Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Ciência da Computação

# A Video Descriptor Based on Multivariate Gaussians for Human Action Recognition

**Ana Paula Schiavon**

JUIZ DE FORA
NOVEMBRO, 2017

# A Video Descriptor Based on Multivariate Gaussians for Human Action Recognition

Ana Paula Schiavon

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Orientador: Marcelo Bernardes Vieira

JUIZ DE FORA
NOVEMBRO, 2017

# A Video Descriptor Based on Multivariate Gaussians for Human Action Recognition

Ana Paula Schiavon

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Marcelo Bernardes Vieira
Doutor em Ciência da Computação

Saulo Moraes Villela
Doutor em Engenharia de Sistemas e Computação

Rodrigo Luis de Souza da Silva
Doutor em Engenhara Civil

JUIZ DE FORA
30 DE NOVEMBRO, 2017

*To my family.*

# Resumo

O reconhecimento da ação humana é um campo de pesquisa em Visão Computacional que estuda maneiras de descrever ações humanas usando conteúdo visual de vídeos. Esta área possui várias aplicações como em sistemas de vigilância, indexação de vídeo e interface humano-computador. Neste trabalho é apresentado um descritor global de movimento baseado em distribuições Gaussianas multivariadas estimadas a partir de histogramas de gradientes orientados tridimensionais extraídos dos vídeos. Como o espaço de distribuições Gaussianas multivariadas não é linear, estratégias baseadas em Álgebra de Lie foram utilizadas para incorporar o espaço de Gaussianas em tal espaço visando utilizar operações Euclidianas. Por fim, avaliamos nosso descritor para as bases de dados KTH, MuHAVi e SKIG utilizando um classificador de máquinas de vetores suporte.

**Palavras-chave: Reconhecimento de ação humana, Álgebra de Lie, Distribuição Gaussiana multivariada, Histograma de gradientes orientados**

# Abstract

Human action recognition is a research field in Computer Vision which studies ways to describe human actions using visual video content. This area has several applications, such as surveillance systems, video indexing and human-computer interfaces. In this work we present a global motion descriptor based on multivariate Gaussian distributions estimated from three-dimensional histograms of oriented gradients extracted from videos. As the multivariate Gaussian distributions space is not linear, strategies based on Lie Algebra were used to embed the Gaussian space into a linear space aiming to use Euclidean operations. Finally, we evaluated our descriptor for the KTH, MuHAVi and SKIG datasets using a support vector machine classifier.

**Keywords: Human action recognition, Lie algebra, Multivariate Gaussian distribution, Histogram of oriented gradients**

# Agradecimentos

À Deus, pela vida. À minha mãe Luzia, pelo amor e apoio sempre presentes. Aos meus irmãos de sangue, Eliana, Valdivino, Juliana, Adriana, Miguel, Jussara, e aos de alma, Sabrini, Ana Julia, Paula, Mario e Welinton, agradeço por todo apoio e pelos momentos de diversão que só existem porque vocês existem. Aos amores da tia, Henrique, Miguel e Giovana, por serem as coisinhas mais gostosas desse mundo.

Agradeço também aos demais familiares pela confiança e incentivo. Aos amigos que a universidade proporcionou, Ana Paula, Gisele, Ludmila, Luiza, Míria e Rebeca, por todos os encontrinhos, comidas e fofocas. Ao Fernando, pela sincera amizade.

À Virginia, Helena e Fábio, agradeço por toda ajuda. Ao meu orientador Marcelo, pelos ensinamentos, paciência e por ter me dado a oportunidade de trabalhar no GCG.

Finalmente agradeço àqueles que não foram citados mas que contribuíram de alguma forma para a realização deste trabalho.

*“Para ser grande, sê inteiro: nada*

*Teu exagera ou exclui.*

*Sê todo em cada coisa. Põe quanto és*

*No mínimo que fazes.*

*Assim em cada lago a lua toda*

*Brilha, porque alta vive.”*

*Ricardo Reis*

# Contents

**List of Figures** **7**

**List of Tables** **8**

**Abbreviations List** **9**

**List of Symbols** **10**

**1 Introduction** **11**

1.1 Theme Presentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

1.2 Problem Definition . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

1.3 Objectives . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

**2 Related Works** **14**

**3 Fundamentals** **17**

3.1 Histogram of Oriented Gradients 3D . . . . . . . . . . . . . . . . . . . 17

3.2 Global Tensor Descriptor . . . . . . . . . . . . . . . . . . . . . . . . . 17

3.3 Multivariate Gaussian Distribution . . . . . . . . . . . . . . . . . . . . 18

3.4 Lie Algebra . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

3.5 L$^2$EMG Descriptor . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

**4 Proposed Method** **22**

4.1 Computing HOG3D . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

4.2 Estimating the Gaussian for a Frame . . . . . . . . . . . . . . . . . . . 24

4.3 Embedding Gaussian into a Linear Space . . . . . . . . . . . . . . . . . 25

4.4 Gaussian for a Video . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

4.5 Video Descriptor . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

**5 Results and Discussion** **27**

5.1 Datasets . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

5.1.1 KTH . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

5.1.2 MuHAVi . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

5.1.3 SKIG . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28

5.2 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

5.2.1 KTH . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

5.2.2 MuHAVi . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33

5.2.3 SKIG . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36

**6 Conclusion** **39**

**Bibliography** **41**

# List of Figures

4.1 Flowchart of our descriptor. . . . . . . . . . . . . . . . . . . . . . . . . . 23
4.2 HOG3D computation. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
4.3 Triangular weight vector of size $l = 3$. It is used to filter the frame information. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

5.1 KTH dataset. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
5.2 MuHAVi dataset. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
5.3 SKIG dataset. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
5.4 Confusion matrix for the best result in MuHAVi dataset. The recognition rate in this case is 89.92%. . . . . . . . . . . . . . . . . . . . . . . . . . 34

# List of Tables

5.1 Grid-search parameters . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
5.2 Training Protocols . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
5.3 Accuracy for KTH dataset varying $l$ and $\epsilon$ parameters. The triangular weight vector $\beta$ is adjusted according to its size $l$. . . . . . . . . . . . . . 30
5.4 Confusion matrix of the best result for KTH dataset. The recognition rate in this case is 90.73%. . . . . . . . . . . . . . . . . . . . . . . . . . . 31
5.5 Comparison between the zero mean-vector and non-zero mean vector for KTH dataset. Experiments were performed using $l = 13$ and $\epsilon = 10$. . . . . 32
5.6 Lie Algebra influence for KTH dataset. Experiments were performed using $l = 13$ and $\epsilon = 10$. . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
5.7 Method Comparison . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
5.8 KTH Comparison . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
5.9 Accuracy for MuHAVi dataset varying $l$ and $\epsilon$ parameters. The triangular weight vector $\beta$ is adjusted according to its size $l$. . . . . . . . . . . . . . 33
5.10 Comparison between the zero mean-vector and non-zero mean vector for MuHAVi dataset. Experiments were performed using $l = 19$ and $\epsilon = 100$. . 35
5.11 Lie Algebra influence for MuHAVi dataset. Experiments were performed using $l = 19$ and $\epsilon = 100$. . . . . . . . . . . . . . . . . . . . . . . . . 35
5.12 MuHAVi Comparison . . . . . . . . . . . . . . . . . . . . . . . . . . 35
5.13 Accuracy for SKIG dataset varying $l$ and $\epsilon$ parameters. The triangular weight vector $\beta$ is adjusted according to its size $l$. . . . . . . . . . . . . . 36
5.14 Confusion Matrix of the SKIG dataset without reflection. The recognition rate is 87.50%. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
5.15 Comparison between the zero mean-vector and non-zero mean vector for SKIG dataset. Experiments were performed using $l = 19$ and $\epsilon = 100$. . . . 37
5.16 Lie Algebra influence for SKIG dataset. Experiments were performed using $l = 19$ and $\epsilon = 100$. . . . . . . . . . . . . . . . . . . . . . . . . . . 38
5.17 SKIG Comparison . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38

# Abbreviations List

| | |
|---|---|
| BoV | Bag-of-Visual-Words |
| CNN | Convolutional Neural Network |
| DE-LogE | Direct Embedding Log-Euclidean |
| HAR | Human Action Recognition |
| HOF | Histogram of Optical Flow |
| HOG | Histogram of Oriented Gradients |
| HOG3D | Three-dimensional Histogram of Oriented Gradients |
| IE-LogE | Indirect Embedding Log-Euclidean |
| $L^2$EMG | Local Log Euclidean Multivariate Gaussian |
| MBH | Motion Boundary Histograms |
| MEI | Motion Energy Images |
| MHI | Motion History Images |
| SIFT | Scale-Invariant Feature Transform |
| SIFT3D | Three-dimensional Scale-Invariant Feature Transform |
| SVM | Support Vector Machine |

# List of Symbols

| | |
|---|---|
| $\mathcal{N}(\mu, \Sigma)$ | Multivariate Gaussian Distribution |
| $E$ | Embedded Multivariate Gaussian Distribution |
| $\vec{h}$ | Histogram of Oriented Gradients |
| $D$ | Video Descriptor |
| $Sym(n)$ | Group of the $n \times n$ symmetric matrices |
| $Sym^+(n)$ | Group of the $n \times n$ symmetric and positive definite matrices |
| $Ut(n)$ | Group of the $n \times n$ upper triangular matrices |
| $PDUT(n)$ | Group of the $n \times n$ upper triangular matrices with positive diagonal entries |
| $A(n+1)$ | Group of the $(n+1) \times (n+1)$ upper triangular matrices |
| $A^+(n+1)$ | Group of the $(n+1) \times (n+1)$ upper triangular matrices with positive diagonal entries |
| $N(n)$ | Space of $n$-dimensional Gaussians |

# 1 Introduction

## 1.1 Theme Presentation

Human Action (or Activity) Recognition (HAR) has been one of the main concerns of the computer vision community. Its goal is to identify correctly which action is being performed by a human in a video sequence. In other words, we are interested in classifying human actions according to a set of predefined labeled actions. The human action recognition field has several applications, such as human-computer interaction systems, games, video retrieval, monitoring of the elderly in smart homes, and especially in security systems, allowing the automatic surveillance of crowded places such as airports and shopping centers.

According to Aggarwal and Ryoo (2011), activities can be categorized into different levels: gestures, actions, interactions and group activities. Gestures are movements of a person's body part. "Raising a leg" and "stretching an arm" are examples of gestures. Actions can be seen as the activities composed of multiple gestures, such as "running", "walking". Interactions are human actions involving two persons or objects, for example, "making tea" and "two persons fighting". Group activities are the actions that involve multiple persons or objects. "A group of persons marching together" and "two groups fighting" are examples of group activities.

With the advance of deep learning methods, researchers recently turned their eyes on how to train a deep neural network capable of recognizing human actions in videos. Deep models (WANG; QIAO; TANG, 2015) have been achieving outstanding results and outperforming handcrafted state-of-the-art methods. Although these architectures have demonstrated impressive results, Nguyen, Yosinski and Clune (2015) proved that they can be fooled, indicating that handcrafted methods are still interesting to explore. Focused on feature engineering, the classical framework for human action recognition is basically composed of three steps: feature extraction, descriptor creation, and action classification. In the feature extraction step, we need extract the visual and temporal information from videos. For this, we can use feature descriptors like Histogram of Oriented Gradients (HOG) (DALAL; TRIGGS, 2005), Histogram of Optical Flow (HOF) (DALAL; TRIGGS; SCHMID, 2006), Motion Boundary Histograms (MBH) (DALAL; TRIGGS; SCHMID, 2006). The descriptor creation step is responsible to convert the feature descriptors into motion descriptors able to describe human actions. In the action classification, a classifier such as Support Vector Machine (SVM) (CORTES; VAPNIK, 1995) is used to classify the descriptors resulted from the previous step.

In this work, given extracted features, in the descriptor creation step we combine the proposals of Li et al. (2017) and Perez et al. (2012). Li et al. (2017) proposed an image descriptor associating one pixel with a multivariate Gaussian distribution whose covariance matrix and mean vector are estimated in its neighborhood. Perez et al. (2012) proposed a video descriptor based on histograms of gradient computed over the spatio-temporal domain and accumulated into orientation tensors. Works like Perez et al. (2012), Mota et al. (2013), Mota et al. (2014) evidenced that orientation tensors are good motion descriptors by their power of aggregation and because they are compact forms of representing a movement. They can also be viewed such as a covariance matrix from a Gaussian with mean vector zero. This encourages us to explore how strategies based on multivariate Gaussian distributions can be applied to the human action recognition problem and what is the impact of the mean vector of Gaussians to the recognition.

In the action classification step, we evaluate our generated descriptors for the actions contained in the datasets KTH, MuHAVi and SKIG through the SVM classifier, widely used in human action recognition works.

## 1.2 Problem Definition

Consider an image $I(x, y)$ as a two-dimensional matrix where each cell $(x, y)$ of the matrix contains the brightness intensity of the pixel with spatial coordinates $(x, y)$. A video can be seen as a concatenation of images (called frames) over the temporal domain, represented as a three-dimensional matrix $v(x, y, t)$, where $t$ indicates a frame position.

The problem of describing a video can be defined as: given a space of videos $\mathcal{S}$, we are interested in creating a function $f : \mathcal{S} \rightarrow \mathbb{R}^n$, where $\mathbb{R}^n$ is a space of $n$-dimensional descriptors, so that descriptors of similar videos are also similar if compared using the Euclidean norm.

## 1.3 Objectives

The main objective of this work is to create a video descriptor based on a multivariate Gaussian distribution which is able to describe human actions. The secondary objectives are:

1. Evaluate the impact of the mean vector to recognition, verifying if the Gaussians with mean vector different of zero are capable of improving the recognition and if yes, how much is possible;

2. Evaluate our method for MuHAVi, SKIG and KTH datasets.

# 2 Related Works

Shape and silhouette were the first features used for human action recognition. Silhouettes are robust to color, texture and contrast changes. However, they are sensitive to the viewpoint and is still difficult to obtain a person segmentation. Bobick and Davis (2001) introduced the Motion Energy Images (MEI) and the Motion History Images (MHI) methods. The goal is a single image representing the motion information. MEI uses a binary image to describe the presence of the motion and the MHI uses a greyscale image to describe how the motion occurs. This method was the first to introduce the idea of temporal templates for human action recognition. Blank et al. (2005) extended the MEI template to the space-time, resulting in a volumetric extension of MEI, where a 3D surface is mapped in a 2D image. This extension adds robustness to viewpoint variations.

The most popular feature descriptors for human action recognition are Histogram of Oriented Gradients (HOG) and Histogram of Optical Flow (HOF). Gradient and optical flow methods are sensitive to material properties, texture and illumination. The gradient is robust to camera movements, whereas the optical flow is not. However, objects in the background may be confused in the gradient methods. The main motivation to use optical flow methods is the access to the temporal information, extracted through the temporal gradient in other approaches. Polana and Nelson (1992) used for the first time the optical flow as a motion descriptor, to recognize natural events, such as the motion of trees and water. Later, Polana and Nelson (1994) applied the optical flow to recognize periodic human actions. The optical flow magnitudes were accumulated in a spatio-temporal grid of non-overlapping bins. Introduced by Dalal, Triggs and Schmid (2006), the Motion Boundary Histogram (MBH) image descriptor is an extension of the optical flow. MBH is computed through the spatial derivative of the optical flow. These descriptors are able to capture the relative motion of different limbs while resisting background motions.

Laptev et al. (2008) detect interest points in a video and compute histogram descriptors of space-time volumes in the neighborhood of detected points. These volumes are subdivided into cuboids and for each cuboid are computed histograms of oriented gradients and optical flow. Similarly to the SIFT descriptor (LOWE, 1999), the histograms are normalized and concatenated into HOG and HOF descriptor vectors. Introduced by Klaser, Marszałek and Schmid (2008), HOG3D is an extension of HOG to spatio-temporal domain inspired by the SIFT3D descriptor (SCOVANNER; ALI; SHAH, 2007). To bin a gradient in 3D, the HOG3D descriptor uses convex regular polyhedrons instead of spherical coordinates used in SIFT3D.

Due to the possible different dimensions, scales, and the high dimensionality, comparing local descriptors may be not suitable. In this context, the Bag-of-Visual-Words (BoV) model is attractive. In the BoV, a codebook is generated using a clustering algorithm and each cluster is then associated with a codeword. A histogram of codeword frequencies represents the video descriptor. Laptev et al. (2008) build a spatio-temporal bag-of-features using a set of spatio-temporal features. The BoV representation assigns each feature to the closest codeword and computes the histogram of codeword frequencies over a space-time volume.

Wang et al. (2011) proposed a descriptor based on dense trajectories, that captures the local motion information of the video. Dense points from each frame are sampled and tracked based on displacement information from a dense optical flow field. A dense representation guarantees a good coverage of the main movement and surroundings.

Perez et al. (2012) use a global video descriptor based on histograms of gradients computed over the spatio-temporal domain. The histograms are calculated over a grid and then they are accumulated into orientation tensors. They used an orientation tensor as a descriptor by their power of aggregation and because it is a compact form of representing a movement. Similarly, Mota et al. (2013) compute three-dimensional histograms of oriented gradients in equally sized blocks throughout the video sequence. These histograms are encoded into orientation tensors, and then, they are concatenated to create a video descriptor. Focusing on block matching algorithms, Maia et al. (2015) proposed an approach that divides an image into blocks and encodes displacement vector provided from the block matching for each block into orientation tensors aiming to generate the final self-descriptor. The final descriptor is a sum of the frame tensors. Figueiredo et al. (2016) proposed a video self-descriptor based on sparse trajectory clustering. The displacement vectors are obtained through the cross product between the block matching vector and the gradient for each frame, resulting in block trajectories that contain the temporal information. The block matching vectors are used to cluster the trajectories according to their shape. This information is encoded into orientation tensors to generate the final descriptor.

Recently, deep models have been achieving outstanding results and outperforming handcrafted state-of-the-art methods. The convolutional neural network (CNN) is the most used deep architecture in computer vision tasks. The CNN is a neural network where the neurons are composed by convolutional and pooling layers. These layers provide robustness across spatial variations. Aiming to equip the CNN with temporal information, Ji et al. (2013) introduced the 3D convolutional networks that use 3D kernels to extract features from spatio-temporal dimensions. Simonyan and Zisserman (2014) introduced the multiple-stream deep convolutional networks. These networks use two parallel networks to separate appearance from motion information. An extension of the two-stream network was proposed in Wang, Qiao and Tang (2015), that use dense trajectories traced over convolutional feature maps aggregated using the Fisher vector, that consists in characterizing a sample by its deviation from the generative model. Although these architectures have demonstrated impressive results, in a recent study, Nguyen, Yosinski and Clune (2015) used images unrecognizable by humans in their tests and a deep neural network predicted as recognizable objects with over 99% of confidence, indicating that deep approaches can be fooled and the handcrafted methods are still interesting to explore.

Li et al. (2017) proposed an image descriptor associating one pixel with a multivariate Gaussian distribution estimated in the neighborhood. They proved that the space of Gaussians can be provided with a Lie group structure by defining a multiplication operation on this manifold. The space of Gaussians is isomorphic to a subgroup of the upper triangular matrix. Then the authors proposed two methods to embed this space into a linear space respecting the geometry of Gaussians thus enabling use the Euclidean operations.

# 3 Fundamentals

## 3.1 Histogram of Oriented Gradients 3D

The brightness gradient indicates a local intensity variation and is approximated by the application of a derivative filter. According to Perez et al. (2012), the three-dimensional gradient of the $j$-th video frame $I_j$ at pixel $p$ can be defined as:

$$\vec{g_t}\,(p) = [dx\ dy\ dt]\ = \left[\frac{\partial I_j(p)}{\partial x}\ \frac{\partial I_j(p)}{\partial y}\ \frac{\partial I_j(p)}{\partial t}\right], \tag{3.1}$$

or equivalently, in spherical coordinates:

$$\vec{s_t}\,(p) = [\rho_p\ \theta_p\ \varphi_p]\ , \tag{3.2}$$

where $\theta_p\ \epsilon\ [0, \pi]$, $\varphi_p\ \epsilon\ [0, 2\pi)$ and $\rho_p = \|\vec{g_t}\,(p)\|$.

The gradient for all $n$ points of an image $I_j$ are quantized in a three-dimensional histogram of gradients (HOG3D) $\vec{h_j} = \{h_{k,q}\}$ and

$$h_{k,q} = \sum_p \rho_p \cdot \omega_p. \tag{3.3}$$

where $k\ \epsilon\ [1, n_{b_\theta}]$, $q\ \epsilon\ [1, n_{b_\varphi}]$, $n_{b_\theta}$ and $n_{b_\varphi}$ are the number of bins for $\theta$ and $\varphi$ respectively, $\{p\ \epsilon\ I_j \mid k = 1 + \left[\frac{nb_\theta \cdot \theta_p}{\pi}\right]\ ,\ q = 1 + \left[\frac{nb_\varphi \cdot \varphi_p}{2\pi}\right]\}$ are the points whose angles map to the $k$ and $q$ bins and $\omega_p$ is a per pixel weighting Gaussian factor.

## 3.2 Global Tensor Descriptor

Perez et al. (2012) proposed a method for human action recognition based on the combination of Histograms of Oriented Gradients and orientation tensors. The proposal is to make a simple global tensor descriptor using only the information extracted from the HOG3D.

The first step is to partition each frame $I_j$ from a video $v$ in $x$ and $y$ directions by a uniform grid with $n_x$ and $n_y$ nonoverlapping blocks. Each block can be viewed as a distinct video. The next step is to calculate the HOG3D to all video frames. For each frame the histogram is calculated to each block, resulting in histograms $\vec{h}_j^{a,b}$, $a \in [1, n_x]$ and $b \in [1, n_y]$. In a power normalization process, to empirically reduce interframe brightness unbalance, all elements $a_k$ of the histogram $\vec{h}_j^{a,b}$ are adjusted to $a_k^\alpha$. The frame tensor is computed as the sum of all block tensors:

$$\mathbb{T}_{I_j} = \sum_{a,b} \vec{h}_j^{a,b} \vec{h}_j^{a,b^T}. \tag{3.4}$$

Then, the global tensor descriptor is given by:

$$\mathbb{T}_v = \sum \mathbb{T}_{I_j}. \tag{3.5}$$

To enforces horizontal gradient symmetries that occur in the video, the video frame is flipped horizontally and the same process is executed. The global tensor descriptor adds also the flipped frame information. To be able to compare different videos, $\mathbb{T}_v$ is normalized with a $L_2$-norm.

## 3.3 Multivariate Gaussian Distribution

Given a random vector variable $X = [X_1, X_2, ..., X_n]^T$, it has a multivariate Gaussian distribution if its probability density function is given by

$$\mathcal{N}_x(\mu, \Sigma) = |2\pi\Sigma|^{-1/2} \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)), \tag{3.6}$$

where $x \in X$, $\mu = E[X] = [E[X_1], E[X_2], ..., E[X_n]]^T$ is the mean vector, $\Sigma = E[(X - \mu)(X-\mu)^T]$ is the covariance matrix and $|\cdot|$ is the matrix determinant.

## 3.4 Lie Algebra

A Lie Algebra $\mathfrak{g}$ over a field $\mathbb{F}$ is a vector space equipped with a product $[\cdot,\cdot]$: $\mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ called Lie bracket that satisfies the following axioms:

1. bilinearity: $[ax + by, z] = a[x, z] + b[y, z]$ and $[z, ax + by] = a[z, x] + b[z, y]$
2. skew symmetry: $[x, y] = -[x, y]$
3. Jacobi Identity: $[x, [y, z]] + [y, [z, x]] + [z, [x, y]] = 0$

for all $x, y, z \in \mathfrak{g}$ and $a, b \in \mathbb{F}$.

A Lie group $G$ is a group provided with the structure of a differential manifold such that the inverse and the multiplication are smooth functions. A Lie group homomorphism $\phi : G \to G'$ is a smooth function that satisfies

$$\phi(a \cdot b) = \phi(a) \circ \phi(b), \tag{3.7}$$

where $G, G'$ are Lie groups, $a, b \in G$. If $\phi$ is a bijective function and $\phi^{-1}$ is smooth, $G$ is isomorphic to $G'$ and they are equivalent. Given a Lie Algebra of a Lie group, we can transfer Lie Algebra properties to the Lie group. This process allows us to describe Lie groups, that are typically non-linear, through the linear algebra embedded in Lie Algebra.

Matrices groups are attractive for the computer vision community. The group $PDUT(n)$ of $n \times n$ upper triangular matrices with positive diagonal entries have the set of all $n \times n$ upper triangular matrices, $Ut(n)$, as its Lie algebra. The Lie algebra for the group $Sym^+(n)$ of the $n \times n$ symmetric and positive definite matrices is the set $Sym(n)$ of all $n \times n$ symmetric matrices. According to Li et al. (2017), the space $N(n)$ of $n$-dimensional Gaussians $\mathcal{N}(\mu, \Sigma)$ is isomorphic to a subgroup of the upper triangular matrices

$$A^+(n+1) = \left\{ A_{\mu,Z} \triangleq \begin{bmatrix} Z & \mu \\ 0^T & 1 \end{bmatrix} | Z \in PDUT(n), \mu \in \mathbb{R}^n \right\}. \tag{3.8}$$

The group

$$A(n+1) = \left\{ A_{t,X} \triangleq \begin{bmatrix} X & t \\ 0^T & 0 \end{bmatrix} | X \in Ut(n), t \in \mathbb{R}^n \right\} \tag{3.9}$$

is the Lie Algebra of $A^+(n+1)$. This result is extremely important for this work because is based on Lie group isomorphisms, which keeps safe algebraic and topological structure of the spaces involved. It allows us to embed Gaussians into a linear space and to combine them using Riemannian operations.

## 3.5 L$^2$EMG Descriptor

Li et al. (2017) proposed an image descriptor, called Local Log-Euclidean Multivariate Gaussian (L$^2$EMG), that associates one-pixel point with a multivariate Gaussian distribution to characterize the first- and second-order statistics in the local neighborhood. Inspired by SIFT and HOG descriptors, the L$^2$EMG descriptor is continuous and can model high-order statistics whereas these histogram-based descriptors only estimate zero-order statistics.

Aiming to estimate Gaussians, Li et al. (2017) considered $I_j$ as an input image and $f(p)$ the $n$-dimensional vector of features computed over the image $I_j$. Each pixel $p$ of the image $I_j$ is represented by a multivariate Gaussian $\mathcal{N}(\mu(p), \Sigma(p))$ with mean vector $\mu(p)$ and covariance matrix $\Sigma(p)$. Let $P(p)$ be a $r \times r$ image patch centered at $p$. The estimated Gaussian can be written as the Equation 3.6 with

$$\begin{aligned} \mu(p) &= \frac{1}{r^2}\sum_i \sum_j f(P_{i,j}(p)) \\ \Sigma(p) &= \frac{1}{r^2-1}\sum_i \sum_j (f(P_{i,j}(p)) - \mu(p))(f(P_{i,j}(p)) - \mu(p))^T \end{aligned} . \tag{3.10}$$

After estimating the multivariate Gaussian distributions, these Gaussians can be embedded into a linear space through one of the following methods: Direct Embedding Log-Euclidean (DE-LogE) and Indirect Embedding Log-euclidean (IE-LogE), this last one based on the left and right coset. Let $H$ be a closed subgroup of the group $G$. A left coset of $H$ in $G$ is a subset $aH = \{a \cdot h | h \in H \text{ and } a \in G\}$ and the right coset of $H$ in $G$ is a subset $Ha = \{h \cdot a | h \in H \text{ and } a \in G\}$.

The first embedding method, called direct embedding Log-Euclidean (DE-LogE), maps $A^{+}(n+1)$ via matrix logarithm to the linear space $A(n+1)$. For the DE-LogE computation, $\Sigma = L^{-T}L^{-1}$, where $L$ is the Cholesky factor of $\Sigma^{-1}$. As the covariance matrix $\Sigma(p)$ may be rank-deficient, a small positive number $\epsilon$ is added to the diagonal elements. The embedding matrix can be written as:

$$\log(A_{\mu,L^{-T}}) = \log \begin{bmatrix} L^{-T} & \mu \\ 0^T & 1 \end{bmatrix}. \tag{3.11}$$

The second method, called indirect embedding Log-euclidean (IE-LogE) maps $A^{+}(n+1)$ via the coset and polar decomposition into $Sym^{+}(n+1)$, and then into the linear space $Sym(n+1)$. For IE-LogE based on the left coset is computed:

$$P_{\mu,L^{-T}} = \begin{bmatrix} \Sigma + \mu\mu^T & \mu \\ \mu^T & 1 \end{bmatrix} = O\,\mathrm{diag}\,(\lambda_i)O^T, \tag{3.12}$$

where diag $(\lambda_i)$ is the diagonal matrix consisting of eigenvalues $\lambda_i = 1, ..., n+1$ and $O$ is an orthogonal matrix consisting of eigenvectors corresponding to $\lambda_i$ and the embedding matrix is defined by:

$$\log(P_{\mu,L^{-T}}) = O\,\mathrm{diag}\left(\frac{1}{2}\log(\lambda_i)\right)O^T. \tag{3.13}$$

Similarly, for IE-LogE based on the right coset is computed:

$$P'_{\mu,L^{-T}} = \begin{bmatrix} L^{-1}L^{-T} & L^{-1}\mu \\ \mu^T L^{-T} & \mu^T\mu + 1 \end{bmatrix} = O'\,\mathrm{diag}\,(\lambda'_i)O'^T, \tag{3.14}$$

where diag $(\lambda'_i)$ is the diagonal matrix consisting of eigenvalues $\lambda'_i = 1, ..., n+1$ and $O^{'}$ is an orthogonal matrix consisting of eigenvectors corresponding to $\lambda'_i$ and the embedding matrix is defined by:

$$\log(P'_{\mu,L^{-T}}) = O'\,\mathrm{diag}\left(\frac{1}{2}\log(\lambda'_i)\right)O'^T. \tag{3.15}$$

# 4 Proposed Method

Our method combines the works of Perez et al. (2012) and Li et al. (2017) aiming to create a video descriptor for the human action recognition problem. Instead of pixels, we are interested in estimating multivariate Gaussian distribution for all frames using the HOG3D information and combining the frame Gaussians to estimate a Gaussian for the video.

Since that Gaussians are composed by a mean vector and a covariance matrix, notice that Equations 3.4 and 3.5 are covariances provided from Gaussians with zero mean vector. Although there exists Lie algebra for the space of orientation tensors, it is inappropriate to compare Gaussians with mean vector different of zero. It occurs because this space would contemplate only the covariance matrices, disregarding the mean vectors and implying into compare covariance matrices instead of Gaussians. In this context, the Lie algebra for the Gaussians space, presented by Li et al. (2017), allow us to adequately compare them. Whereas Perez et al. (2012) was interested in orientation tensors as descriptors, we are interested in Gaussians as video descriptors.

Figure 4.1 illustrates the flowchart of our descriptor. Given an input video, the first step is to compute the HOG3D features for all frames. These histograms are used to estimate the Gaussian for each frame, considering a triangular filter vector used to weight the frames. With the Gaussians calculated, they are embedded into the linear space $A(n+1)$ through the DE-LogE method. The Gaussian for the video is a sum of Gaussians of all frames. Finally, the descriptor is a concatenation of the covariance matrix and the mean vector from the estimated Gaussian of the video. The SVM classifier is used to the classification task.

## 4.1 Computing HOG3D

Aiming extract more information from the videos, we compute the HOG3D for a frame $I_j$. The Figure 4.2 illustrates the HOG3D computation. Firstly, the frame is subdivided

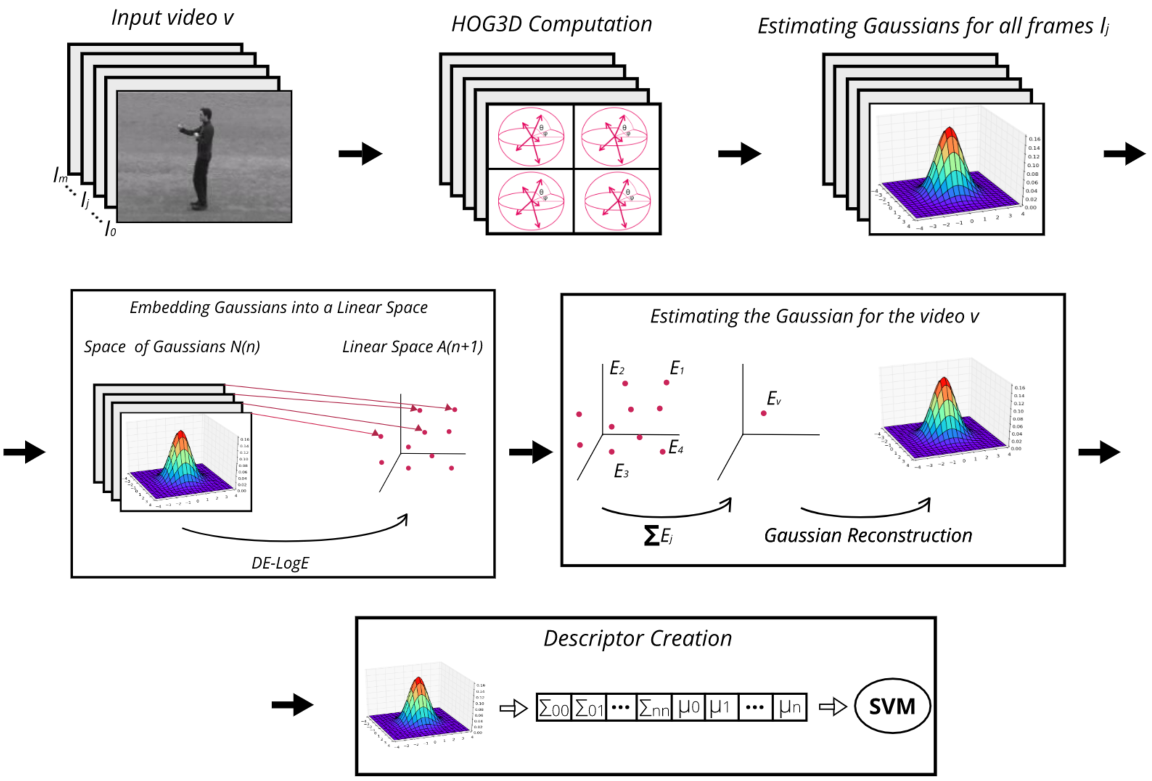

Figure 4.1: Flowchart of our descriptor.

in $n_x \times n_y$ uniform blocks. Each block yields one HOG3D $\vec{h}_j^{a,b}$, $a \in [1, n_x]$ and $b \in [1, n_y]$, resulting in $n_x \cdot n_y$ histograms of size $n_{b_\theta} \cdot n_{b_\varphi}$ for each frame, where $n_{b_\theta}$ and $n_{b_\varphi}$ are the number of bins of the histogram. Similarly to Perez et al. (2012), power normalization is applied on each histogram.

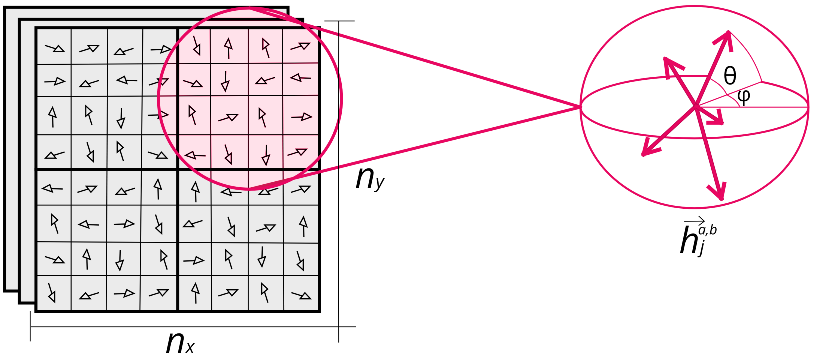

Figure 4.2: HOG3D computation.

## 4.2 Estimating the Gaussian for a Frame

To compute the Gaussian for the frame $I_j$ the information provided from its temporal neighborhood is considered. A triangular filter $\beta = \beta_0, \beta_1, ...\beta_l$ of size $l$ is centered in $I_j$ so that the histograms of the considered interval $[j - \frac{l}{2}, j + \frac{l}{2}]$ be weighted according to its distance to the frame $I_j$. Hence, frames more distant are less influential than others near to the frame $I_j$. Figure 4.3 illustrates this triangular weight vector of size $l = 3$ over a frame sequence.

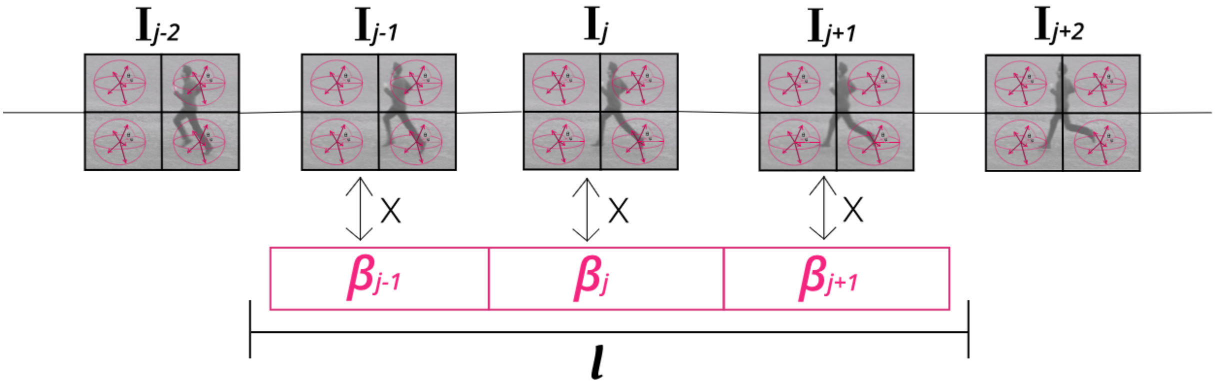

Figure 4.3: Triangular weight vector of size $l = 3$. It is used to filter the frame information.

The Gaussian $\mathcal{N}_{I_j}(\mu(I_j), \Sigma(I_j))$ for a frame $I_j$ is given as in Equation 3.6, where the mean vector $\mu(I_j)$ is given as:

$$\mu(I_j) = \frac{1}{n_x n_y l \|\beta\|_2} \sum_{a}^{n_x} \sum_{b}^{n_y} \sum_{i=j-\frac{l}{2}}^{j+\frac{l}{2}} \beta_i \vec{h}_i^{a,b}, \tag{4.1}$$

and the covariance matrix $\Sigma(I_j)$ is given as:

$$\Sigma(I_j) = \frac{1}{(n_x n_y l - 1) \cdot \|\beta\|_2} \sum_{a}^{n_x} \sum_{b}^{n_y} \sum_{i=j-\frac{l}{2}}^{j+\frac{l}{2}} (\beta_i \vec{h}_i^{a,b} - \mu(I_j))(\beta_i \vec{h}_i^{a,b} - \mu(I_j))^T, \tag{4.2}$$

where $\|\beta\|_2$ indicates the $L_2$-norm of the vector $\beta$. As the covariance matrix $\Sigma(I_j)$ may be rank-deficient, an $\epsilon$ value is added to the diagonal elements.

## 4.3 Embedding Gaussian into a Linear Space

With the Gaussian $\mathcal{N}_{I_j}(\mu(I_j), \Sigma(I_j))$ computed for a frame $I_j$, it is embedded into the space $A(n+1)$ through the DE-LogE embedding method. We choose this method for simplicity and because it provides good results. So

$$E_{I_j} = \log \begin{bmatrix} L_{I_j}^{-T} & \mu(I_j) \\ 0^T & 1 \end{bmatrix}, \tag{4.3}$$

where $L_{I_j}$ is the Cholesky factor of $\Sigma^{-1}(I_j)$ and $\mu(I_j)$ is the mean vector of $\mathcal{N}_{I_j}$.

## 4.4 Gaussian for a Video

The embedded Gaussian $E_v$ for a video $v$ is computed through the Gaussians of the $m$ frames $Ij$ of $v$. Given a set of embedded Gaussians, $E = [E_{I_1}, E_{I_2}, ..., E_{I_m}]$, the embedded Gaussian $E_v$ is given by the sum of all elements of $E$:

$$E_v = \sum_{j}^{m} E_{Ij}. \tag{4.4}$$

As well as in Perez et al. (2012), we can also add the horizontal flipped video information. The same process is executed for the horizontal flipped video, resulting in Gaussians $E' = [E_{I'_1}, E_{I'_2}, ..., E_{I'_m}]$. Thus, if the flipped video is considered, the embedded Gaussian $E(v)$ is given by the sum of all elements of $E$ and $E'$:

$$E_v = \sum_{j}^{m} E_{Ij} + E_{I'j}. \tag{4.5}$$

To be able to compare different videos, $E_v$ is normalized with a $L_2$-norm.

## 4.5 Video Descriptor

To compute the video descriptor we bring back $E_v$ to the space of Gaussians, resulting in $\mathcal{N}_v(\Sigma(v), \mu(v))$.

The video descriptor $D$ is given by the concatenation of the linearized covariance matrix and the mean vector of $\mathcal{N}_v$:

$$D = [\Sigma(v)|\mu(v)]\,. \tag{4.6}$$

It is important to notice that if the histogram is a column vector of size $n_{b_\theta}n_{b_\varphi}$, the covariance matrix is a symmetric matrix with size $n_{b_\theta}n_{b_\varphi} \times n_{b_\theta}n_{b_\varphi}$. It means that only its upper triangular coefficients are needed. So, we store only $n_{b_\theta}n_{b_\varphi} \cdot (\frac{n_{b_\theta}n_{b_\varphi}+1}{2}) + n_{b_\theta}n_{b_\varphi}$ elements in total.

# 5 Results and Discussion

## 5.1 Datasets

### 5.1.1 KTH

The KTH (SCHULDT; LAPTEV; CAPUTO, 2004) dataset contains 2391 sequences acquired over a homogeneous background with a static camera. It contains six types of human actions (*walking*, *jogging*, *running*, *boxing*, *hand waving*, *hand clapping*) performed by 25 people in four different ambients. The resolution is 160×120 pixels and the framerate is 25fps.

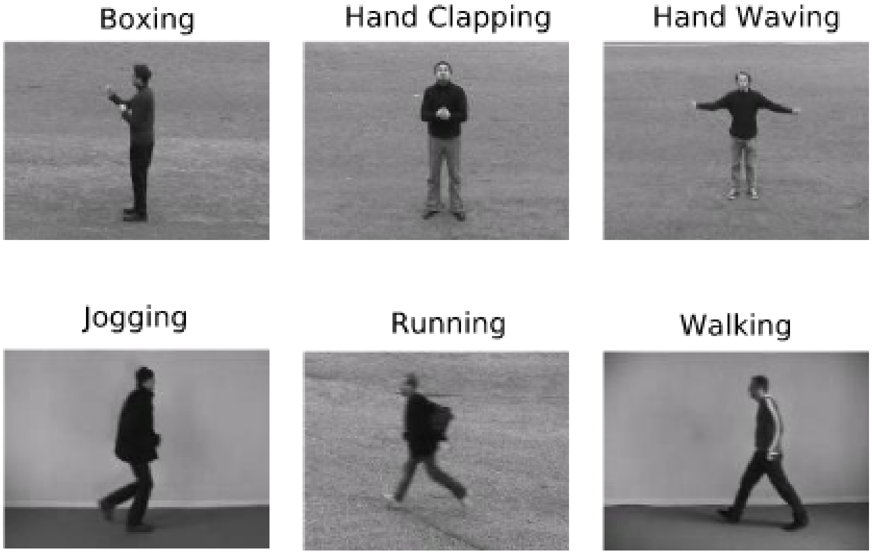

Figure 5.1: KTH dataset.

### 5.1.2 MuHAVi

The MuHAVi (SINGH; VELASTIN; RAGHEB, 2010) dataset contains 17 action class (*walk turn back*, *run stop*, *punch*, *kick*, *shot gun collapse*, *pull heavy object*, *pickup throw object*, *walk fall*, *look in car*, *crawl on knees*, *wave arms*, *draw graffiti*, *jump over fence*, *drunk walk*, *climb ladder*, *smash object*, *jump over gap*) performed by 7 persons, totalling 119 videos. The actions are surrounding by 8 cameras and occur in a closed scenario. Only the information from the camera 4 is considered. The video resolution is 720×576 pixels and the framerate is 25fps.

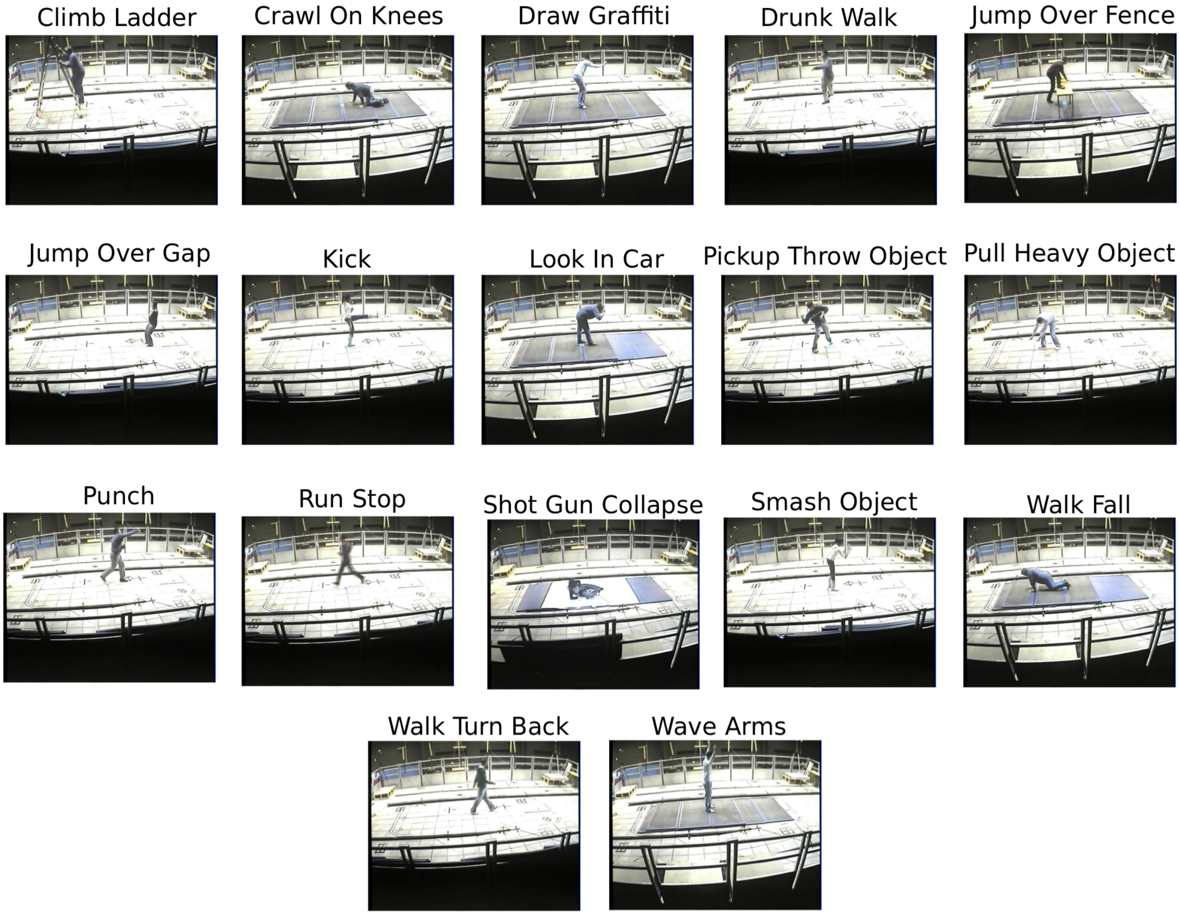

Figure 5.2: MuHAVi dataset.

### 5.1.3 SKIG

The SKIG (LIU; SHAO, 2013) dataset contains 1080 gesture sequences captured with a Kinect sensor from 6 subjects performing 10 hand gestures (*circle*, *triangle*, $up - down$, $right - left$, *wave*, "*Z*", *cross*, *come here*, *turn around*, *pat*) in three hand postures: fist, index and flat. The sequences were performed under three different backgrounds (wooden board, white plain paper and paper with characters) and 2 illumination conditions (strong light and poor light).

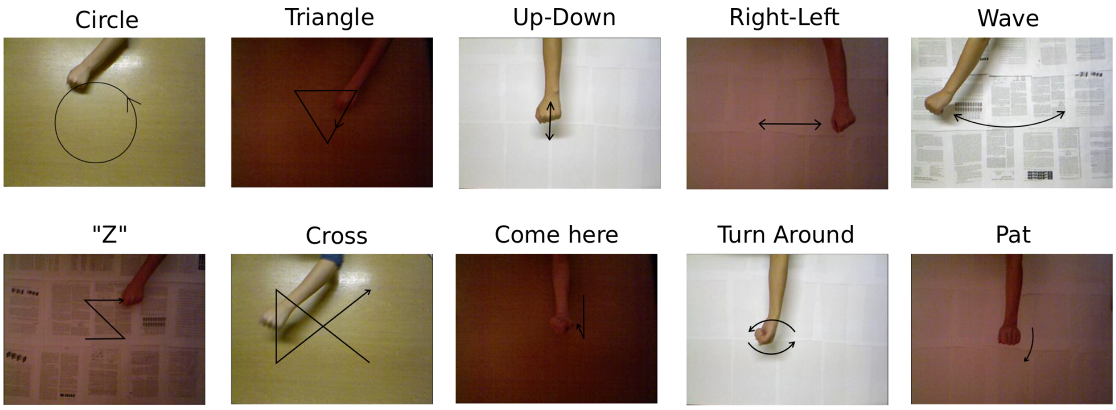

Figure 5.3: SKIG dataset.

## 5.2 Results

We evaluated our method using three datasets: KTH, MuHAVi and SKIG. To this, we fixed the parameters for the HOG3D computation: $n_{b_\phi} = 16$, $n_{b_\theta} = 8$, $n_x = 8$, $n_y = 8$, $\alpha = 0.72$. These parameters were found by Perez et al. (2012), for KTH dataset. We fixed it for KTH in order to compare our method with Perez et al. (2012) method. Since Perez et al. (2012) not evaluated MuHAVi and SKIG datasets, we maintain the same parameters found for KTH because to evaluate HOG3D parameters is not in our scope. In order to find the best values for the parameters $l$, $\beta$ and $\epsilon$, we performed tests varying some sizes $l$ for the triangular weight vector $\beta$ and for the parameter $\epsilon$ and, as well as in Perez et al. (2012), we also performed experiments adding horizontal reflected dataset information. Experiments comparing the impact of the mean vector and Lie Algebra to recognition were performed for all datasets.

We used the Scikit-learn (PEDREGOSA et al., 2011) implementation of the SVM classifier with a Radial Basis Function (RBF) kernel, with a one-vs.-rest strategy. Table 5.1 contains the grid-search parameters and Table 5.2, the training protocols for each dataset.

Table 5.1: Grid-search parameters

| **Parameters** | **Values** |
|---|---|
| C | $10^{-5}$, $10^{-4}$, $10^{-3}$, $10^{-2}$, $10^{-1}$, 1, 10, $10^{2}$, $10^{3}$, $10^{4}$, $10^{5}$ |
| $\gamma$ | $10^{-5}$, $10^{-4}$, $10^{-3}$, $10^{-2}$, $10^{-1}$, 0.5, 0.7812,1, 1.3889, 3.125, 10, 12.5, $10^{2}$, $10^{3}$, $10^{4}$, $10^{5}$ |

Table 5.2: Training Protocols

| **Dataset** | **Protocol** | **Split** |
|---|---|---|
| KTH | Grid-search<br>and fixed train-validation-test split | train : 8 persons<br>validation: 8 persons<br>test: 9 persons |
| MuHAVi | Grid-search with 3-fold cross-validation<br>and leave-one-group-out for performance testing | 7 persons split |
| SKIG | Grid-search with 3-fold cross-validation<br>and fixed training-testing sets for performance evaluation | train : 4 persons<br>test: 2 persons |

### 5.2.1 KTH

In order to find the best combination of the triangular weight vector size $l$ and the adjust parameter $\epsilon$, we evaluate eleven different sizes to $l$: 1, 3, 5, 7, 9, 11, 13, 15, 19 and 21 frames. These experiments were performed varying three values for $\epsilon$: 1, 10 and 100. Table 5.3 contains the accuracy that was found for each combination. We can notice that the weighting yields some improvement, outperforming 90.00% of accuracy with only 5 frames. The parameter $\epsilon$ varies according to the number of frames considered, being smaller when the number of frames increases. The best combination found on the evaluated interval was $l = 13$ and $\epsilon = 10$, with 90.75% of average recognition rate.

Table 5.3: Accuracy for KTH dataset varying $l$ and $\epsilon$ parameters. The triangular weight vector $\beta$ is adjusted according to its size $l$.

| $\boldsymbol{l}$ | $\beta$ | $\epsilon$ | **Accuracy** |
|---|---|---|---|
| 1 | [1] | 1 | 85.75% |
| | | 10 | 83.43% |
| | | 100 | 85.75% |
| 3 | [1, 2, 1] | 1 | 86.33% |
| | | 10 | 89.46% |
| | | 100 | 89.57% |
| 5 | [1, 2, 3, 2, 1] | 1 | 89.11% |
| | | 10 | 90.38% |
| | | 100 | 87.37% |
| 7 | [1, 2, 3, 4, 3, 2, 1] | 1 | 89.34% |
| | | 10 | 90.61% |
| | | 100 | 88.76% |
| 9 | [1, 2, 3, 4, 5, 4, 3, 2, 1] | 1 | 89.80% |
| | | 10 | 90.50% |
| | | 100 | 89.22% |
| 11 | [1, 2, 3, 4, 5, 6, 5, 4, 3, 2, 1] | 1 | 89.57% |
| | | 10 | 88.30% |
| | | 100 | 89.57% |

| | | | |
|---|---|---|---|
| 13 | [1, 2, 3, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1] | 1 | 89.46% |
| | | 10 | **90.73%** |
| | | 100 | 89.11% |
| 15 | [1, 2, 3, 4, 5, 6, 7, 8, 7, 6, 5, 4, 3, 2, 1] | 1 | 89.80% |
| | | 10 | 90.38% |
| | | 100 | 88.88% |
| 17 | [1, 2, 3, 4, 5, 6, 7, 8, 9, 8, 7, 6, 5, 4, 3, 2, 1] | 1 | 89.80% |
| | | 10 | 90.38% |
| | | 100 | 88.06% |
| 19 | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1] | 1 | 89.80% |
| | | 10 | 90.38% |
| | | 100 | 87.95% |
| 21 | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1] | 1 | 90.15% |
| | | 10 | 90.50% |
| | | 100 | 87.72% |

Table 5.4 has the confusion matrix for the best result found to KTH. The vertical direction has actual labels and the horizontal direction has the predicted labels. The grayscale varies according to the percentage of samples labeled as being of a certain class and the diagonal entries indicate the percentage of the samples correctly predicted. The accuracy 90.50% is obtained through of the average of the values of its diagonal. The actions *jogging* and *running* get mixed up because they are very similar, with persons moving horizontally through the frame. The main difference between these actions is the movement speed.

Table 5.4: Confusion matrix of the best result for KTH dataset. The recognition rate in this case is 90.73%.

| | Boxing | Clapping | Waving | Jogging | Running | Walking |
|---|---|---|---|---|---|---|
| Boxing | **97.20** | 2.80 | 0.00 | 0.00 | 0.00 | 0.00 |
| Clapping | 5.60 | **94.44** | 0.00 | 0.00 | 0.00 | 0.00 |
| Waving | 3.47 | 0.70 | **95.83** | 0.00 | 0.00 | 0.00 |
| Jogging | 0.00 | 0.00 | 0.00 | **79.17** | 13.19 | 7.64 |
| Running | 0.00 | 0.00 | 0.00 | 20.14 | **79.86** | 0.00 |
| Walking | 0.00 | 0.00 | 0.00 | 2.08 | 0.00 | **97.92** |

Table 5.5 shows the recognition rates for our method using zero mean vector and non-zero mean vector for with reflection and without reflection cases. We can notice that for this dataset, adding reflection information, zero mean vector was better than the case when the mean vector is different of zero. We believe that it occurs due to the movements domain, being adequate to use Gaussians with zero mean vector when the dataset contains

a considerable number of actions performed over all frame, such as KTH.

Table 5.5: Comparison between the zero mean-vector and non-zero mean vector for KTH dataset. Experiments were performed using $l = 13$ and $\epsilon = 10$.

| Method | Mean Vector | Accuracy |
|---|---|---|
| Without reflection | zero | 90.73% |
| Without reflection | non-zero | 90.73% |
| With reflection | zero | 90.03% |
| With reflection | non-zero | 87.49% |

We can see in Table 5.6 the results obtained to the performed experiments in order to evaluate the Lie Algebra influence to combine Gaussians. Notice that combining Gaussians through a linear space yield considerable improvement, more evident in the case with reflection. It occurs because the Lie Algebra used is based on Lie group isomorphisms, which keeps safe algebraic and topological structure of the spaces involved.

Table 5.6: Lie Algebra influence for KTH dataset. Experiments were performed using $l = 13$ and $\epsilon = 10$.

| Method | Lie Algebra | Accuracy |
|---|---|---|
| Without reflection | yes | 90.73% |
| Without reflection | no | 90.03% |
| With reflection | yes | 87.49% |
| With reflection | no | 84.70% |

Table 5.7 has a comparison between the best results obtained by Perez et al. (2012) and by our method. As well as in Perez et al. (2012), we combine information from the horizontally reflected dataset. We obtained 87.49% of accuracy, whereas Perez et al. (2012) obtained 92.01%. For the dataset without reflection we not only reached but also outperformed Perez et al. (2012) in 1.39%. It evidences that Gaussians are capable of improving the recognition.

Table 5.7: Method Comparison

| Method | Without reflection | With reflection |
|---|---|---|
| Perez et al. (2012) | 89.34% | **92.01%** |
| Our Method | **90.73%** | 87.49% |

Table 5.8 shows some works from literature with their respective accuracy. Our method achieved competitive recognition rates for this dataset, with a simple approach and with a computational cost lower than methods based on deep learning.

Table 5.8: KTH Comparison

| Method | Accuracy |
|---|---|
| Klaser, Marszałek and Schmid (2008) | 85.30% |
| Wang et al. (2011) | 94.20% |
| Perez et al. (2012) | 92.01% |
| Ravanbakhsh et al. (2015) | **95.60%** |
| Our method | 90.73% |

### 5.2.2 MuHAVi

For the MuHAVi dataset we performed the same experiments as in KTH, in order to find the best values for the size $l$ of the weight vector $\beta$ and for the $\epsilon$ parameter. We evaluate eleven values for $l$: 1, 3, 5, 7, 9, 11, 13, 15, 17, 19 and 21, according to Table 5.9. Notice that the best size for the weight vector $\beta$ is $l = 19$, achieving 89.92%. We can also see that the $\epsilon$ parameter have more variation, being $\epsilon = 100$ the best value when $l = 19$.

Table 5.9: Accuracy for MuHAVi dataset varying $l$ and $\epsilon$ parameters. The triangular weight vector $\beta$ is adjusted according to its size $l$.

| $l$ | $\beta$ | $\epsilon$ | **Accuracy** |
|---|---|---|---|
| 1 | [1] | 1 | 81.51% |
| | | 10 | 86.55% |
| | | 100 | 84.03% |
| 3 | [1, 2, 1] | 1 | 85.71% |
| | | 10 | 85.71% |
| | | 100 | 86.55% |
| 5 | [1, 2, 3, 2, 1] | 1 | 86.55% |
| | | 10 | 87.39% |
| | | 100 | 82.35% |
| 7 | [1, 2, 3, 4, 3, 2, 1] | 1 | 86.55% |
| | | 10 | 88.24% |
| | | 100 | 86.55% |
| 9 | [1, 2, 3, 4, 5, 4, 3, 2, 1] | 1 | 87.39% |
| | | 10 | 87.39% |
| | | 100 | 87.39% |
| 11 | [1, 2, 3, 4, 5, 6, 5, 4, 3, 2, 1] | 1 | 87.39% |
| | | 10 | 85.71% |
| | | 100 | 86.55% |
| 13 | [1, 2, 3, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1] | 1 | 85.71% |
| | | 10 | 86.55% |
| | | 100 | 89.08% |
| 15 | [1, 2, 3, 4, 5, 6, 7, 8, 7, 6, 5, 4, 3, 2, 1] | 1 | 86.55% |
| | | 10 | 85.71% |
| | | 100 | 89.08% |

| | | | |
|---|---|---|---|
| 17 | [1, 2, 3, 4, 5, 6, 7, 8, 9, 8, 7, 6, 5, 4, 3, 2, 1] | 1 | 85.71% |
| | | 10 | 85.71% |
| | | 100 | 89.08% |
| 19 | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1] | 1 | 87.39% |
| | | 10 | 87.39% |
| | | 100 | **89.92%** |
| 21 | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1] | 1 | 88.24% |
| | | 10 | 86.55% |
| | | 100 | 86.55% |

Figure 5.4 contains the confusion matrix for MuHAVi dataset. We can notice that the majority of the actions has 100.00% of accuracy, with only actions with similar movements mixed up, such as *run stop* and *walk turn back*.

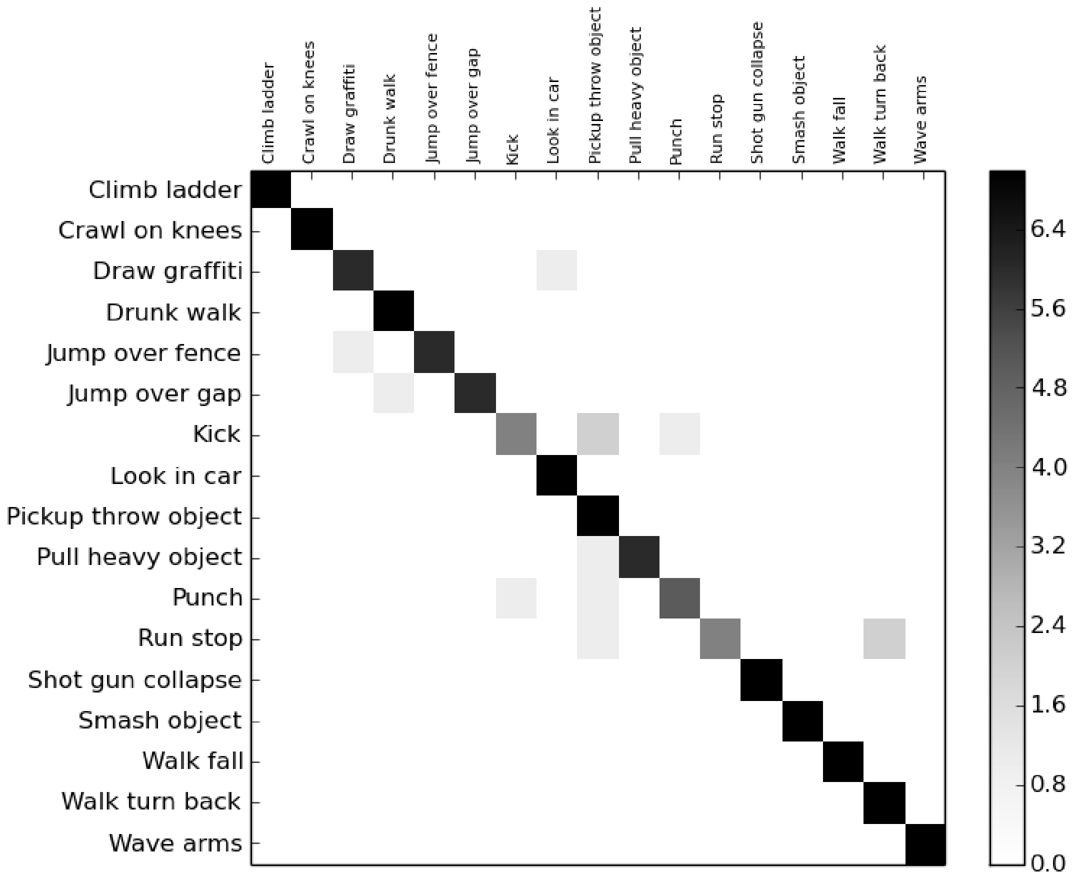

Figure 5.4: Confusion matrix for the best result in MuHAVi dataset. The recognition rate in this case is 89.92%.

In order to evaluate the mean vector for this dataset, we performed experiments using the best parameters previously found and adding reflection information. Table 5.10 shows that for all tests we have no accuracy difference. We believe that it occurs due the number of samples of the MuHAVi dataset be small, causing insignificant impact in this case.

Table 5.10: Comparison between the zero mean-vector and non-zero mean vector for MuHAVi dataset. Experiments were performed using $l = 19$ and $\epsilon = 100$.

| Method | Mean Vector | Accuracy |
|---|---|---|
| Without reflection | zero | 89.92% |
| Without reflection | non-zero | 89.92% |
| With reflection | zero | 89.92% |
| With reflection | non-zero | 89.92% |

Table 5.11 has the recognition rates obtained for MuHAVi dataset using the best parameters found, $l = 19$ and $\epsilon = 100$, and with Lie Algebra and without it. It is clear to see that combining Gaussians through a linear space for this dataset is crucial to obtain good results.

Table 5.11: Lie Algebra influence for MuHAVi dataset. Experiments were performed using $l = 19$ and $\epsilon = 100$.

| Method | Lie Algebra | Accuracy |
|---|---|---|
| Without reflection | yes | 89.92% |
| Without reflection | no | 77.31% |
| With reflection | yes | 89.92% |
| With reflection | no | 77.31% |

Table 5.12 shows some works from literature with their respective accuracy. Although MuHAVi is a multicamera dataset, consisting of 8 cameras in total, we achieved competitive recognition rates using only the information of the camera 4 in comparison with other methods of the literature.

Table 5.12: MuHAVi Comparison

| Method | Accuracy |
|---|---|
| Moghaddam and Piccardi (2010) | 80.40% |
| Karthikeyan et al. (2011) | 88.23% |
| Moghaddam and Piccardi (2014) | 92.00% |
| Alcântara, Moreira and Pedrini (2014) | 89.08% |
| Alcantara et al. (2017) | **92.40%** |
| Our method | 89.92% |

### 5.2.3 SKIG

Similarly to KTH and MuHAVi, we performed experiments in the SKIG varying the size $l$ of the triangular weight vector $\beta$ and $\epsilon$. For $l$, we also evaluate eleven values: 1, 3, 5, 7, 9, 11, 13, 15, 17, 19 and 21 frames and for the parameter $\epsilon$, three values: 1, 10 and 100. Table 5.13 shows that the best size for $\beta$ is $l = 19$. The $\epsilon$ parameter found for the SKIG dataset differs from the KTH and MuHAVi, being 100 the best value found for all variations of $l$. We can see that the weighting yields more improvement for this dataset that for KTH and MuHAVi. It occurs due differences in the movements domain. The hand gestures in the SKIG are smoother and somewhat alike.

Table 5.13: Accuracy for SKIG dataset varying $l$ and $\epsilon$ parameters. The triangular weight vector $\beta$ is adjusted according to its size $l$.

| $l$ | $\beta$ | $\epsilon$ | **Accuracy** |
|---|---|---|---|
| 1 | [1] | 1 | 67.22% |
| | | 10 | 71.11% |
| | | 100 | 75.56% |
| 3 | [1, 2, 1] | 1 | 81.57% |
| | | 10 | 80.83% |
| | | 100 | 84.44% |
| 5 | [1, 2, 3, 2, 1] | 1 | 81.39% |
| | | 10 | 82.50% |
| | | 100 | 84.72% |
| 7 | [1, 2, 3, 4, 3, 2, 1] | 1 | 81.67% |
| | | 10 | 82.78% |
| | | 100 | 85.00% |
| 9 | [1, 2, 3, 4, 5, 4, 3, 2, 1] | 1 | 83.06% |
| | | 10 | 83.61% |
| | | 100 | 84.72% |
| 11 | [1, 2, 3, 4, 5, 6, 5, 4, 3, 2, 1] | 1 | 84.72% |
| | | 10 | 85.00% |
| | | 100 | 86.39% |
| 13 | [1, 2, 3, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1] | 1 | 85.28% |
| | | 10 | 86.39% |
| | | 100 | 86.94% |
| 15 | [1, 2, 3, 4, 5, 6, 7, 8, 7, 6, 5, 4, 3, 2, 1] | 1 | 86.11% |
| | | 10 | 86.94% |
| | | 100 | 87.22% |
| 17 | [1, 2, 3, 4, 5, 6, 7, 8, 9, 8, 7, 6, 5, 4, 3, 2, 1] | 1 | 86.67% |
| | | 10 | 87.22% |
| | | 100 | 86.94% |

| | | | |
|---|---|---|---|
| 19 | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1] | 1 | 86.67% |
| | | 10 | 86.67% |
| | | 100 | **87.50%** |
| 21 | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1] | 1 | 86.94% |
| | | 10 | 85.93% |
| | | 100 | 86.94% |

Table 5.14 has the confusion matrix for the best result for SKIG dataset. The action $up - down$ is the most confusing. It occurs because in the actions $up - down$ and *pat* are quite similar, with all movement occurring in small regions in the middle of the frame.

Table 5.14: Confusion Matrix of the SKIG dataset without reflection. The recognition rate is 87.50%.

| | Circle | Triangle | UpDown | RightLeft | Wave | "Z" | Cross | Come | Turn | Pat |
|---|---|---|---|---|---|---|---|---|---|---|
| Circle | **88.89** | 0.00 | 11.11 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Triangle | 0.00 | **97.22** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 2.78 |
| UpDown | 0.00 | 0.00 | **50.00** | 0.00 | 0.00 | 0.00 | 0.00 | 2.78 | 0.00 | 47.22 |
| RightLeft | 0.00 | 0.00 | 0.00 | **97.22** | 0.00 | 0.00 | 0.00 | 0.00 | 2.78 | 0.00 |
| Wave | 2.78 | 0.00 | 0.00 | 0.00 | **97.22** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| "Z" | 8.33 | 0.00 | 0.00 | 0.00 | 0.00 | **86.11** | 5.56 | 0.00 | 0.00 | 0.00 |
| Cross | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **100.00** | 0.00 | 0.00 | 0.00 |
| ComeHere | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **83.33** | 0.00 | 16.67 |
| TurnAround | 0.00 | 0.00 | 2.78 | 0.00 | 0.00 | 0.00 | 0.00 | 11.11 | **83.33** | 2.78 |
| Pat | 0.00 | 0.00 | 5.56 | 0.00 | 0.00 | 0.00 | 0.00 | 2.78 | 0.00 | **91.67** |

Table 5.15 shows the recognition rates obtained for the experiments performed in order to evaluate the impact of the mean vector for recognition. We can see that, different from KTH and MuHAVi, in this case, the non-zero mean vector yields some improvement for the recognition. We believe that it occurs due to the movements domain, and, unlike KTH, Gaussians with non-zero mean vector are more suitable when the dataset contains more similar actions, with a considerable number of actions performed in small regions of the frame.

Table 5.15: Comparison between the zero mean-vector and non-zero mean vector for SKIG dataset. Experiments were performed using $l = 19$ and $\epsilon = 100$.

| Method | Mean Vector | Accuracy |
|---|---|---|
| Without reflection | zero | 86.94% |
| Without reflection | non-zero | 87.50% |
| With reflection | zero | 86.94% |
| With reflection | non-zero | 87.50% |

We can see in Table 5.16 the results obtained to the performed experiments in order to evaluate the Lie Algebra influence to combine Gaussians. We can notice that as well as in KTH and MuHAVi, combining Gaussians through a linear space yield some improvement, as expected.

Table 5.16: Lie Algebra influence for SKIG dataset. Experiments were performed using $l = 19$ and $\epsilon = 100$.

| Method | Lie Algebra | Accuracy |
|---|---|---|
| Without reflection | yes | 87.50% |
| Without reflection | no | 86.67% |
| With reflection | yes | 87.50% |
| With reflection | no | 86.67% |

Table 5.17 shows some works from literature with their respective accuracy. We can see that our method not outperforms methods based on deep neural networks, but achieves good recognition rates with a simpler approach, with a computational cost inferior to the cost of the deep methods.

Table 5.17: SKIG Comparison

| Method | Accuracy |
|---|---|
| Liu and Shao (2013) | 84.60% |
| Nishida and Nakayama (2015) | 91.60% |
| Li, Zhang and Jin (2017) | **96.70%** |
| Our method | 87.50% |

# 6 Conclusion

In this work, we presented a new motion descriptor based on Gaussians for human action recognition. The main objective of this work was to create a video descriptor based on a multivariate Gaussian distribution that was capable of describing human actions. In order to evaluate the impact of the mean vector of the Gaussian to recognition, we performed experiments to verify if a mean vector different of zero would be capable of improving the recognition and how much would be possible. To this purpose, we used Lie Algebra aiming to combine Gaussians preserving the algebraic and topological structure of the spaces involved. Analyzing our results, we notice that the non-zero mean vector can yield some improvement in some cases and that combining Gaussians through a linear space is more appropriate than combining Gaussians through the space of Gaussians.

We evaluate our method through three datasets: KTH, MuHAVi and SKIG, in two scenarios: with and without horizontal reflection. We also evaluate some combinations of the parameters $\epsilon$ and $l$, for each dataset. For KTH, we found $\epsilon = 10$ and $l = 13$, achieving 90.73% of accuracy using the dataset without reflection and 87.49% adding reflection. Experiments evaluating the mean vector showed that for KTH is more appropriate to use Gaussians with zero mean vector. Although we do not outperform the best result obtained by Perez et al. (2012), we outperform when we consider only without reflection case, indicating that Gaussians are capable of improving the recognition. For MuHAVi dataset, with $\epsilon = 100$ and $l = 19$ we achieved 89.92% of accuracy. For this dataset the mean vector and the reflection had no difference, we believe that it occurred due to the number of samples. For SKIG we achieved 87.50% with $\epsilon = 100$ and $l = 19$. The weighting, in this case, yielded more improvement that for KTH and MuHAVi datasets. It occurred due differences in the movements domain since the hand gestures in the SKIG are smoother and somewhat alike. For MuHAVi and SKIG datasets the horizontal reflection information not added anything to the recognition, whereas for KTH it was worse.

Although deep neural networks have demonstrate impressive results, they have a high computational cost, mainly in the training step. Our method achieved competitive recognition rates for all evaluated datasets, with a simple approach and less costly that deep methods.

As future works we intend to include the IE-logE embedding method in order to compare how good it is performance, if it presents some sort of improvement comparing with the DE-LogE method. We can also use this method for local approaches, for example, estimating local Gaussians for a video.

# Bibliography

AGGARWAL, J. K.; RYOO, M. S. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, ACM, v. 43, n. 3, p. 16, 2011.

ALCÂNTARA, M. F.; MOREIRA, T. P.; PEDRINI, H. Real-time action recognition based on cumulative motion shapes. In: IEEE. *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on.* [S.l.], 2014. p. 2917–2921.

ALCANTARA, M. F. de et al. Action identification using a descriptor with autonomous fragments in a multilevel prediction scheme. *Signal, Image and Video Processing*, Springer, v. 11, n. 2, p. 325–332, 2017.

BLANK, M. et al. Actions as space-time shapes. In: IEEE. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on.* [S.l.], 2005. v. 2, p. 1395–1402.

BOBICK, A. F.; DAVIS, J. W. The recognition of human movement using temporal templates. *IEEE Transactions on pattern analysis and machine intelligence*, IEEE, v. 23, n. 3, p. 257–267, 2001.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, Springer, v. 20, n. 3, p. 273–297, 1995.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: IEEE. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on.* [S.l.], 2005. v. 1, p. 886–893.

DALAL, N.; TRIGGS, B.; SCHMID, C. Human detection using oriented histograms of flow and appearance. In: SPRINGER. *European conference on computer vision.* [S.l.], 2006. p. 428–441.

FIGUEIREDO, A. M. de O. et al. A video self-descriptor based on sparse trajectory clustering. In: SPRINGER. *International Conference on Computational Science and Its Applications.* [S.l.], 2016. p. 571–583.

JI, S. et al. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 35, n. 1, p. 221–231, 2013.

KARTHIKEYAN, S. et al. Probabilistic subspace-based learning of shape dynamics modes for multi-view action recognition. In: IEEE. *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on.* [S.l.], 2011. p. 1282–1286.

KLASER, A.; MARSZAŁEK, M.; SCHMID, C. A spatio-temporal descriptor based on 3d-gradients. In: BRITISH MACHINE VISION ASSOCIATION. *BMVC 2008-19th British Machine Vision Conference.* [S.l.], 2008. p. 275–1.

LAPTEV, I. et al. Learning realistic human actions from movies. In: IEEE. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on.* [S.l.], 2008. p. 1–8.

LI, C.; ZHANG, X.; JIN, L. Lpsnet: A novel log path signature feature based hand gesture recognition framework. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2017. p. 631–639.

LI, P. et al. Local log-euclidean multivariate gaussian descriptor and its application to image classification. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 39, n. 4, p. 803–817, 2017.

LIU, L.; SHAO, L. Learning discriminative representations from rgb-d video data. In: *IJCAI*. [S.l.: s.n.], 2013. v. 4, p. 8.

LOWE, D. G. Object recognition from local scale-invariant features. In: IEEE. *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. [S.l.], 1999. v. 2, p. 1150–1157.

MAIA, H. A. et al. A video tensor self-descriptor based on variable size block matching. *Journal of Mobile Multimedia*, Rinton Press, Incorporated, v. 11, n. 1&2, p. 090–102, 2015.

MOGHADDAM, Z.; PICCARDI, M. Histogram-based training initialisation of hidden markov models for human action recognition. In: IEEE. *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*. [S.l.], 2010. p. 256–261.

MOGHADDAM, Z.; PICCARDI, M. Training initialization of hidden markov models in human action recognition. *IEEE Transactions on Automation Science and Engineering*, IEEE, v. 11, n. 2, p. 394–408, 2014.

MOTA, V. F. et al. A tensor motion descriptor based on histograms of gradients and optical flow. *Pattern Recognition Letters*, Elsevier, v. 39, p. 85–91, 2014.

MOTA, V. F. et al. Combining orientation tensors for human action recognition. In: IEEE. *Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI-Conference on*. [S.l.], 2013. p. 328–333.

NGUYEN, A.; YOSINSKI, J.; CLUNE, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2015. p. 427–436.

NISHIDA, N.; NAKAYAMA, H. Multimodal gesture recognition using multi-stream recurrent neural network. In: SPRINGER. *Pacific-Rim Symposium on Image and Video Technology*. [S.l.], 2015. p. 682–694.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

PEREZ, E. A. et al. Combining gradient histograms using orientation tensors for human action recognition. In: IEEE. *Pattern Recognition (ICPR), 2012 21st International Conference on*. [S.l.], 2012. p. 3460–3463.

POLANA, R.; NELSON, R. Low level recognition of human motion (or how to get your man without finding his body parts). In: IEEE. *Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on*. [S.l.], 1994. p. 77–82.

POLANA, R.; NELSON, R. C. Qualitative recognition of motion using temporal texture. *CVGIP: Image understanding*, Elsevier, v. 56, n. 1, p. 78–89, 1992.

RAVANBAKHSH, M. et al. Action recognition with image based cnn features. *arXiv preprint arXiv:1512.03980*, 2015.

SCHULDT, C.; LAPTEV, I.; CAPUTO, B. Recognizing human actions: a local svm approach. In: IEEE. *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on.* [S.l.], 2004. v. 3, p. 32–36.

SCOVANNER, P.; ALI, S.; SHAH, M. A 3-dimensional sift descriptor and its application to action recognition. In: ACM. *Proceedings of the 15th ACM international conference on Multimedia.* [S.l.], 2007. p. 357–360.

SIMONYAN, K.; ZISSERMAN, A. Two-stream convolutional networks for action recognition in videos. In: *Advances in neural information processing systems.* [S.l.: s.n.], 2014. p. 568–576.

SINGH, S.; VELASTIN, S. A.; RAGHEB, H. Muhavi: A multicamera human action video dataset for the evaluation of action recognition methods. In: IEEE. *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on.* [S.l.], 2010. p. 48–55.

WANG, H. et al. Action recognition by dense trajectories. In: IEEE. *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on.* [S.l.], 2011. p. 3169–3176.

WANG, L.; QIAO, Y.; TANG, X. Action recognition with trajectory-pooled deep-convolutional descriptors. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* [S.l.: s.n.], 2015. p. 4305–4314.