Università degli Studi di Genova

Robotics, Brain and Cognitive Sciences Department,
Istituto Italiano di Tecnologia

A thesis submitted in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY

Learning visual cues of interaction for humanoid robots

Doctoral Course in Cognitive Robotics, Interaction and Rehabilitation Technologies
Doctoral Program in Bioengineering and Robotics

Author:
Alessia Vignolo

Supervisors:
Prof. Francesca Odone
Prof. Giulio Sandini
Dr. Alessandra Sciutti

February 23, 2018

Abstract

One of the fundamental skills supporting safe and comfortable interaction between humans is their capability to understand intuitively each other's actions and intentions. At the basis of this ability is a special-purpose visual processing that human brain has developed to comprehend human motion. Among the first "building blocks" enabling the bootstrapping of such visual processing is the ability to detect movements performed by biological agents in the scene, a skill mastered by human babies in the first days of their life. After, they refine the ability to understand actions until they get to the point of being able to interact in a correct way.

In this thesis we present computational models based on the assumption that such visual abilities must be based on local low-level visual motion features which are independent of shape, such as the configuration of the body, and perspective. We first design a computational model to detect biological motion in the scene and we implement it on the humanoid robot iCub, embedding it into a software architecture that leverages the regularities of biological motion also to control robot attention and oculo-motor behaviors. In essence, we put forth a model in which the regularities of biological motion link perception and action enabling a robotic agent to follow a human-inspired sensory-motor behavior. We then take a step forward towards action understanding, by building a system that can segment actions into motion primitives and use them to individuate similarities among different actions and among different visual perspectives.

As a result we propose a computational model of the perceptual primitives supporting infants' social skills development, designed to be implemented on a robotic platform in order to facilitate mutual understanding, safety and goal prediction during human-robot interaction.

Contents

Chapter 1

Introduction

Goals. Robots are progressively entering our houses: robotic devices as vacuum cleaners, pool cleaners and lawn mowers are becoming more and more commonly used and the growth of robotics in the consumer sector is expected to continuously increase in the near future[1]. The fields of applications for robotics will influence not only domestic activities, but also entertainment, education, monitoring, security and assistive living, leading robots to frequent interactions with untrained humans in unstructured environments. The success of the integration of robots in our everyday life is subordinated to the acceptance of these novel tools by the population. The level of comfort and safety experienced by the users during the interaction plays a fundamental role in this process. A key challenge in current robotics has then become to maximize the naturalness of human-robot interaction (HRI), to foster a pleasant collaboration with potential non-expert users. To this aim, a promising avenue seems to be endowing robots with a certain degree of social intelligence, to enable them to behave appropriately in human environments.

In order to design robots that interact in a natural way, it is important to endow them with functionalities that are typical of humans. In this context, *human infants* represent an important source of inspiration. Indeed, even if endowed with limited sensory-motor capabilities and no explicit knowledge of social norms, infants can already quite proficiently coordinate with their peers [2] and caregivers [84], even in absence of language. Moreover, from the restricted social abilities exhibited in the very first months of life, humans are able to develop a full fledged social competence in adulthood. The partial skills exhibited by a baby can therefore represent the minimum set of abilities necessary to enable the bootstrapping of more complex interactive expertise. Endowing robots with analogous "social building blocks" represents therefore the starting point in the attempt to replicate complex HRI skills, favoring the establishment of a simple yet efficient intuitive understanding in the naive user. A long-

[1] EU Strategic Road Map 2014-2020

term goal is to make the interaction between robots and humans more natural, providing the robot with the capability of moving and perceiving in a human-like fashion. As a consequence, during the interaction, in the human partner the same perceptual mechanisms, which are unconsciously and naturally activated during a human-to-human interaction, would be activated.

In particular, human neonates show a natural predisposition towards *biological motion* [80]: despite the limited visual information available to them, they can perceive the presence of other humans moving near them. Since infancy, indeed, vision represents a fundamental sense for interaction, and this particular predisposition is one of the keys for developing a good capability in the human-to-human interaction. The observation of actions of our partners, as reaching, grasping etc, can be enough to deduce what will be their next move and to adapt accordingly the subsequent motion to interact and collaborate. Hence, an agent apparently simple and with limited visual capabilities has already "in nuce" all the skills needed to allow for the development of the complex action understanding abilities proper of human adults.

In this context, the objective of the thesis is to study, design and validate *computational models* of the perceptual primitives which are important for the development of social skills and motion understanding ability in infants.
The scope of the thesis is highly multidisciplinary, as it addresses a developmental robotics task inspired by studies of infant cognitive development and aimed at implementing models, based on computer vision and machine learning techniques, on a complex engineering system such as a humanoid robot. The reference architecture of the thesis is the iCub humanoid platform [52], where the designed vision models have been implemented. This choice allows for a systematic evaluation of the proposed solution and their use for natural human-robot interaction.

State of art. The reference field of this work is **developmental robotics**, a collaborative and interdisciplinary approach to robotics that is directly inspired by the developmental principles and mechanisms observed in children's cognitive development. It builds on the idea that the robot, using a set of intrinsic developmental principles regulating the real-time interaction of its body, brain, and environment, can autonomously acquire an increasingly complex set of sensorimotor and mental capabilities [15].
Between all the capabilities, we focus on the **motion perception** one. In particular, biological motion plays an important role in human perception: stimuli that follow biological kinematics are processed by specific areas in the brain [86] and are easier to be anticipated during human–human interaction [67, 23]. Conversely, movements that do not comply with

biological motion rules are likely to be misperceived, both visually and proprioceptively [33, 92]. Similarly, in the context of human–robot interaction it has been suggested that the lack of biological plausibility in the motion of a humanoid robot could lead to a sense of eeriness and disgust, precluding the possibility for a natural interaction [16] and reducing the coordination with the partner [4]. Another peculiarity of biological motion perception is its precocity. The ability to discriminate biological motion from non-biological motion in humans is in fact present from birth, together with a natural propensity to orient attention toward biologically moving stimuli [80].
All these considerations suggest that biological motion understanding represents one of the basic perceptual properties that support the development of human social interaction skills.

Biological motion understanding is inherently linked to the topic of action analysis. Several are the approaches that have been adopted so far to understand human activities in the fields of **computer vision**. One option is the use of RGBD sensors [43, 82]: the additional information carried by the depth channel gives the opportunity to enrich the process of extracting the body structure and robustly interpret human activity. Potential problems for these systems occur when the visibility of the partner is limited, for instance due to occlusions or because the agent is partially out of the scene. In these cases, the difficulty in matching the 3D input with the human skeleton might limit the efficacy of this approach, making it less appropriate for cluttered environments. Another option is the use of RGB sensors [104, 21].

Despite the amount of work in computer vision [108, 1, 66], it is often not possible to implement directly these methods in **robotics**, due to the specific constraints of the robotic platform. For instance high resolution images are ideally needed to facilitate human activity recognition, while robots are in general equipped with relatively lower resolution cameras, in order not to overload their network, usually prioritized for real-time behavior, as locomotion. Moreover, in robotics it is often not possible to adopt any approximation usually viable for fixed cameras (e.g., a stable background) as they should move in their environment. Last, for robotics aimed at interaction it is imperative that the processing is real-time, even at the expenses of precision. Indeed, a non-perfect but rapid evaluation could in general trigger a robot exploration action, enabling also a rapid correction in case of error. On the contrary a slow scene processing hinders completely the possibility for an interaction. These features bring strong constraints on the video analysis approaches appropriate for interactive robotics, and suitable solutions have to be adopted [25].

Main contributions of the thesis. In this thesis we take inspiration from developmental science to design the principles of a hierarchical framework replicating the developmental stages of human visual perception and supporting social intelligence (see Table (1.1) which is a part of a table extracted from [87]). The Table (1.1) highlights how infants in the first stages analyze the movement per se (i.e. gaze and biological motion detection) while in the following ones perceive the movements as actions (i.e. recognition of the pointing, goal understanding, etc.). The sentences in bold are the ones explicitely related to the development of motion perception abilities.

Innate skills	
Newborns	Newborns gaze longer when the person looks directly at them [28]
Newborns	**Newborns prefer biological motion [80]**
Newborns	Newborns are attracted to people (i.e face/voice) [29]
Early Development	
3 months	Infants engage mutual gaze with adults, i.e. both agents attend to each other's eyes simultaneously [42]
6 months	Infants can perceive approximate direction of attention of others (i.e. to the left or to the right) [14]
9 months	Infants can accurately detect the direction of the adult's gaze [42]
12 months	**Infants start to understand pointing as an object-directed action [105]**
12 months	Infants anticipate with gaze the goal of a feeding action [34]
Later development	
18 months	Children start to follow adults' gaze outside their own field of view [42]
18 months	**Children can infer what another person is trying to achieve, even if the attempt is unsuccesful [51, 3]**
18 months	Children start to follow adults' gaze outside their own field of view [42]
18 months	**Children altruistically (instrumentally) help adults when they are having problems in achieving a goal [103]**

Table 1.1: Infant abilities list.

Taking inspiration from this, it is possible to derive a hierarchical framework composed by different steps (represented in the flow of Figure 1.1): biological motion recognition; identification of groups of similar actions (or action categorization); action recognition.

In the thesis, we contribute to the development of this framework considering the following points:

- **Biological and non-biological motion models.** Inspired by neonates that have this capability, we build a computational model to distinguish between biological and non-biological motion by exploiting low-level motion features. In the experimental analysis we also show how the presence of biological movements can be derived by very partial observations – in extreme cases just by observing a tool manually operated by

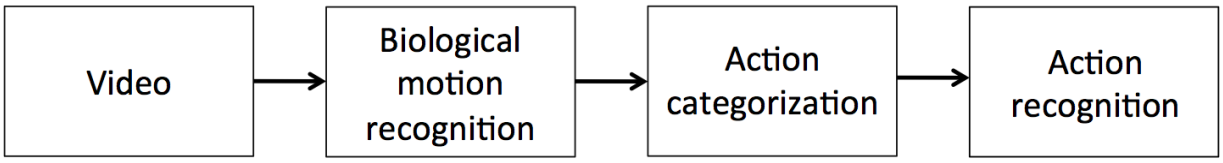


Figure 1.1: Flow for action understanding.

a user or, in principle, the movements of a human shadow, where other methods of human detection would fail.

- **Implementation of the biological motion detection on iCub.** We then implement the detector of biological movements on the robot iCub.We showed how the detector reliably discriminates between biological and non-biological motion with a good response rate and makes the robot direct the attention towards the biological motion. Such a functionality, achieved by a neat procedure and without the need for a time-consuming interpretation of the scene, was missing in the iCub system and is original with respect to the state of the art.

- **Action analysis.** By building on this capability of recognizing biological motion as proxy for the localization of interactive partners, we then focus on the capability of understanding classes of actions in order to prepare the interaction. By leveraging on motion primitives, a well-known concept of motor control, we build a system that can categorize the actions based on a representation of them as a combination of the motion primitives automatically discovered by data.

Structure of the document. The rest of the document is organized as follows:

- Chapter 2 is devoted to present in details our method to discriminate between biological and non-biological motion, followed by an extensive experimental analysis;

- Chapter 3 is dedicated to present the iCub architectural framework that hosts our method of biological motion detection on the robot and to show both the results produced on the method while working online and the effect on the robot action;

- Chapter 4 contains the details of our method for representing actions as a suitable combination of motion primitives, previously discovered from data, and an experimental session on action recognition first carried out on a single view point and then with an intra-view analysis;

- Chapter 5 is finally left to a discussion on possible future outcomes.

Chapter 2

Biological motion

2.1 Introduction

In this chapter, we consider the natural predisposition of newborns to notice potential interacting partners in their surroundings, which is manifested by a preference for biological motion [80] and for faces looking directly to them [27] over other visual stimuli. Interaction in its simplest form seems therefore constituted by a sensitivity to some properties of others' motion and to their direction of attention.

Drawing inspiration from these observations, we propose a video-based computational method for biological motion detection, which will be next implemented on the humanoid robot iCub [52], to guide robot attention toward potential interacting partners in the scene. We focus on a method purely based on motion, which does not require any a priori knowledge of human shape or skeleton, nor detecting faces and hands [12, 32].

In essence, we put forth a model in which the regularities of biological motion link perception and action enabling a robotic agent to follow a human-inspired sensory-motor behavior. This way, we address two fundamental components necessary to facilitate the understanding of robots by human users:

1. On the *perception* side, we make the robot *find the same types of stimuli salient* as a human (e.g., [10]). In particular, we propose a computational tool to make the robot sensitive to human activity, a very relevant type of motion for human observers.

2. On the *action* side, we enable the robot to *direct its attention to human activity* through a biologically-inspired oculo-motor mechanism [9]. This way the robot can reorient its gaze towards where the human partners are acting. Such eye shift can also represent an intuitive form of communication, revealing where the robot is focusing and potentially informing the human partner of its availability to interact [63].

The use of a common, biologically-inspired, perceptual and motor framework facilitates the human partner's understanding and prediction of the future actions of its robot counterpart.

To design a system sensitive to the regularities typical of biological movements we draw inspiration from the laws governing human motor control. We consider in particular the *Two-Thirds Power Law*, since there is an evidence that human neonates are sensitive to it since the first days after birth [50]. The law is a well-known invariant of human movements [97, 92, 88, 72] describing the regular relationship between the instantaneous tangential velocity and the radius of curvature of human end-point movements [35, 98, 46]. There have been experimental evidences, particularly for handwriting [46, 99], that in biological movements velocity and curvature show a strong mutual influence. The low-level motion descriptor we adopt, based on the same dynamic features, is meant to capture such connections even on complex and noisy data. These motion features are particularly important in motion perception of infants as they are close to the motionese features that have been discussed in [7, 8, 58, 100]: humans, when interact with children, exaggerate their movements to make them more legible by their interacting partners, modifying, among others, the velocity and the curvature of the movements.

The goal of this section is, then, the analysis of video sequences in order to discriminate between biological and non-biological motion, adopting a motion representation inspired by the Two-Thirds Power Law. Our contribution is the application of the Two-Thirds Power law, that has been always studied in the context of motor control, in the field of video analysis for the first time, and in more general contexts (not just concerning planar movements but 3D movements too).

To handle the wide intra-class variability of biological stimuli, we propose the use of a structured motion descriptor that accounts for multiple temporal resolutions of the measurements. A careful, automatic selection of such resolutions allows us to easily adapt our model to a variety of scenarios.

We test the method on a wide set of variations including different sensors, points of view, types of behaviors and dynamics. In particular, its efficacy in generalizing to new scenarios, including scene observation from different visual perspectives and in the presence of severe occlusions, is demonstrated.

The methods have been presented in [89, 90, 91, 59].

2.2 State of art

Several are the approaches that have been adopted so far to *perceive* and *detect* the presence of human activities. In the following we discuss state-of-the-art methods grouped by categories according to the type of sensor and/or type of information used, while enhancing the novelties of our approach.

One potential approach to detect humans is to endow robots of **specific sensors** such as RFID or thermal sensors (see, for instance, [18]). In spite of high performances, this solution requires ad-hoc hardware, usually not available in common robotics platform, limiting the range of possible scenarios. Their relatively high cost is another factor that may harm a large-scale diffusion – which is however desirable for future family or companion robots.

For the reasons above, we focus here on approaches based on more traditional **RGB and depth sensors**. Although the proliferating of works in the computer vision community, the constraints and limitations of robotics settings make it difficult to directly employ methods successfully applied to other domains. Robots are in general equipped with relatively low resolution cameras, in order not to overload their network, while standard computer vision approaches may rely on high resolution images. Moreover, interactive robots require a fast processing to support interaction: a perfect classification performance becomes useless if it is achieved not rapidly enough to enable appropriate robot reaction. In this respect, the speed-precision trade-off in HRI is often unbalanced toward speed, as a rapid, yet not precise estimation still allows the robot to continue the collaboration, while adjustment of the initial guess may always be achieved exploiting the evolution of the interaction itself.
With these constraints in mind, we cite here examples of use of RGBD sensors [43, 82], promoted in recent years by the widespread availability of low-cost, highly-portable sensors. This approach provides a richer information on the body structure, helping the understanding of the performed activity, but to the price of low success when the visibility of the partner is limited and it is not possible to match the 3D input with the human skeleton.

More related to our work is a third category of approaches, based on the analysis of 2D video signals acquired with the **RGB sensors** of the robot cameras [104, 21].

Most 2D video analysis methods for human detection currently adopted in robotics rely on appearance or shape features, for instance detecting faces and hands in the scene [12, 32]. However, these approaches have severe limitations as scene complexity grows, for instance when the clutter in the environment increases or the light conditions become more challenging. Shape-based or part-based methods are likely to fail when the human body is only partially visible – as in presence of occlusions – while detectors based on faces are not appropriate for

close interaction scenarios, as those involving precise manipulation on a tabletop.

Although still based on 2D signals, our approach substantially differs from previous works, as we strictly focus only on the motion properties of the stimuli. A purely motion-based human detection system makes it possible to detect the presence of humans in the vicinity just by observing the effects of their behaviour on the environment, as for instance, the movement of the manipulated tool – a use-case that to the best of our knowledge has not been considered so far in the related literature. Note that, while motion detection is common in robotics applications, oftentimes as a preliminary step for further analysis, *human detection through motion* requires a selectivity to biological motion, which is usually absent in common robotic systems.

There is wide evidence that humans are better at predicting stimuli moving according to biological motion; whereas they present a distorted perception when behaviors subvert these kinematics rules [33, 67, 23, 92]. Also in the specific context of HRI, it has been demonstrated that the adoption of biological plausible motion by a humanoid robot can lead to a more natural coordination with its actions [4] and potentially to a more pleasant interaction (see [76]). Conversely, the execution of non-biological motion by a humanoid robot has been suggested as a possible cause for the Uncanny Valley effect [55], i.e., to the occurrence of a sense of eeriness and disgust toward the robot, precluding the possibility for a natural interaction [16]. Human-like motion benefits interaction also when it is applied to gaze behavior, for instance facilitating the regulation of conversations (e.g. [56]), the coordination of shared plans in collaboration [6] and the prediction of robot goals (e.g. [70]). Drawing inspiration from these evidence, to maximize the efficacy of the human activity detection module, our proposed architecture leverages the regularities of biological motion also for the preparation and execution of the robot saccadic action. This way, the robotic oculo-motor action triggered by the perception module informs the human partner in an intuitive way about the internal attentional status of the robot.

2.3 The 2/3 Power Law

In order to design a model able to detect the regularities of biological motion, we identified some low-level motion features, whose co-variation characterise human movements. To identify them, we took inspiration from the Two-Thirds Power Law, and we derive the computational counterpart of the analytical quantities involved in the law. In this section, we present a review of the law.

Given a planar point which moves along a trajectory over time, its movement can be described considering the geometric *shape of the trajectory* and the *law of motion*, which controls how the position of the point varies. A motion which is associated with a physical event, is always the consequence of their mutual influence. In the case of our interest, that is the human motion, the central nervous system at the peripheral musculoskeletal plant constrains a movement to exhibit a set of invariant features, which put in relationship the shape with the kinematics [35, 98, 46]. A principle that governs human motion is, for instance, the *isogony principle*: it states that humans tend to cover equal angles in the same amount of time, independently of the arc length of the spatial trajectory, putting in a strong relationship the velocity profile with the curvature of human motion.

These statements have been formalised in the so-called *Two-Thirds Power Law* [98, 46], which can be formulates as

$$V(t) = K(t) \left(\frac{R(t)}{1 + \alpha R(t)} \right)^{\beta} \tag{2.1}$$

where $V(t)$ is the tangential velocity at time t and $R(t)$ the radius of curvature at the same instant t. Experimentally, it has been observed that β exponents are close to the empirical value of $\frac{1}{3}$ for a large class of human motion, but in particular for 2D handwritings [98, 46, 96]: this fact is a manifestation of the regularity of human motion.

The other two quantities involved in the law are $\alpha \in [0, 1]$ and $K(t) \geq 0$. The first is equal to zero when the trajectory does not present points of inflection, otherwise it depends on the average velocity of the motion. The latter, also known as *velocity gain factor*, depends on tempo and length of the motion [95, 94]. Even if the role of $K(t)$ is still not perfectly clear, in [45, 93] it has been shown that its value is constant for long segments of the trajectories while it changes usually where there are points of inflections of junctions.

In case $\alpha = 0$ the law can be simplified as $V(t) = K(t)R(t)^{\beta}$ from which we can derive the alternative formulation $A(t) = K(t)C(t)^{1-\beta}$ where $A(t) = \frac{V(t)}{R(t)}$ is the angular velocity and $C(t) = \frac{1}{R(t)}$ is the curvature.

In [93] the authors mathematically prove that when the form of the trajectory to be followed is elliptical, the sequence of human-compliant positions must be generated by harmonic functions with the same frequency, which is the mode of production humans spontaneously select for drawing movements, also known as *Lissajous* model of trajectory production [96].

If we consider ellipsoidal shapes with the center in the origin of a cartesian coordinates system, and the main axis laying on the x and y axis of the system, the shapes can be

mathematically described in terms of the parametric equations

$$\begin{cases} x(t) = A\cos(\Phi(t)) \\ y(t) = B\sin(\Phi(t)) \end{cases} \tag{2.2}$$

which depend on the length A and B of the main axis, and on $\Phi(t)$, representing the angular parameter (which here acts as the *law of the motion*) as a function of the time instant t.

The spatial derivatives of the positions (i.e. the ideal velocity) can be analytically computed as

$$\begin{cases} V_x(t) = -A\sin(\Phi(t))\Phi'(t) \\ V_y(t) = B\cos(\Phi(t))\Phi'(t) \end{cases} \tag{2.3}$$

while the spatial curvature is

$$C(t) = \frac{1}{A^2B^2}\left(\frac{x(t)^2}{A^4} + \frac{y(t)^2}{B^4}\right)^{-\frac{3}{2}}. \tag{2.4}$$

A *Lissajous* model can be easily obtained by discretizing the trajectory using equally-spaced angles $\Phi(t) \in [0, 2\pi]$, using a sampling step Δ: at time t, $\Phi(t) = t\Delta$, with $t = \{0, 1, \ldots, N\}$ and $N = \frac{2\pi}{\Delta}$.

In the next section, we start by discussing the computational derivation of the motion features inspired by the law.

2.4 A temporal multi-resolution biological motion descriptor

In this section we start with a brief summary of an instantaneous motion description we adopt as a building block for our method [61]. Then, we review the proposed multi-resolution method [90] which efficiently combines measurements that may span different temporal portions of an image sequence.

2.4.1 Instantaneous motion representation

Taking inspiration from the Two-Thirds Power Law, we can derive a set of motion features which empirically estimate the analytical quantities related by it. We do not use directly the law but features inspired by it as our data (which are visual data) are more complicated and noisy than the data used traditionally to evaluate the law (that are motoric data). We report in Figure 2.1 the key steps of our low-level layer of motion representation starting

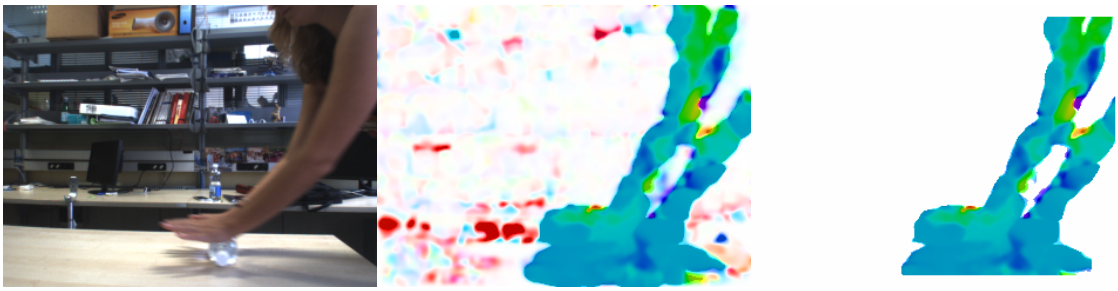
from video sequences. At each time instant t, the *optical flow* (that is the pattern of the apparent motion of image objects between two consecutive frames caused by the movement of the object) is computed using a dense approach [26] which provides an estimate of the apparent motion vector in each image point (Figure 2.1b). The optical flow magnitude is thresholded to enhance locations with significant motion. Isolated pixels and small regions, which are likely to be generated by noise, are rejected by first applying a *perceptual grouping* – in which only locations whose neighbouring pixels are also marked as moving are kept in the analysis – and then discarding small groups. We then obtain a *motion map* whose largest connected component (henceforth referred to as $\mathcal{R}(t)$) becomes the candidate region for motion recognition (Figure 2.1c), under the assumption that only a single interesting source of motion is observed in the scene at each time instant. Let $(u_i(t), v_i(t))$ be the optical flow components associated with point $\mathbf{p}_i(t) \in \mathcal{R}(t)$, and N the size of the region, i.e. the number of pixels in it. We compute a set of motion features, according to the formulations in Table (2.1), which empirically estimate the analytical quantities related by the Two-Thirds Power Law. We finally describe the region $\mathcal{R}(t)$ with a feature vector $\mathbf{x}_t \in \mathbb{R}^4$ by averaging the features over all the region elements:

$$\mathbf{x}_t = \frac{1}{N}\left[\sum_i \hat{V}_i(t), \sum_i \hat{C}_i(t), \sum_i \hat{R}_i(t), \sum_i \hat{A}_i(t)\right] \tag{2.5}$$

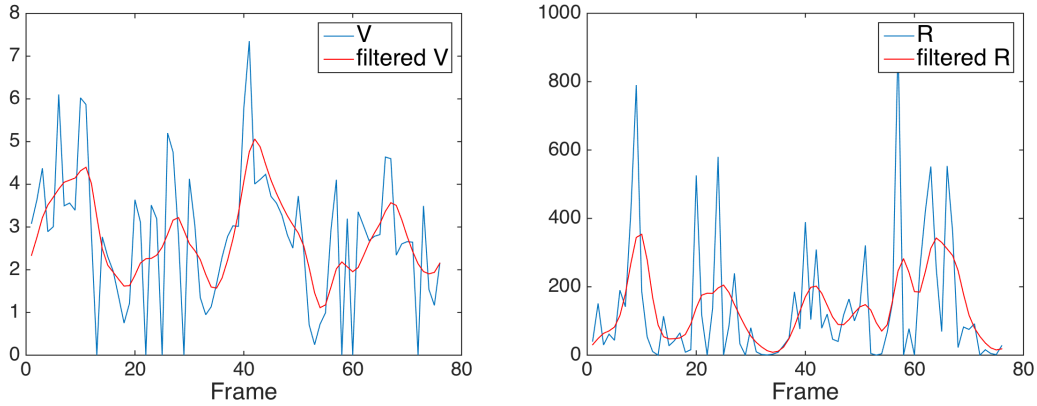
Figure 2.1, on the bottom line, shows the behavior of two of the computed features (velocity and radius of curvature) across a period of time lasting 80 frames. As expected, the peculiarities of the performed movements are best appreciated by observing it for some time.

Feature	**Formula**
Tangential velocity	$\hat{\mathbf{V}}_i(t) = (u_i(t), v_i(t), \Delta_t)$
Tangential velocity magnitude	$\hat{V}_i(t) = \sqrt{(u_i(t)^2 + v_i(t)^2 + \Delta_t^2}$
Acceleration	$\hat{\mathbf{A}}_i(t) = (u_i(t) - u_i(t-1),$ $v_i(t) - v_i(t-1), 0)$
Curvature	$\hat{C}_i(t) = \frac{\|\hat{\mathbf{V}}_i(t) \times \hat{\mathbf{A}}_i(t)\|}{\|\hat{\mathbf{V}}_i(t)\|^3}$
Radius of curvature	$\hat{R}_i(t) = \frac{1}{\hat{C}_i(t)}$
Angular velocity	$\hat{A}_i(t) = \frac{\hat{V}_i(t)}{\hat{R}_i(t)}$

Table 2.1: Empirical formulations of the spatio-temporal dynamic features (Δ_t is the temporal displacement between observations of two adjacent time instants).



(a) Frame (b) Optical flow magnitude (c) Segmentation



(d) Velocity (e) Radius of curvature

Figure 2.1: Above, the key steps of our low-level motion representation: the original frame (a), the map of the optical flow magnitude (b) and the segmented region (c). Below, a visual comparison between raw (blue) and filtered (red) features: the velocity (d) and the radius of curvature (e).

2.4.2 Multi-resolution motion representation over time

Since a *meaningful event* lasts more than one temporal tick (e.g. in the action of *Moving an object*, the meaningful information is not just in a specific time instant but rather in the whole duration of the action), we integrate the instantaneous motion representation over a fixed temporal frame w. To this purpose, we consider a set of w subsequent measurements $[\mathbf{x}_{t-w}, \ldots, \mathbf{x}_t]$ and compute a running average of each feature across time, obtaining a new motion descriptor $\hat{\mathbf{x}}_t(w)$.

The choice of an appropriate size for the temporal window is critical and highly dependent on the specific dynamic event. For this reason we adopt a multi-resolution approach, where different temporal windows are jointly adopted, and we propose an adaptive procedure where we learn from examples the best combination of temporal windows.

More in details, let us consider a maximum temporal window extent $w_T^{MAX} \in \mathbb{N}$, such that $w_T^{MAX} > 1$, and a selection of potentially interesting time windows w_T defined as elements of

a set $W = \left\{ w \in \mathbb{N} | w \geq 1 \wedge w \leq w_T^{MAX} \right\}$.
At a certain time instant t we may have a temporal sequence of observations $S_t \in \mathbb{R}^{4w_T^{MAX}}$ as

$$S_t = \left[\mathbf{x}_{t-w_T^{MAX}}, \ldots, \mathbf{x}_t \right]. \tag{2.6}$$

We apply a bank of *running average* filters – of widths selected from the range in the set W – to each feature separately. The result is a set of motion descriptors $\hat{\mathbf{x}}_t(w_T)$ referring to different time periods w_T and such that

$$\begin{aligned} &\hat{\mathbf{x}}_t(1) = \mathbf{x}_t \\ &\hat{\mathbf{x}}_t(w_T) = \mathcal{RA}(S_t|_{w_T}, w_T), \text{ for } 1 < w_T \leq w_T^{MAX} \end{aligned} \tag{2.7}$$

where $\mathcal{RA}$ is the running average filtering while the notation $S_t|_{w_T}$ denotes the restriction of sequence S_t to the last w_T elements. The leftmost part of Figure 2.2 reports a sketch of this filtering procedure.

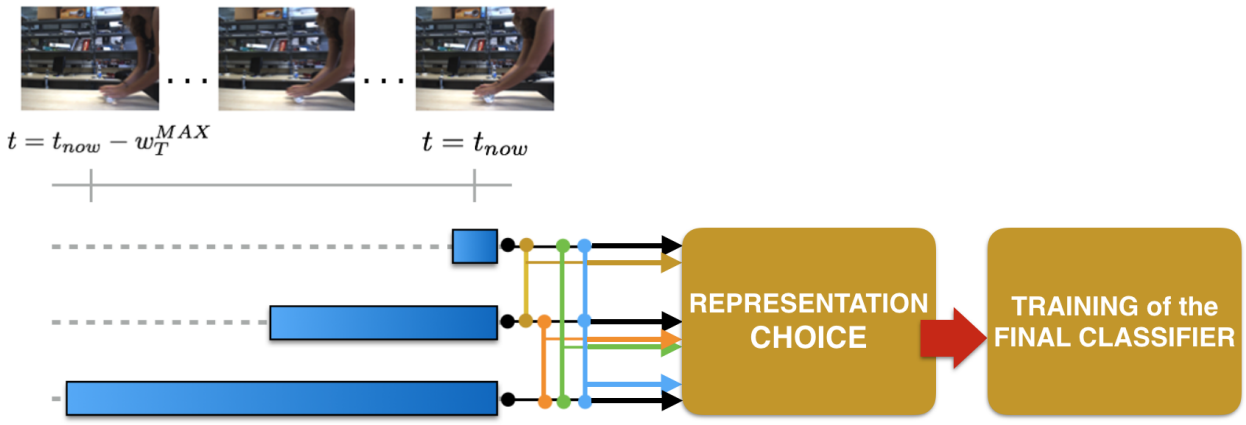


Figure 2.2: A visual sketch of our pipeline. From left: in each image of a sequence we detect the moving region and compute the features. We then compute a running average of those features over different temporal windows (3 in this example, identified by the blue ranges). Then, we evaluate different temporal models which may be composed by more than one temporal resolution and select the most appropriate.

Starting from the set of motion descriptors of Equation (2.7) we obtain many possible temporal multi-resolution motion descriptors $\{\mathcal{F}_t^i\}_i$:

$$\mathcal{F}_t^i = \oplus \delta^i(w_T) \hat{\mathbf{x}}_t(w_T), \text{ for all } w_T \in W \tag{2.8}$$

where $\oplus$ denotes the concatenation between feature vectors, while $\delta^i(w_T) \in \{0, 1\}$ is a binary weight representing the presence or absence of the corresponding filtered vector in the final

descriptor.

Thus, as a final step, we need to select an appropriate and minimal combination of different temporal windows, considering that a multi-resolution descriptor will allow us to deal with different type of dynamic events, but many different temporal windows would carry a similar amount of information. The core of the selection process is detailed in the next section, as it is intertwined with the actual motion classification step.

2.4.3 Biological motion representation and classification

We formulate the problem of recognizing biological motion from video sequences as a binary classification problem. To this purpose, given a certain temporal scheme denoted with i^*, we consider a training set

$$Z = \{(\mathcal{F}_k^{i^*}, y_k) \in X \times Y\}_{k=1}^n \tag{2.9}$$

where $\mathcal{F}_k^{i^*} \in X \subseteq \mathbb{R}^d$ is a given temporal multi-resolution descriptor (input)[1], while $y_k \in Y = \{-1, 1\}$ is the associated output label (1 for biological samples and -1 for negative non-biological samples). The size d depends on the specific $\mathcal{F}_k^{i^*}$ considered. Henceforth, we will refer to $\mathcal{F}_k^{i^*}$ as $\mathcal{F}_k$.

To learn the relationship between input and output in a predictive way, we adopt a *Regularized Least Squares* (RLS) binary classifier which amounts at minimizing the following functional

$$f_Z = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{k=1}^{n} (y_k - f(\mathcal{F}_k))^2 + \lambda ||f||_{\mathcal{H}} \tag{2.10}$$

where $\mathcal{H}$ is a Reproducing Kernel Hilbert Space with a positive semi-definite kernel function K, and λ a regularization parameter that controls the trade-off between the data term and the smoothness term. At run time, a new datum $\mathcal{F}$ is associated with an estimated label obtained by the sign of $f_Z^\lambda(\mathcal{F})$, with

$$f_Z^\lambda(\mathcal{F}) = \sum_{k=1}^{n} \alpha_k K(\mathcal{F}, \mathcal{F}_k) \tag{2.11}$$

where $\alpha = (\mathbf{K} + n\lambda I)^{-1}\mathbf{y}$ is an n-dimensional vector of unknowns, while $\mathbf{K}$ is the associated kernel matrix. In the model selection procedure better detailed in Section 2.5.1, we train a set of classifiers each one associated with a different combination of motion features $\mathcal{F}$. The best multi-resolution motion descriptor is selected in a data-driven manner, by ranking the validation error achieved by the different classifiers.

[1]We omit the index t of Equation (2.8) for readability.

2.5 Offline experimental analysis

In this section we discuss the experiments we performed primarily on video sequences acquired with the iCub humanoid robot [52], using the machine learning library GURLS for an efficient implementation of RLS [83]. We first describe the method assessment, where we discuss the choice of kernel function and strategy for data filtering. To evaluate the sensitivity of the method to the acquisition sensor, we also considered test sets captured with a common web-cam and a hand-held camera (Canon EOS 550D).

The method developed builds on the optical flow, therefore in the experiments we make some assumptions keeping in mind the optical flow limitations known from literature: in particular, we consider videos with textured moving objects in order to avoid the camouflage effect. Moreover, we consider videos with just one source of motion in the scene, even if, in principle, in case there are several sources of motion, it is possible to identify all the regions, track them and apply the same method for motion recognition on individual regions in parallel.

In the following, we first discuss in details the training procedure. Second, we show the generalization capability of our approach by discussing its appropriateness on a selection of tests including new dynamic events, new scenarios, and on data acquired by a different sensor.

2.5.1 Training the motion classifier

The training phase of a motion classifier includes *(i)* a *model selection* in which the classification parameters and the most appropriate *multi-resolution representation* are chosen; and *(ii)* training of the final classifier based on the previously selected model.

The dataset

Our training set is composed of indoor videos of three subjects observed by the iCub eyes while performing repetitions of given actions from a repertoire of dynamic movements typical of a human-robot interaction setting. The choice of acquiring a collection of videos in-house is due to the absence, to the best of our knowledge, of a benchmark explicitly designed for purposes similar to ours. More in details, we consider the following actions:

- *Rolling dough* (9 movements, ∼300 frames — Figure 2.3a)
- *Pointing* a finger towards a certain 3D location (7 movements, ∼330 frames — Figure 2.3b)

- *Mixing in a bowl* (29 movements, ∼190 frames — Figure 2.3c)
- *Transporting* an object from and to different positions on a table (6 movements, ∼300 frames — Figure 2.3d)
- *Writing* on a paper sheet (3 movements, ∼300 frames — Figure 2.3e)

As for the non-biological examples, we consider a selection of dynamic events which can be observed indoors:

- *Wheel with a random pattern* (∼300 frames — Figure 2.3f)
- *Wheel with a zig-zag pattern* (∼300 frames — Figure 2.3g)
- *Balloon* (∼300 frames — Figure 2.3h)
- *Toy Top* turning on a table (∼300 frames — Figure 2.3i)
- *Toy Train* (∼398 frames — Figure 2.3j)

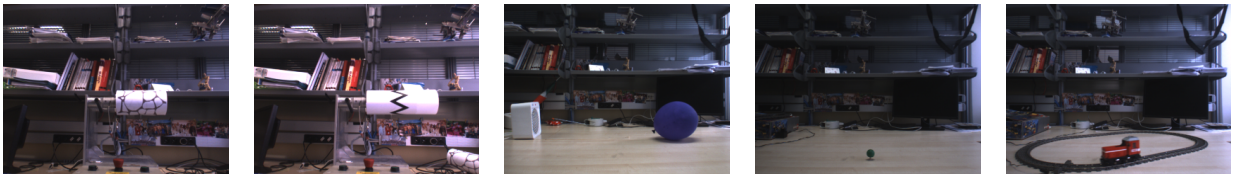
For each dynamic event we acquired two videos. Henceforth, we will adopt the notation $\{V_{S_i1}\}$ and $\{V_{S_i2}\}$, $i = 1, 2, 3$, to denote, respectively, the sets of first and second video instance of subject S_i. Similarly, $\{V_{N1}\}$ and $\{V_{N2}\}$ are the two sets of videos containing non-biological events.

In the following, the training set used for training the classifier and selecting the model includes $\{V_{S_i1}\}$ for $i = 1, 2, 3$, $\{V_{N1}\}$, and $\{V_{N2}\}$. Details on how they are divided are described where appropriate. Instead, $\{V_{S_i2}\}$, $i = 1, 2, 3$, are left out and used as a first test in Section 2.5.2. The images have size 320×240 and have been acquired at an approximate rate of 15 fps. The cameras we used in our work (both the robot and the opposite view webcam used for the test) have a relatively low resolution.

Method assessment

We first compare the adoption of linear and RBF kernel in RLS combined with the use of two different strategies for filtering the motion features over time. The first approach relies on filtering the horizontal and vertical velocity components (V_x and V_y) only (i.e. the elements from which all the other features are derived), while with the second one we filter each feature in Equation (2.5) (here after identified as V, C, R and A) separately. In the comparison, we consider a finite set of possible temporal windows, $W = \{1, 10, 15, 30\}$. For each of them we train a separate classifier on a data set composed as $\{V_{S_11}\} \cup \{V_{N1}\}$. Next, we evaluate the classification performances on a validation set in which we collect $\{V_{S_12}\} \cup \{V_{N2}\}$.

(a) Rolling dough (b) Pointing (c) Mixing in a bowl (d) Transporting (e) Writing



(f) Wheel (random pattern) (g) Wheel (zig-zag pattern) (h) Balloon (i) Toy-top (j) Toy train

Figure 2.3: (a-e) Biological movements included in the training set.
(f-j) Non-biological movements included in the training set.

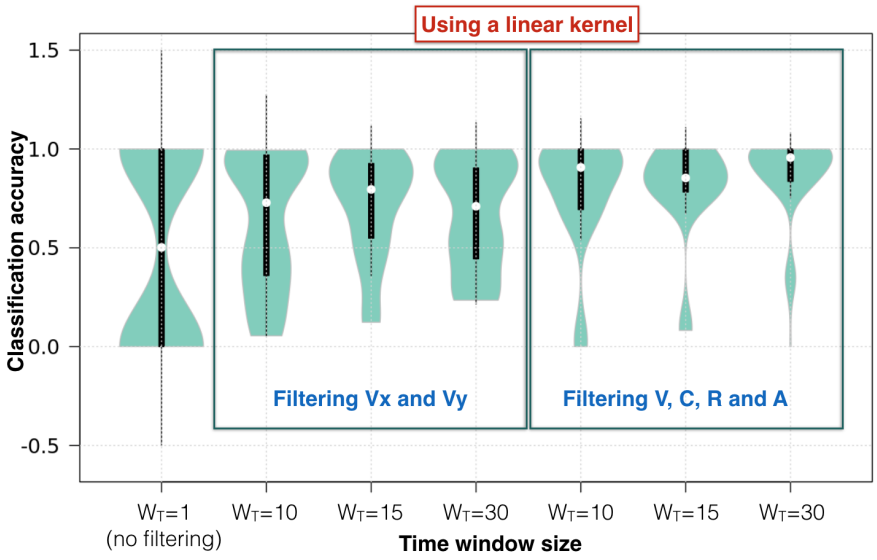
Figure 2.4 shows the results we obtained in terms of violin plots, enhancing the probability density of the data at different values. A visual comparison between Figure 2.4a and 2.4b clearly shows the benefit of using the RBF kernel, while the violin plots of the right-most parts of both figures speaks in favor of filtering the dynamic features separately. This observation is enforced reporting in Table (2.2) the average classification accuracies obtained with the RBF kernel. We include in the evaluation a further accuracy (second column of the table) obtained by classifying videos (instead of single motion descriptors, i.e. single frames) with respect to the majority of labels associated with it over time. The results clearly show the robustness of the representation schema.

Following these conclusions, in the next experiments the features are filtered separately and the classifier is equipped with an RBF kernel.

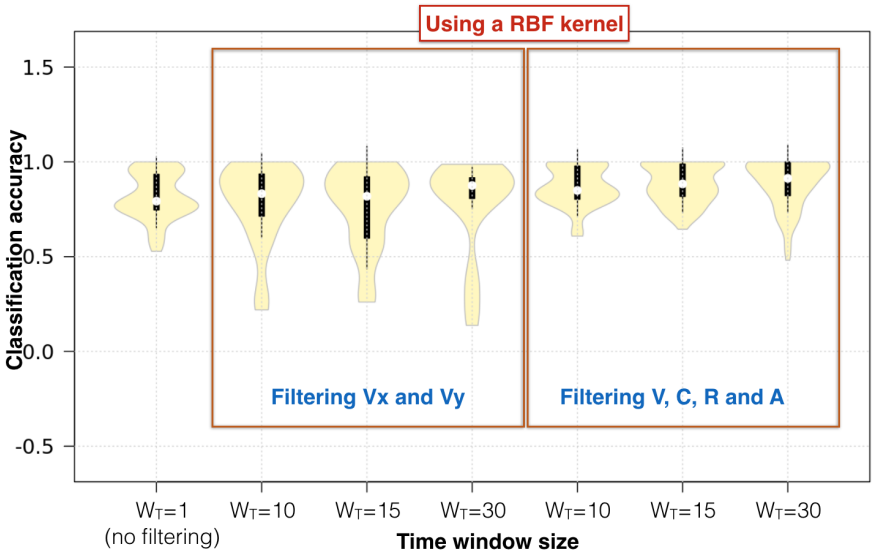
Model selection

The main purpose of the model selection (see a visual sketch in Figure 2.5) step is to choose the most appropriate temporal multi-resolution representation, from a large set of N choices. This will allow us, at run time, to compute that representation only.
We perform the selection in a data-driven manner, where for each representation considered, we obtain an average validation accuracy by adopting a *Leave-One-Subject-Out* approach.

Leave One Subject Out procedure. For a given multi-resolution representation (Equation (2.8)) we represent all data accordingly, then we partition the training set each time leaving the videos of one subject $\{V_{S_i 1}\}$ as positive examples of a validation set. As for



(a)



(b)

Figure 2.4: Comparison of different conditions for building the representation scheme, using a linear kernel (a) and using a RBF kernel (b). In both cases, left and right parts of the plots refer to filtering the velocity components only, or all the features separately.

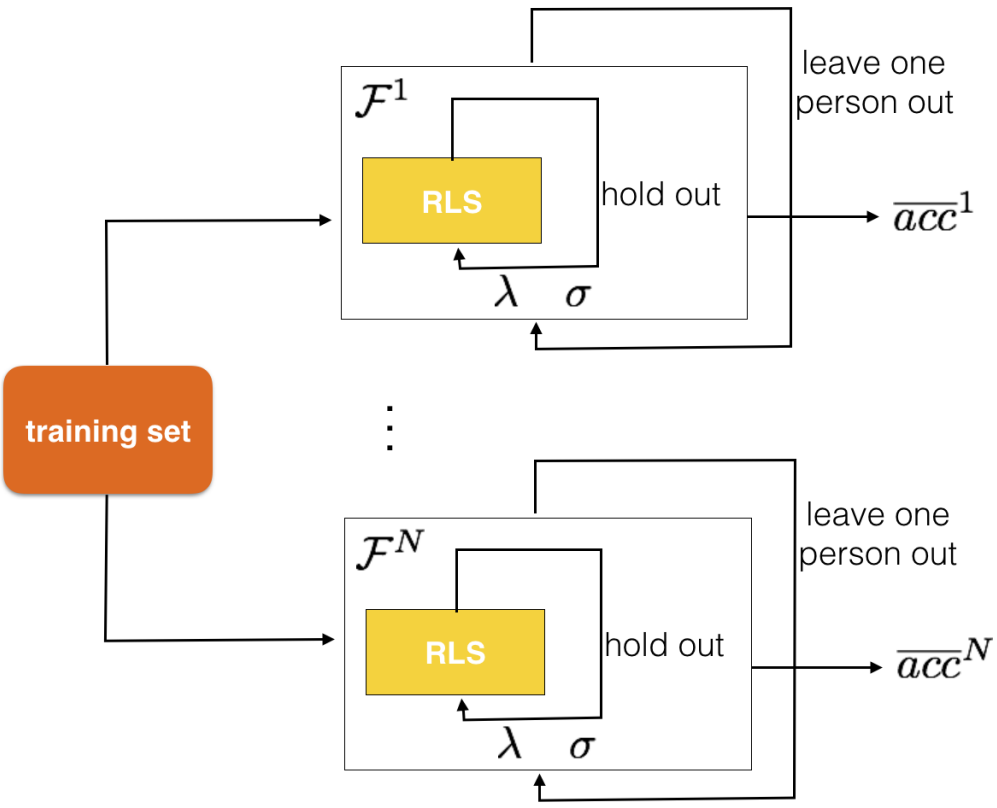


Figure 2.5: A visual sketch representing the core procedure of our model selection.

Method	**Filter width**	**Avg. Acc. ± std.dev.**	**Video hit rate**
No filter	($W_T = 1$)	0.81 ± 0.13	0.72
On V_x and V_y	W_T=10	0.78 ± 0.22	0.71
	W_T=15	0.76 ± 0.23	0.68
	W_T=30	0.77 ± 0.27	0.80
On V, C, R and A	W_T=10	0.88 ± 0.10	0.92
	W_T=15	0.89 ± 0.10	0.86
	W_T=30	0.87 ± 0.13	0.84

Table 2.2: Averages and standard deviations of accuracies across 10 different runs of the training phase using the RBF kernel. The filtering applied to each feature shows higher robustness.

Scheme	**Average accuracy**	**Overall accuracy**	**Precision**	**F-measure**	**Hit rate video**	**Overall ranking**
$\hat{\mathbf{x}}_t(1) \oplus \hat{\mathbf{x}}_t(15) \oplus \hat{\mathbf{x}}_t(30)$	$\mathbf{0.87} \pm 0.13$	$\mathbf{0.86} \pm 0.01$	$\mathbf{0.91} \pm 0.01$	$\mathbf{0.69} \pm 0.20$	0.82 ± 0.05	5.25
$\hat{\mathbf{x}}_t(1) \oplus \hat{\mathbf{x}}_t(10) \oplus \hat{\mathbf{x}}_t(30)$	$\mathbf{0.87} \pm 0.14$	$\mathbf{0.86} \pm 0.01$	0.90 ± 0.01	0.68 ± 0.20	$\mathbf{0.86} \pm 0.04$	5.03
$\hat{\mathbf{x}}_t(1) \oplus \hat{\mathbf{x}}_t(30)$	$\mathbf{0.87} \pm 0.12$	$\mathbf{0.86} \pm 0.01$	$\mathbf{0.91} \pm 0.01$	0.68 ± 0.20	$\mathbf{0.86} \pm 0.04$	4.00
$\hat{\mathbf{x}}_t(1) \oplus \hat{\mathbf{x}}_t(10) \oplus \hat{\mathbf{x}}_t(15) \oplus \hat{\mathbf{x}}_t(30)$	0.86 ± 0.13	0.85 ± 0.02	0.90 ± 0.02	0.68 ± 0.20	0.81 ± 0.07	2.08
$\hat{\mathbf{x}}_t(1) \oplus \hat{\mathbf{x}}_t(15)$	0.85 ± 0.12	0.85 ± 0.02	0.90 ± 0.01	0.68 ± 0.20	0.81 ± 0.07	1.67
$\hat{\mathbf{x}}_t(1) \oplus \hat{\mathbf{x}}_t(10) \oplus \hat{\mathbf{x}}_t(15)$	0.85 ± 0.13	0.85 ± 0.02	0.89 ± 0.02	0.68 ± 0.20	0.82 ± 0.07	1.54
$\hat{\mathbf{x}}_t(1) \oplus \hat{\mathbf{x}}_t(10)$	0.84 ± 0.12	0.84 ± 0.01	0.89 ± 0.01	0.67 ± 0.19	0.77 ± 0.05	1.08
$\hat{\mathbf{x}}_t(10) \oplus \hat{\mathbf{x}}_t(15) \oplus \hat{\mathbf{x}}_t(30)$	0.85 ± 0.15	0.84 ± 0.02	0.89 ± 0.02	0.67 ± 0.20	0.78 ± 0.06	1.08
$\hat{\mathbf{x}}_t(10) \oplus \hat{\mathbf{x}}_t(30)$	0.84 ± 0.15	0.83 ± 0.02	0.88 ± 0.02	0.67 ± 0.20	0.76 ± 0.07	0.87
$\hat{\mathbf{x}}_t(15) \oplus \hat{\mathbf{x}}_t(30)$	0.83 ± 0.15	0.83 ± 0.02	0.88 ± 0.02	0.67 ± 0.19	0.74 ± 0.05	0.83
$\hat{\mathbf{x}}_t(10) \oplus \hat{\mathbf{x}}_t(15)$	0.80 ± 0.14	0.81 ± 0.03	0.86 ± 0.02	0.65 ± 0.19	0.61 ± 0.12	0.73

Table 2.3: A quantitative evaluation of the combined time descriptors.

the negatives, the set $\{V_{N1}\}$ is always used as a training and the set $\{V_{N2}\}$ as a validation. This allows us to obtain an average validation accuracy. With this partitioning, the training set is composed of ~ 3000 points. Notice that within each run of the training procedure we include a *hold out* process (with $M = 10$ different partitioning), with a balanced training, that allows us to select the parameters σ (RBF Kernel parameter) and λ (RLS regularization parameter).

Detecting the best representation. Let N be the number of the different multi-resolution representations considered. This number depends on the cardinality of the set of potentially interesting time windows W (see Section 2.4.2). We set $W = \{1, 5, 10, 15, 20, 25, 30\}$, and forced the final temporal descriptor (Equation (2.8)) to be concatenation of at most 3 different temporal windows (Equation (2.7)). We chose a maximum size temporal window of 30 frames – equivalent to 2 seconds – as this temporal period already affords complex action processing in human brain [85].

The step size of 5 frames between adjacent windows is due to the intrinsic nature of the data. The choice of considering at most 3 temporal windows is suggested by the need of controlling the amount of data redundancy. Under these assumptions, we obtain $N = \#W + \binom{\#W}{2} + \binom{\#W}{3} = 63$.
In Figure 2.6 we show the performances of each representation scheme, ranked in descending order with respect to the average validation accuracy. The bars are color-coded with respect to the number of concatenated temporal representations (from dark to light: 3, 2, 1). In general three temporal windows appear to be more descriptive, and in particular the ones including different temporal ranges (short-medium-long) are ranked first.

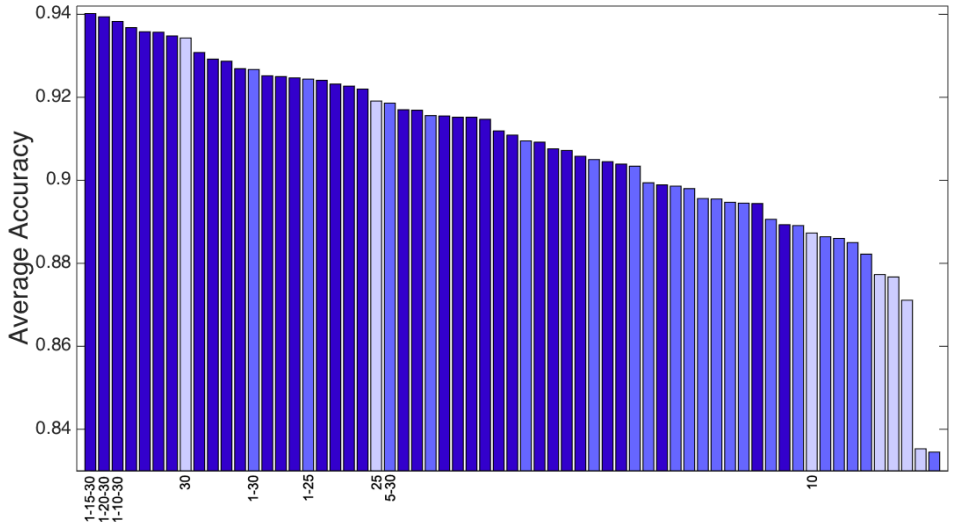


Figure 2.6: Average accuracy for each representation (dark blue: concatenation of three temporal windows; medium blue: concatenation of two temporal windows; light blue: a single temporal window.

With this analysis, we conclude that the temporal multi-resolution representation that concatenates the raw features vector with the filtered measures on temporal windows $w_T = 15$ and $w_T = 30$ is the best-performing, leading to a final feature vector of length 12. Figure 2.7 shows the classification accuracies of the selected multi-resolution representation, compared to the cases a single filter width is adopted, on the validation set. A first observation is that there is not a single temporal window appropriate for all the events: for instance, the single filter width $w_T = 30$ performs quite well in all cases but one (sequence *Mixing*, case (c) of Figure 2.7), as the very fast dynamics of the movement requires smaller window sizes for filtering the signals. Indeed, shorter time windows provide better performances in this case.

Overall, the multi-resolution descriptor reports more stable performances, with higher

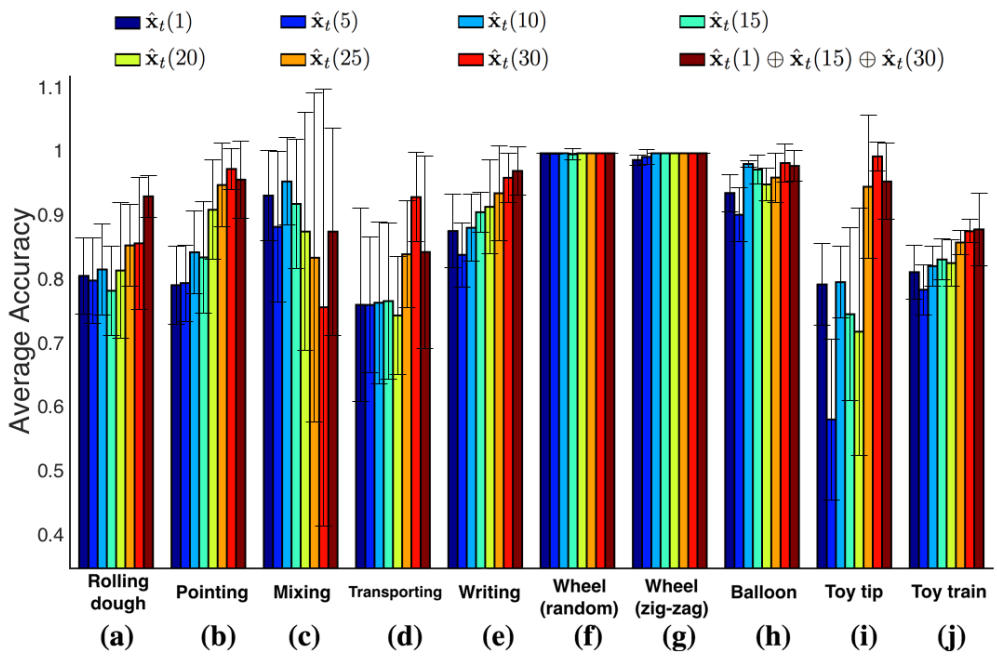


Figure 2.7: A comparison between the selected temporal multi-resolution descriptor and all the different time widths considered independently. The results refer to the actions of Figure 2.3.

average accuracies and lower standard deviations (see Table (2.4)). This speaks in favour of the capability of our approach to cope effectively with dynamic events of variable temporal duration when no prior information is available.

Training the final classifier

Now we have selected the most appropriate temporal representation r^*, we may build the final classifier. To this purpose, we consider the whole training set and run a final training procedure using the r^* representation (1-15-30), and performing model selection in order to set σ^* and λ^* again with a balanced hold out procedure, with $M = 10$ trials. The obtained classifier is adopted to evaluate the capability of our method to generalize to new data, as discussed in the following sections.

2.5.2 Testing the motion classifier

In this section we report the results of our testing analysis (see Figure 2.8). The experiments we carried out aim at testing the validity of the model on new data, including data containing very different appearance of dynamics with respect to the training set.
We organized the experiments in different test trials, to discuss the robustness of our model

Representation	Average accuracy	Standard deviation acc.
$\hat{\mathbf{x}}_t(1)$	0.87	0.09
$\hat{\mathbf{x}}_t(5)$	0.84	0.12
$\hat{\mathbf{x}}_t(10)$	0.89	0.09
$\hat{\mathbf{x}}_t(15)$	0.88	0.10
$\hat{\mathbf{x}}_t(20)$	0.88	0.10
$\hat{\mathbf{x}}_t(25)$	0.92	0.06
$\hat{\mathbf{x}}_t(30)$	0.93	0.08
$\hat{\mathbf{x}}_t(1) \oplus \hat{\mathbf{x}}_t(15) \oplus \hat{\mathbf{x}}_t(30)$	**0.94**	**0.05**

Table 2.4: Average and standard deviation accuracy of the temporal single-resolution representations and the best performing multi-resolution scheme.

on scenarios of increasing complexity. At first, we perform an assessment of the method on the same actions of the training set but using different videos (Test I in the following). Then, we proceed considering conditions that vary with respect to the training set: we focus on movements included in the training set but characterized by different speeds or trajectory patterns (Test II); actions in critical situations of visibility (as in presence of occlusions, limited spatial extent of the observed motion, and even when just the shadow is in the camera field of view (Test III); different human actions (on the fronto-parallel plane or performed in depth with respect to the camera) recorded with the robot (Test IV) and with a hand-held camera placed in front of the robot, to test the influence of the acquisition sensor and of the viewpoint (Test V). On average, each video lasts about $20''$.
In the following, we discuss in details each test.

Positive examples. We focus on biological movements and consider the training actions, adopting the second set of videos of each subject, i.e. $\{V_{S_12}\}, \{V_{S_22}\}, \{V_{S_32}\}$. As expected, the method performs very well (see the graph in Figure 2.8a), with an average accuracy, across subjects, of 0.98 ± 0.03.

Negative examples with changing speed. We consider variations of the apparent motion with respect to the training set. Case 1: the three training subjects performing faster training actions *Rolling dough, Transporting*; Case 2: the *Wheel* (with one of the appearance patterns of the training set) and the *Toy train* (with the same trajectory of the training set) with different speed and the *Wheel* with a different pattern (first picture in Figure 2.8b); Case 3: the *Toy train* covering a circular trajectory (second picture in Figure 2.8b) as opposed to the ellipsoidal path considered in the training set (Figure 2.3j), with slower and faster

velocity profiles (at approximately half and twice the velocity of the training set). The accuracies, reported in the graph in Figure 2.8b show again very appropriate values, although an influence of the variations applied in Case 3 can be observed. This may be explained with a partial lack on information when the conditions become too severe (presence of high velocity, limited spatial extent of the apparent moving region).

Occlusions and distant dynamics. We focus here on some critical scenarios.

- **Case 1**: A training subject performs actions included in the training set (see an example in the first picture in Figure 2.8c) and a new one (*Waving*) with partial occlusions;
- **Case 2**: A training subject performs actions not included in the training set (*Walking, Waving hand*) far from the camera (second picture in Figure 2.8c);
- **Case 3**: Observing the shadow of an action included in the training set (*Pointing*) (third picture in Figure 2.8c) and a new one characterised by a whole-body motion (*Walking*) as opposed to the upper-body movements considered in the training set.

The accuracies are reported in the graph in Figure 2.8c. Cases I and III show how our method is tolerant to the presence of severe occlusions and, to some extent, is able to deal with indirect information, such as the one produced by the shadow of a moving object. As expected, both situations produce good results, with a relatively small decay in the performances. On the contrary, Case II shows a greater decrease in performance, probably due to the too limited extension of the apparent motion caused by the large distance of the motion from the camera.

Novel dynamic events. We consider here actions executed on the fronto-parallel plane and movements performed in depth, on a transverse plane. As for fronto-parallel dynamics, we focus in particular on handwriting, considering the following sub-categories: frontal drawings of smooth symbols (as ellipses, infinite, see the first picture in Figure 2.8d, **Case 1**), hearts, **Case 2**), sharp symbols (as rectangles and lines, **Case 3**), unconstrained text writings (**Case 4**).
Concerning the movements in depth, we identified the following scenarios: a user performing natural, unconstrained movements (**Case 5**, see an example in the second picture of Figure 2.8d); drawing smooth symbols on a table (**Case 6**, see an example in the third picture of Figure 2.8d); drawing hearts on a table (**Case 7**); drawing of sharp symbols on a table (**Case 8**); free text writing on a table (**Case 9**); natural movements towards the robot (hi5, handshake, **Case 10**).
We considered both smooth and sharp shapes in order to test the method in case of continuous

movements similar to the ones on which the Two-Thirds Power Law has been already tested in the literature (smooth shapes), and in case of other type of movements as the discontinuous ones (sharp shapes). The accuracies are reported in the graph of Figure 2.8d. We can observe a very good accuracy in the fronto-parallel cases (from Case 1 to Case 4). Regarding the movements in depth, we can observe there is a decay in the performances in Case 5, as it includes very different movements with respect to training, with some even involving complex forces (like in the action of hammering); Case 6, the actions of drawing smooth shapes, shows a very good performance, while it decreases in Case 7 and Case 8, respectively the actions of drawing hearts and sharp shapes; the accuracy is very good in Case 9 and 10, respectively the action of writing on a table, that indeed was in the training set (even if the video has been acquired in different place and time), and the actions towards the robot.

View-point changes with different sensors. We consider the same movements adopted for Test IV, but observed from an opposite point of view and using two different sensors (a common web-cam and the camera Canon EOS 550D). The use of different sensors and the change of perspective lead to the generation of optical flow fields that may differ significantly from the ones adopted for the training phase.
We organized the tests considering the same classification adopted in Test IV. Planar movements have been observed with a web-cam (320 x 240 pixels, 20 fps), while the sequences of actions in depth have been acquired with the Canon (320 x 240 pixels, 30 fps).
The accuracies are reported in the graph of Figure 2.8e. We can observe that changing the point of view there is a decay in the performance in the fronto-parallel cases (Case 1-Case 4), except for a small increase in Case 3, while there is an increase in the performance in all the cases of movements in depth except for Case 10. The movements with inflection points (drawing movements of smooth shapes, both on the fronto-parallel plane and in depth), that should be robust to the change of the point of view, are Case 1 (decrease of 0.29) and Case 6 (increase of 0.13).

2.6 Discussion

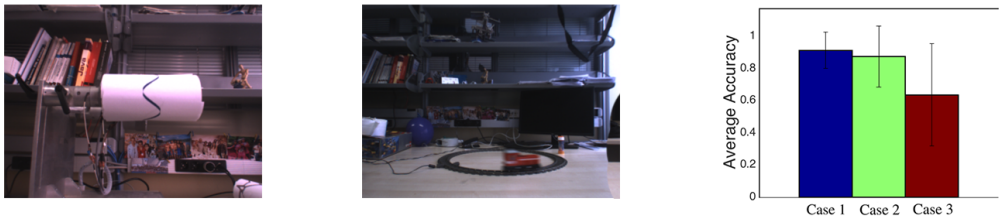
In this chapter we presented a computational model for discriminating between biological and non-biological movements in video sequences, leveraging a well known regularity of human motor control. Notwithstanding the large heterogeneity of the dynamics of the movements that can be encountered in everyday life situations, we proposed a temporal multi-resolution descriptor, purely based on low-level motion features. We showed that this descriptor has on average a better performance than any single-resolution descriptor, as the latter fails in

capturing the large variability of possible dynamics of the movements.

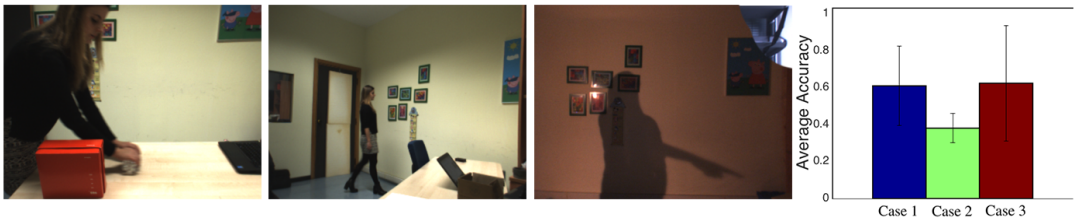
We demonstrated the descriptor to be effective also for events of a variable temporal duration and to generalize well to new and challenging scenarios. It should be noticed that our approach does not require any appearance-based detection of the human partner, as the regularities of biological motion are extracted independently of the agent's shape. This feature guarantees the possibility to recognize human activities also when the agent is not visible or severely occluded, e.g. observing a shadow or a visible tool moved by a hidden agent.

Given the promising results derived from the offline testing reported in the previous sections, in the next chapter we propose a version of the method able to work online and to be integrated on the software framework of the robot iCub.

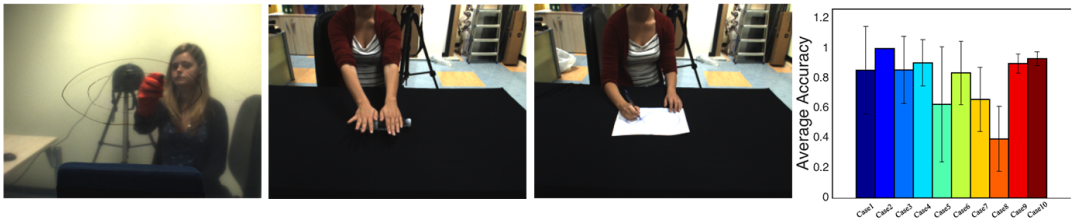
(a) Test I on new videos of biological movements observed during training. Each bar refers to a training subject.



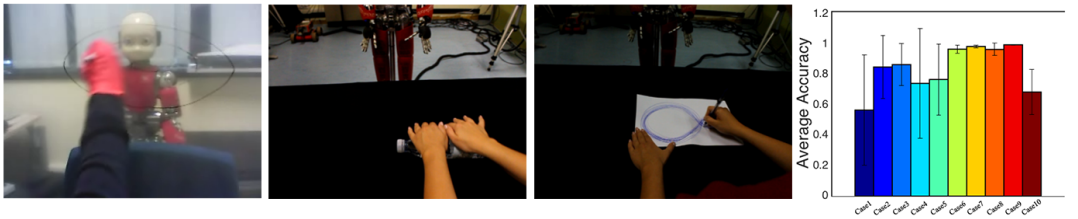
(b) Test II on movements with changing speed.



(c) Test III on videos with occlusions, far events and shadows.



(d) Test IV on novel dynamic events.



(e) Test V on novel dynamic events observed from a different view-point.

Figure 2.8: Frames from sequences adopted in the Tests and graphs with the average classification accuracy (see the text for details on different cases for each Test). The images are the ones that have been really used for the analysis: the low quality is due to both the resolution and the bad lighting condition. In some cases, the presence of a transparent board placed between the camera and the observed scene further affects the quality of the images.

Chapter 3

Implementation of the biological motion detector on iCub

3.1 Introduction

The possibility to exploit the method described in the previous chapter for robot perception is then validated by implementing the method in a module integrated in the software framework of the iCub humanoid robot [91]. The module implements an engineered variation of our method – appropriately handled to work online – and is used to enhance the robot visual attention system [71, 62], endowing the robot with the ability to rapidly redeploy attention in the scene on actions performed by human agents with a biologically plausible saccadic behaviour. The advantage of the solution is that attention is biased towards moving human agents even when they are not visible in the scene. At the same time, the natural robot gaze motion can act as an implicit communication signal, informing the collaborators of its current attentional state. In this paper, a detailed analysis of the results of the integration between the motion classification and the attentive system is done by separating the two stages of perception and action: this gives a better idea of when, during the robot pipeline, and why the robot fails or is not perfectly precise in the discrimination task between the biological and non-biological movements. Moreover, we also analyze the velocity profile of the fixation point to reach the target.

3.2 Implementation in the iCub framework

The final goal is to embed the human activity detection system in a more structured architecture supporting natural attention redeployment and gaze behavior by the robot.

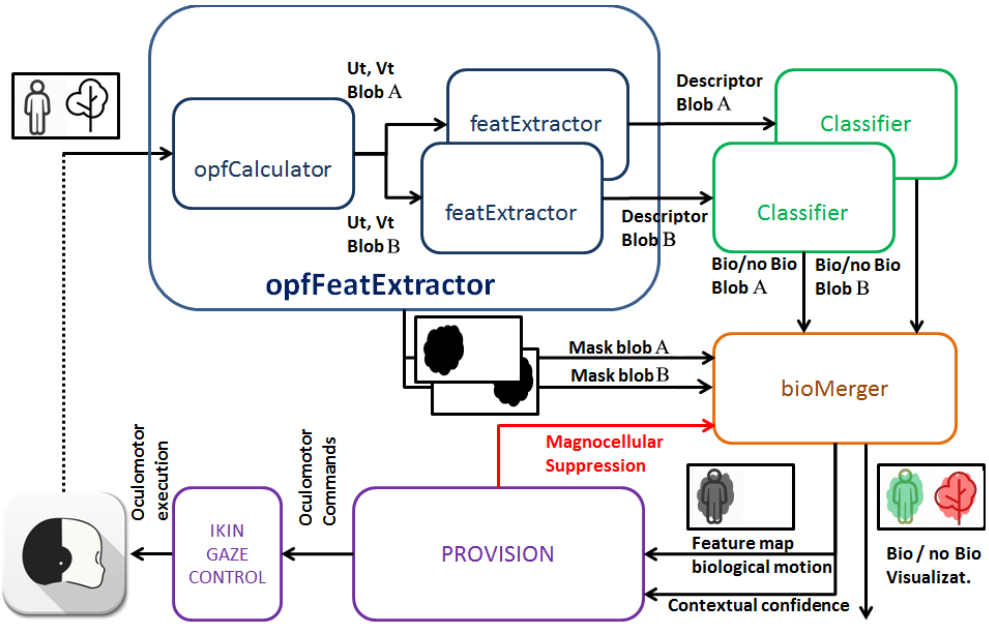


Figure 3.1: The iCub architecture where our method has been implemented.

The software framework of the solution is designed to leverage the modularity supported by the middleware Yarp [53] and to enable two different computation stages: the perception of biological movement and the synthesis of biological oculomotor actions. Modularity guarantees optimal computation distribution on the network resources and scalability of the solution. In Figure 3.1, we show the structure of the framework indicating how interconnections between modules are structured to produce a natural behaviour in the iCub. The frameworks is designed to close the sensor-action loop through the execution of oculomotor actions based on salient loci in the stream of input images.

In the following, we review each module in details. Although our solution may account for a generic number N of moving entities in the scene, without loosing in generality, we focus on the case $N = 2$ to exemplify the system behaviour. In particular, we choose the two biggest moving blobs in the scene and we track them within the visual field in time with the assumption that they do not overlap in corresponding regions of the visual field. The modules explained in Sections 3.2.1, 3.2.3 are part of our contributions, while the ones explained in Sections 3.2.2, 3.2.4, 3.2.5 are part of the pre-existing framework.

3.2.1 OpfFeatExtractor

The module resembles the early stage of visual pathways. The analysis is based on images of size 320×240 acquired from the cameras embedded in the eyes. The module comprises two

classes of parallel computing, opfCalculator and featExtractor, that with reference to Section 2.4, correspond to the functionalities of motion segmentation and description, respectively. The parallelization of the necessary computation demand in multiple threads guarantees online performance.

Two instances of the featExtractor class analyze the most salient and persistent blobs in the image plane, henceforth named A and B blob. The correctness of the data transferring from the opfCalculator to the featExtractor is guaranteed by supervised access (Yarp::Sig::Semaphore) to the two shared resources, srA = $[U_t, V_t, blobA]$ and srB =$[U_t, V_t, blobB]$. The opfCalculator module provides maps of the horizontal (U_t) and vertical (V_t) components of the optical flow on the whole image and the masks of A ($blobA$) and B ($blobB$) blobs to the rest of the network via tcp ports. In addition to the blobs descriptors, the opfFeatExtractor also provides two mono-cromatic images, the binary maps marking the locations of blobs A and B in the image plane.

3.2.2 Classifier

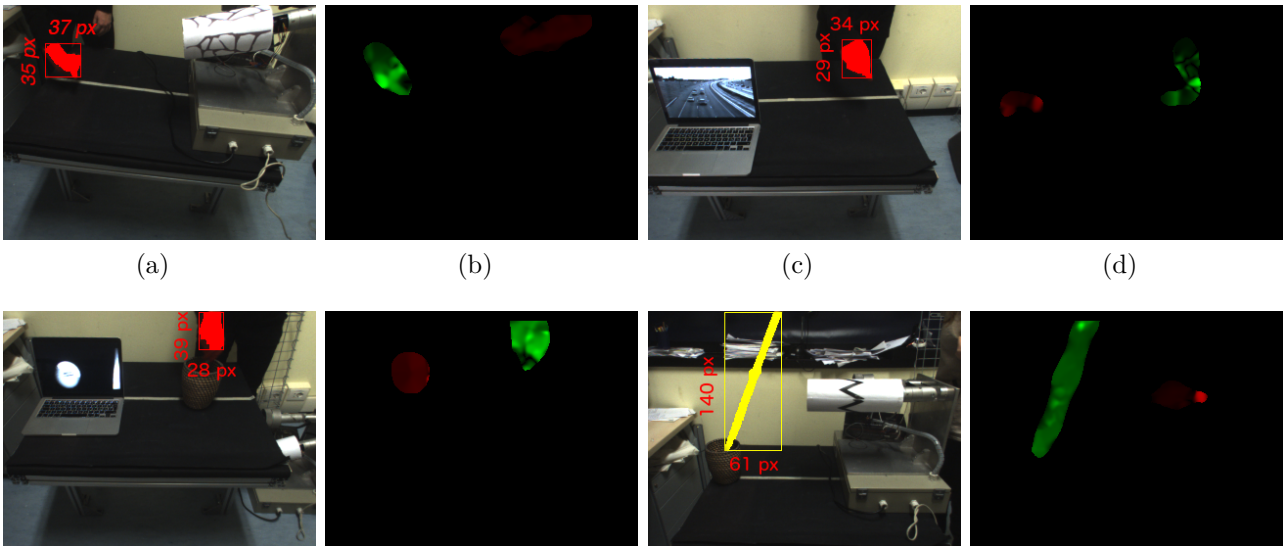
The Classifier is a module that wraps few novel functions around the Machine Learning library GURLS [83]. The module is programmable from remote (RPC port) allowing the user to control the modules, triggering different functionalities, the most relevant being the *training* of the model, and the *online recognition* to classify new observations. When model training functionality is activated, information coming from the opfFeatExtractor module is collected in a training set. When an appropriate amount of data is available, the module invokes a GURLS function to train a binary classifier using RLS (see Section 2.4.3). After the training, the model is adopted for online recognition, when at each time instant, new observed stimuli are described and classified. Classification is instantaneously based on the RLS score, generating a vote for the biological class if the score is positive, or for the non-biological class in case it is negative.

In order to partially correct instability of the final classification due to temporary failures, votes are collected into a temporal buffer of size 15. At each time instant, the final output of the classifier is based on a statistic of the votes in the buffer: when the majority of them (at least the 60%) is for a certain class, then the new observed event is labeled as an instance of that class, otherwise the system returns a temporary uncertain response as feedback.

Two classification processes receive, as input, the descriptors of A and B blobs and they give, as output, the classification results of A and B respectively. The two classifications are made by using a single classifier.

3.2.3 BioMerger

The bioMerger module synchronizes the feedbacks of two classification modules (for both A and B) with the binary masks provided by the opfFeatExtractor module. Consequently, the module generates a color image of size 320×240 where the detected blobs A and B are color-coded according to their associated labels as depicted in Figure 3.2b,d,f,h. In addition, the module prepares a topographic feature map designed to compete in the visual attention system PROVISION (PROactive VISion attentiON) [71]. The spatial map (320×240 grayscale image) indicates, with different level of saliency, the spatial locations where the biological movement is detected. The bioMerger streams a top-down command to the attentive system modifying the weight of biological motion in the competition for the attention, according to the confidence of the classification.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3.2: (a,c,e,g) Representation of the setup and color segmentation for biological motion samples.
(b,d,f,h) Biological (green) and non-biological (red) movements detected in (a,c,e,g).

3.2.4 PROVISION

The PROVISION is a log-polar attention system based on the computation model for attentive systems proposed in [38]. Through the combination of two fundamental processes, Winner-Take-All (WTA) and Inhibition-of-Return (IoR), the visual attention selects the most significant location in the saliency map. The selection of the saliency winning lo-

cation activates a ballistic oculomotor action (saccade) that brings the salient stimulus in the camera center of the drive eye. In this contribution we enhanced the collection of feature maps with an additional feature map responding to the presence of biological motion in the image plane. The mechanism triggers PROVISION autonomous focus of attention redeployment towards the biological movement which in turn triggers a oculomotor command to the IkinGazeControl. The PROVISION system provides to the rest of the network a command of suppression of the movement perception. The process resembles the suppression of the magnocellular visual pathway [13] and avoids excessive activation of the visual pathway caused by the egomotion during the saccade. The bioMerger leverages the PROVISION command of suppression to idle the process of extraction of the biological movement feature. This assures a stable perception-action loop comprising the extraction of the optical flow, the classification and the execution of oculomotor actions, such as saccades.

3.2.5 IkinGazeControl

The biological control [73] accounts for both the neck and eye control. The combination of two independent controls guarantees the convergence of the fixation point on the target. The controller solves the fixation tasks by implementing a biologically inspired kinematic controller that computes the robot joint velocities in order to generate minumum-jerk, quasi-straight trajectory of the fixation point. The controller is also enriched with additional models of biological oculomotor actions such as vestibular ocular reflex and passive gaze stabilization. The PROVISION system gives instructions to the IkinGazeController that autonomously coordinates 3 degrees-of-freedom (DoF) neck and 3-DoF eye system to show natural behaviour in the robot gaze. The rate obtained is compliant with the temporal dynamics of saccades in human attentive systems and the process as a whole resembles the infant predisposition to bias attention towards biological movement in the scene.

3.3 The method at work on the robot

In this section we present the experimental analysis performed online on the robot. We start analyzing the accuracy for the classification of biological motion, even in presence of different moving stimuli in the observed scene. Later, we will discuss the integration with the attention system and biological control system of the oculomotor action in the humanoid robot iCub.

3.3.1 Experiments on online learning

We describe the classification performances obtained on the robot. We first observe that, in typical applications involving proactive robots, it is fundamental to provide reliable training and classification in a reasonable time span. In the reported experiment, we show how this is achieved by parallelizing tasks in the software infrastructure.

To facilitate reproducibility both the biological and the non-biological stimuli are presented on a table (64cm of height) and at a distance of 64cm from the origin of the iCub frame of reference.

Training is performed starting from an initial condition without a priori knowledge, meaning that the robot lacks the abilities of discriminating between biological and non-biological motion. The training is performed online on the robot, replicating the situation where the operator interactively instructs the robot. Model selection is also performed online.

We first train the robot on the set of biological and non-biological categories already adopted for the offline analysis (see Section 2.5). In this case, we apply a filtering with a Gaussian mask to partially correct instantaneous noisy information that might affect the overall analysis. In our experiments we fix the width of the Gaussian mask to M =9, which we found to be a good compromise between accuracy of the results and computational efficiency. On average, each video lasts about $20''$.

We test the classification system by proposing a single stimulus from a subset of representative events categories in different portions of the iCub field of view. The obtained results are shown in the first part of Table (3.1). During the evaluation, we determine whether each received packet matches the expected response (column *AccuracyA*). The average accuracy obtained in case of a single stimulus with biological and non-biological motion is 0.98. The reported accuracy is obtained in asynchronous evaluation periods (column *Time*). A relevant aspect for robotic applications, requiring adaptability to context change in the environment, is the transmission rate (column *RateA*), which is reasonable for oculomotor actions such as saccades.

Finally, we consider an experimental scenario where two stimuli (A and B) are presented on different portions of the field of view. The analysis of the classification quality is reported in the second part of Table (3.1), where we show accuracy and transmission rate corresponding to the A and B stimulus, and the overall evaluation time. The average accuracy obtained in case of two stimului with biological and non-biological motion is 0.94. The classification of the motion is uncertain at the initial transient, due to the filtering of the features vector and the instability of the classification. In the table we report the accuracy excluded the initial time windows necessary for stable filtering of the result. Averaging the accuracies obtained on biological and non-biological samples in one stimulus and two stimuli cases, we obtain an

Stimuli	AccuracyA	RateA[pkt/s]	AccuracyB	RateB[pkt/s]	Time[s]
clouds	1.0	4.30	–	–	40
leaves	0.94	4.40	–	–	40
rolling dough	0.96	4.02	–	–	40
transporting	1.0	4.19	–	–	40
clouds-rolling dough	1.0	3.13	1.0	3.13	30
clouds-transporting	0.98	3.27	1.0	3.26	30
leaves-rolling dough	0.81	3.51	0.90	3.57	30
leaves-transporting	0.85	3.83	1.0	4.11	30

Table 3.1: Online classification results with one stimulus and two stimuli.

accuracy of 0.95.

Despite the number of classifications has increased, the decrease of the rate is limited and it has no effect on the gaze control. In fact, the software framework is designed to be scalable and the computation demand of multiple classifications is distributed across the processing node in the network. Overall, this set of experiments produces convincing results for what concerns accuracy. We only have a degradation on the pair *leaves + rolling dough* stimuli, due to discontinuities of the stimulus provided.

Figure 3.3 shows how the classifier generates response messages, for a biological stimulus (*rolling dough*). The score provided by the classifier is accumulated over a 15 frames temporal window. In this case, the response is constantly 1.0 indicating a correct classification as biological movement. The brief undetermined classification (classifier response: 0.0) is due to scores below zero in the previous window of 15 frames, as depicted in picture. The system recovers after few iterations and the classification returns to provide correct response giving evidence of robustness.

3.3.2 Experiment on integration with PROVISION and Gaze Control

The classification system is designed to reliably provide results to a broad range of software applications in the iCub network. To facilitate its use, we integrate the classification output with the masks produced by the opfFeatExtractor into a single mask. The mask is produced and provided to the network by our new bioMerger module. In this experiment, the output of the bioMerger module interfaces with pre-existing software: PROVISION and iKin Gaze Control [64]. The biological movement detector provides a feature map of biological movement and the level of confidence associated with classification.

The integration experiment described here includes two different stages: perception and action. For both the evaluation stages we produce a biological and a non-biological movement (distractor) and we determine how the position of the salient biological stimulus evolves over time. To determine the ground truth for the localization of the human activity in the scene, we adopt a color segmentation module and we perform experiments with a human subject wearing colored gloves. The localization based on color relies on a source of information alternative to the one exploited by our algorithm (i.e., motion), thus representing a dependable estimation for comparison (see Figure 3.2). For each case, beyond measuring the perceptual error, we run multiple saccades and extract the statistics on the errors due to the control stage.

In the evaluation of perception quality, we compare our estimated (u, v) position of the salient stimulus in the image plane provided by PROVISION with the segmentation of the moving region detected by the color segmentation (Figure 3.4). In this phase, no oculomotor command is generated and the fixation point of the robot is at the center of the scene in F=[-0.5, 0.0, -0.35]m where the frame of reference is located and oriented according to the iCub standards. In Figure 3.5 we show the distance in pixels between the two different locations. Notice that a mean distance in the range [20-40] pixels corresponds to a metric range [4-8] cm, given the distance of the camera from the stimuli (64cm).

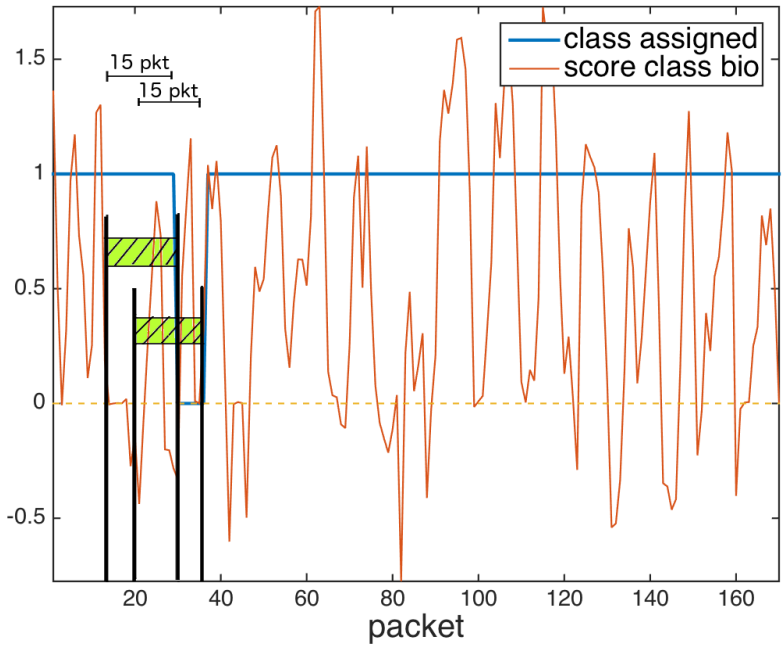


Figure 3.3: An example of the classifier generating response messages – *Rolling dough* case (bio = 1, nonbio = -1,"?" = 0).

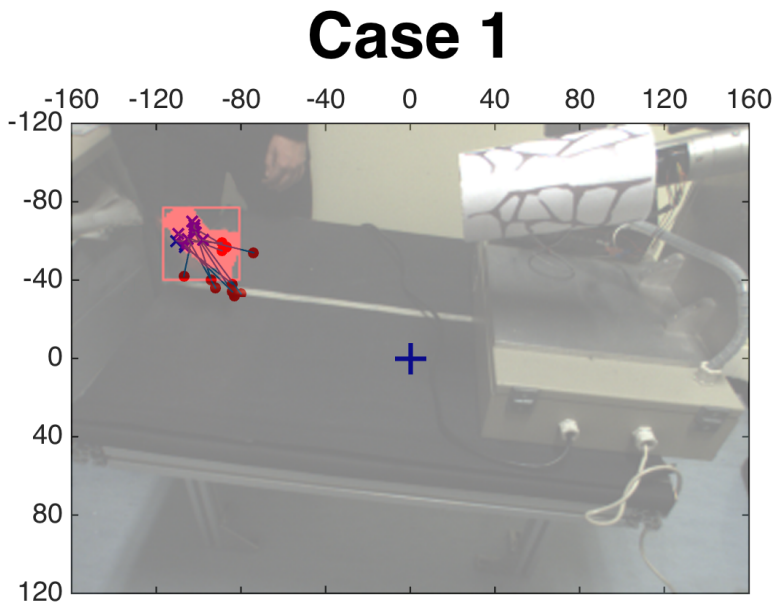


Figure 3.4: An example of perception errors, computed as the distances between the locations identified by the color segmentations (blue crosses) and the corresponding positions individuated by the biological motion detector (red dots). The dots and the crosses are printed on a more transparent version of the scene extracted from the robot point of view before the beginning of the oculo-motor action.

Case	Stimuli (A-B)	corr/tot
Case 1	**gesturing**-wheel random	11/11 sac.
Case 2	leaves-**writing subject1**	10/11 sac.
Case 3	cars-**gesturing**	15/15 sac.
Case 4	bouncing ball-**mixing**	11/12 sac.
Case 5	**mixing,no person**-wheel zigzag	11/11 sac.
Case 6	wheel random-**writing subject2**	13/13 sac.

Table 3.2: Number of correct saccades in integration experiment.

In the evaluation of the action quality, we analyze how the biological movement detector biases the proactive attentive system of the humanoid robot iCub. We assume that the classification of the motion is done when the camera of the robot is not moving, but we experimentally observed that a small movement of the eyes is acceptable, as small movements are discarded from the analysis thanks to the thresholding of the optical flow. The visual attentive system generates a saccade command and once the controller plans the relative saccade, the oculomotor action is executed bringing the center of the robot eye (fovea) to the most salient stimulus, winning in the competition between perceptual features. Considering

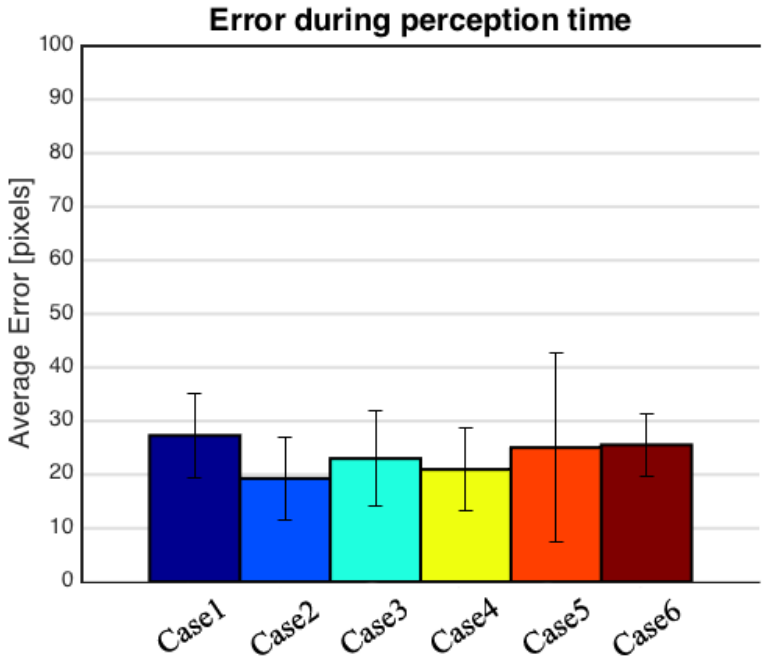


Figure 3.5: Average perception error in all cases. For the detailed list see Table (3.2).

the known distance of the stimulus from the stereo cameras, the gaze controller moves the fixation point to the target of interest (the biological movement in the scene). In PROVISION, a post-saccadic refinement mechanism based on visual feedback control can potentially refine the saccade. However we disabled such additional control to avoid unclear measurements on two distinct and concurrent visual processes on the robot. Notice that, as shown in Table (3.2), the system performs incorrect saccades in two different cases. The two incorrect saccades are due to a misclassification of the biological stimulus by the Classifier module: they have been discarded for both the evaluation of the perception and action quality.

In Figures 3.6 we show the position given by the color segmentation when the saccade starts (blue dots), and the trajectory of the position given by the saccadic commands (red line) towards the fovea (0,0), from when the saccade starts up to 2.5 seconds (as, from Figure 3.7 where we represent the velocity of the fixation point during the executing of the saccade; in other words, we can consider the time of its approaching to the target as the duration of the saccade). The semitransparent image overlapping the trajectory is to consider as the snapshot of one visual scene taken right before the oculomotor command saccade is triggered.

We measure the control error by computing the distance between the center of the fovea (0,0) and the position given by the saccadic command (red line of Figure 3.6) for the six typologies of trials in the previous perception stage. In Figure 3.8, we show the error from

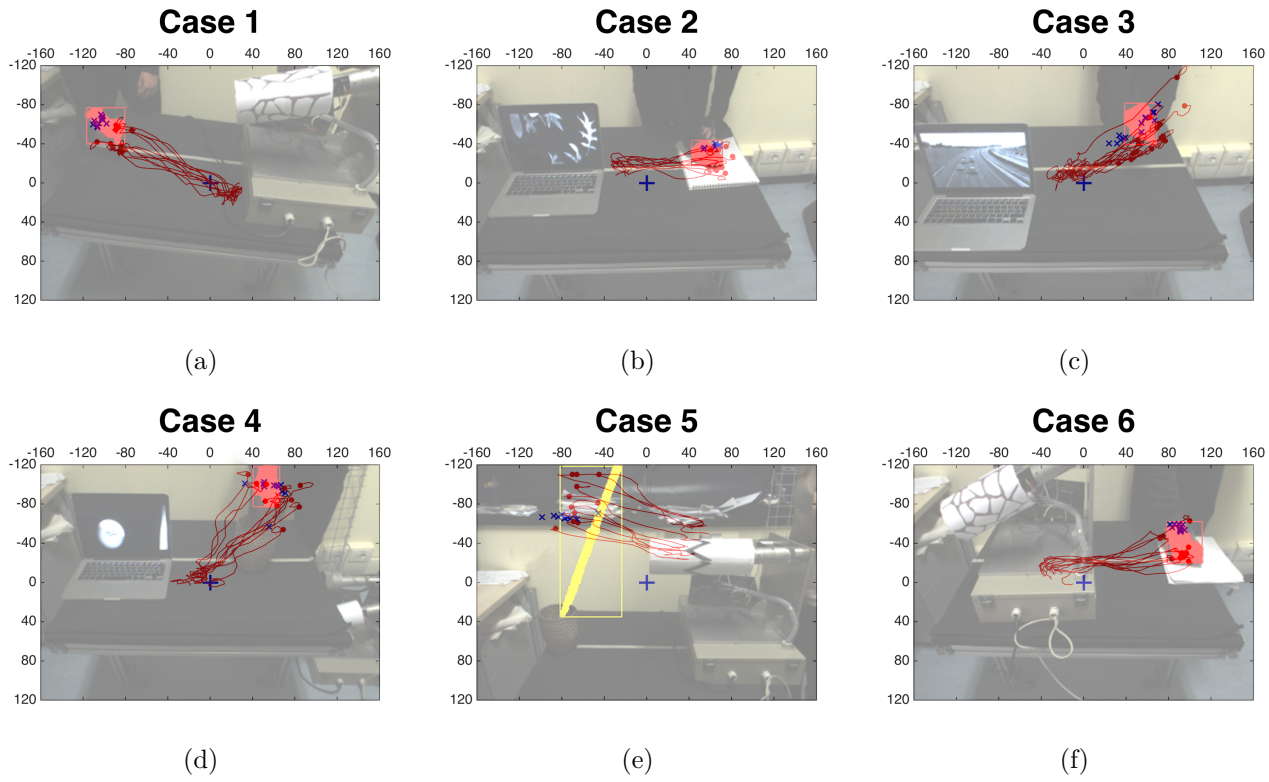


Figure 3.6: The position given by the color segmentation when the saccade starts (blue crosses), and the trajectory given by the saccadic commands (red line) towards the fovea (0,0).

the moment the saccade starts up to about 6 s. In the graphs of the control error the mean of the error (blue solid line) reaches immediately the quality threshold of 40pixels. The threshold is set according to our estimation that at a distance of 0.68m, 40pixel error is interpreted as a correct saccade from a human observer. The responses in Case1, Case3 and Case5 show overshoots, due to the relative position of the biological stimulus with respect to the resting position. All the responses converge to control errors below the 40 pixel threshold guaranteeing the expected quality of the control of the saccade to the biological movement. The oscillations after the transient are due to the response of the color segmentation that tracks moving stimuli after the end of the saccade and it is not related to the quality of the saccade generated by the system.

Then, we measure the distance between the center of the fovea (0,0) and the centroid of the color detection system (Figure 3.9): this can be referred as a global error, as it includes both the perception error and the control error.

Case5 is a very peculiar case as the color segmentation gives us the position of the center

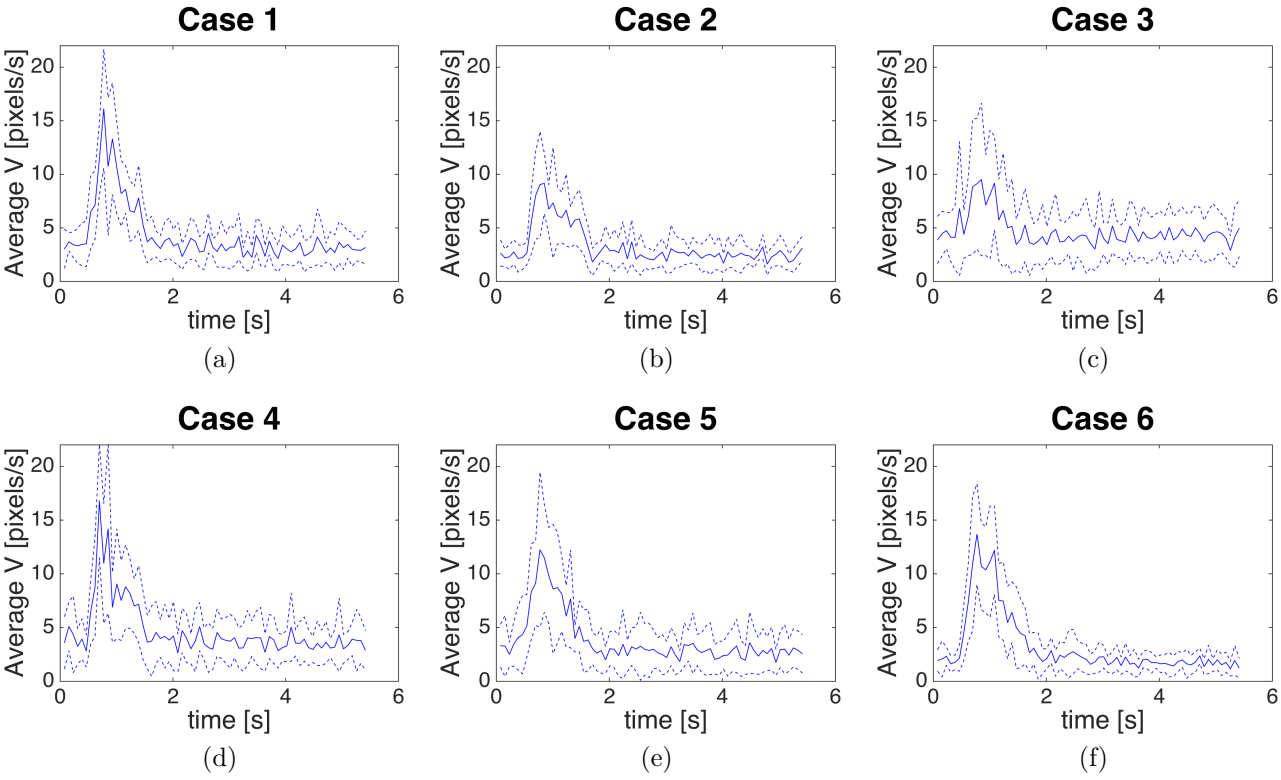


Figure 3.7: The blu line is the average velocity of the fixation point to reach the target for each case, the blue dotted line is the average velocity ± standard deviation.

of the rectangle around the stick (Figure 3.6e), while our module will give as oculomotor command the most salient position of the saliency map, which corresponds to the position of maximum optical flow. This leads to have a larger standard deviation in the perception error (Figure 3.5, Case 5) and a larger control error (Figure 3.8e). However, considering the goal of detecting humans in the scene, our method could be considered actually more accurate than color segmentation.

3.4 Discussion

This computational model can therefore enable an artificial agent to detect the presence of humans in its surrounding to provide the appropriate pro-social behavior, as we demonstrate by implementing it on the humanoid robot iCub. The video at this link https://youtu.be/wQ39oUq1eaA shows some real-world experiments of the proposed computational model.

The saccadic action performed by the robot as a consequence of the detection of human

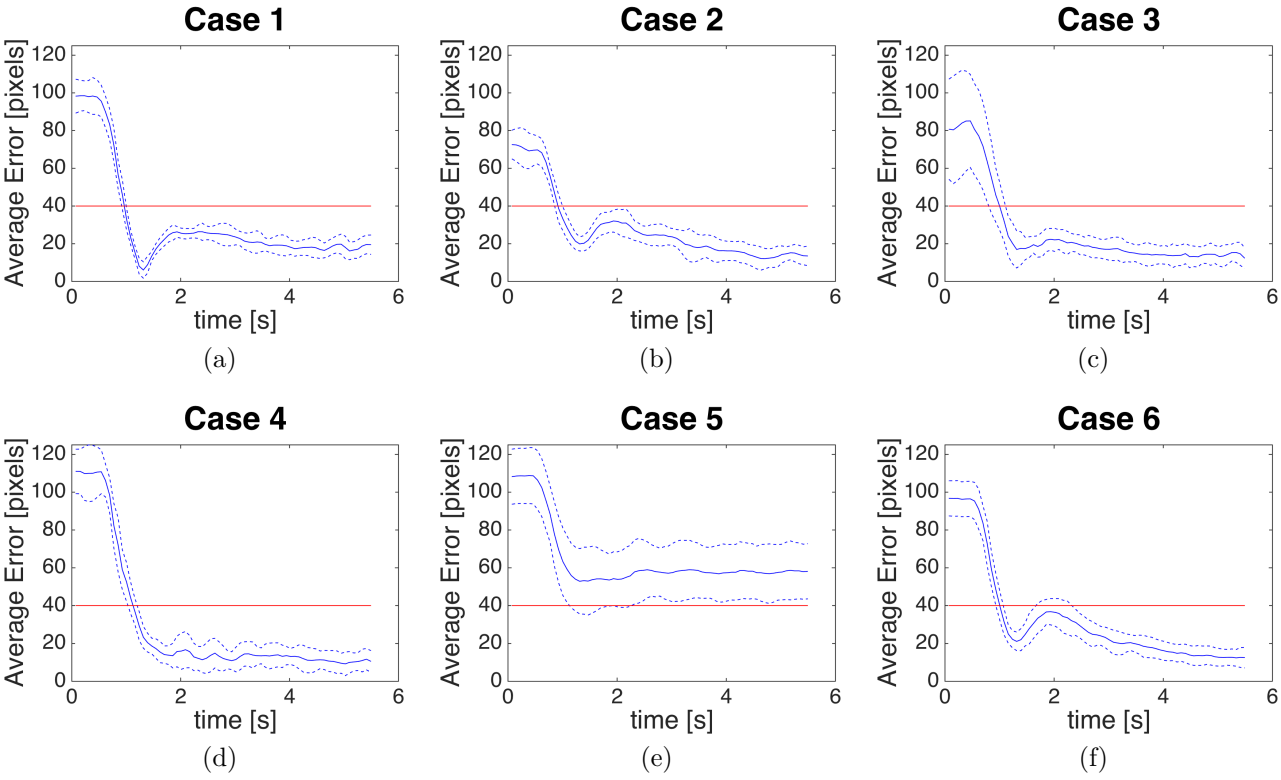


Figure 3.8: Control error. Distance between the position given by our module as saccadic command and the fovea (0,0). The blu line is the average error, the blue dotted line is the average error ± standard deviation. The red line is a threshold of 40 pixels.

activity in the scene, beyond providing the robot with a better view on the area where it is more probable that an interaction could start, also informs the human partner about the internal attentional status of the robot in the most intuitive approach. This type of gaze-based intuitive communication, commonly adopted in conversational agents and social robotics, has recently gained impact also in the field of small manufacturing, where Baxter (by Rethink robotics) exploits a screen with (non-functional) eyes, just to reveal its focus of attention. In our system the matching between the actual function of the eyes (i.e., cameras) and their ostensive value, increase even more the intuitive interpretation of the iCub actions.

In this respect, our work represents the first building block of the social abilities of the robot, that in the future will be exploited to categorize actions into different classes, an issue that we have started to address in [60]. Such capability can be of strategic interest for a broad community aiming at enabling effective interaction between human, robots and intelligent machines.

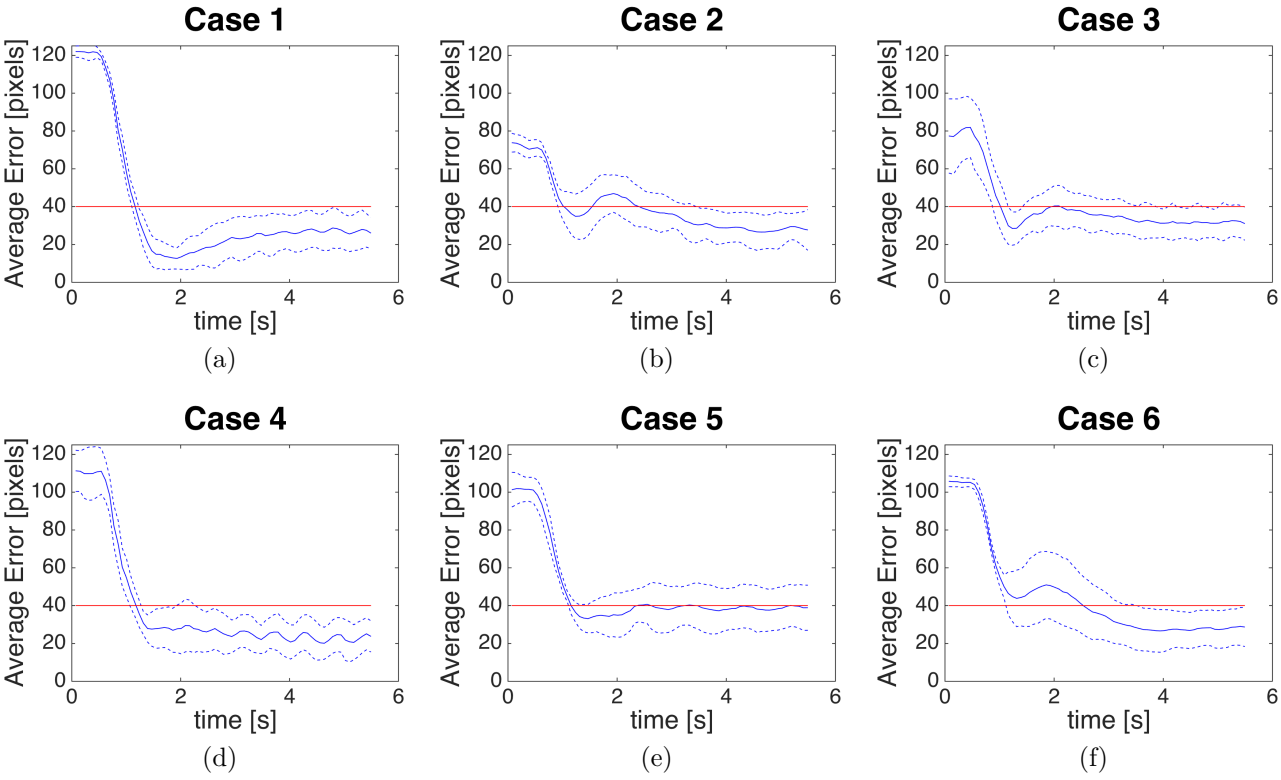


Figure 3.9: Global error. Distance between the centroid of the biological motion (given by color segmentation) and the fovea (0,0). The blu line is the average error, the blue dotted line is the average error ± standard deviation. The red line is a threshold of 40 pixels.

Chapter 4

Understanding and recognising actions

4.1 Introduction

In the previous chapter we showed that it is possible from a low level description of a visual motion to determine whether it belongs to a living being or to an inanimate object. This partition represents a fundamental building block in the development of interactive skills, but does not provide any information on the meaning of the observed movement. The goal of this chapter is to assess whether an extension of the description we proposed could help understanding which type of action is occurring in front of the robot, once the motion has been recognized as biological. Our goal is not classical action recognition, where possible classes of interest are clearly a priori identified. Rather we aim at individuating groups of actions sharing some similarity relationships, in terms of kinematics (for instance one group could be made by continuous actions and another one by actions with a starting point and an ending point), and at enabling the robot to recognize to which of these emerging groups belongs the action it is seeing. With this goal in mind we explore the concept of visual motion primitives, intended as a limited number of action sub-components necessary and sufficient to describe and reconstruct a wide range of different complex actions. We propose a kinematic definition of primitives based on simple visual features as velocity and curvature of the apparent motion (see below Section 4.4), and we verify whether such description enables the detection of similarity among different classes of actions or of the similarity between actions observed from different visual perspectives (Section 4.7).

The idea of motion primitives is an important topic in fields such as Human-Machine Interaction (HMI) and has applications in several areas, like intelligent systems, ambient

intelligence, natural interfaces and robotics. In the latter, the primitives have been mostly applied to motor control problems [75], as they are combined for making the robot able to perform manipulation actions on objects. Our innovation is exploiting the concept of motion primitives in a perception setting, with the idea, in the future, of using it for building a unique paradigm where perception and action are strictly linked to each other: in this scenario, the robot will understand the human motion by observing it and decomposing it in primitives, and will be able to perform the motion itself.

In particular, we propose a system that, starting from a video sequence, is able to learn in an automatic way some motion primitives and to represent the video sequences as a proper combination of them. To this purpose, we use the dictionary learning technique.

4.2 State of art

The literature about gesture, action, and activity recognition, in the field of computer vision is huge: we refer the reader to [108, 1, 66] for a complete survey of the topic. Regarding the works on action recognition for the specific application in robotics, we refer to [25]. Here we will focus more on works about the recognition of actions and of human poses based on optical flow and on works that share our goal, that is "actions categorization", and in particular on research related to the idea of motion primitives.

Optical flow has been used in several 2D and 3D model-based methods with the goal of estimating the human pose from videos. Methods [11, 41, 79, 101], for instance, link the 2D image motion to the parameters of articulated human figures. In [24] the authors generate training flow fields from different views using a synthetic character and motion capture data. By applying PCA, a low-dimensional representation of the flow is built, and simple activities are represented in that low-dimensional space. Efros at al. [22] use optical flow to estimate pose of low resolution people in video. As the flow information is limited and noisy, it is treated as a spatio-temporal pattern, which becomes a motion descriptor, used to query a database for the nearest neighbor with a similar pattern and 2D and 3D pose that is known. In this method, similar sequences of fully body poses in the database are required. In Bissacco et al. [5] they use a boosted regression method to recognize pose from image and motion features, which are derived by image differences and not really optical flow. Some recent methods use optical flow to augment traditional 2D pose estimation methods. In [31] the authors use optical flow to help in the segmentation of body parts while reasoning about pose, segmentation and motion. In [39] the authors use images and flow to train a deep convolutional neural network (CNN) to estimate upper body pose. Nevertheless, these methods use optical flow just as an additional information, and rely mostly on other image

cues.
In our work, we investigate how far we can go by using just features computed from the optical flow. Regarding the problem of actions classification, the classical approaches [108, 1, 66] in computer vision aim at achieving the best possible performance by using a rich information extracted from the video. A direct qualitative comparison with these methods is not feasible as the goal is different: indeed, we use a low level description based on primitives focusing just on the kinematics and in particular on velocity and curvature of the movement. Our goal is to discuss what we can achieve with this low level information, which resembles the low ability of infants in motion perception. Moreover, the use of kinematic features leads towards another topic of great interest for robotics, that is the possibility, for the robot, of not just understanding the human motion in front of it but also to be able to perform this motion itself: the ambition is, indeed, to design unique methods to model both perception and action, while usually these are two different problems addressed with different methods.

Regarding motion primitives, although there is no common definition of them (often referred to also as segments), an accepted idea is that actions can be naturally decomposed, in phases and subgoals [81]. According to the motion primitive paradigm, each action phase corresponds to a primitive. For instance, for manipulation tasks, there are typically a series of action phases on which objects are grasped, moved, brought into contact with other objects, and finally released [30, 17]. Most of previous works on this topic in robotics refer to the problem of motion planning using combinations of primitives [36, 44, 48], and are often applied to manipulation tasks, in some cases with the interesting approach of learning sub-actions and/or their goals [107, 81]. While the effort in reproducing human motion for robot motor planning is apparent, very little has been done to reveal and exploit the biological mechanisms allowing for an efficient use of the motion primitives from a perception standpoint.
From the human perception point of view, in [37] it has been shown that humans, if being asked to segment hand/arm actions, base their choice on where to place segmentation marks on low level kinematic information, i.e. change in direction, velocity and acceleration of the wrist. From a computational point of view, in [68] the authors propose a computational representation of human action to capture the view-invariant dynamic instants where a change in the speed and direction of the trajectory occurs.
We decided to focus more on kinematics, that is the intervals between two dynamic instants instead of the dynamic instants themselves: these intervals in time represent the basis on which the primitives are built.

4.3 Adaptive data representation

In order to pursue the goal of representing actions through motion primitives, these have to be found out. In this section, we introduce the dictionary learning technique, a common approach to learn adaptive representation from data, that can be used to discover the motion primitives.

4.3.1 Dictionary Learning

The purpose of dictionary learning is to learn a set of atoms, that is a dictionary $\mathbf{D}$, that can capture the essence of all the data of the considered dataset, such that every datum can be approximately expressed as a linear combinations of elements in the dictionary as $\mathbf{x} \simeq \mathbf{Du}$: this mapping brings the descriptors into a common reference frame, allowing a more effective comparison between them over time. The dictionary $\mathbf{D}$ is a matrix $K \times d$, where K is the number of atoms of the dictionary and d number of features contained in each descriptor. Typically $K > d$, therefore D is a over-complete basis.
For doing dictionary learning in an unsupervised way, typically, the K-Means [19] technique is adopted. It is based on minimizing the following reconstruction error:

$$\begin{aligned} &\min_{\mathbf{D},\mathbf{U}} \|\mathbf{X} - \mathbf{DU}\|_F^2 \\ &\text{s.t.} \quad \text{Card}(\mathbf{u}_i) = 1, |\mathbf{u}_i| = 1, \mathbf{u}_i \succeq 0, \forall i = 1, \ldots, T \end{aligned} \tag{4.1}$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T]$ is the set of descriptors used for learning the dictionary, $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_T]$ are the cluster membership codes, with each $\mathbf{u}_i \in \mathbb{R}^K$, K dictionary size. The constraint $\text{Card}(\mathbf{u}_i) = 1$ means that only one element of $\mathbf{u}_i$ is non-zero, and $\mathbf{u}_i \succeq 0$ means that the element must be greater than zero, i.e. each local descriptor belongs to one cluster. Another way to do dictionary learning is to usa a Gaussian Mixture Model (GMM) [74] that generalizes the K-Means algorithm by learning a probability density function $p(\mathbf{x}_i \,|\theta) = \sum_{k=1}^{K} p(\mathbf{x}_i|\mu_k, \mathbf{\Sigma}_k)\pi_k$, where π_k are the mixing coefficients and $p(\mathbf{x}_i|\mu_k, \mathbf{\Sigma}_k)$ is a d dimensional Gaussian. Each descriptor $\mathbf{x}_i$ is then soft-assigned to clusters:

$$u_{ik} = \frac{p(\mathbf{x}_i|\mu_k, \mathbf{\Sigma}_k)\pi_k}{\sum_{j=1}^{K} p(\mathbf{x}_i|\mu_j, \mathbf{\Sigma}_j)\pi_j} \quad k = 1, \ldots, K. \tag{4.2}$$

The Mixture Models learning is done by an Expectation-Maximization (EM) algorithm.
In the case of sparsity based methods [106], the dictionary is learned by minimizing the

following objective function:

$$\min_{\mathbf{D},\mathbf{U}} \|\mathbf{X} - \mathbf{D}\mathbf{U}\|_F^2 + \lambda\|\mathbf{U}\|_1 \tag{4.3}$$

where $\|\cdot\|_F$ is the Frobenius norm. L_1-norm yields to results characterized by sparsity and robustness. Another possibility is to use other penalties like the L_0-norm, but in this case it becomes an NP-hard problem where the convergence to an optimal solution by the algorithm is not taken for granted. It is worth noting that fixing $\mathbf{U}$, the optimization of Equation (4.3) becomes a least square problem, while fixing $\mathbf{D}$, it becomes a linear regression with the sparsifying norm L_1. Given a fixed $\mathbf{D}$, an efficient algorithm to solve the problem in Equation (4.3) is the *feature-sign search* one. This algorithm searches for the sign of the coefficients $\mathbf{U}$; indeed, considering only non-zero elements the problem is reduced to a standard unconstrained quadratic optimization problem (QP), which can be solved analytically. Moreover it performs a refinement of the signs if they are incorrect. For the complete procedure we refer the reader to [49]. In practice, the algorithm employed for learning the dictionary is not crucial, and dictionaries learned with K-means are enough to obtain the best results.

4.3.2 Coding

Once a dictionary is learnt, the input features $\mathbf{x}_1, \ldots, \mathbf{x}_M \in R^d$ are mapped into a new space of codes $\mathbf{u}_1, \ldots, \mathbf{u}_M \in R^K$. Several coding operators can be used, each of them producing different output. For instance, BOW-like methods give as output the amount of atom contribution to the linear combination or the visual word occurrences. These coding operators minimize the following reconstruction error:

$$\begin{aligned} \mathbf{u}_i = &\arg\min_{\mathbf{u}} \|\mathbf{x} - \mathbf{D}\mathbf{u}\|^2 + \lambda \mathrm{R}(\mathbf{u}) \\ &\text{s.t.} \quad \mathrm{C}(\mathbf{u}) \end{aligned} \tag{4.4}$$

where, depending on which regularization terms $\mathrm{R}(\mathbf{u})$ and constraints $\mathrm{C}(\mathbf{u})$ are used, different algorithms can be derived:

Vector Quantization (VQ) [47] This algorithm, given a dictionary $\mathbf{D}$, minimizes the following reconstruction error of $\mathbf{x}_i$:

$$\begin{aligned} &\min_{\mathbf{u}_i} \|\mathbf{x}_i - \mathbf{D}\mathbf{u}_i\|^2 \\ &\text{s.t.} \quad \mathrm{Card}(\mathbf{u}_i) = 1, |\mathbf{u}_i| = 1, \mathbf{u}_i \succeq 0 \end{aligned} \tag{4.5}$$

that is the same as Equation (4.1) with a fixed dictionary.

Sparse Coding (SC) [106] In this algorithm, each descriptor is represented by using just a subset of atoms, the more relevant ones. It minimizes:

$$\min_{\mathbf{u}_i} \|\mathbf{x}_i - \mathbf{D}\mathbf{u}_i\|^2 + \lambda\|\mathbf{u}_i\|_1 \tag{4.6}$$

where the L_1-norm makes the learned representation able to capture the main patterns of the descriptors and produces sparsity. The parameter λ is chosen as a trade-off between the signal approximation and the sparsity. This equation corresponds to Equation (4.3) with a fixed dictionary.

Locality-constrained Linear Coding (LLC) [102] Given that locality produces sparsity and not viceversa, one can give priority to locality, and this algorithm does this. Here, just a subset of coded vector $\mathbf{u}_i$, whose components are related to the k nearest neighbors of the input $\mathbf{x}_i$ and the atoms in $\mathbf{D}$ are selected. The codes are obtained minimizing:

$$\begin{aligned} &\min_{\overline{\mathbf{u}}_i} \|\mathbf{x}_i - \overline{\mathbf{D}}\overline{\mathbf{u}}_i\|^2 \\ &\text{s.t.} \quad \mathbf{1}^{\mathrm{T}}\overline{\mathbf{u}}_i = 1 \end{aligned} \tag{4.7}$$

where $\overline{\mathbf{u}}_i$ are the components of $\mathbf{u}_i$ related to the k nearest neighbors of $\mathbf{x}_i$ in $\mathbf{D}$, denoted with $\overline{\mathbf{D}}$. Codes entries that are not associated with any neighborhood in the dictionary are set to zero.

New coding methods that have been proposed recently do not minimize any function but rather rely on high-level statistics: in this way, large and informative representations can be achieved even with small dictionaries. Between these methods, we mention Super Vector coding [109], Fisher Vector [74] or its non-probabilistic version, VLAD [40]. They are too demanding for real-time and robotics applications due to the final descriptor size.

In conclusion, in order to better distinguish the classes, we decide for a sparse representation of the data, choosing the Sparse Coding technique.

4.4 Motion representation using visual primitives.

In this section, we discuss how to represent an action as a suitable combination of visual *motion primitives*. First we devise a method to segment an action into *sub-movements* and we derive the *motion primitives* as *atoms* of a *dictionary* learnt from data. Then we discuss how to combine such primitives to obtain a meaningful description of the action.

To do that, we go through the following steps:

- **Data extraction.** At each time instant t, we first extract the *optical flow* for each point of the moving region, we compute the average of it and then we compute the tangential velocity and the curvature [69] of human end-point movements, as described in Section 2.4. At this stage, each video is represented by temporal sequences of tangential velocity and curvature.

- **Segmentation.** We split the velocity sequence over time in *sub-movements* detecting, in an automatic way, points that correspond to a *Start*, *Stop*, *Change* in the dynamics of the action, which refer to the dynamic instants cited in Section 4.2. We used two rules of segmentation for different type of actions:

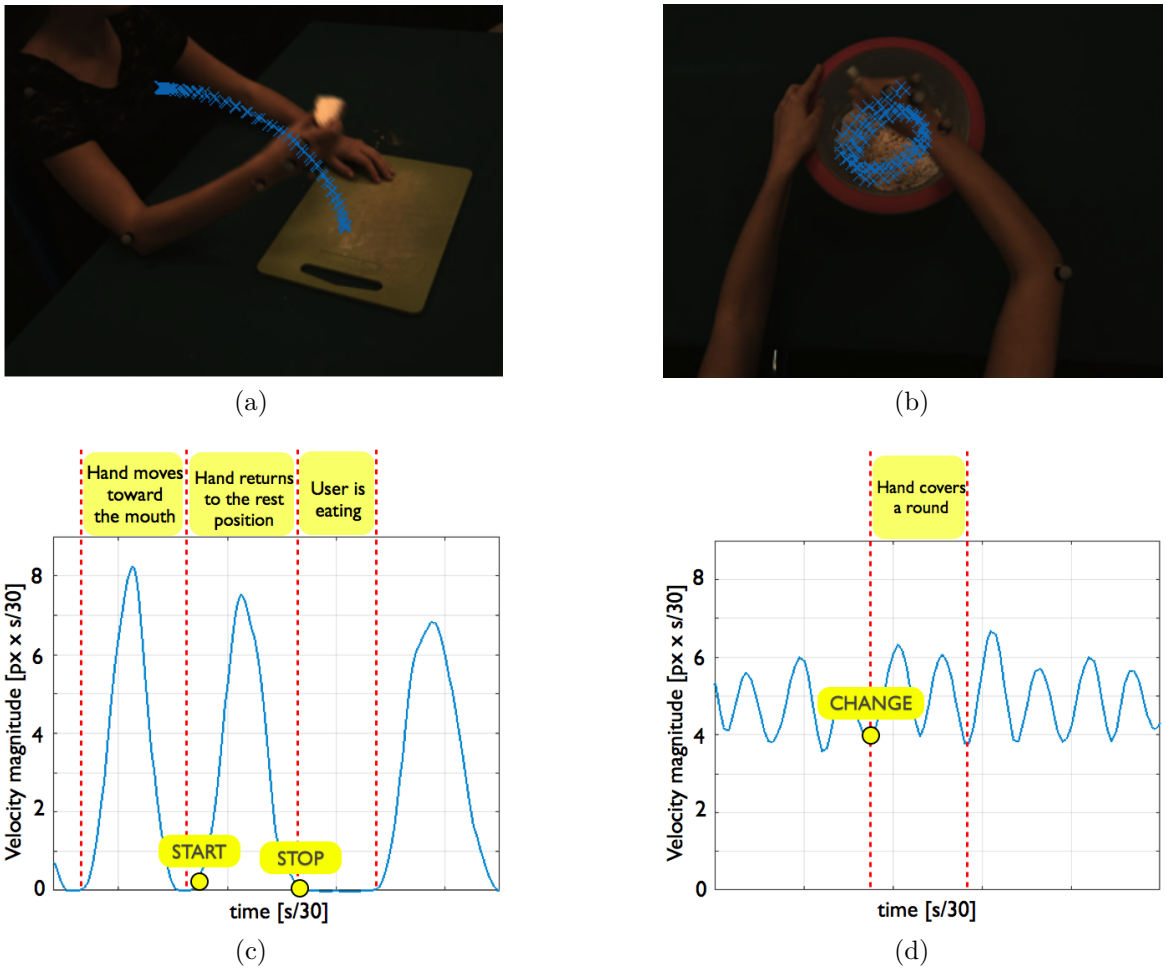
  - **Rule 1**: the velocity is segmented where it goes to 0, and this rule can be applied to actions which are not continuous, which have points of *Start* and *Stop* (like *Eating*, see Figures 4.1a, 4.1c). For actions that do not have points of *Start* and *Stop*, the Rule 2 is applied.

  - **Rule 2**: the velocity is segmented where there is a maximum in the curvature, as it corresponds to a local minimum of the velocity, and this rule can be applied to segment continuous and repetitive actions that do not present points of *Start* and *Stop* but point of *Change* in the dynamics (like *Mixing*, see Figures 4.1b, 4.1d).

  As a result of the segmentation procedure, we obtained the *sub-movements* of the velocity as depicted in the examples of Figure 4.1. For each of the two datasets of *sub-movements* (each one corresponding to a different rule of segmentation), we build a *dictionary* of *motion primitives*.

- **Representation of sub-movements with dictionary.**

  - **One dictionary for action.** First of all, we treat each action separately and we build a dictionary for each action. For doing it, for each action we take all the sub-movements obtained from the segmentation and we do a K-means clustering obtaining a dictionary of atoms (with a number of atoms equal to the value of K considered for the K-means). In this case where a dictionary is built starting from one action, a low value of K should be enough to cover all the sub-movements of the action.
  - **Unique dictionary.** Then, we treat all the actions together and we do the clustering for the sub-movements obtained with the segmentation of all the actions



(a) (b) (c) (d)

Figure 4.1: (a,c) Video sequence of *Eating*, a movement with dynamic instants of *Start* and *Stop*, where the Rule 1 is applied to segment it.
(b,d) Video sequence of *Mixing*, a continuos movement with no point of *Start* and *Stop*, but with points of *Change* in dynamics, where the Rule 2 is applied to segment it. One round of the hand includes two *sub-movements*.

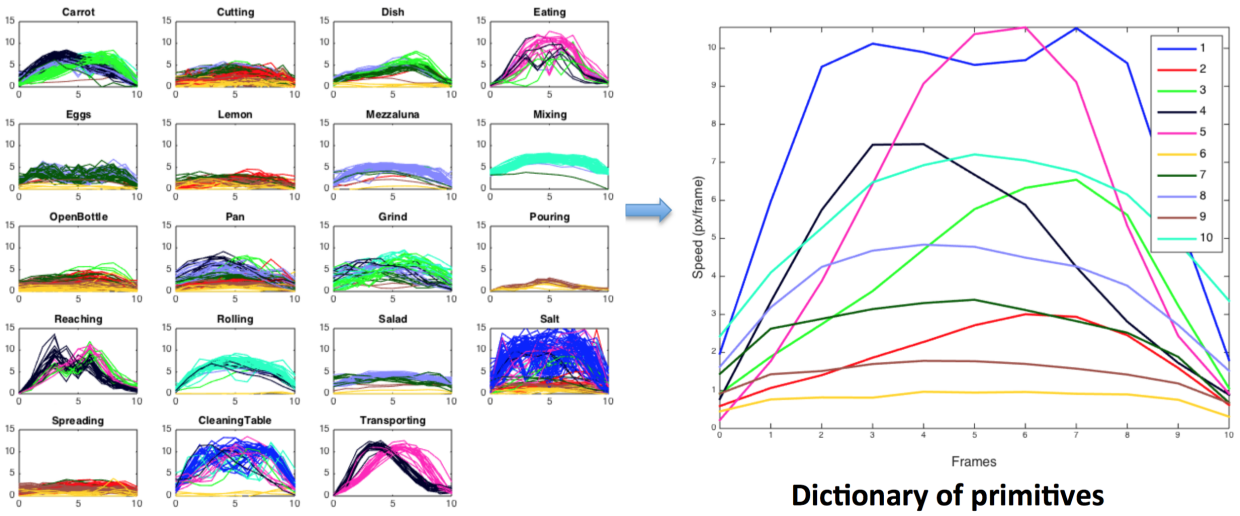


Figure 4.2: Building a unique dictionary of motion primitives.

of the training set, building a unique dictionary (case example in Figure 4.2). In this case, the value of K should be increased as we have to deal with a lot of different actions at the same time and we need a number of atoms that is enough to represent all of them. After building the dictionary, each sub-movement of the training set is then reconstructed as an approximation of a linear combination of some of the atoms in the dictionary, using the sparse coding technique described in Section 4.3.2, and represented as the sequence of weights used for each atom in the reconstruction (case example in Figure 4.3).

At the end of this procedure, given a video V representing a given action, we finally describe each sub-movements v_i of V as the feature vector $[u_1, u_2, ...u_K]$, where u_j are the coefficients/weights assigned to each atom (some of them are equal to 0) and K is the number of atoms of the dictionary.

4.5 Analysing motion primitives

Given the data representation described in the previous section, our purpose is to capture similarities between actions.

First, we address the problem with an **unsupervised approach**, where we perform a K-means clustering of the data, with different values of K, to see how the sub-movements of the actions are grouped together inside the clusters. It is important to notice that this K-means procedure is unrelated and subsequent to the one described in the previous section. From the results of this step we infer the complexity of the problem, that is characterized by

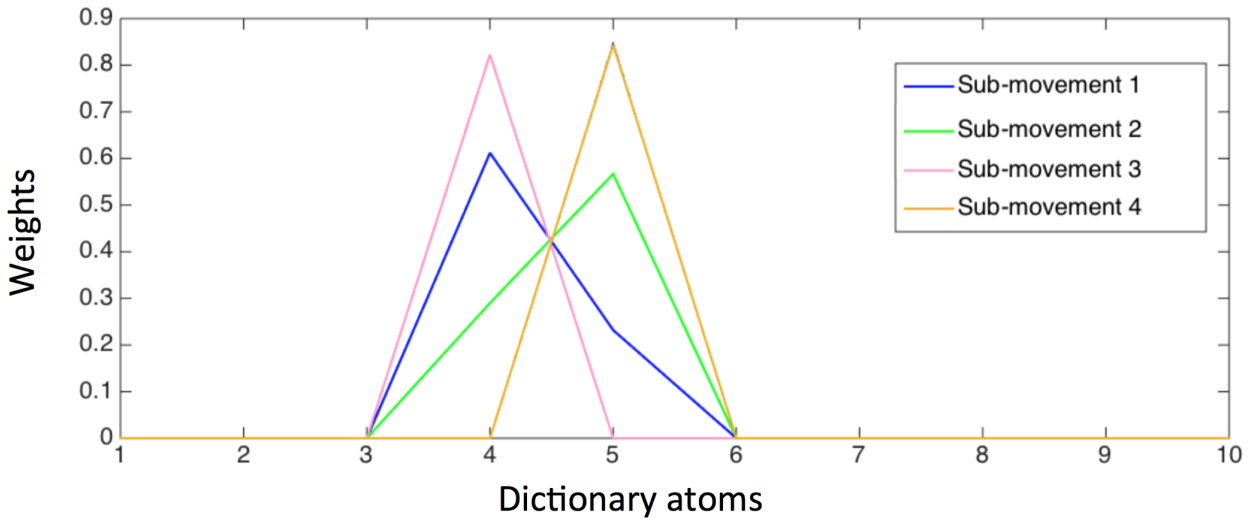


Figure 4.3: Representation of the sub-movements of the actions as linear combination of dictionary atoms (sparse coding).

intra-class variability, as sub-movements of the same action look different from one another, and by inter-class similarity, as sub-movements of different actions look similar.

Then, following a **supervised approach**, we perform a classification of the actions. We first train a binary classifier per class, with the samples of that class as positive samples and all other samples as negatives: given an action, in the testing phase, the model can say if a new data is of that action or not. After that, we build a multi-class classifier with one-vs-all approach, where a binary classifier per class is built, and prediction is then performed by running these binary classifiers and choosing the prediction (i.e the action) with the highest confidence score. To learn the relationship between input and output in a predictive way, we adopt a *Regularized Least Squares* (RLS) classifier. Both the binary classification and the RLS have been already explained in Section 2.4.3 in detail.

4.6 Experimental results on several dictionaries

In this section we present some early results obtained on video sequences of a multi-view dataset we acquired (Appendix A), composed of 19 actions (reported in Table (4.1)). The purpose of these early experiments is to investigate our data following the procedure described in Section 4.4 using one dictionary for action, focusing only on one point of view (the frontal one).

For each action we take all the sub-movements obtained from the segmentation of the video sequences of the training set and we do a K-means clustering obtaining a dictionary

1	Grating the carrot
2	Cutting the bread
3	Cleaning a dish
4	Eating
5	Beating eggs
6	Squeezing the lemon
7	Cutting with a mezzaluna
8	Mixing
9	Open the bottle
10	Turning the omelette
11	Pestling
12	Pouring water
13	Reaching an object
14	Rolling the dough
15	Washing the salad
16	Salting
17	Spreading cheese on a bread
18	Cleaning the table
19	Transporting an object

Table 4.1: Dataset of cooking actions.

of atoms (with a number of atoms equal to the value of K_D considered for the K-means). We repeat it with $K_D = 1, 2, 3$ as the actions we consider are mostly characterized by 1, 2 or 3 motion primitives. Then, in order to understand which dictionary is the best one for each action, we reconstruct each sub-movement using the dictionaries obtained, and we then take into consideration the dictionary which corresponds to the first K_D with a mean error of reconstruction < 0.3 (if, for a certain action, all the dictionaries have an higher error, we take the dictionary related to the maximum K_D, that is $K_D = 3$). We then reconstruct each sub-movement of the test set as a linear combination of the atoms of the dictionaries chosen for each action. In the error matrix of Figure 4.4, the mean errors of reconstruction for each action using all the dictionaries are reported: in the cell (i, j), for instance, there is the mean error obtained by reconstructing the sub-movements of the action i with the dictionary obtained for the action j. We can observe that it is possible to identify some blocks in the matrix, meaning that identifying some similarities between actions is feasible.

These are just early results that allow us to understand better our problem, but that are not conclusive with respect to the goal we have, that, instead, will be addressed in a deeper way in the next section.

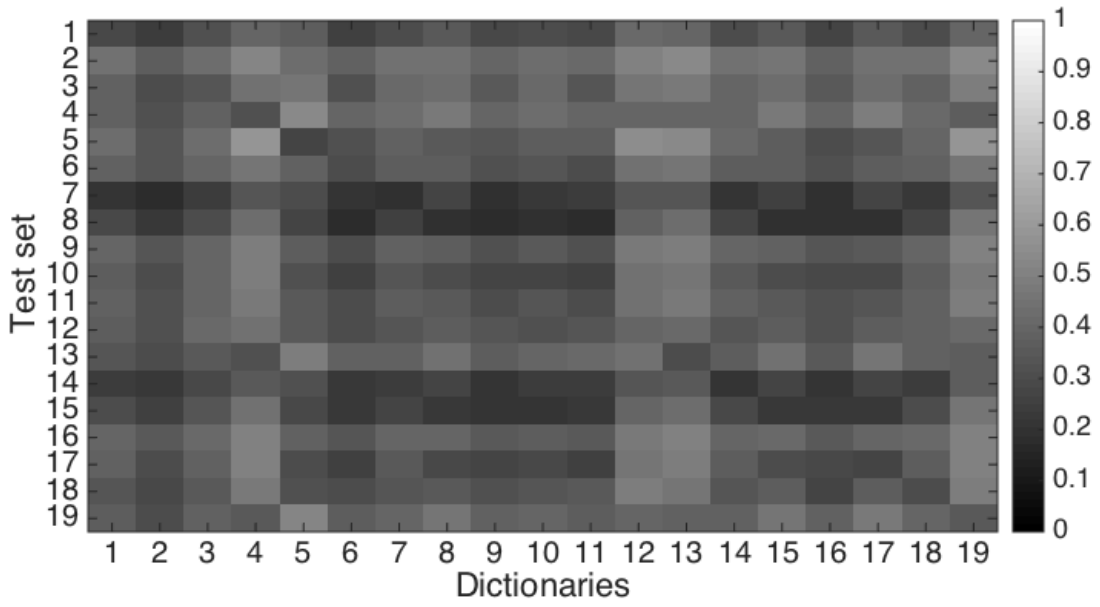


Figure 4.4: Error matrix obtained with the reconstructions of sub-movements of actions with the learnt dictionaries.

4.7 Experimental results on single dictionary

In this section we present the results obtained on video sequences of a multi-view dataset we acquired (Appendix A), composed of 19 actions (reported in Table (4.1)), by using one single dictionary of motion primitives, that means by treating all the actions together.

The purpose of these experiments is to evaluate if it is possible to represent complex dynamic events with the tools used so far. We start from addressing the problem of actions similarity, focusing only on one point of view (the classical one, i.e. the frontal point of view), and we end by comparing different points of view.

We first do unsupervised learning to carry on an analysis to understand the complexity level of the data we are dealing with and how much separate the classes are, in order to infer how the actions taken into consideration are related to each other. Then, we discuss the results obtained by doing supervised learning, in a multi-class classification framework, where each class is represented by an action.

All these results concern the frontal point of view, resembling the case in which a robot has to understand the action performed by a human in front of it, or the case of an infant observing the mother. Finally, following a developmental approach, we discuss how adding data recorded by different points of view can improve the classification results.

4.7.1 Unsupervised learning

To represent the data, we follow the procedure described in Section 4.4. First of all, we split the velocity curves of the actions in sub-movements, and we perform a K-means clustering with $K_D = 15$ clusters, obtaining a dictionary of 15 atoms (Figure 4.5). The size of the dictionary is fixed to 15 as we empirically observed that it can describe well all the sub-movements of the actions in the dataset we used. From Figure 4.5, we can observe symmetric and asymmetric bell-shaped profiles, already known in human movement science [54]. All the primitives are composed by a single-peaked bell-shaped velocity profile, while just one (the 6th atom in Figure 4.5) is double-peaked. Regarding the first group of primitives, it is possibile to have an idea of the quantitative differences between them with the Table (4.2).

Given this representation, we group the data with a K-means clustering with $K_D = 19$ to see how the sub-movements of all the actions are distributed inside the clusters and how much the clusters are similar to each other. At the beginning, the choice of K_D is 19 because the classes are 19, and we would like to give them the possibility to be all divided in different clusters, without forcing some of them to be grouped together. In Figure 4.6 we report the distances between the centroids of the clusters: we can observe that some clusters are more linked to some others (dark cells), and that other ones are really different with respect to anyone else (whiter rows and columns). Then, as we have also the labels of the clustered data, we can see which classes fall inside each cluster: in the histograms of Figure 4.7, for each cluster we count how many data of a certain class have been associated to that cluster. Observing Figure 4.7, we may derive a list of the classes that are representative of a certain cluster (Table (4.3)), considering:

- the first class with more samples if
  - the second class contains $< 80\%$ of samples of the first one;
- the first and the second class with more samples if
  - the second class contains $> 80\%$ of samples of the first one,
  - the third class contains $< 80\%$ of samples of the second one;
- the first, the second and the third class with more samples if
  - the second class contains $> 80\%$ of samples of the first one,
  - the third class contains $> 80\%$ of samples of the second one,
  - the fourth class contains $< 80\%$ of samples of the second one,

given that all the classes accepted contain more samples than a certain threshold th (set to 10).

The cells of Table (4.3) that are empty refer to clusters that contain no class which meets the previous requirements.

From Figure 4.7 and Table (4.3) we can observe how some classes, in particular 2 and 9 are contained in several clusters and they appear to have a high intra-class variability, meaning that they have samples that are different to each other and that are well represented by different centroids, or that contain primitives which occur in different actions.

To simplify the problem, we eliminate some information in order to have a better understanding of what remains: we replicate the K-means clustering eliminating these two classes and with $K_D = 17$, obtaining Table (4.4). Here the same problem is observed with the class 17, so we do the same considering all the classes but 2, 9 and 17, with $K_D = 16$ and $K_D = 8$. We use also $K_D = 8$ to force some classes to be grouped together, to study the similarities between them. In this case, as the number of samples of each clusters is higher because there are less clusters, we also increase the threshold th of the minimum amount of samples contained in one cluster to the value of 15. From the histograms obtained with both values of K_D, we then list here the classes that can be considered grouped (together with some other classes or alone) in a certain cluster (Tables (4.5) and (4.6)).

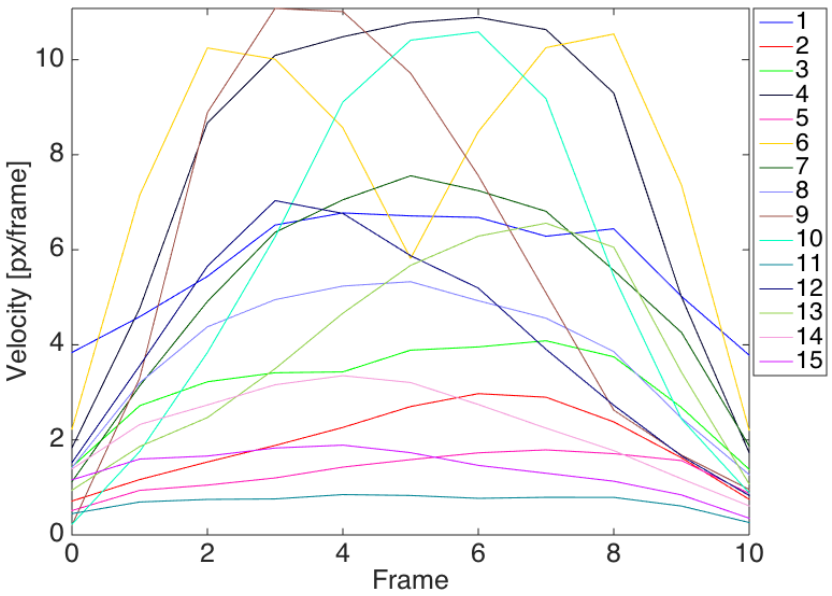


Figure 4.5: Dictionary of 15 atoms.

From the tables we can observe that some clusters contain some classes together (e.g.

Primitive	Maximum speed (px/frame)	Factor b
1	6.78	0.45
2	2.97	0.64
3	4.09	0.73
4	10.89	0.64
5	1.79	0.73
7	7.56	0.55
8	5.33	0.55
9	11.08	0.36
10	10.58	0.64
11	0.85	0.45
12	7.03	0.36
13	6.56	0.73
14	3.35	0.45
15	1.89	0.45

Table 4.2: Quantitative description of the learnt primitives with single-peaked, bell-shaped velocity profiles of Figure 4.5 in terms of maximum speed (peak speed) and factor b, computed as the acceleration time (i.e. the time to peak) divided by the total duration time of the primitive.

Custer	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Classes	10,17	9	1		1,2	9,2	14	9	9	2	5	8		19,13,2	2	7	14	9	9

Table 4.3: All the 19 classes grouped in 19 clusters.

cluster 6 of Table (4.6) contains classes 8 and 15) while others contain just one class (e.g. cluster 8 contains samples of class 7). It is worth noting that classes of similar actions, the classes 8,15 (*Mixing*, *Washing the salad*) and 4,13,19 (*Eating*, *Reaching an object*, *Transporting an object*) are grouped in two clusters. From this analysis carried out in an unsupervised setting, we can conclude that our problem involves classes with a high intra-class variability and classes very similar to each other. Given that premise, we can now move to the next section, aware that the multi-class problem we are addressing is quite complicated.

4.7.2 Supervised learning

In this section, we try to classify the actions in a supervised manner. To this purpose we consider the data represented as described in Section 4.4 and their labels and we build a classifier. We adopt the machine learning library GURLS for an efficient implementation of RLS and we consider a classifier equipped with an RBF kernel.

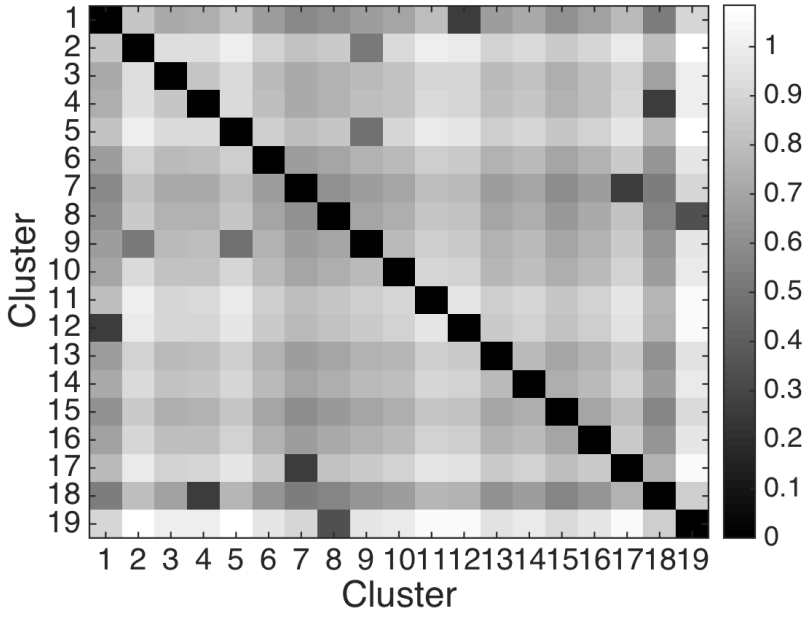


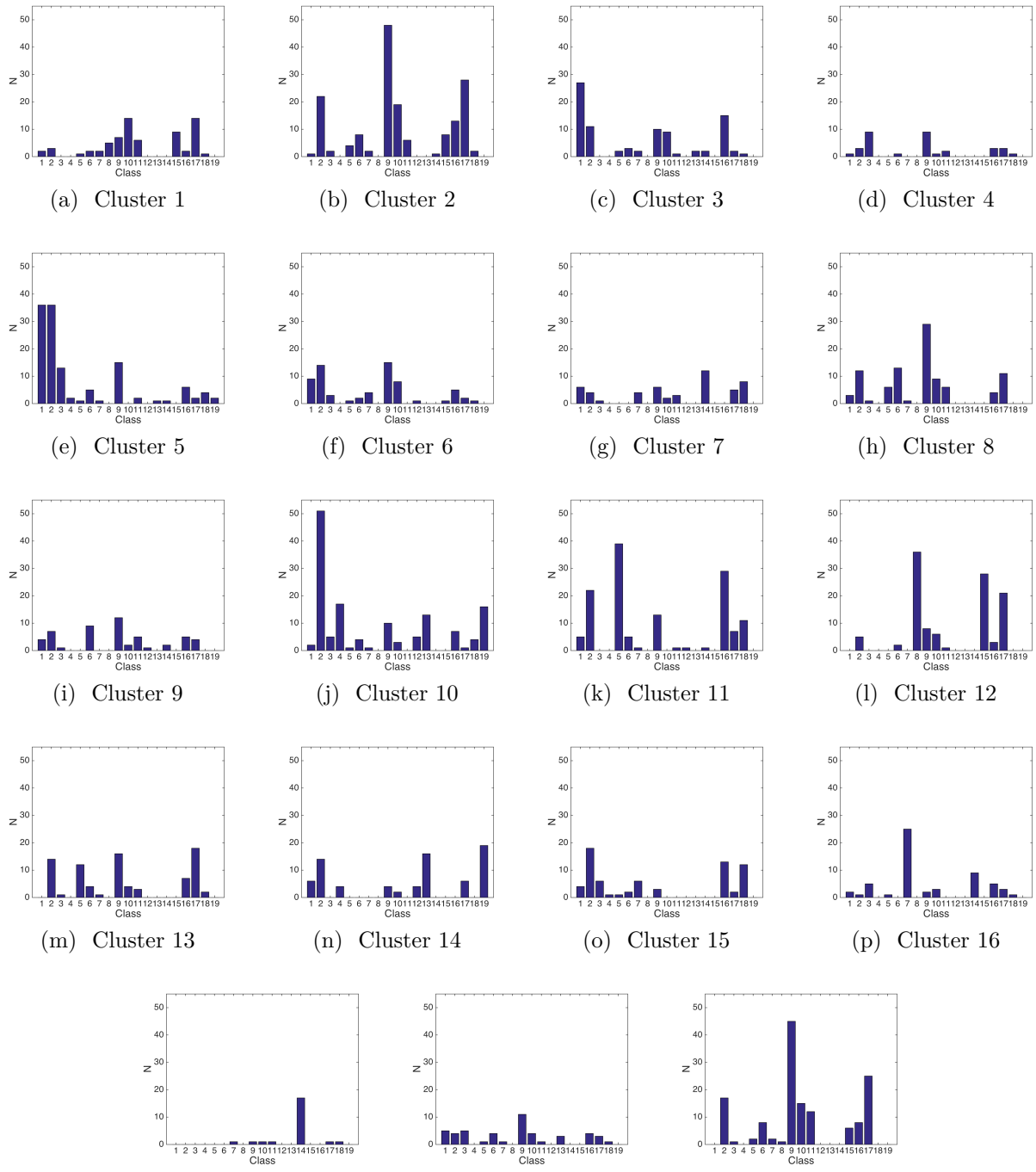
Figure 4.6: Matrix of the euclidian distances between the centroids of the clusters for the frontal view data.

Custer	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Classes	17	18,16	1	8,15	19	1	13	17	14	6,17,10		19	3		7	5	17

Table 4.4: All the 19 classes but 2 and 9 grouped in 17 clusters.

We use the whole training set and run a training procedure doing model selection in order to set σ^* and λ^* with a balanced hold out procedure, with $M = 3$ trials.

Binary classification First of all, in order to measure the complexity of the classes and to observe if some classes are easier to be distinguished than others, we train several binary classifiers. More in detail, for each class we train a binary classifier to discriminate between the data of that class and the rest of data, building as training set all the data of the class and the same amount of data randomly chosen from the rest of the set, to have a balanced training set. The performance of the classification of the test set, computed as the average accuracy (*Macroavg* of Table (4.7) with number of classes/actions $N = 2$), are shown in Table (4.8). We can observe that some classes (e.g. class 4,5,7,8,13,19) are easier to be classified than others (e.g. 2,6,9,11,16). This is in line with the results obtained in the unsupervised classification experiments (Tables (4.6) and (4.5)) where classes 8,15 and 4,13,19 where quite well distinguishable from others.



(a) Cluster 1 (b) Cluster 2 (c) Cluster 3 (d) Cluster 4

(e) Cluster 5 (f) Cluster 6 (g) Cluster 7 (h) Cluster 8

(i) Cluster 9 (j) Cluster 10 (k) Cluster 11 (l) Cluster 12

(m) Cluster 13 (n) Cluster 14 (o) Cluster 15 (p) Cluster 16

(q) Cluster 17 (r) Cluster 18 (s) Cluster 19

Figure 4.7: Histograms of the distribution of the classes for each cluster obtained with a K-means with K=19. All 19 classes considered, frontal view point.

Custer	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Classes	1	10	5	4,19,13	10	3	10	1	5,16	5	8	16,18	19,13		7	14

Table 4.5: All the 19 classes but 2,9,17 grouped in 16 clusters.

Custer	1	2	3	4	5	6	7	8
Classes	1	1	1 0	5,16	10	8,15	13,14,19	7

Table 4.6: All the 19 classes but 2,9,17 grouped in 8 clusters.

Performance	**Formula**
Macroavg (average accuracy)	$\frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{P_i}$, where N is the number of classes
PrecRec (average precision)	$\frac{1}{11}\sum_{r\in\{0,0.1,\ldots,1.0\}} p_{interp}(r)$, where $p_{interp}(r) = max_{\widetilde{r}:\widetilde{r}\geq r} p(\widetilde{r})$, p is precision, $p = \frac{t_p}{(t_p+f_p)}$, r is recall, $r = \frac{t_p}{(t_p+f_n)}$
Overall accuracy	$\frac{\sum_{i=1}^{N} TP_i}{n}$, where n is the number of samples of the test set

Table 4.7: Different quantities to evaluate the classification performance.

Then, we train a unique multi-class classifier with the purpose of discriminating each action from the others in a multi-class framework. In the training phase, the GURLS library accepts as input parameter the type of performance to consider during the model selection, *Macroavg* and *PrecRec* (the formulas are reported in Table (4.7)).

We decided to use *PrecRec* as it seems to suffer less from the problem of overfitting to the training data, as the performance on the classification on the training set and on the test set are more similar to each other than in the *Macroavg* case. The obtained classifier is then used to evaluate the capability of our method to discriminate between actions on new data. We carry out classification experiments on the test set, composed by the same actions of the training set but of different videos. At the end of the test phase, the GURLS library gives as output the scores of each class.

Scores over time In Figure 4.8 there is an example of scores of all the sub-movements of an action (*Transporting an object*) over time: we can observe how, even in cases, like this one, where the maximum score in average is the one associated with the correct class, it still goes up and down over time, showing that some sub-movements of the actions are more descriptive than others. It is worth noting that *Reaching an object* and *Eating* have also a

Class	1	2	3	4	5	6	7	8	9	10
Macroavg	0.79	0.61	0.71	0.85	0.82	0.52	0.86	0.82	0.61	0.71

Class	11	12	13	14	15	16	17	18	19
Macroavg	0.55	0.68	0.86	0.76	0.72	0.58	0.71	0.67	0.80

Table 4.8: Average accuracy of binary classifiers on test set for each class.

high score, confirming the similarity of the kinematics of the three actions.

Multi-class classification The label associated with the data is the class with the maximum score. We report the results of this classification experiment in Figure 4.9, that shows the confusion matrices of the average precision (definition in Table (4.7)) obtained on the training set and on the test set, where in the cell in position (i, j) there is the percentage of sub-movements of the class i-th that have been labeled as class j-th.

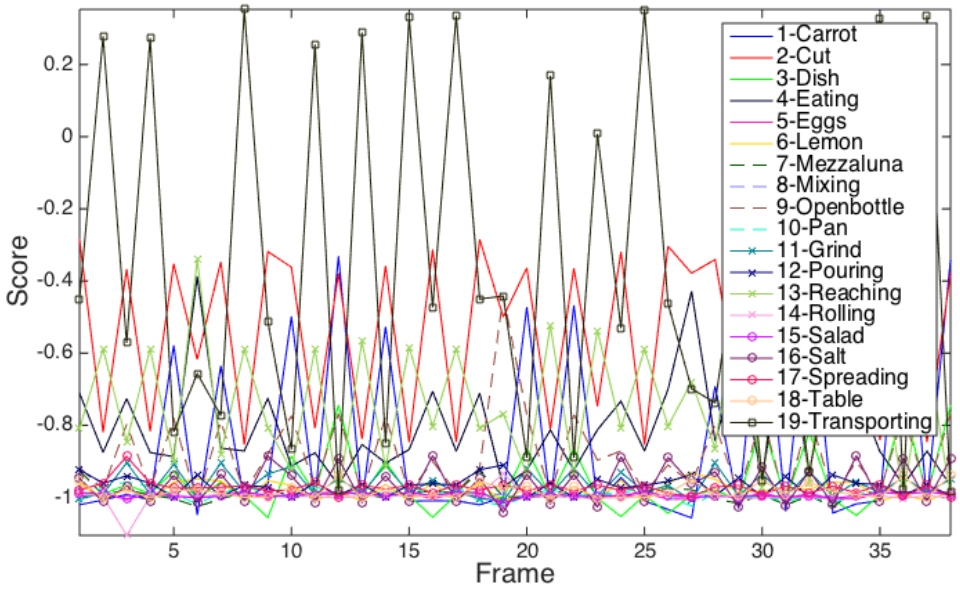
In the first row of Table (4.9) the accuracies have been reported: more in detail, we computed the overall accuracies and the average accuracies (formulas in Table (4.7)).

It can be noticed how the performance, although well above chance, is not very high. This was anticipated in Section 4.7.1, where we noticed a high intra-class variability and the presence of similar classes.

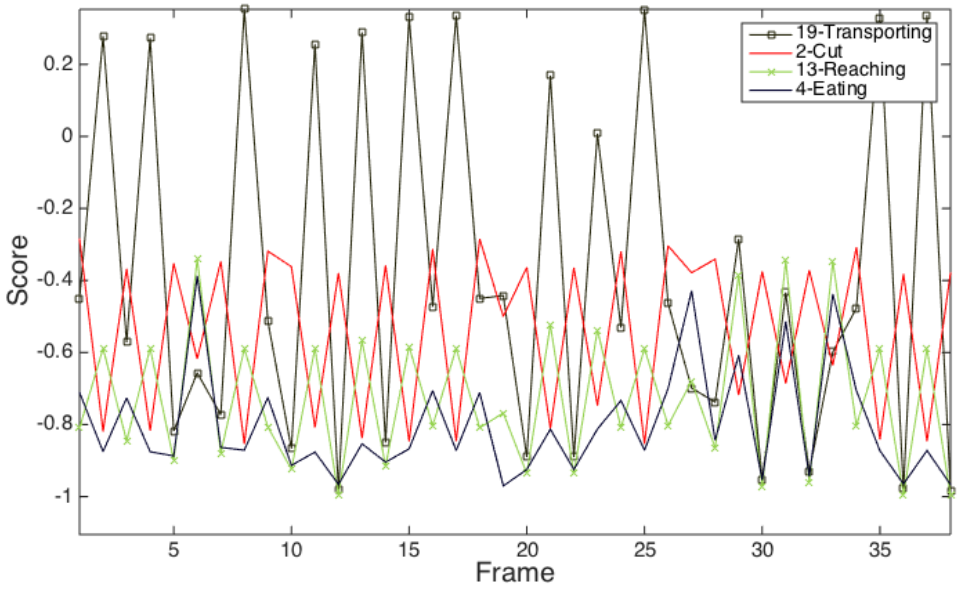
However, if we make the system wait for the whole video of the action to be finished to classify it, then the possibility to have a correct classification increases. Also, if we consider as correctly classified sample even if the correct class is one of the first three classes with the highest score, then the performance will improve significantly (Table (4.10)).

We then tried to decrease the number of classes to observe if, in case of a multi-class problem composed by less classes, the performances get better. The results obtained in Section 4.7.1 can lead the decision about which classes should be eliminated: we did the same unsupervised analysis of Section 4.7.1 for all the points of view, and we found out that the classes that are more difficult are 2, 9, 17 and also 4, 6, 7, 11, 12, 13. According to these results, first we eliminate the classes 2, 9, 17 (results in Figure 4.10 and the second row of Table (4.9)) and then also 4, 6, 7, 11, 12, 13 (results in Figure 4.11 and the third row of Table (4.9)), improving significantly the performance.

We performed the same experiments with the videos acquired by the other viewpoints (egocentric and lateral). From the confusion matrices obtained we build graphs of similarity reported in Figure 4.12. We take into consideration the cells with a value higher than a threshold of 0.15, and we represent them as a connection between two nodes (each one is an action). The edges are oriented from the action to the class: for instance, in Figure 4.12c,



(a)



(b)

Figure 4.8: (a) Scores of all the classifiers for each sub-movement of the *Transporting an object* sequence.
(b) Scores of the first 4 classifiers with the highest score (in average). The maximum score is associated to the correct class, that is *Transporting an object*.

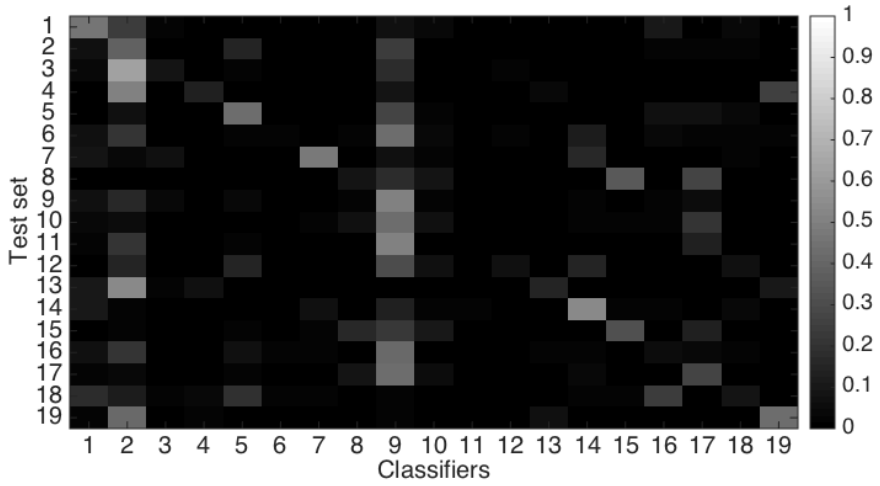


Figure 4.9: Confusion matrix of the average precision obtained with the multi-class classifier on the test set.

the edge from node 18 to node 16 means that more than 15% of samples of action 18 have been classified as class 16 (this connection is represented by the cell with row 18 and column 16 in Figure 4.11). The width of the edge is representative of the value of the cell (the higher the value, the thicker the edge), and the nodes are positioned close to nodes that are similar to them. We can observe that in the three viewpoints, in the case of 19 classes there are some nodes (in particular 2 and 9) that are in the center of the graphs, making confusion in understanding the similarities between the actions. In the case of 16 and 10 classes, the graphs are clearer and some interesting connections can be noted: in Figure 4.12c we can observe that the classes 8 (*Mixing*), 10 (*Turning the omelette*) and 15 (*Washing the salad*), are connected to each other as well as 5 (*Beating eggs*) and 18 (*Cleaning the table*), and also 1 (*Grating the carrot*) with 3 (*Cleaning a dish*) and 1 with 19 (*Transporting an object*), while 14 (*Rolling the dough*) is well distinguishable from the others. Some of these similarities can be found even in the other viewpoints: in Figures 4.12f there are connections between 5 and 18, 1 and 3 (both characterized by a fast vertical periodic movement), 1 and 19, while in Figures 4.12i there are connections between 1 and 3, 1 and 19, 8 and 15 (both characterized by circular movements). Looking to Figures 4.12b and Figures 4.12e is possible to notice also another interesting connection, that is between 4 (*Eating*), 13 (*Reaching an object*) and 19 (*Transporting an object*), that are characterized by very similar movement of the hand, with a starting and an ending point.

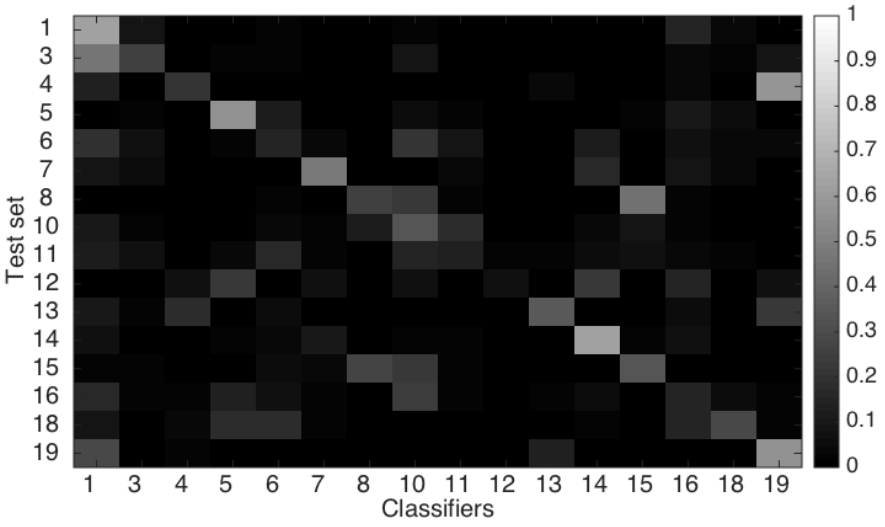


Figure 4.10: Confusion matrix of the average precision obtained with the multi-class classifier on 16 classes on the test set.

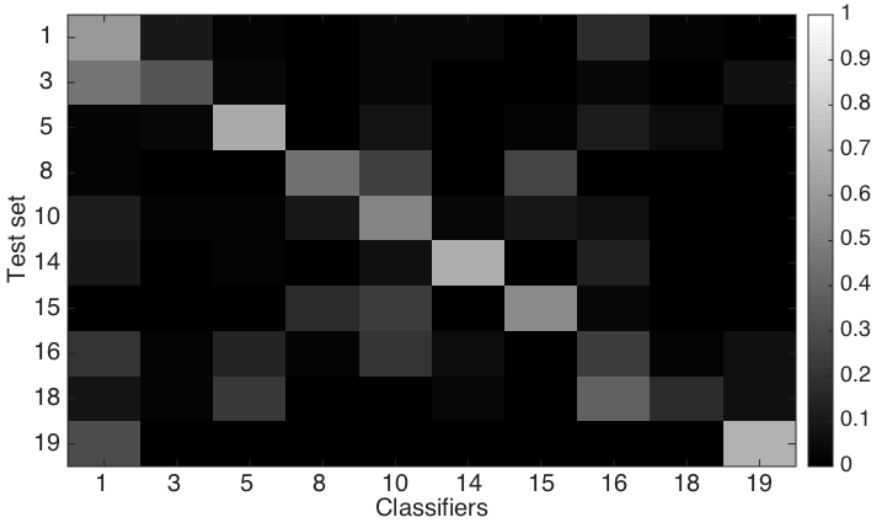


Figure 4.11: Confusion matrix of the average precision obtained with the multi-class classifier on 10 classes on the test set.

N. of classes	Training set		Test set		Chance
	Overall acc.	Average acc.	Overall acc.	Average acc.	
19	0.45	0.42	0.29	0.24	0.05
16	0.55	0.54	0.36	0.34	0.06
10	0.61	0.64	0.49	0.49	0.1

Table 4.9: Accuracy of multi-class classification on training set and test set.

N. of classes	Training set		Test set		Chance
	Overall acc.	Average acc.	Overall acc.	Average acc.	
19	0.81 (0.45)	0.78 (0.42)	0.60 (0.29)	0.53 (0.24)	0.05

Table 4.10: Accuracy of multi-class classification on training set and test set considering the classification of a sample as correct if the correct class has a score which is within the third maximum score. 19 classes case. In brackets there is the accuracy value in case you consider the classification as correct just when the correct class gives the maximum score (first row of Table (4.9)).

4.7.3 Intra-view analysis

In this section we perform some experiments to investigate the view-invariance property of the system, by using the data recorded from different points of view. In this experiment we follow a developmental approach, taking inspiration from a child (person 1, egocentric view) that observes himself, the mother (person 2, frontal view) and another person (person 3, lateral view).

In Figure 4.13 different scenarios are shown. In TR1 case, for instance, we want to evaluate situations that emulate the infant which observes his own actions to recognize them (TR1 TE1), or to recognize these actions performed by a person in front of him (TR1 TE2) or to recognize actions performed by another person (TR1 TE3). We then did the same with the other viewpoints (TR2 and TR3).

Following again the developmental inspiration, we asked ourselves if learning with two different viewpoints (for instance TR12, similar to the child that learns by observing himself and himself in the mirror or the mother in front that imitates him) could bring advantages, even in the capability of recognizing the same actions from a different viewpoint. In TR12$\hat{}$ case, we emulate the child that learns by observing movements performed by himself and movements similar but not identical performed by the mother (as "$\hat{}$" means that the video has been recorded in a different time). In TR13 case, we emulate the child that learns by observing movements performed by himself and by a third person.

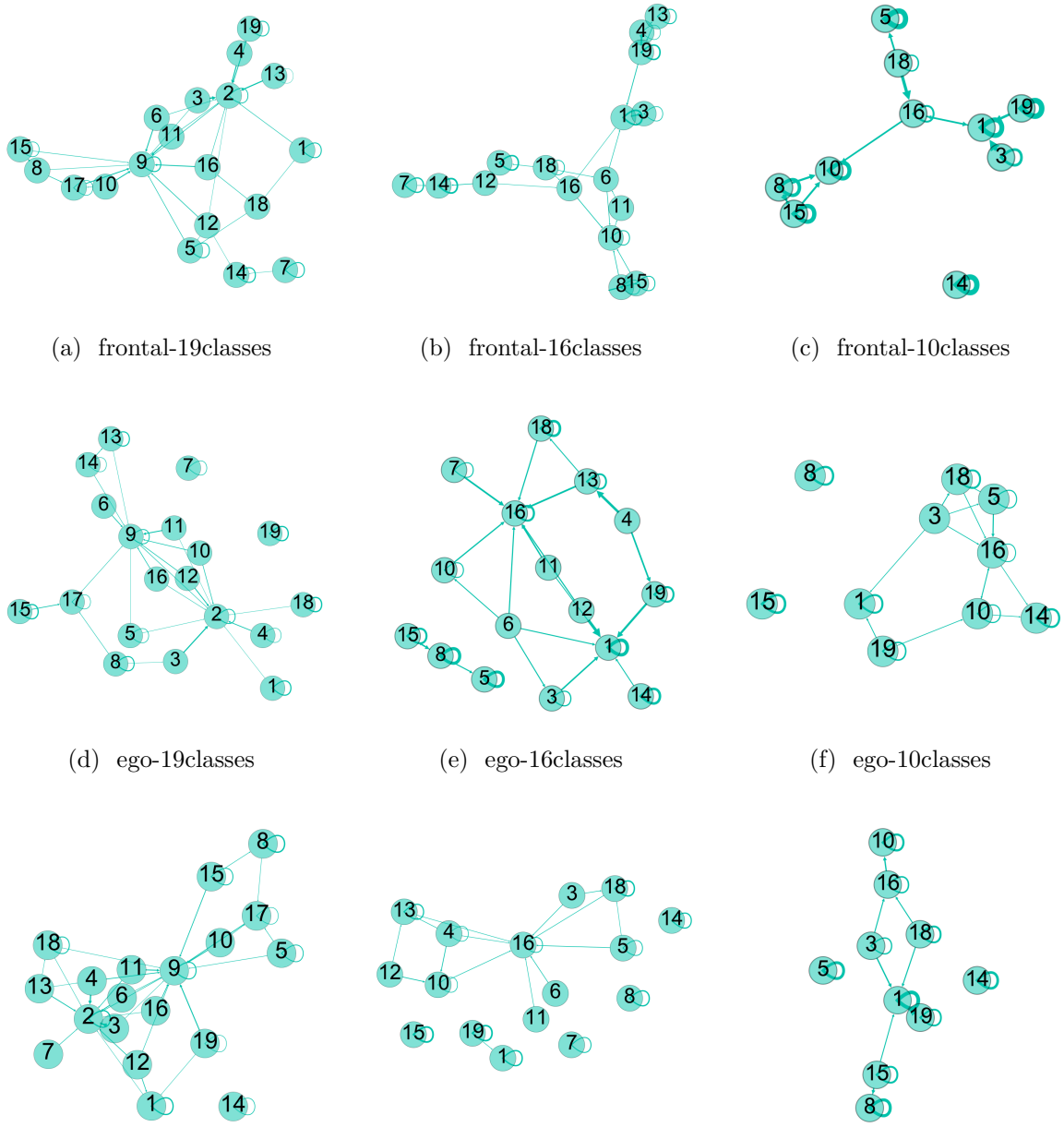
We can observe that the best performances, as expected, are obtained by training and testing the system on the same view. Nevertheless, we can observe that in TR12, TR12^, TR13 the performance over the three bars are more consistent across views that in the cases with a singular view as training set. This means that training on two views bring advantages, as, although there is a loss in the performance with respect to the previous condition on single view, a good generalization is achieved.

4.8 Discussion

In this chapter we did a step forward towards motion understanding, estimating the similarity between actions. We proposed a representation of actions based on the use of a dictionary: first, the system segments the action in the video in sub-movements and learns a dictionary of motion primitives in an automatic way, producing symmetric and asymmetric bell-shaped velocity profiles. Then, the actions are represented as a suitable combination of the motion primitives learnt.

We analyzed the descriptiveness of our representation by carrying out experiments in unsupervised and supervised manner, finding out some similarity relationships between different actions. We also analyzed different viewpoints, reasoning about view-invariance and showing that putting together more viewpoints leads to have better performance in the classification results, even in case of test on a different viewpoint: this result can be linked to human development of action understanding, which is facilitated by occasions in which the child and the mother act together [57].

Even if the goal of this work was categorizing actions into groups, it is possible to improve the current method to use it for actual action recognition. For doing it, several possibilities could be adopted: first of all, the description of the motion can be enriched by using also the other features and not only the speed. Furthermore, it is possible to use all the features extracted for each pixel of the region of interest without collapsing the whole information in a centroid by averaging them: this could be useful to extract the movements related to different segments of an articulated object (e.g. an arm) or to study in parallel the movement of different objects (e.g. two hands). Another idea is reasoning about the reliability of the answers of the classifier in different parts of the video: indeed, some parts of the recorded actions are more important and peculiar of the action itself than others. Taking into consideration this distinction between different parts of the action could lead to give more weight to the classification answers of peculiar parts of a certain action. Another improvement could be considering also simple spatial features, as the shape of the trajectory or the direction along which the motion is performed.



(a) frontal-19classes (b) frontal-16classes (c) frontal-10classes

(d) ego-19classes (e) ego-16classes (f) ego-10classes

(g) lateral-19classes (h) lateral-16classes (i) lateral-10classes

Figure 4.12: Graphs of similarity between actions.

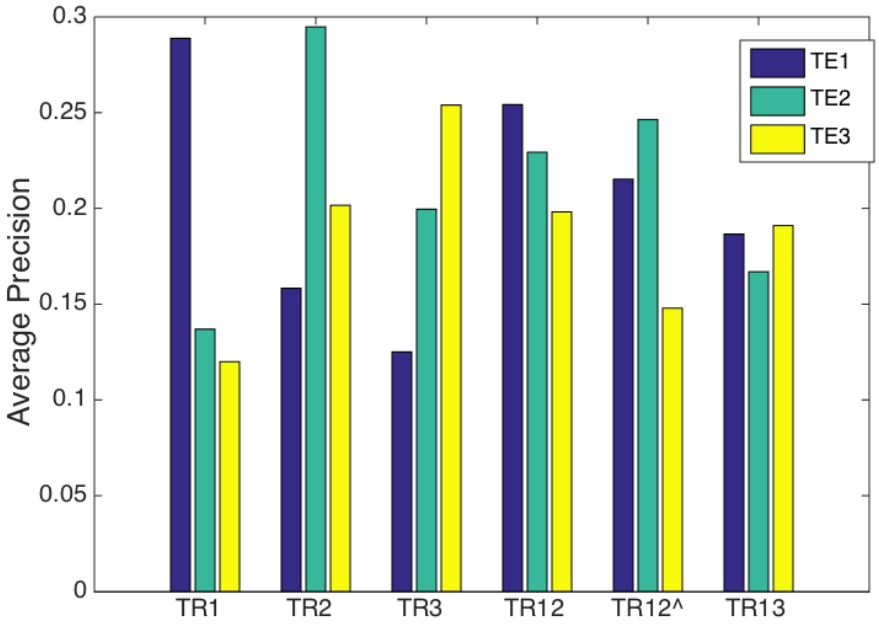


Figure 4.13: Average precision in multi-class classification

Chapter 5

Conclusion and future work

In this thesis, we investigated a problem of developmental robotics, devising computational models of the visual primitives at the basis of social interaction in humans. Our inspiration roots on the very first stage of development, where the limited amount of visual information suggests that human beings have the capability to accomplish simple pro-social tasks on the basis of rather coarse motion models. We took inspiration from the Two-Thirds Power Law, validating its applicability to video analysis problems. We built a model able to discriminate between biological and non-biological motion, demonstrating the possibility to exploit our method to perform human activity detection also in complex scenarios, where traditional appearance-based approaches (such as skin or face detection methods) would fail. Our approach is robust to severe occlusions or to indirect representation of the agent motion in the scene (as during the observation of agents' shadows).

Moreover, we implemented an online version of the method on a robotic intelligent system, which leverage the human detection skill and appropriately orient the focus of attention in order to establish an interaction with the human counterpart. In particular, the integration with the attention system allowed us to endow the robot with the ability redeploy the fixation point on biological activities in the scene.

We also exploited biological motion regularities to discover motion primitives and to visually represent the observed actions as a combination of them, categorizing the actions into different classes. The system is able to detect groups of coherent actions and, to a certain extent, in a coherent way across views.

In summary, we addressed the two problems of biological motion recognition and action categorization of Figure 5.1. The latter has been implemented only offline, while its porting on iCub is left for a future work. For this, some parts of the biological motion classification method implemented on iCub can be used: in particular, the information on which the method builds on are already extracted online from iCub cameras, as they are the same as

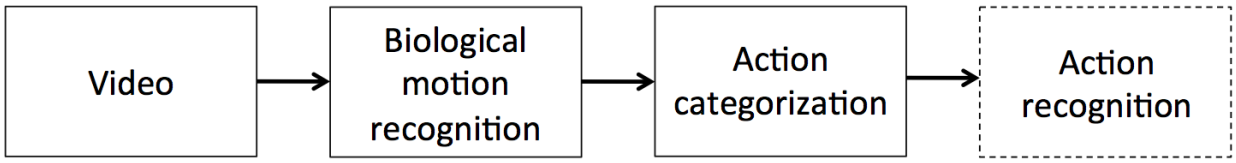


Figure 5.1: Flow for action understanding. Steps in full line have been implemented and discussed in the thesis, the ones in dashed line are left for future works.

in the method for biological motion detection, and the multi-class classification is already implemented in GURLS, the library used for the binary classification of biological and non-biological motion. The missing parts are the segmentation procedure, the clustering for the motion primitives identification, and the representation of the data with sparse coding.

Regarding the future work, we know that humans can infer a lot of information from the way an action is performed, while robots still miss this capability, limiting the intuitiveness and the naturalness of the interaction [78]. The important variables of this implicit interaction are low level information as velocity and curvature: they can reveal the effort of a person [77], the willingness of making the partner understand the action (like in the case of signaling [65]) and even the emotional status (e.g. aggressiveness or kindness of the action [20]). In this context, a future work can be the use of the methods developed in the thesis to determine this features of the action during an interaction: this will enhance artificial agent's competence in interpreting its operative context having a better understanding of the actions performed by the human and will allow it to provide the appropriate pro-social behavior.

For the future, we also aim at improving our method for understanding actions by using not just visual but even motoric information: to this purpose, we have already collected a multimodal dataset (Appendix B) at the University of Maryland.

Appendices

Appendix A

Multi-view dataset

We acquired a dataset composed of indoor videos of one subject performing cooking actions. We have used three identical high resolution IP cameras, mounted on three tripods so that in all acquisitions we have a still uniform background and moving foreground objects. Figure A.1 shows the setup and example video frames. The dataset includes repetitions of the same action observed from three different viewpoints: a frontal view (A), a lateral view (B), and an egocentric view, obtained by a camera mounted slightly above the subject's head (C). We recorded different types of cooking actions, which included the use of tools and food ingredients: 1-*Grating the carrot*, 2-*Cutting the bread*, 3-*Cleaning a dish*, 4-*Eating*, 5-*Beating eggs*, 6-*Squeezing the lemon*, 7-*Cutting with a mezzaluna*, 8-*Mixing*, 9-*Open the bottle*, 10-*Turning the omelette*, 11-*Pestling*, 12-*Pouring water*, 13-*Reaching an object*, 14-*Rolling the dough*, 15-*Washing the salad*, 16-*Salting*, 17-*Spreading cheese on a bread*, 18-*Cleaning the table*, 19-*Transporting an object*. No specific constraints have been imposed to the volunteer with the exception of the request of containing the actions within a fixed working space. For each dynamic event we acquired two videos. The images have size 640×480 and have been acquired at an approximate rate of 30 fps.

Figure A.1: Acquisition setup

Appendix B

Multi-sensor dataset

Multi-view and motoric dataset. The dataset collection has been performed in three different main sections: 1) actions with IMU sensor glove (NuGlove), 2) actions with force sensor glove (or force sensor knife), 3) actions with bare hands. In each section, three different objects are involved: 1) a sponge, 2) a cup, 3) a potato (and the tool knife). 15 subjects were trained to perform the actions. Detailed explanations about the tools used to record the data are provided here.

The **NuGlove** is an instrumented gesture recognition glove made by AnthroTronix company that can be used either as a controller for robotic devices or to track individual hand movements. The NuGlove is an orientation-based glove, meaning that it uses the natural movements of the operator's hand/arm as the input, but also includes gyroscopes and magnetometers, allowing for 9-axis detection and control in each finger. When combined with the accelerometers, the overall system allows for greater movement detection and differentiation. Python is used as the programming languages and operation system used is Linux/Ubuntu. It is equipped with USB cable for data transferring (wired connection). The glove is equipped with 7 IMU sensors on the back side of the glove (one on each finger, and one on the dorsum of the hand and on the wrist area). Each sensor provides 7 outputs: 3 accelerations along each axis and 4 quaternions. As a result, the final output of the glove consists of 50 columns of data. The first column is the time stamp, the next 28 columns are the quaternions data and the next 21 columns are the acceleration data. A figure of the NuGlove is presented on the left of Figure B.1.

To measure the pressure applied by different part of the hand on the object or on the tool, we used a **force sensor glove** (in the case of the actions with the sponge and the cups) and a **force sensor knife** (in the case of the actions with the knife and the potato). The glove is equipped with 7 force sensors on the inertial side of the glove (one on each finger, and two on the palm area) but since in most actions the little finger is not involved or if

involved it has mostly a performance similar to the ring finger, the data of the sensor on this finger is not considered. The knife is equipped with 6 force sensors. As a result, the final output of the glove and the knife consists of 7 columns of data. The first column is the time stamp (useful for the synchronization with the other sensors) and the next 6 columns are the pressure data in lb (Columns 2 to 7). The sensors mounted are FlexiForce A201 sensors, that allow a maximum force of 4448N (0-1000 lb). A figure of the sensorized glove and knife is presented on the right of Figure B.1.

The two **cameras** used are Xtion PRO LIVE cameras, that have one RGB, one Depth and two Microphone sensors. Depth image size is VGA (640x480) with 30 fps and the resolution is SXGA (1280*1024).

The actions we asked to the subjects to perform are the following 10 actions for each object:
Actions with a sponge: Washing, Spot cleaning, Scratching (removing a small spot), Flipping, Squeezing, Sponging up water, Cleaning, Folding it (in half), Twisting it (requires both hand), Squeeze down.
Actions with a cup: Drinking, Pounding, Shaking, Moving (to a different location), Pouring, Rotating (by lifting it up and holding it on the rim), Flipping, Turning in place, Rolling (it back and forth), Scooping (up sugar).
Actions with a knife and a potato: Peeling, Chopping (One single shot of cutting), Slicing, Mincing, Dicing, Carving (a triangle), Pressing, Transferring, Making a hole (removing a small piece by a rotation of the knife), Coring (circular motion for cylindrical hole).

Figure B.1: IMU sensor glove, force sensor glove and knife.

Publication list

The work described in 2 and 3 has been published in the following international conferences and journals.

— Vignolo A., Noceti N., Rea F., Sciutti A., Odone F., Sandini G., 'Detecting Biological Motion for Human-Robot Interaction: A Link between Perception and Action', Journal Frontiers in Robotics and AI, 2017

— Malafronte D., Goyal G., Vignolo A., Odone F., Noceti N., 'Investigating the use of space-time primitives to understand human movements', International Conference on Image Analysis and Processing (ICIAP), Catania, Italy, September 11-15, 2017

— Vignolo A., Sciutti A., Rea F., Noceti N., Odone F., Sandini G., 'Computational Vision For Social Intelligence', Association for the Advancement of Artificial Intelligence (AAAI) Spring Symposium Series, Computational Principles of Natural and Artificial Intelligence, Palo Alto (CA), US, March 27-29, 2017

— Vignolo A., Rea F., Noceti N., Sciutti A., Odone F., Sandini G., 'Biological movement detector enhances the attentive skills of humanoid robot iCub', IEEE/RAS International Conference of Humanoids Robotics (Humanoids), Cancun, Mexico, November 15-17, 2016

— Vignolo A., Noceti N., Rea F., Sciutti A., Odone F., Sandini G., 'Human activity detection on the robot iCub', International workshop on Human-Friendly Robotics (HFR), Genova, Italy, September 29-30, 2016

— Vignolo A., Noceti N., Odone F., Rea F., Sciutti A., Sandini G., 'A computational model of biological motion detection based on motor invariants', Journal of Perception, Abstract for European Conference on Visual Perception (ECVP), Barcelona, Spain, August 28 - September 1, 2016

— Vignolo A., Noceti N., Sciutti A., Rea F., Odone F., Sandini G., 'The complexity of biological motion. A temporal multi-resolution motion descriptor for human detection in video', IEEE International Conference on Developmental Learning and Epigenetic Robotics (ICDL-EPIROB), Cergy-Pontoise, France, September 19-22, 2016

— Vignolo A., Noceti N., Rea F., Sciutti A., Odone F., Sandini G., 'Towards a friendly humanoid robot: Computational models of biological motion perception', International Workshop on Cognitive Development for Friendly Robots and Rehabilitation, Genoa, Italy, December 2-3, 2015

— Noceti N., Sciutti A., Vignolo A., Rea F., Odone F., Sandini G., 'Modeling the development of visual perception with computational vision', Journal of Perception, Abstract for European Conference on Visual Perception (ECVP), Liverpool, UK, August 23-27, 2015

— Sandini G., Noceti N., Vignolo A., Sciutti A., Rea F., Verri A., Odone F., 'Computational

Model of Biological Motion Detection: a path toward view-invariant action understanding', Journal of Vision, 015; 15(12):497-497. doi: 10.1167/15.12.497, Abstract for the Vision Science Society Meeting, St. Pete Beach, Florida, May 15-20, 2015

— Sandini G., Noceti N., Vignolo A., Sciutti A., Rea F., Verri A., Odone F., 'Modeling Visual Features to Recognize Biological Motion: A Developmental Approach', MODVIS - Computational and Mathematical Models in Vision, St. Pete Beach, Florida, May 13-15, 2015

Bibliography

[1] J.K. Aggarwal and M.S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, 2011.

[2] J.B. Asendorpf and P.M. Baudonniere. Self-awareness and other-awareness: Mirror self-recognition and synchronic imitation among unfamiliar peers. In *Developmental Psychology*, pages vol. 29, pp. 88–95, 1993, 1993.

[3] F. Bellagamba and M. Tomasello. Re-enacting intended acts: Comparing 12- and 18-month-olds. *Infant Behavior and Development*, 22:277–282, 1999.

[4] A. Bisio, A. Sciutti, F. Nori, G. Metta, L. Fadiga, G. Sandini, and T. Pozzo. Motor contagion during human-human and human-robot interaction. *PloS one*, 9(8):e106172, 2014.

[5] A. Bissacco, M. H. Yang, and S. Soatto. Fast human pose estimation using appearance and motion via multi-dimensional boosting regression. *CVPR*, pages 1–8, 2007.

[6] J.-D. Boucher, U. Pattacini, A. Lelong, G. Bailly, F. Elisei, S. Fagel, and J. Ventre-Dominey. I reach faster when i see you look: Gaze effects in human-human and human-robot face-to-face cooperation. *Frontiers in Neurorobotics*, 6, 2012.

[7] R. J. Brand and W. L. Shallcross. Infants prefer motionese to adult-directed action. *Developmental science*, 11(6):853–861, 2008.

[8] R. J. Brand, W. L. Shallcross, M. G. Sabatos, and K. P. Massie. Fine-grained analysis of motionese: Eye gaze, object exchanges, and action units in infant-versus adult-directed action. *Infancy*, 11(2):203–214, 2007.

[9] C. Breazeal, A. Edsinger, P. Fitzpatrick, and B. Scassellati. Active vision for sociable robots. *IEEE Transactions on systems, man, and cybernetics-part A: Systems and Humans*, 31(5):443–453, 2001.

[10] C. Breazeal and B. Scassellati. A context-dependent attention system for a social robot, 1999.

[11] C. Bregler and J. Malik. Tracking people with twists and exponential maps. *CVPR*, pages 8–15, 1998.

[12] L. Brethes, P. Menezes, F. Lerasle, and J. Hayet. Face tracking and hand gesture recognition for human-robot interaction. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 2, pages 1901–1906 Vol.2, 2004.

[13] D. Burr, M.C. Morrone, and J. Ross. Selective suppression of the magnocellular visual pathway during saccadic eye movements. *Nature*, 371:511–513, 1994.

[14] G. Butterworth. The ontogeny and phylogeny of joint visual attention. *Natural theories of mind: Evolution, development, and simulation of everyday mind*, 1991.

[15] A. Cangelosi and M. Schlesinger. Developmental robotics: from babies to robots. *The MIT Press*, 2015.

[16] T. Chaminade and G. Cheng. Social cognitive neuroscience and humanoid robotics. *Journal of physiology, Paris*, 103(3-5):286–95, 2009.

[17] L.T. Ching, Y. Yang, C. Fermuller, and Y. Aloimonos. Using a minimal action grammar for activity understanding in the real world. 2012.

[18] M. Correa, G. Hermosilla, R. Verschae, and J. Ruiz-del Solar. Human detection and identification by robots using thermal and visual information in domestic environments. *Journal of Intelligent and Robotic Systems*, 66:223–243, 2012.

[19] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, and C. BrayLixin. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004.

[20] G. Di Cesare, L. Sparaci, A. Pelosi, L. Mazzone, G. Giovagnoli, D. Menghini, E. Ruffaldi, and S. Vicari. Differences in action style recognition in children with autism spectrum disorders. *Frontiers in psychology*, 2017.

[21] R. Dillmann. Teaching and learning of robot tasks via observation of human performance. *Robotics and Autonomous Systems*, 47:109–116, 2004.

[22] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. *ICCV*, pages 726–733, 2003.

[23] C. Elsner, T. Falck-Ytter, and G. Gredebäck. Humans anticipate the goal of other people's point-light actions. *Frontiers in psychology*, 3, 2012.

[24] R. Fablet and M. Black. Automatic detection and tracking of human motion with a view-based representation. *ECCV*, 1:476–491, 2002.

[25] S. R. Fanello, I. Gori, G. Metta, and F. Odone. Keep it simple and sparse: Real-time action recognition. *J. Mach. Learn. Res.*, 14(1):2617:2640, 2013.

[26] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Proc.s of the 13th Scandinavian Conference on Image Analysis*, SCIA'03, pages 363–370, 2003.

[27] T. Farroni, G. Csibra, F. Simion, and M. H. Johnson. Eye contact detection in humans from birth. *Proceedings of the National Academy of Sciences*, 99(14):9602–9605, 2002.

[28] T. Farroni, G. Csibra, F. Simion, and M.H. Johnson. Eye contact detection in humans from birth. *Proceeding of the National Academy of Sciences (PNAS)*, 99:9602–9605, 2002.

[29] T. Farroni, M.H. Johnson, E. Menon, L. Zulian, D. Faraguna, and G. Csibra. Newborns' preference for face relevant stimuli: Effect of contrast polarity. *Proceeding of the National Academy of Sciences (PNAS)*, 102:17245–17250, 2005.

[30] J. Flanagan, M. Bowman, and R. Johansson. Control strategies in object manipulation tasks. *Current Opin. Neurobiol.*, 16:650–659, 2006.

[31] K. Fragkiadaki, H. Hu, and J. Shi. Pose from flow and flow from pose estimation. *CVPR*, pages 2059–2066, 2013.

[32] A. Gaschler, S. Jentzsch, M. Giuliani, K. Huth, J. de Ruiter, and A. Knoll. Social behavior recognition using body posture and head pose for human-robot interaction. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2128–2133, 2012.

[33] G. Gavazzi, A. Bisio, and T. Pozzo. Time perception of visual motion is tuned by the motor representation of human actions. *Scientific reports*, 3, 2013.

[34] G. Gredebäck and A. Melinder. Infants' understanding of everyday social interactions: a dual process account. *Cognition*, 114:197–206, 2010.

[35] P. H. Greene. Problems of organization of motor systems. *Progress in theoretical biology*, 2:123–145, 1972.

[36] K. Hauser, T. Bretl, K. Harada, and J.C. Latombe. Using motion primitives in probabilistic sample-base planning for humanoid robots. *Int. Work. on the Algorithmic Foundations of Robotics*, pages 507–522, 2008.

[37] P. E. Hemeren and S. Thill. Deriving motor primitives through action segmentation. 2011.

[38] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:1254–1259, 1998.

[39] A. Jain, J. Tompson, Y. LeCun, and C. Bregler. Modeep: A deep learning framework using motion features for human pose estimation. *ACCV*, pages 302–315, 2014.

[40] H. Jegou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010.

[41] S. Ju, M. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated motion. *Face and Gesture*, pages 38–44, 1996.

[42] F. Kaplan and V. Hafner. The challenges of joint attention. In *Proc. of Int. Work. on Epigenetic Robotics*, 2006.

[43] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *International Journal of Robotics Research*, 32:951–970, 2013.

[44] D. Kulic, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura. Incremental learning of full body motion primitives and their sequencing through human motion observation. *The International Journal of Robotics Research*, 31(3):330–345, 2012.

[45] F. Lacquaniti, C. Terzuolo, , and P. Viviani. *Global metric properties and preparatory processes in drawing movements*. 1984.

[46] F. Lacquaniti and C. Terzuolo. The law relating the kinematic and figural aspects of drawing movements. *Acta Psychologica*, 54:115–130, 1983.

[47] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, volume 2, pages 2169–2178, 2006.

[48] D. Lee and C. Ott. Incremental kinesthetic teaching of motion primitives using the motion refinement tube. *Autonomous Robots*, 31(2-3):115–131, 2011.

[49] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *NIPS*, 2007.

[50] D. Méary, E. Kitromilides, K. Mazens, C. Graff, and E. Gentaz. Four-day-old human neonates look longer at non-biological motions of a single point-of-light. *PloS one*, 2(1):e186, 2007.

[51] A. N. Meltzoff. Understanding the intentions of others: Re-enactment of intended acts by 18-month-old children. *Developmental Psychology*, 31:838–850, 1995.

[52] G. Metta, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. Von Hofsten, K. Rosander, M. Lopes, J. Santos-Victor, et al. The icub humanoid robot: An open-systems platform for research in cognitive development. *Neural Networks*, 23(8):1125–1134, 2010.

[53] G. Metta, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, Santos-Victor von Hofsten C., A. J., Bernardino, and L. Montesano. The icub humanoid robot: An open-systems platform for research in cognitive development. *Neural Networks*, 23:1125–1134, 2010.

[54] P. Morasso. Spatial control of arm movements. *Experimental brain research*, 42(2):223–227, 1981.

[55] M. Mori. The uncanny valley. *Energy*, 7:33–35, 1970.

[56] B. Mutlu, T. Kanda, J. Forlizzi, J. Hodgins, and H. Ishiguro. Conversational gaze mechanisms for humanlike robots. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 1(2):12, 2012.

[57] Y. Nagai, Y. Kawai, and M. Asada. Emergence of mirror neuron system: Immature vision leads to self-other correspondence. In *IEEE International Conference on Development and Learning (ICDL)*, volume 2, pages 1–6, 2011.

[58] Y. Nagai and K. J. Rohlfing. Computational analysis of motionese toward scaffolding robot action learning. *IEEE Transactions on Autonomous Mental Development*, 1(1):44–54, 2009.

[59] Nature. Research Highlights. `https://www.nature.com/articles/d41586-017-00807-3`, 2017. [Online].

[60] N. Noceti, A. Sciutti, F. Rea, F. Odone, and G. Sandini. Estimating human actions affinities across views. In *VISAPP*, 2015.

[61] N. Noceti, A. Sciutti, and G. Sandini. Cognition helps vision: Recognizing biological motion using invariant dynamic cues. In *ICIAP*, pages 676–686, 2015.

[62] F. Orabona, G. Metta, and G. Sandini. Object-based visual attention: a model for a behaving robot. *Computer Vision and Pattern Recognition-Workshops*, 2005.

[63] O. Palinko, A. Sciutti, L. Schillingmann, F. Rea, Y. Nagai, and G. Sandini. Gaze contingency in turn-taking for human robot interaction: Advantages and drawbacks. In *24th IEEE International Symposium on Robot and Human Interactive Communication*, Kobe, Japan, August 31 - September 4 2015.

[64] U. Pattacini and et al. An experimental evaluation of a novel minimum-jerk cartesian controller for humanoid robots. In *International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[65] G. Pezzulo, F. Donnarumma, and H. Dindo. Human sensorimotor communication: A theory of signaling in online social interactions. *PloS one*, 8(11):e79876, 2013.

[66] R. Poppe. *A survey on vision-based human action recognition.* 2010.

[67] T. Pozzo, C. Papaxanthis, J. L. Petit, N. Schweighofer, and N. Stucchi. Kinematic features of movement tunes perception and action coupling. *Behavioural brain research*, 169(1):75–82, apr 2006.

[68] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *International Journal of Computer Vision*, 50(2):203–226, 2002.

[69] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. In *International Journal of Computer Vision*, volume 50(2), pages 203–226, 2002.

[70] F. Rea, P. Muratore, and A. Sciutti. 13-year-olds approach human-robot interaction like adults. In *IEEE International Conference Developmental Learning and Epigenetic Robotics*, 2016.

[71] F. Rea, G. Sandini, and G. Metta. Motor biases in visual attention for a humanoid robot. In *IEEE/RAS International Conference of Humanoids Robotics*, 2014.

[72] M.J.E. Richardson and T. Flash. Comparing smooth arm movements with the two-thirds power law and the related segmented-control hypothesis. *Jour. of Neuroscience*, 22(18):8201–8211, 2002.

[73] A. Roncone, U. Pattacini, G. Metta, and L. Natale. A cartesian 6-dof gaze controller for humanoid robots. *Proceedings of Robotics: Science and Systems, Ann Arbor, MI*, 2016.

[74] J. Sanches, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. In *IJCV*, 2013.

[75] S. Schaal. *Dynamic Movement Primitives -A Framework for Motor Control in Humans and Humanoid Robotics*. Springer, 2006.

[76] A. Sciutti, A. Bisio, F. Nori, G. Metta, L. Fadiga, T. Pozzo, and G. Sandini. Measuring human-robot interaction through motor resonance. *International Journal of Social Robotics*, 4(3):223–234, 2012.

[77] A. Sciutti, L. Patané, F. Nori, and G. Sandini. Understanding object weight from human and humanoid lifting actions. *IEEE Transactions on Autonomous Mental Development*, 2014.

[78] A. Sciutti and G. Sandini. Interacting with robots to investigate the bases of social interaction. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2017.

[79] H. Sidenbladh, M. Black, and D. Fleet. Stochastic tracking of 3d human fi

gures using 2d image motion. *ECCV*, pages 702–718, 2000.

[80] F. Simion, L. Regolin, and H. Bulf. A predisposition for biological motion in the newborn baby. *Proceedings of the National Academy of Sciences*, 105(2):809–813, 2008.

[81] F. Stulp, E.A. Theodorou, and S. Schaal. Reinforcement learning with sequences of motion primitives for robust manipulation. *Trans. on Robotics*, 28(6):1360–1370, 2012.

[82] J. Sung, C. Ponce, B. Selman, and A. Saxena. Unstructured human activity detection from rgbd images. In *IEEE International Conference on Robotics and Automation*, pages 842–849, 2012.

[83] A. Tacchetti, P. K. Mallapragada, M. Santoro, and L. Rosasco. Gurls: a least squares library for supervised learning. *The Journal of Machine Learning Research*, 14(1):3201–3205, 2013.

[84] M. Tomasello, M. Carpenter, J. Call, T. Behne, and H. Moll. Understanding and sharing intentions: The origins of cultural cognition. In *Behavioral and brain sciences*, pages 28(05), 675–691, 2005.

[85] B. A. Urgen, M. Plank, H. Ishiguro, H. Poizner, and A. P. Saygin. Temporal dynamics of action perception : The role of biological appearance and motion kinematics. In *34th Annual Conference of the Cognitive Science Society*, pages 2469–ï¿½2474, 2012.

[86] L. M. Vaina, J. Solomon, S. Chowdhury, P. Sinha, and J. W. Belliveau. Functional neuroanatomy of biological motion perception in humans. *Proceedings of the National Academy of Sciences*, 98(20):11656–11661, 2001.

[87] D. Vernon. *Artificial Cognitive Systems*. MIT Press, 2014.

[88] S. Vieilledent, Y. Kerlirzin, S. Dalbera, and A. Berthoz. Relationship between velocity and curvature of a human locomotor trajectory. *Neuroscience Letters*, 305(1):65 – 69, 2001.

[89] A. Vignolo, N. Noceti, F. Rea, A. Sciutti, F. Odone, and G. Sandini. Detecting biological motion for human-robot interaction: A link between perception and action. *Frontiers in Robotics and AI*, 2017.

[90] A. Vignolo, N. Noceti, A. Sciutti, F. Rea, F. Odone, and G. Sandini. The complexity of biological motion. a temporal multi-resolution motion descriptor for human detection in videos. In *IEEE International Conference Developmental Learning and Epigenetic Robotics*, 2016.

[91] A. Vignolo, F. Rea, N Noceti, A Sciutti, F Odone, and G Sandini. Biological movement detector enhances the attentive skills of humanoid robot icub. In *IEEE-RAS International Conference on Humanoid Robots*, 2016.

[92] P. Viviani, G. Baud-Bovy, and M. Redolfi. Perceiving and tracking kinesthetic stimuli: further evidence of motor-perceptual interactions. *J. Exp. Psychol Hum Percept Perform*, 23(4):1232–1252, 1997.

[93] P. Viviani and M. Cenzato. Segmentation and coupling in complex movements. *Journal of Experimental Psychology: Human Perception and Performance*, 11:828–845, 1985.

[94] P. Viviani and T. Flash. Minimum-jerk, two-thirds power law, and isochrony: converging approaches to movement planning. *Journal of Experimental Psychology: Human Perception and Performance*, 21(1):32–, 1995.

[95] P. Viviani and G. McCollum. The relation between linear extent and velocity in drawing movements. *Neuroscience*, 10(1):211–218, 1983.

[96] P. Viviani and R. Schneider. A developmental study of the relationship between geometry and kinematics in drawing movements. *Journal of Experimental Psychology: Human Perception and Performance*, 17(1):198–, 1991.

[97] P. Viviani and N. Stucchi. Biological movements look uniform: evidence of motor-perceptual interactions. *J Exp Psychol Hum Percept Perform.*, 18(3):603–623, 1992.

[98] P. Viviani and C. Terzuolo. Trajectory determines movement dynamics. *Neuroscience*, 7(2):431–437, 1982.

[99] P. Viviani and C. Terzuolo. Trajectory determines movement dynamics. *Neuroscience*, 7:431–437, 1982.

[100] A. L. Vollmer, K. S. Lohan, J. Fritsch, B. Wrede, and K. Rohlfing. Which motionese parameters change with children's age? *Poster presented at the International Conference on Development and Learning (ICDL)*, June 2009.

[101] S. Wachter and H. Nagel. Tracking persons in monocular image sequences. *CVIU*, 74(3):174–192, 1999.

[102] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[103] F. Warneken and M. Tomasello. The roots of human altruism. *British Journal of Psychology*, 100(3):455–471, 2009.

[104] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115:224–241, 2011.

[105] A.L. Woodward and J.J. Guajardo. Infants' understanding of the point gesture as an object-directed action. *Cognitive Development*, 17:1061–1084, 2002.

[106] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[107] Y. Yang, I. Saleemi, and M. Shah. Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions. *PAMI*, 35(7):1635–1648, 2012.

[108] S. Zhang, Z. Wei, J. Nie, L. Huang, S. Wang, and Z. Li. A review on human activity recognition using vision-based method. 2017.

[109] X. Zhou, K. Yu, T. Zhang, and T.S. Huang. Image classification using super-vector coding of local image descriptors. In *ECCV*, 2010.