Teaching by Touching: Interpretation of Tactile Instructions for Motion Development

Fabio DallaLibera, Fransiska Basoeki, Takashi Minato, Hiroshi Ishiguro and Emanuele Menegatti

***Abstract*— Touch is an important means for communication among humans. Sport instructors or dance teachers often use touch to adjust students' postures in a very intuitive way. Using tactile instructions appears thus to be a very appealing modality for developing humanoid robot motions as well. Spontaneous interpretation of tactile instructions given by users reveals itself to be a complex task for artificial systems. This paper describes a proof of concept system for robot motion creation based on tactile interaction. The system is interesting for two reasons. Firstly, it shows the feasibility of using tactile instructions for motion development. Secondly, it can be used as a tool for studying the way humans intuitively use touch to communicate. This, in turn, will allow the development of better algorithms for predicting the meaning of tactile instructions. Results of a pilot experiment are discussed, and a first set of features of tactile communication, yielded by the analysis of the data collected, is identified.**

I. INTRODUCTION

Touch is an important but often overlooked communication means used by humans. It is very rich: intensity, frequency, velocity, abruptness, contact time, surface of contact are just some of its features [1]. Touch is fundamental in the interaction between infants and their caregivers [2]. It is used to communicate both emotions and specific information, like the presence or absence of a caregiver or the identity of the person touching the infant. Studies with toddlers [3] show that touch is so informative that the sole information on touch can be used to measure the "quality of interaction". At older ages, tactile communication maintains its importance. For instance, in dance, haptic interaction is fundamental for the coordination between partners [4].

However, to date, few quantitative studies have been performed on human tactile interaction [1], mainly due to difficulties in actually measuring the numerous parameters of touches in social contexts. For instance, the way sports coaches or dance instructors use touch to communicate with their trainees remains completely unexplored. This aspect is of high interest for robotics. In fact, motion development is still one of the most time consuming tasks. Letting users correct robots' movements in the same way sport instructors modify the postures of their students appears to be a very appealing way for intuitively teaching movements to robots.

F. DallaLibera is a Research Fellow of the Japan Society for the Promotion of Science at Osaka University

T. Minato is with the JST ERATO Asada Project, Osaka University, 2-1 Yamada-oka, Suita, Osaka, 565-0871, Japan

F. Basoeki and H. Ishiguro are with the Department of Systems Innovation, Graduate School of Engineering Science, Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka, 560-8531, Japan

E. Menegatti is with the Department of Information Engineering, University of Padova, Via Gradenigo 6/B, 35131, Padova, Italy

This paper presents the possibility of using the interpretation of tactile instructions for the development of motions. The advantages of *interpreting* tactile instructions are discussed in Section II, which compares the presented idea to the approaches found in literature. A proof of concept system implementation, based on supervised learning of the meaning of tactile instructions, is described in Section III.

Data acquired from the system usage are analyzed in Section IV. The main purpose of the experiment is studying how humans employ touch to teach motions. These results, although preliminary, are important because they can be used for future development of better interpreters of the meaning of tactile instructions. In particular, the general knowledge extracted can be employed to reduce the need of teaching the meaning of tactile instructions, that is essential in the completely supervised learning approach presented here. Section V concludes the paper by summarizing the results and describing future work.

II. RELATED WORKS

In the field of the generation of robot movements, human-robot interaction is often used as a way for transferring knowledge from humans to robots. This transfer of knowledge can decrease the prohibitive learning times of self-exploration. Plenty of solutions, that often go under the name of Programming by demonstration [5] or Robot coaching [6] were presented.

A very diffused approach, motion retargeting, consists in acquiring the movement of a human performer and adapting it to the robot [7]. This technique presents several inconveniences. Generally the equipment is expensive, requires a careful setup and is not accessible to most users. Recording requires the availability of an actor able to perform the desired movement. Finally, differences between humans and robots require an intensive adaptation of the human motion, that strongly degrade the appearance of the final motion. Similarly, on-line control of robots using the data from Microsoft Kinect was recently presented as a possible inexpensive solution.

Another way for humans to teach motions to robots is through direct physical interaction [8]. When using this setup the possibilities and limitations of the robot, in terms, for instance, of joint range or maximum torque, can be easily felt by the user [9]. The idea appeared very early in the field of industrial robot arms, being referred to as direct teaching, guiding, or play back. This approach is nowadays largely employed and continues to draw attention, both for industrial manipulators and mobile robots [10], [11].

Similarly, when dealing with humanoid robots, kinesthetic demonstration [12] is often employed. Within this approach, users directly grasp and move a robot's limbs, providing demonstrations of the task that are sufficient for the robot to extract a probabilistic model of the movement.

At first glance, the approach presented in this paper, called *TbT, Teaching by Touching*, resembles kinesthetic demonstration. However, the two approaches are very different. When teaching by kinesthetic demonstration, the robot moves passively in response to the force applied to it. More specifically, during kinesthetic demonstration the robot motors are usually switched off. As possible alternatives, a subset of the motors can be made passive only when necessary [13], or compliant actuators can be employed [14].

TbT regards tactile interaction as a communication means, instead of considering it merely as a way to set a robot's posture. Specifically, the tactile information is considered as an instruction that carries the intention of the user. In fact, while kinesthetic demonstration resembles the interaction between a puppeteer and a puppet, TbT aims to mimic the interaction between a coach and a human trainee, who interprets the instructions received.

TbT presents several advantages over kinesthetic demonstration. A single touch can be associated to the simultaneous movement of both arms and legs, while it would be very difficult to move the four limbs of a robot simultaneously with classical kinesthetic demonstration. Additionally, with large robots, kinesthetic demonstration may be infeasible, if compliance and gravity compensation are not adopted.

Most importantly, when the robot *interprets* the meaning of tactile instructions, it can apply a set of corrections assumed to be useful. For instance, when receiving instructions on how to modify the motion, the robot could apply small modifications in order to satisfy criteria like dynamical stability, perfect symmetry between the right and left joints (often desired but difficult to be realized by direct manipulation), minimization of the body oscillations, of the load on the knee servomotors, and so forth.

Conceptually, kinesthetic demonstration could be considered as a special case of TbT. In fact, it is possible to make the robot interpretation correspond exactly to the effect that the application of the force would have on a robot with passive motors. Additionally, we need to notice that kinesthetic demonstration is essentially just the result of the forces applied in the interaction over the robot structure. In contrast, within the TbT approach, any feature of touch could be considered as a component of tactile instructions. Therefore, an advanced "somatic alphabet", such as the one presented in [15], could be used as the instruction input. Similarly, the input does not need to be limited to tactile sensors located on the robot surface. For instance, torque sensors in the actuators may be considered as well.

A preliminary evaluation of the TbT approach, conducted with simulated touch sensors, was provided in [16], showing TbT advantages in terms of reduction of the motion development time. This paper presents the actual implementation, with a robot equipped with tactile sensors on its whole body.

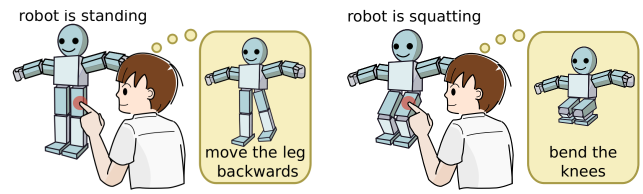


Fig. 1. Different touch meaning in different contexts. The same touch corresponds to two different meanings depending on the context.

A detailed analysis of the tactile instructions provided in a pilot experiment is also given.

III. System Implementation

Although human trainees are able to spontaneously grasp the meaning of tactile instructions given by their coaches, the interpretation reveals itself to be very complex for artificial systems. In the first instance, the meaning of tactile instructions is both context and user dependent. Fig. 1 provides an example of context dependency. If users press the upper part of the leg when the robot is standing they could imply that the robot should bend the leg backwards. However, when the robot is squatting, the same touch on the leg could mean that the robot should bend its knees further. Furthermore, experiments with a simulator [16] showed that when asked to interact freely with the robot, different people tend to give different meaning to similar tactile instructions.

Given these difficulties in the interpretation of tactile instructions, and the absence of models in literature, we adopted machine learning for the construction of the mapping from tactile instructions and their context to motion modifications. Thanks to this choice the system completely adapts to the user, who is free to create associations between touch patterns and movement modifications. Furthermore, data collected during the interaction can be analyzed, and general policies underlying the user's way of associating touch patterns to motion modifications can be extracted. More specifically, in our system, the learning of the mapping occurs during motion development. In this way the user is not requested to teach the meaning of the touch instructions in a specifically devised session, but can provide them in contexts that naturally appear during the motion development.

A conceptual schema of the motion development process is reported in Fig. 2. The user observes the motion, and provides a touch instruction to modify the movement or add new features to it. For simplicity, the current system uses a key-frame based description of the motion, i.e. the movement is defined as a sequence of postures that the robot must assume over time. Motion modifications therefore correspond to the editing of the posture assumed by the robot at a selected instant of the motion. If the robot correctly interprets the tactile instruction, then the user just keeps developing the motion. If the robot does not respond as expected, instead, the user shows the robot the desired motion modification.

Specifically, in the actual system implementation, wrong estimations of tactile instructions are signaled to the robot by using a pedal, as shown in Fig. 3. While the pedal is being

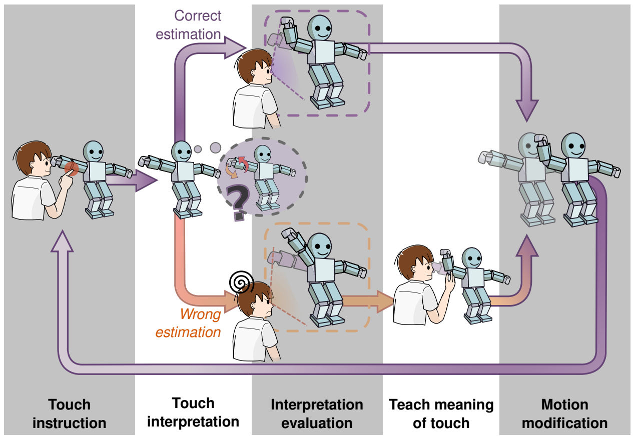


Fig. 2. Schema for motion development. Users touch the robot to edit the posture of a keyframe. The robot interprets the meaning of the tactile instruction and changes its posture accordingly. Users then evaluate the change in the robot's motor positions. If the movement corresponds to their intention then they will continue to develop the motion, otherwise they will teach the robot the meaning of the touch instruction.

pressed, the user can show the correct interpretation using two auxiliary communication means: directly setting each angle through a GUI and using kinesthetic demonstration. We stress that this doesn't bring the drawbacks of kinesthetic demonstration previously listed to the TbT approach. In fact:

1) The meaning of a touch needs to be taught just once, and can thereafter be reused. The user faces thus the drawbacks of kinesthetic demonstration much less frequently.
2) Any modality, like speech recognition, can be used.

When the pedal is released, a new association between the touch pattern given when the pedal was first pressed and the movement provided while holding the pedal down is stored in a database. Initially, this database is empty and the robot does not respond to any touch instruction, but by incrementally teaching the meaning of new instructions the tactile instruction interpretation is improved more and more.

In practice, this database of examples of the mapping from touch instructions and their context to posture modifications is used by a supervised learning algorithm to estimate the meaning of tactile instructions. A locally weighted learning algorithm [17], and in particular Kernel Regression with a specifically devised kernel, is used to predict the desired motion modification for a given touch input.

As briefly stated above, the meaning of tactile instructions is context dependent. In the current implementation, the context consists of the angular positions of all the motors and the robot's orientation. Joint angles allow discriminating, for instance, squatting positions from standing positions, which could require different interpretations of the tactile patterns as shown in Fig. 1. Similarly, the robot's orientation is important for tactile interpretation because, for example, the meaning of the touch instructions may change depending on whether the robot is standing or lying down.

Formally, let us assume the robot to have n tactile sensors and m motors. Let us then use $o = 2$ variables to describe the robot's orientation, expressed as the inclination and the orientation around the vertical and computed as in [18]. The interpretation of a touch instruction is a function that given as input $I_* \in \mathbb{R}^{n+m+o}$, i.e. the touch pattern and its context, provides an output $M_* \in \mathbb{R}^m$ that expresses an angle modification for each of the joints. Let us denote by E the number of examples of the mapping from touch instructions with their context to posture modifications. For each of these examples let us consider its input $I_i \in \mathbb{R}^{n+m+o}$ and the corresponding output $M_i \in \mathbb{R}^m$, $1 \leq i \leq E$. The estimation of the desired joint modification can be calculated as a function of the stored outputs:

$$M_* = \sum_{i=1}^{E} \omega(I_*, I_i) M_i$$

where the function $\omega(I_*, I_i)$ gives the similarity between the system input I_* and the input of the i-th example I_i. We require $\omega(I_*, I_i)$ to meet the following criteria:

1) The stronger the user pushes the sensors, the further the robot joints are rotated.
2) The more the touch pattern and context of an example differ from the system input, the less the movement associated to that example contributes to the output.
3) The examples whose touch pattern includes the pressure of sensors that are not pushed in the touch instruction I_* provide no contribution to the output.

As an example of the importance of the last rule, imagine the robot to be in a sitting position with its legs stretched forward. Assume that a user pushes the foot toes and the heel of one leg simultaneously, teaching the robot to bend the knee and bring the legs close to the body. Suppose that at a later moment the robot is touched only on the toe portion of its foot. Many users could desire to associate this touch with a simple foot rotation. The third criterion specified above prevents the complete leg for being moved, given the absence of heel pressure in I_*.

We set $\omega(I_*, I_i)$ as

$$\omega(I_*, I_i) = \begin{cases} 0 & if\ \exists s : s \in \Psi_i \land s \notin \Psi_* \\ \dfrac{\prod_{s \in \Psi_i} \bar{I}_*^{(s)} / \bar{I}_i^{(s)}}{1 + \sqrt{\|\tilde{I}_* - \tilde{I}_i\|_2^2 + \sum_{s : \notin \Psi_i} (\bar{I}_*^{(s)})^2}} & otherwise \end{cases}$$

where, assuming to normalize all the components of I_i and I_* by their respective variance in the dataset of E examples,

- $\bar{I}_i \in \mathbb{R}^n$ denotes the set of components of the i-th input corresponding to the tactile sensors, and $\tilde{I}_i \in \mathbb{R}^{m+o}$ denotes the remaining components of I_i, i.e. the context. Analogous definitions are given for $\bar{I}_*$ and $\tilde{I}_*$.
- $\bar{I}_i^{(s)} \in \mathbb{R}$ denotes one of the components of $\bar{I}_i$, i.e. the force applied to the s-th tactile sensor, $1 \leq s \leq n$. An analogous definition is given for $\tilde{I}_*^{(s)}$.
- Ψ_i denotes the set of sensors pushed in the i-th example, i.e. $\Psi_i = \{s : \bar{I}_i^{(s)} > 0\}$.

Essentially, the condition $\exists s : s \in \Psi_i \land s \notin \Psi_*$ is used to satisfy the third criterion, while the numerator and the denominator of the fraction are used to comply with the

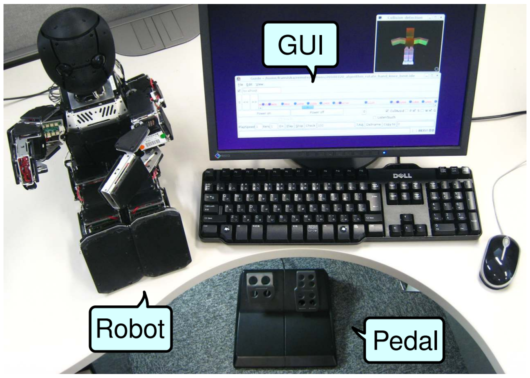


Fig. 3. System implementation. A pedal allows the users to easily switch between the *motion development* mode and the *touch meaning provision* without having to move their hands away from the robot.

first and the second criterion, respectively. Actually, a set of different ω functions were tested by the same user in the development of a walking motion, and the one that led to the least number of unexpected robot responses was chosen.

IV. EXPERIMENT

In order to test the feasibility of the proposed approach, four users were asked to teach a motion to a humanoid robot using the proposed touch interface. Specifically, the subjects developed *Algorithm Exercise*, a famous dance appearing in a Japanese TV show for children. This dance was chosen because it is complex enough to require the user to teach a large number of different postures to the humanoid but it is simple from the view point of balancing the robot. Snapshots of the realized motion[1] are shown in Fig. 6.

The robot employed in the experiments is VStone M3-Neony, a 22 DOF humanoid robot equipped with 90 tactile sensors on its whole body. The location of its actuators and tactile sensors is reported in Fig. 4.

The tactile instruction corresponds to a 114-dimensional vector, which indicates the pressure of each sensor, the robot's posture and its orientation. Given the very high dimensionality of the input, a great variety of responses to tactile instructions can be taught.

The test subjects — three males and one female — are Japanese Engineering students at Osaka University. They are all right handed, and their age ranges from 23 to 25 years. These users are familiar with VisiON 4G, a robot that has a structure similar to the one of M3-Neony but lacks tactile sensors. Furthermore, the subjects had never used the TbT interface and did not know its underlying concepts. The subjects could effectively use our system to develop the motion, with an explanation of the system usage that required less than 5 minutes. The time required for development greatly varied between the users, ranging from 1.5 to 8 hours. Actually, for dancing motions, there is no clear criterion for determining when the motion is acceptable, and users decide that the motion is satisfactory with very different criteria.

[1]A video is available at
http://robotics.dei.unipd.it/~fabiodl/video.php?alg

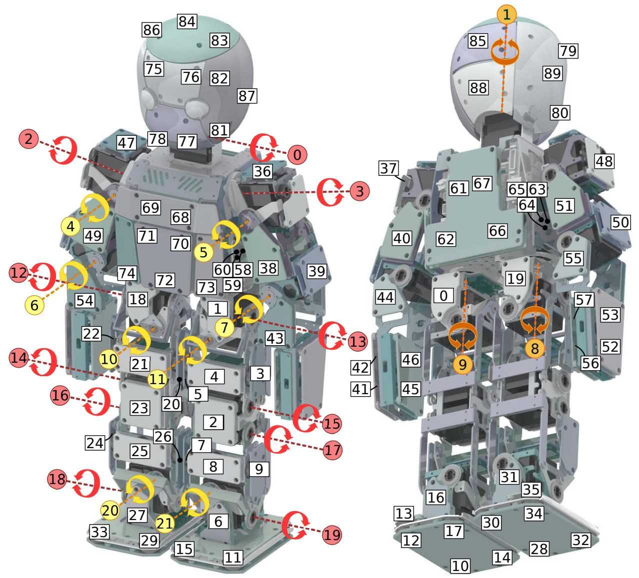


Fig. 4. Location of the motors and tactile sensors of M3-Neony. Motor IDs are shown inside circles, while sensor IDs are shown inside rectangles.

For instance, a user tried to reproduce the motion shown with very high precision in movements and timings, while another focused mainly on the motion smoothness, and a third terminated its work as soon as the gestures of the robot corresponded to the human ones. Future works will also test the realization of motions with quantitative goals, like "kick a ball at least 50 cm far".

During motion development, the users provided the meaning of an average of 95.75 (standard deviation 6.55) tactile instructions. Actually, once instructions are taught, these can not only be reused, but also combined, simply by pressing multiple sensors simultaneously, as the equations previously reported indicate. For instance, if the robot is taught to look downwards when its chin is pressed and to turn its head leftwards when its right cheek is touched, then if its chin and right cheek are pushed together, the robot will shift its gaze toward its own left foot. In the following discussion let us denote by the word *instruction* a basic association between sensors and angle changes taught, and by the term *touch* a tactile pattern applied to the robot, that, as in the previous example, can consist of multiple *instructions*.

The subjects provided an average of 867.5 *touches*, that were translated into an average of 1181.2 *instructions*, showing that the users actually exploited the superposition of several tactile *instructions* in the same *touch*. Fig. 5 reports the ratio between the number of touch meanings taught and the number of touches provided. This ratio decreases over time, showing that the users need to teach fewer and fewer instructions because they can effectively reuse the ones they already taught.

As a first analysis of the mappings between touch instructions and motor posture changes taught by each user, we calculated the mutual information between the value of each sensors and the rotation given for each of the motors. To compute the mutual information, we initially discretized

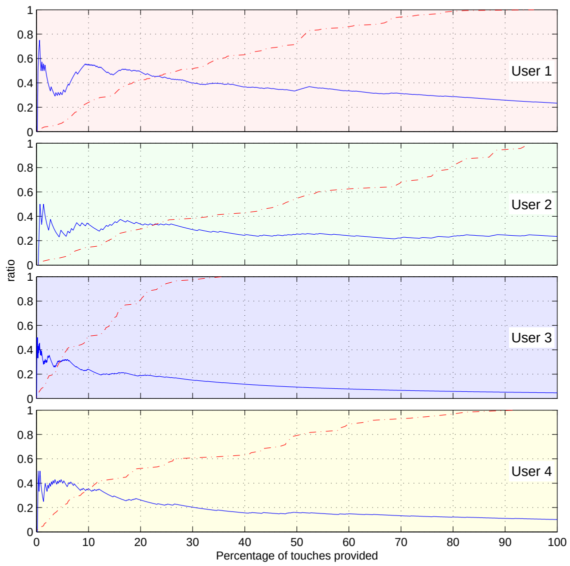
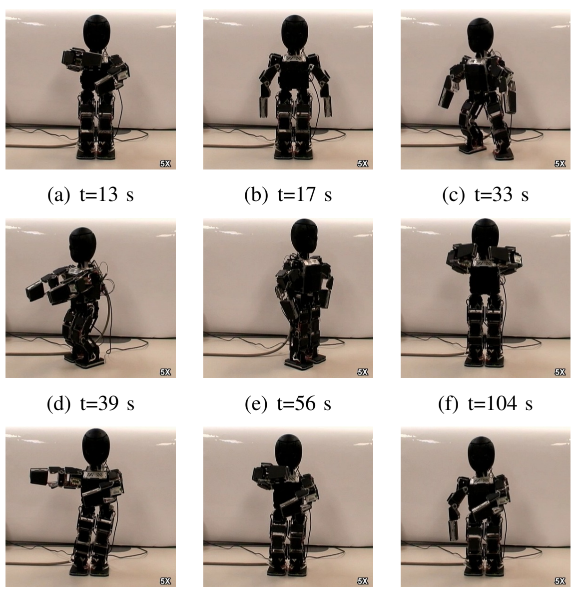


Fig. 5. Touch meanings provision over time for the four users. Each panel refers to a user, as indicated on the right side. The x axis represents the cumulative percentage of touches provided throughout the whole experiment. The solid line shows the ratio between the touch meanings taught and the number of touches provided. The dashed line shows the ratio between the number of meanings taught and the total number of meanings taught by the user during the whole experiment.



(a) t=13 s (b) t=17 s (c) t=33 s

(d) t=39 s (e) t=56 s (f) t=104 s

(g) t=127 s (h) t=173 s (i) t=183 s

Fig. 6. Snapshots of the robot performing the target movement.

the data. In detail, each sensor information was reduced to a binary value, 0 for a pressure less than 20% of the maximum force measurable by the sensor and 1 otherwise. Similarly, each motor change information was set to 0 if the user moved the motor less than 5 degrees, to -1 if the movement exceeded 5 degrees in clockwise direction and +1 if the movement exceeded 5 degrees in counter clockwise direction. The threshold of 20% for the touch sensor was chosen empirically, observing that when users touch a sensor they commonly apply a force much higher than the 20% of the maximum measurable force, while noise is far below this threshold. Similarly, the threshold of 5 degrees for the motor rotations was selected observing that intentional joint angle changes exceed 5 degrees while unintentional ones are contained within this threshold.

Fig. 7 illustrates the results, normalized as in [19]. We notice that the users mainly touched the sensors on a limb to move motors on the same limb. However, the subjects didn't restrict themselves to a one to one correspondence between joints and sensors. Several sensors were used to actuate a single joint and conversely a single sensor actuated several joints. We also notice that different users tend to provide different mappings, even if the experimental conditions (robot, task and subjects' background) are the same. This confirms the user dependence of the touch protocol [16], and therefore the need for using different mappings for different users.

Interestingly, statistical dependence between the sensors on the top of the head and the leg motors for some users can be observed. Direct inspection of the data shows that the first user taught the robot to squat in response to touches on the head. Similarly the fourth user employed the sensors on the back of the robot's head to make the robot lean forwards and sensors on the front of the head to make the robot lean backwards. For the first user we also notice statistical dependence between the sensors placed on the side of the robot's body and the corresponding leg. By examining the data, it was found that sensors on the side were used to make the robot rotate the corresponding leg and bring the knee outwards on that side (see Fig. 6(c)). Statistical dependence between the sensors on the upper part of the left leg ($s00.lHipB$ and $s01.lHipF$) and motors of the right leg ($m12.rHipP$) derives from the fact that often when the posture of one leg was changed the other leg was moved as well to maintain the balance.

Multiple motors are usually moved by a single touch instruction. We thus checked whether consistencies in the relationship between the changes of different motors can be found. Computation of the mutual information between couples of motors yields the results reported in Fig. 8. We notice a very strong statistical dependence between motors belonging to the same limb. Statistical dependence between the two legs is also observable. As discussed above, this can be explained by the desire of the test subjects to keep the robot balanced throughout the whole experiment.

The high statistical dependencies between different motors suggest that the motor changes could actually lie in a low-dimensional manifold of the whole 22-dimensional motor space. In particular, we analyzed how well the motor change relative to the e-th example fits in a linear subspace of dimension q constructed from the motor modifications taught in the previous $e-1$ examples. In detail, we took the motor changes specified in the first $e-1$ examples $M_1 \dots M_{e-1}$ and applied Principal Component Analysis (PCA). We then considered the e-th example motor change M_e, and projected the resultant vector on the subspace defined by the first q principal components $v_1 \dots v_q$. Finally,

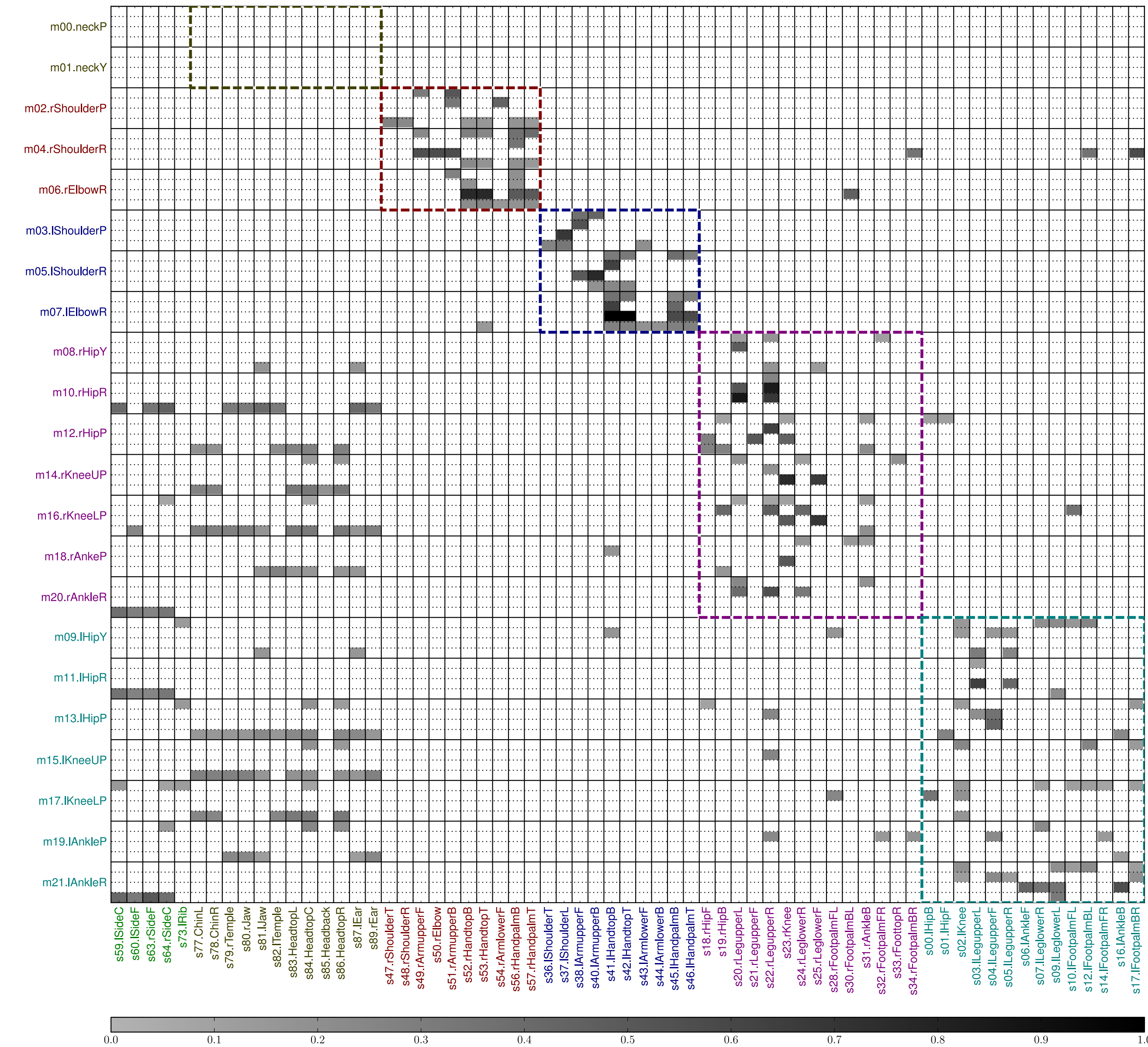


Fig. 7. Mutual information between sensor values and motor angle changes for the four subjects. Columns correspond to sensors, rows to motors. Each label consists of an id (see Fig. 4), followed by a shorthand name. The color indicates the robot part. The entries that correspond to sensor and motors of the same robot part are highlighted by a dashed rectangle. The intersection between a column (sensor) and a row (motor) is divided into four sections by dotted lines. Each section corresponds to one of the users, with the topmost section corresponding to User 1 and the bottommost section corresponding to User 4. The color of each section indicates the normalized mutual information value. For clarity, only mutual information values higher than 0.01 are shown, and only sensors that have mutual information value higher than 0.01 with at least one motor are reported.

we calculated the infinity norm of the reconstruction error $\epsilon_q(e) = \left\| M_e - \sum_{i=1}^{q} M_e^T v_i v_i \right\|_\infty$.

Table I reports the reconstruction error for different settings of q, $1 \le q \le 22$, averaged over all the examples e, $1 \le e \le E$. For comparison, the error obtained by applying PCA on the whole set of examples $M_1 \dots M_E$ is also reported. Precisely, the columns with header P.E. (previous examples) and A.E. (all examples) report the projection error on the subspace computed from $M_1 \dots M_{e-1}$ and from $M_1 \dots M_E$, respectively. We notice that the difference is very limited, indicating that considering the meanings taught by the user for previous instructions can help in predicting the subspace where the meaning of new instructions lie.

As widely known, motions of humans and humanoids can often be described in a low-dimensional subspace of the joint space as well [20]. We therefore analyzed whether the motion modification M_e can be projected with little errors on the subspace where the robot movement lies. Specifically, for each motor modification M_e, we identified all the postures that the user brought the robot to before teaching M_e. We applied PCA to these postures and determined the principal components $\bar{v}_1 \dots \bar{v}_q$. The average reconstruction error norm obtained by projecting the posture modification M_e on the subspace defined by $\bar{v}_1 \dots \bar{v}_q$ is also reported in the columns

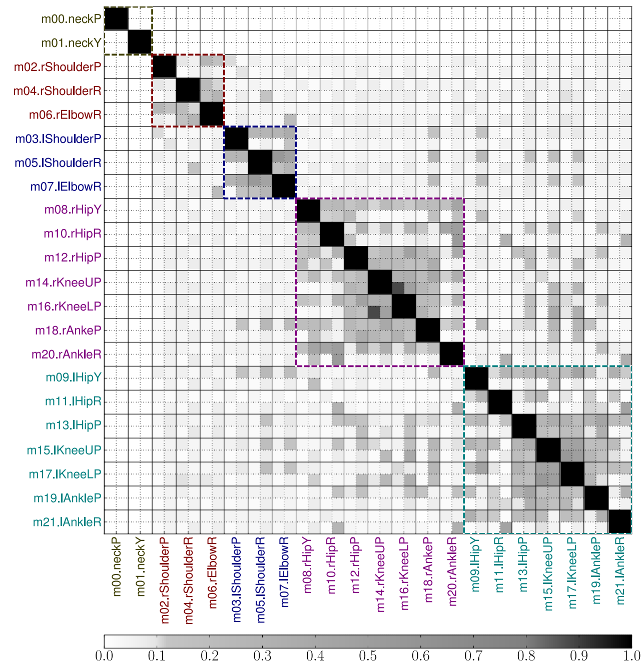


Fig. 8. Mutual information between couples of motors for the four subjects. Each label consists of an id, corresponding to the ones reported in Fig. 4, followed by a shorthand name. The colors of the labels indicates the robot part. The entries that correspond to motors of the same robot part (e.g. same limb) are denoted by a dashed square. The intersection between a column (sensor) and a row (motor) is divided into four sections by dotted lines. Each section corresponds to the data of one of the users. Specifically, the top-left, top-right, bottom-left, bottom-right sections correspond, respectively, to User 1, 2, 3 and 4. The color of each section indicates the normalized mutual information value.

with header P.P. (previous postures) of Table I.

We notice that, except for $q < 5$, the reconstruction error is comparable to the one for the projection on the subspaces constructed using the motor change information M_e, $1 \leq e \leq E$. Intuitively, this means that when the users set the postures that realized the target motion, they restricted their instructions to movements similar to those that compose the target motion itself, instead of setting them in a completely free manner (i.e. using motor changes in the whole motor space). The target motion could thus be used as additional information for a better estimation of the tactile instruction meaning. For instance, once an initial guess of the motion modification is computed, this could be projected onto the subspace where the motion being developed lies. Clearly these are preliminary results that need to be verified extensively, and that could be influenced by the task choice.

V. Conclusions and Future Work

This paper presented the idea of using the *interpretation* of tactile instructions for robot motion development. A proof of concept system was implemented, and data collected by making people interact with the system were analyzed. The system is interesting for two aspects.

Firstly, it shows that tactile interaction is a feasible method for the development of robot motions. Preliminary experiments with four students showed in fact that all the subjects were able to develop the motion without problems, and could use the teaching by touching system without any training.

Secondly, our system shows the possibility of using humanoids as a tool for studying how humans employ touch to communicate specific knowledge. More in detail, the system realizes the mapping between tactile instructions and motion modifications by instance based learning. These instances can be analyzed to identify general features of the mapping. In turn, this allows the development of better interpreters that require users to teach the meaning of less instructions. In other terms, studying the data acquired with the supervised learning system presented here allows moving toward an unsupervised touch instruction interpreter, or at least to reduce the number of examples that must be provided. Furthermore, the data collected by the system could in future be analyzed from the perspective of human sciences. Clearly, measuring features of human-humanoid communication is much easier than measuring human-human communication, because the robot's state can be easily observed and controlled. Although humans and humanoids are not identical, similarity in their shape and tendency of humans to anthropomorphize inanimate objects [21] may allow us to shed light on human-human communication by observing human-humanoid communication.

TABLE I

Reconstruction error for different number of dimensions

Dim	User 1			User 2			User 3			User 4		
	P.E.	A.E.	P.P.	P.E.	A.E.	P.P.	P.E.	A.E.	P.P.	P.E.	A.E.	P.P.
1	30.0	28.7	46.4	24.0	21.1	52.8	20.3	19.6	35.5	35.0	31.5	57.0
2	26.9	24.9	39.5	21.5	18.4	37.7	18.3	16.9	29.0	30.9	27.8	46.5
3	23.9	22.0	35.5	19.4	16.1	29.4	16.9	14.9	19.0	27.0	23.4	30.7
4	22.3	20.3	25.7	15.9	13.6	26.0	15.4	13.5	17.5	25.4	20.9	23.8
5	20.1	17.8	22.3	13.4	10.9	17.9	14.2	12.1	14.8	22.7	17.8	20.3
6	18.9	16.1	19.5	11.4	9.4	14.1	12.6	11.0	12.7	19.4	16.0	18.3
7	17.6	13.7	17.1	10.6	8.1	10.8	11.3	10.1	10.9	14.8	13.6	15.1
8	16.1	12.6	14.5	8.6	7.1	9.9	10.0	8.6	9.7	13.2	10.1	12.0
9	15.0	10.5	13.2	7.7	6.0	8.2	9.4	8.0	8.4	12.4	8.5	10.5
10	13.6	9.7	11.4	7.3	5.0	7.7	9.0	7.2	6.9	11.2	7.1	9.9
11	12.3	8.6	10.2	6.9	4.2	5.8	8.3	6.2	5.8	9.7	5.8	9.3
12	10.9	7.2	9.2	6.0	3.9	4.7	7.6	5.4	4.7	8.5	4.9	8.8
13	9.5	6.5	7.3	5.3	3.2	4.2	6.8	4.5	4.2	7.9	4.2	7.5
14	8.6	5.5	5.9	4.9	2.6	3.7	5.7	3.7	3.6	7.0	3.4	6.5
15	7.4	4.2	4.9	4.3	2.2	3.3	5.1	3.1	2.9	6.3	2.7	5.2
16	6.3	3.2	4.0	3.8	1.8	3.0	4.2	2.4	2.3	5.7	2.1	4.7
17	5.3	2.6	3.0	3.2	1.3	2.6	3.6	1.7	1.8	4.9	1.5	4.0
18	4.2	2.3	2.2	2.6	0.9	2.2	2.5	1.3	1.4	4.5	1.2	3.1
19	3.1	1.2	1.6	1.6	0.6	1.4	1.8	0.7	0.9	3.8	0.8	2.3
20	1.2	0.4	0.8	0.7	0.1	0.9	1.0	0.3	0.6	2.1	0.5	1.3
21	0.4	0.1	0.1	0.4	0.0	0.1	0.5	0.0	0.2	1.0	0.2	0.3
22	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

The analysis of the experimental data shows that specific algorithms for interpretation of tactile instructions are required, since the interpretation cannot be reduced to a simple one-to-one mapping between sensor and joint angle modifications. Furthermore, results confirmed user dependence of the mappings. However, general features were also identified:

1) Sensors on one limb are very frequently employed to move joints of the same limb.
2) Associations between different parts, e.g. sensors on the head to make the robot bend the knees, should

also be considered by the touch instruction interpreter.

3) The angle modifications that should be performed in response to a tactile instruction can be constrained to a linear subspace of the whole motor command space.
4) The subspace generated by the keyframe of the motions under development is (at least for some tasks) a good subspace for the representation of the movements expected in response to a touch instruction.

This results provide hints for the development of better tactile instruction interpreters. In detail, in a probabilistic framework, the pressure of a group of sensors on a limb should be mapped with higher probabilities to movements that regard the same limb. However, the system should include the possibility of representing “higher level” behaviors, like squatting down when the head top is touched. Furthermore, the relationship between the responses to instruction and the motion subspace suggests to give higher probabilities to instruction responses that lie on the motion subspace.

Additionally, data analysis shows that refining the mapping online, as proposed in this paper, is a feasible solution. Specifically, for all the subjects, the need for teaching tactile instruction meanings decreases over time.

Future work will concentrate on the evaluation of the system. In particular, the motion development time for naive users will be compared with the one of literature approaches. At the same time, users enjoyment will be evaluated. We can in fact expect some users to prefer interacting with an active robot rather than with a passive robot or a GUI.

Actually, for a fair comparison, the system should be compared with other approaches after it had already been trained, i.e. after the database between tactile instructions and motion modification had already been populated to some extent. We also stress that the great advantage of the presented system is the possibility to seamlessly introduce a set of heuristics, like auto-balancing, that improve the response to tactile instruction. The current system implementation does not exploit this feature on purpose, to show that motion development through TbT does not require the implementation of any particular heuristic.

Along these lines, we chose to employ the simplest motion representation possible, a key-frame based representation. This leads to the natural assumption that motion modifications consist in posture modifications. However, we stress that TbT is not restricted to a keyframe based motion representation. In fact, in [22] we presented, only in simulation, the idea of having the robot controlled by a Central Pattern Generator (CPG) and of modifying the complete movement by changing the CPG parameters by touching. We also note that, although the users teach static postures, the system can be used to develop dynamic motions, like jumps [16].

A limitation of the current touch interpreter is that the touches are considered independently, and not as a stream of information. Future work will address this topic.

References

[1] M. J. Hertenstein, “Touch: Its communicative functions in infancy,” *Human Development*, vol. 45, pp. 70–94, 2002.

[2] T. Field, *Touch*. MIT Press, Bradford Books, 2003.

[3] F. Tanaka, A. Cicourel, and J. R. Movellan, “Socialization between toddlers and robots at an early childhood education center,” *Proc. of the National Academy of Sciences*, vol. 104, no. 46, pp. 17 954–17 958, 2007.

[4] T. Takeda, Y. Hirata, and K. Kosuge, “Hmm-based error recovery of dance step selection for dance partner robot,” in *2007 IEEE Int. Conf. on Robotics and Automation (ICRA 2007)*, Roma, Italy, 2007, pp. 1768–1773.

[5] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, “A survey of robot learning from demonstration,” *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[6] M. Riley, A. Ude, C. Atkeson, and G. Cheng, “Coaching: An approach to efficiently and intuitively create humanoid robot behaviors,” in *IEEE-RAS 6th Int. Conf. on Humanoid Robots (Humanoids 2006)*, Genoa, Italy, 2006, pp. 567–574.

[7] S. Nakaoka, A. Nakazawa, K. Yokoi, H. Hirukawa, and K. Ikeuchi, “Generating whole body motions for a biped humanoid robot from captured human dances,” in *2003 IEEE Int. Conf. on Robotics and Automation (ICRA 2003)*, Taipei, Taiwan, 2003, pp. 3905–3910.

[8] H. S. Raffle, A. J. Parkes, and H. Ishii, “Topobo: a constructive assembly system with kinetic memory,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, ser. CHI ’04, 2004, pp. 647–654.

[9] P. Kormushev, S. Calinon, and D. G. Caldwell, “Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input,” *Advanced Robotics*, 2010.

[10] Y. Maeda, T. Ushioda, and S. Makita, “Easy robot programming for industrial manipulators by manual volume sweeping,” in *2008 IEEE Int. Conf. on Robotics and Automation (ICRA 2008)*, Pasadena, CA, USA, 2008, pp. 19–23.

[11] T. Tsuji and T. Ito, “Command recognition by haptic interface on human support robot,” in *2009 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2009)*, St. Louis, MO, USA, 2009, pp. 3178–3183.

[12] M. Hersch, F. Guenter, S. Calinon, and A. Billard, “Dynamical system modulation for robot learning via kinesthetic demonstrations,” *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1463–1467, 2008.

[13] T. Yoshikai, M. Hayashi, and M. Inaba, “Simultaneous learning and recalling system for wholebody motion of a humanoid with soft sensor flesh,” in *Intelligent Autonomous Systems 10 - IAS-10*, Ottawa, Canada, 2008, pp. 229–237.

[14] O. Ly and P.-Y. Oudeyer, “Acroban the humanoid: playful and compliant physical child-robot interaction,” in *SIGGRAPH ’10: ACM SIGGRAPH 2010 Emerging Technologies*, Los Angeles, CA, USA, 2010, pp. 1–1.

[15] W. D. Stiehl, L. Lalla, and C. Breazeal, “A somatic alphabet approach to sensitive skin for robots,” in *2004 IEEE Int. Conf. on Robotics and Automation (ICRA 2004)*, New Orleans, LA, USA, 2004, pp. 2865–2870.

[16] F. DallaLibera, T. Minato, I. Fasel, H. Ishiguro, E. Pagello, and E. Menegatti, “A new paradigm of humanoid robot motion programming based on touch interpretation,” *Robotics and Autonomous Systems*, vol. 57, no. 8, pp. 846–859, 2008.

[17] W. S. Cleveland and S. J. Devlin, “Locally weighted regression: An approach to regression analysis by local fitting,” *J. of the American Statistical Association*, vol. 83, no. 403, pp. 596–610, 1988.

[18] H. Luinge and P. Veltink, “Measuring orientation of human body segments using miniature gyroscopes and accelerometers,” *Medical and Biological Engineering and Computing*, vol. 43, pp. 273–282, 2005.

[19] A. Strehl and J. Ghosh, “Cluster ensembles — a knowledge reuse framework for combining multiple partitions,” *The J. of Machine Learning Research*, vol. 3, pp. 583–617, 2003.

[20] R. Amit and M. J. Mataric, “Parametric primitives for motor representation and control,” in *2002 IEEE Int. Conf. on Robotics and Automation (ICRA 2002)*, Washington, DC, USA, 2002, pp. 863–868.

[21] B. R. Duffy, “Anthropomorphism and the social robot,” *Robotics and Autonomous Systems*, vol. 42, no. 3-4, pp. 177–190, 2003.

[22] F. DallaLibera, T. Minato, H. Ishiguro, and E. Menegatti, “Direct programming of a central pattern generator for periodic motions by touching,” *Robotics and Autonomous Systems*, vol. 58, pp. 847–854, 2010.