twice) by 9 subjects resulting in 93 video sequences in total. As for the KTH dataset, the background is homogeneous and static. Testing is performed by leave-one-out on a per person basis, i.e., for each fold, training is done on 8 subjects and testing on all video sequences of the remaining subject (as suggested by Scovanner et al. [17]).

Figure 2: Sample frames from video sequences of KTH (top), Weizmann (middle), and Hollywood (bottom) human action datasets.

The **Hollywood actions** dataset [10] contains eight different action classes: answering the phone, getting out of the car, hand shaking, hugging, kissing, sitting down, sitting up, and standing up (see fig. 2, bottom). These actions have been collected semi-automatically from 32 different Hollywood movies. The full dataset contains 663 video samples. It is divided into a clean training set (219 sequences) and a clean test set (211 sequences), with training and test sequences obtained from different movies. In this work, we do not consider the additional noisy dataset (233 sequences).

3.3 Evaluation of descriptor parameters

We optimize the parameters on the KTH actions training/validation set as discussed in subsection 3.1. The optimization converges to the following set of parameters: code book size $V = 4000$; spatial and temporal support $\sigma_0 = 8, \tau_0 = 6$; number of histogram cells $M = 4, N = 3$; number of supporting mean gradients $S = 3$; cut-off value $c = 0.25$; and as polyhedron type icosahedron with full orientation. This results in a descriptor of dimensionality $4 \cdot 4 \cdot 3 \cdot 20 = 960$. Figure 3 shows the influence of the parameters on the average accuracy. The error bars indicate the standard deviation of multiple runs which vary between 3 and 9. In the figure, we plot the accuracy obtained through cross-validation on the training/validation set (denoted as *train*) as well as the accuracy obtained on the test set (denoted as *test*). We observe that the parameter most sensitive to changes is the code book size. The least sensitive parameter is the cut-off value. Taking into account varying results obtained on the training and test sets, it can be stated that reasonable values for the number of histogram cells M, N lie between 2 and 4. The scale parameters (σ_0, τ_0) seem to favor a smaller temporal (6-8 pixels) than spatial (8-10 pixels) support. This is presumably in order to make the descriptor capture smaller motion changes. A reasonable

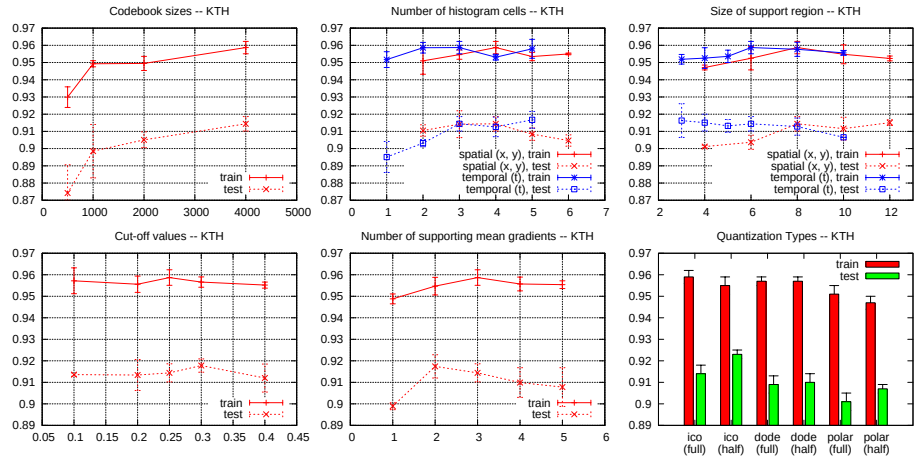

Figure 3: Parameter evaluation for neighboring parameters around the learned optimum value on the KTH actions training/validation set; the average accuracy (with the standard deviation denoted by error bars) is plotted against different parameter settings.

number (S) of supporting mean gradients per dimension seems to be 2 or 3. For the quantization, best results are obtained using the icosahedron with full orientation. Note that we also compare to orientation quantization using polar coordinates, implemented in the spirit of [17]. We quantize the polar coordinates into 6×3 bins, which results in a descriptor dimensionality comparable to the one obtained with the icosahedron. Still, sphere quantization using any of the polyhedrons shows consistently better results.

In the following experiments we use these parameters learned on the KTH dataset. They turned out to be universal enough to obtain state-of-the-art results on other datasets. Note, however, that for optimal performance, parameters might need adaptation to the particular problem. We found that for the Weizmann dataset a larger support area ($\sigma_0 = 12, \tau_0 = 4$) increases the average accuracy from 84.3% to 85.7%. Furthermore, decreasing the number of spatial grids to $M = 3$ and the codebook size to $V = 1500$ (to compensate for slightly reduced dimensionality), we obtain the performance of 90.5%. For the Hollywood dataset, we obtain significant improvement by setting the number of spatial grids to $M = 1$ and adjusting the codebook size to $V = 1000$ (to compensate for heavily reduced dimensionality). Average precision over all classes improves then from 24.7% to 27.3% which is higher than the results for HoGs (27.0%) and HoFs (21.5%). The need to reduce the spatial grids can be well explained with slightly higher variability of the Weizmann dataset and much higher variability of the Hollywood dataset. Note that the Hollywood dataset features much larger intra-class variations. Complex actions are performed by different actors and are captured from different perspectives. Moreover, significant amount of background motion, clutter, occlusions, and shot boundaries are present.

	Local Jets [16]	Gradients+ PCA [19]	HoG [10]	HoF [10]	ours
Accuracy	71.7%	86.7%	81.6%	89.7%	**91.4%** ($\pm$0.4)

Table 1: Average class accuracy on the KTH dataset.

	Shape Context+ Gradients+PCA [15]	3D SIFT [17]	Spin Images [12]	ST Features [12]	ours
Accuracy	72.8%	82.6%	74.2%	68.4%	**84.3%** ($\pm$2.9)

Table 2: Average class accuracy on the Weizmann dataset.

3.4 Comparison to state-of-the-art

We apply the learned set of parameters to the **KTH dataset**. Compared to state-of-the-art descriptors evaluated in a standard bag-of-words (BoW) framework (cf. table 1), our descriptor significantly gains in performance. Results for our single descriptor are even comparable to the best KTH accuracy known to us (91.8%) obtained with a combination of different descriptor and grid types by Laptev et al. [10]. Note that we cannot compare to Jhuang et al. [7] or Wong et al. [20] since their results are based on a non-standard setup. They have used either more training data or split the problem into simpler tasks.

With the set of parameters obtained from the KTH dataset, we also outperform previous BoW-based works on the **Weizmann dataset** (cf. table 2). Liu et al. [12] obtain the best accuracy known to us (90.4%) by combining and weighting multiple feature types. Still, by adapting the parameters to the data set, we match this performance with our single descriptor, see subsection 3.3. Note that we cannot compare to results given by Jhuang et al. [7] or Fathi and Mori [5]. They performed a different evaluation task or included more data given by segmentation masks.

For the **Hollywood dataset**, our descriptor learned on the KTH dataset could not outperform the results reported by Laptev et al. [10] for their 2D HoG descriptor (cf. table 3). However, our descriptor still produces better results than HoF and outperforms the 2D HoG for three out of eight classes. By adjusting a few parameters for the dataset, we obtain a higher accuracy than each of the single descriptors, see subsection 3.3.

4 Summary

In this paper, we have proposed a novel video descriptor based on histograms of 3D gradient orientations[1]. We have extended integral images to integral videos for efficient 3D gradient computation and presented a quantization method for 3D orientations. On this basis, we have constructed a HoG-like 3D descriptor, evaluated all its parameters, and optimized them for action recognition in videos. The performance of the proposed descriptor outperforms the state-of-the-art on two out of three datasets and matches it on the third. In the future, we plan to focus on learning the descriptor parameters on a per-class basis which should provide an additional performance improvement.

	HoG [10]	HoF [10]	ours		HoG [10]	HoF [10]	ours
AnswerPhone	13.4%	**24.6%**	18.6% ($\pm$1.9)	SitDown	29.1%	20.7%	**32.5%** ($\pm$7.2)
HugPerson	**29.1%**	17.4%	19.8% ($\pm$1.1)	SitUp	6.5%	5.7%	**7.0%** ($\pm$0.6)
GetOutCar	21.9%	14.9%	**22.6%** ($\pm$2.1)	Kiss	**52.0%**	36.5%	47.0% ($\pm$0.7)
HandShake	**18.6%**	12.1%	11.8% ($\pm$1.3)	StandUp	**45.4%**	40.0%	38.0% ($\pm$1.3)
Average					27.0%	21.5%	24.7%

Table 3: Average precision on the Hollywood dataset.

[1] The software for our descriptor can be downloaded at: `http://lear.inrialpes.fr/software`.

Acknowledgments. A. Kläser was supported by the European research project CLASS and M. Marszałek by the European Marie-Curie project VISITOR. Experiments were carried out on Grid'5000 funded by the ACI GRID project, INRIA, CNRS, RENATER and other partners.

References

[1] S. Baker, S. Roth, D. Scharstein, M. Black, J. Lewis, and R. Szeliski. A database and evaluation methodology for optical flow. In *ICCV*, 2007.

[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005.

[3] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006.

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

[5] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *CVPR*, 2008.

[6] W. Freeman and M. Roth. Orientation histogram for hand gesture recognition. In *FG*, 1995.

[7] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, 2007.

[8] I. Laptev. On space-time interest points. *IJCV*, 64:107–123, 2005.

[9] I. Laptev and T. Lindeberg. Local descriptors for spatio-temporal recognition. In *SCVMA*, 2004.

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

[11] I. Laptev and P. Perez. Retrieving actions in movies. In *ICCV*, 2007.

[12] J. Liu, S. Ali, and M. Shah. Recognizing human actions using multiple features. In *CVPR*, 2008.

[13] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004.

[14] J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *BMVC*, 2006.

[15] J. C. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *CVPR*, 2007.

[16] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *ICPR*, 2004.

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

[18] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.

[19] S. Wong and R. Cipolla. Extracting spatiotemporal interest points using global information. In *ICCV*, 2007.

[20] S.-F. Wong, T.-K. Kim, and R. Cipolla. Learning motion categories using both semantic and structural information. In *CVPR*, 2007.

[21] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *IJCV*, 73:213–238, 2007.