

Research Article

A Distributed Tactile Sensor for Intuitive Human-Robot Interfacing

Andrea Cirillo, Pasquale Cirillo, Giuseppe De Maria, Ciro Natale, and Salvatore Pirozzi

*Dipartimento di Ingegneria Industriale e dell'Informazione, Università degli Studi della Campania "Luigi Vanvitelli",
Via Roma 29, 81031 Aversa, Italy*

Correspondence should be addressed to Andrea Cirillo; andrea.cirillo@unicampania.it

Received 2 February 2017; Accepted 28 March 2017; Published 27 April 2017

Academic Editor: Calogero M. Oddo

Copyright © 2017 Andrea Cirillo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Safety of human-robot physical interaction is enabled not only by suitable robot control strategies but also by suitable sensing technologies. For example, if distributed tactile sensors were available on the robot, they could be used not only to detect unintentional collisions, but also as human-machine interface by enabling a new mode of social interaction with the machine. Starting from their previous works, the authors developed a conformable distributed tactile sensor that can be easily conformed to the different parts of the robot body. Its ability to estimate contact force components and to provide a tactile map with an accurate spatial resolution enables the robot to handle both unintentional collisions in safe human-robot collaboration tasks and intentional touches where the sensor is used as human-machine interface. In this paper, the authors present the characterization of the proposed tactile sensor and they show how it can be also exploited to recognize haptic tactile gestures, by tailoring recognition algorithms, well known in the image processing field, to the case of tactile images. In particular, a set of haptic gestures has been defined to test three recognition algorithms on a group of 20 users. The paper demonstrates how the same sensor originally designed to manage unintentional collisions can be successfully used also as human-machine interface.

1. Introduction

How users will interact with robots of the future which came out of the factories? This is still an open question, certainly not through a keyboard and a mouse or through a heavy teach pendant. Someone says that speech will be the preferred interaction modality, but some decades ago this was envisioned for the personal computers too, and this did not happen, while nowadays touchpads are by far the most widespread interface of both PCs and other digital devices, from smartphones and tablets to car on-board computers. Tactile interaction is becoming the preferred way to provide commands to our digital assistants and ask them to do something for us. Imagine that such a modality was available also for interacting with robots, then it would be quite natural to command robots by simply touching them. Robots are 6-dimensional machines able to execute complex tasks, so a limited number of simplified touches could not be enough to exploit all robot abilities. The use of complex haptic gestures could be the solution to communicate in a more intuitive modality with robots.

Nowadays, the use of human-machine interfaces (HMI) is exploited to allow robust interaction between the human and the robot. HMI enable the robot to perceive the users and in particular their important communication cues, such as speech, gestures, and head orientation.

A huge number of HMI solutions exist and most of them exploit more than one perception system (multimodal perception). A large portion is constituted by classical vision-based systems [1–3] in which the main drawbacks are the background variability, the bad lighting conditions, and the computational time. Another kind of visual interaction system is the Microsoft Kinect [4], an RGB-D camera that is used to detect human motions, that is, face and hand gestures, head orientation, and arm posture. Such technology is currently used as HMI only in computer games, but Intel has recently launched a similar technology [5], intended to interact with laptops as a complement to the touchpad. Another smaller part is constituted by more complex systems in which different perceptual and communicative cues are fused together in order to implement multimodal dialogue

components for the human-robot interaction. They include systems for spontaneous speech recognition, multimodal dialogue processing, and visual perception of a user, for example, localization, tracking and identification of the face and hand, and recognition of pointing gestures [6–9]. However, all the mentioned approaches use different sensors, that is, camera or Kinect, microphone, and IMUs, and they use computationally expensive frameworks to fuse the data acquired from all the sensors and then take a decision. Hence, integrating such systems in a real-time task in which there is a physical collaboration between the human and the machine, represents, by now, an important challenge.

Alternatively, the tactile interaction offers an intuitive and very fast modality for human-robot interaction. Haptic cues can usually be interpreted very quickly as demonstrated in [10, 11] and tactile sensors can be used to classify different types of touch [12, 13]. The KUKA LWR 4+ has been used in [14] for executing complex tasks in collaboration with humans; switching between task segments and control modalities has been implemented through simple haptic gestures that the user had to apply to the last robot link, for example, pushing or pulling in a certain Cartesian direction. Such approach has only a limited number of gestures due to the limited accuracy in the estimation of contact force vector. The use of a distributed tactile map enlarges the number of recognizable haptic gestures while maintaining a fast response.

Actually, as shown in Dahiya et al. [15], it is not unusual to cover the whole body of the robot or some of its parts with an array or patches of tactile/force sensors to improve the safety in tasks that require humans and robots to collaborate. In fact, in literature several tactile sensors have been presented [16–21] and, as shown in [22], microfabricated devices based on piezoresistive, piezoelectric, and capacitive technology found a large diffusion also in prosthetics and in medicine applications. Actually, most of them have limited reliability, flexibility, and robustness or need a complex circuitry for signal conditioning and acquisition, that is, resistive and piezoresistive technology. Additionally, the use of absorbing and stretchable materials could introduce hysteretic effects. Moreover, a higher spatial resolution is necessary when the users want to communicate with the robots through haptic gestures traced simply by a finger, for example, letters or numbers, instead of more rough gestures such as punches or pats as presented by Kaboli et al. [23].

Starting from the rigid sensor prototype described in [24], the authors of the present paper propose a conformable tactile sensor prototype able to measure a distributed tactile map provided by several interconnected sensing modules constituted by a 2×2 matrix of taxels. The design of the conformable sensor prototype affects the maximum achievable flexibility and wiring of the sensor, so an innovative scanning strategy has been used to reduce the number of components, wires, and PCB layers achieving a more flexible solution. In order to analyze the conformability of the developed sensor, it has been installed on a KUKA LWR 4+ that, by now, is one of the most used robotic arms in the human-robot interaction (HRI) research field. The presented sensor exploits low power consumption optoelectronic devices.

These components made the design of a distributed sensor that needs a very low power to properly work easy, differently from most of the existing optoelectronic solutions. The sensor can be used to reconstruct the 3D contact forces, a tactile map, and the contact point location in which the contact occurs, at the same time. In [25], the authors presented how the proposed sensing solution can be used as distributed force sensor for robot control and collision detection, for example, to handle the interaction between humans and robots. This paper demonstrates how the same sensor can be used to intentionally interact with the robot by exploiting the tactile map.

Inspired to existing solutions, different recognition algorithms have been suitably redesigned for the sensor to optimize the recognition of a set of touch gesture with increasing complexity. The recognition algorithms have been validated through several experimental results, showing how the tactile sensor can be used to communicate with the robot in a natural way. Moreover, this paper discusses the design procedure, which allowed the realization of the flexible version of the sensor. The generalization of this procedure allows maintaining high conformability also for realization of future prototypes with a different design.

The paper is structured as follows: Section 2 describes the design procedure and the technology of the distributed sensor; Section 3 describes how the tactile sensor is used as HMI and it details different recognition algorithms of haptic gestures; Section 4 discusses the conclusions and possible future works.

2. The Distributed Tactile Sensor

This section briefly recalls for completeness the working principle of the tactile sensor and discusses the generalization of the design procedure, partially presented by the authors in [26], which allows obtaining an optimal design of distributed sensors with high conformability.

2.1. The Working Principle. The working principle of the distributed tactile sensor is based on a well-assessed concept, originally used for the development of the force/tactile sensor described in [27], that is, the use of a PCB (Printed Circuit Board) constituted by couples (emitter/detector) of optoelectronic devices to detect the local deformations, generated by an external contact force applied to a deformable layer that covers the optoelectronic layer. For the distributed version, presented in [24], the electronic layer is constituted by an interconnection of a number of identical sensing modules, each being capable of measuring the three components of the contact force applied to it. In particular, each sensing module is constituted by four taxels organized in a 2×2 matrix. Each taxel consists of an optical LED/PT (Light Emitting Diode/PhotoTransistor) couple spectrally matched. A deformable elastic layer is positioned above the 4 optoelectronic couples. This deformable layer has a hemispherical shape on the top side, while on the bottom side it presents four empty cells, vertically aligned with the four optoelectronic couples. At rest, part of the light, emitted by the LEDs and reflected by the four cells, is captured by the PTs.

When an external force is applied to the deformable layer, it produces deformations for all the four taxels constituting the sensing module. These deformations produce variations of the reflected light and, accordingly, of the photocurrents generated by the PTs. The interested reader can find in [24] all the details concerning the realization of both optoelectronic and deformable layers, with illustrative pictures, for the rigid prototype of the distributed sensor.

2.2. The Conformable Distributed Tactile Sensor. The design of the conformable sensor patch, based on flexible PCB technology, is an evolution of the rigid prototype design made possible by two main characteristics of the proposed sensor: the smart scanning control strategy and the low power consumption of a sensing module.

The scanning strategy allows a substantial reduction of the number of wires with respect to the taxel number, which makes the use of a flexible PCB with a limited number of layers corresponding to a reduced thickness possible, thus guaranteeing high conformability of the optoelectronic board and low production cost. The basic idea, inspired by the concept in [28], is to connect the sensing modules in groups sharing 4 A/D channels (4 is the number of taxels of one sensing module), and to switch on and off, with a cyclic pattern, the sensing modules, by ensuring that in each time instant, for each group, only one module is turned on, while all the others that share the same A/D channels are turned off so that the PTs operate as open circuits. This characteristic allows reducing the number of A/D channels necessary to interrogate a sensor patch. Differently from [28], here multiple sensing modules can be directly driven by the same Microcontroller Unit (MCU) digital I/O, without using an external power supply, since each LED works with a forward current of about 1 mA and the voltage supply for all components is the 3.3 V, available from the MCU pins. Hence, since different groups use different A/D channels, sensing modules belonging to different groups can share the same digital I/O as power supply, by reducing also the number of digital I/O necessary to switch on and off the sensing modules during the interrogation. In summary, this scanning strategy allows the reduction of the whole sensor power consumption and of the number of A/D channels and digital I/O required to interrogate the sensor patch, with a consequent simplification of the wiring.

The design of the flexible PCB affects the maximum achievable conformability of the sensor patch; hence some observations are in order. Firstly, the installation of the electronic components on the flexible PCB reduces the flexibility property, depending on both the number and the dimensions of the components. Secondly, the flexibility depends also on the number of layers necessary for the wiring; thus a proper routing of the PCB should be carried out. This requires a suitable redesign of the optoelectronic layer of the original rigid prototype to maximize conformability of the new tactile sensor version.

First of all, note that the sensing modules are only constituted by the optoelectronic components (SFH4080 and SFH3010) that have a SmartLED package 0603 (with

dimensions $1.3 \times 0.8 \times 0.65 \text{ mm}^3$) and additional resistors to drive the LEDs (a resistor for each LED), with package 0402 (with dimensions $1 \times 0.5 \times 0.32 \text{ mm}^3$). By using the scanning strategy described above, each group of sensing modules can share the resistors in series with the PTs. With this choice, the number of resistors needed to convert the photocurrents into the voltages acquired by the A/D channels is reduced from the number of PTs to the number of A/D channels used during the scanning. Furthermore, these resistors can be mounted directly near the A/D channels, by avoiding adding components on the conformable part of the PCB. Since no additional Integrated Circuits (ICs) with cumbersome package are used for the conditioning electronics, the types of components to mount on the flexible PCB, for each taxel, are only three and small enough to maintain an high flexibility of the PCB.

The adopted scanning strategy, in addition to the reduction of the number of components, allows also a simplification of the wiring, by reducing the number of needed layers. By generalizing the adopted interrogation technique (see Figure 1), a total of n sensing modules (corresponding to $4n$ taxels) can be organized in m groups, each one constituted by p sensing modules. Since the sensing modules of each group share 4 A/D channels, the number of external wires needed to interrogate a sensor patch is equal to $4m + p$ (plus one for the ground). As a consequence, to minimize the number of wires needed for a sensor patch, the following constrained optimization problem can be solved:

$$\begin{aligned} \min_{m,p} \quad & (4m + p) \\ \text{subject to:} \quad & mp = n, \quad m, n, p \in \mathcal{N}^+. \end{aligned} \quad (1)$$

The conformable sensor patch presented here has 144 taxels, divided into $n = 36$ sensing modules. By solving the optimization problem (1), the results are $m = 3$ groups (corresponding to $4m = 12$ A/D channels) and $p = 12$ digital I/O, for a total of 24 wires plus one for the ground. With this choice, the routing of a whole tactile sensor can be completed by using a flexible PCB with only 4 layers.

Design of the routing has been carried out by using a semiautomatic routing algorithm. The active surface of the sensor patch (corresponding to the sensing elements) is about $47 \times 47 \text{ mm}^2$, while the 25 wires, needed to interrogate the patch, are routed to a standard connector positioned on the left side. Figure 2(a) reports a picture of the realized PCB highlighting its high flexibility. The solution, after soldering of all the components, maintains a high flexibility that allows the sensor patch to be conformable to a surface with minimum curvature radius of about 3 cm, which is sufficient for covering robot surfaces such as arms, legs, and torso (Figure 2(b)).

For applications where large surfaces have to be covered with a high number of taxels, the distributed tactile sensor proposed in this paper presents very attractive properties also from the power consumption point of view. Each taxel requires a voltage supply of 3.3 V with a current of about 1 mA, for an instantaneous power consumption of 3.3 mW. Since no additional ICs are necessary, with just a few watts

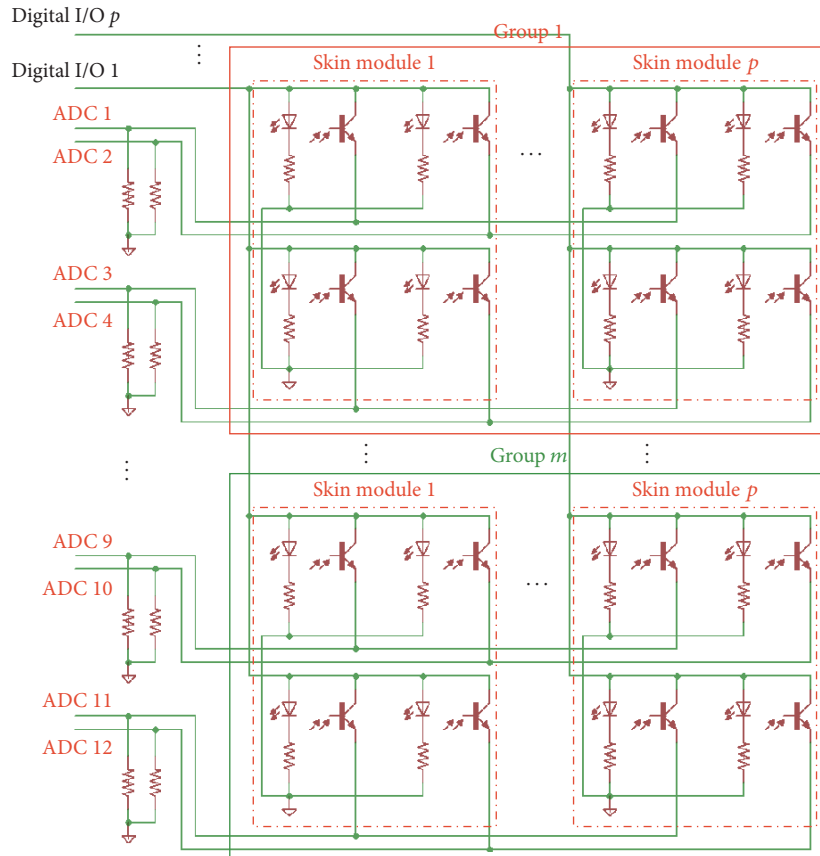


FIGURE 1: General electronic circuit of a sensor patch.

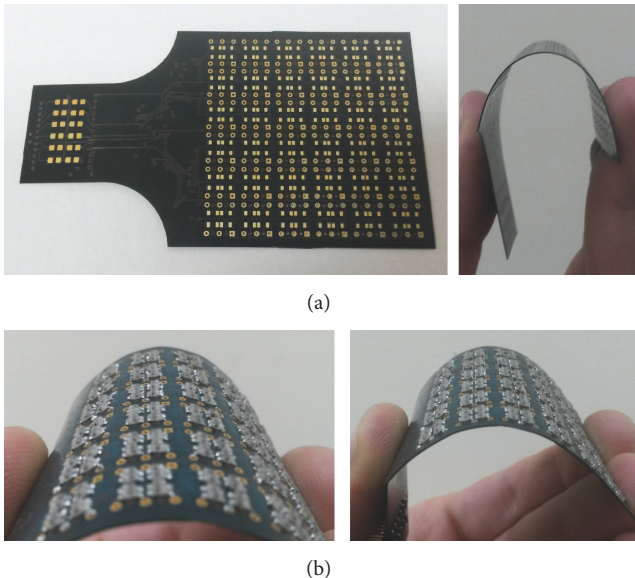


FIGURE 2: Pictures of the realized flexible PCB before (a) and after (b) soldering of optoelectronic components.

of power, thousands of taxels can be driven at the same time. In general, k taxels require power consumption equal to $k \cdot 3.3$ mW. For the sensor patch proposed in this paper,

constituted by 144 taxels, total instantaneous power consumption of 475,2 mW would be needed if all taxels were always switched on. In this case, the power consumption would already be quite limited, but the interrogation technique described above allows a further power saving. In particular, at each time instant, only one sensing module is switched on for each group, corresponding to $4m$ taxels. With the optimal number of groups $m = 3$, only 12 taxels are switched on at the same time, with a total instantaneous power consumption of 39.6 mW, resulting in a reduction of one order of magnitude compared to the previous case. The only limitation can be the minimum sampling frequency necessary to interrogate the whole distributed sensor. For all the 144 taxels of the proposed patch, with the selected MCU, that is, an ARM Cortex M4 STM32F303, a sampling frequency of 150 Hz was obtained. Therefore, the proposed solution is very attractive for battery-powered robotic systems.

2.3. Integration of the Conformable Sensor on a Robot Arm.

Since the sensor working principle depends on the deformations of the silicone layer, if the flexible PCB were conformed to the target shape after bonding of the deformable layer, a residual strain would affect the sensing modules causing the sensor malfunctioning. Hence, in order to ensure the correct operation of the sensor and to keep unaltered the sensor properties, that is, repeatability, hysteresis error, and accuracy, the flexible PCB has to be conformed to the surface

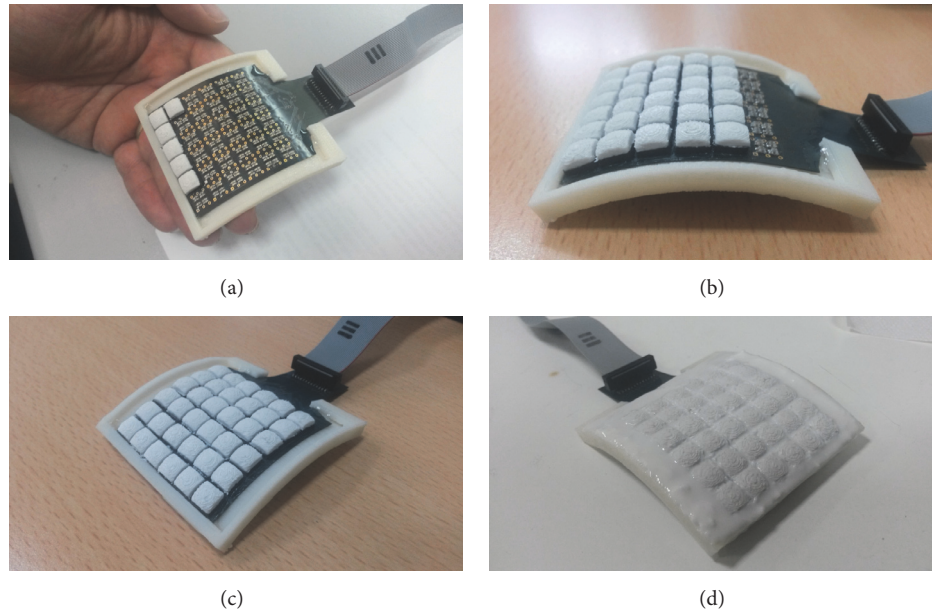


FIGURE 3: Conformable sensor patch during assembly phases: (a) perspective view, (b) side view, (c) perspective view of the complete sensor prototype, and (d) completed sensor prototype.

selected for the final assembly of the sensor patch before the bonding of the silicone layer. To this aim, a mechanical support, designed on the basis of the shape of the surface selected for the final mounting, has to be realized. Afterwards, the PCB is bonded to this support, by conforming it to the desired surface. Finally the silicone caps are bonded to the curved PCB. The sensor used in this paper has been realized to be mounted on a KUKA LWR4+. The details about the design and realization of the mechanical support can be found in [25].

Once the flexible PCB has been fixed to the mechanical support by epoxy resin (see Figure 3(a)), the silicone caps are bonded to each sensing module on the optoelectronic layer (see Figure 3(b)), by obtaining a fully assembled patch (see Figure 3(c)). In order to increase the mechanical robustness of the whole sensor, the silicone caps of all sensing modules have been connected together by using a second silicone molding (see Figure 3(d)). The final assembled sensor provided as raw data 144 voltage signals, corresponding to a 12×12 tactile map that can be directly used as pressure map. In addition, after a suitable calibration detailed in [25], the sensor is able to provide the estimation of the contact points and of the contact force vectors. For all cases, properties such as sensitivity, repeatability, hysteresis, and time response have been analyzed with the methodology reported by the authors in [24], by showing very similar results:

- (i) sensitivity: 0.2 N;
- (ii) repeatability error: 6.77%;
- (iii) hysteresis error: 10.27%;
- (iv) response time: 0.001 s.

3. The Tactile Sensor as HMI

This section shows how the tactile sensor can be actually used as an input device for sending commands to the robot, for example, commands for changing control modality or selecting a task to execute. Different recognition methods, for example, Finite State Machine, Artificial Neural Network [29, 30], Hidden Markov Model [31, 32], and features extrapolation [33], which differ in complexity and performance are described in the literature [34], but most of them are applied to inertial and camera systems.

In this work, three well-known recognition algorithms have been redesigned taking into account the sensor transduction principle and the sensor data collection. The first one is used to recognize gestures that are applied with static contact on the sensor surface, while the other two methods are used to recognize dynamic and more complex touch gestures. The sensor provides 288 bytes corresponding to the voltage signals of the 144 taxels. Starting from the idea behind the classic features extrapolation techniques, different features are computed with the sensor raw data according to the complexity of the gestures to recognize. Moreover, a suitable preprocessing stage and a classifier have been proposed considering the specific feature adopted for each recognition algorithm.

3.1. Static Gestures Recognition. The first method is used to simply show how the sensor information can be exploited to recognize tactile gestures using a simple algorithm. The sensor signals are organized in a 12×12 matrix corresponding to the sensor pressure map. The latter is used as recognition feature. Since only a small set of gestures has been considered, a simple algorithm such as the dot product-based recognition [35] is used to recognize static tactile gestures. This can be

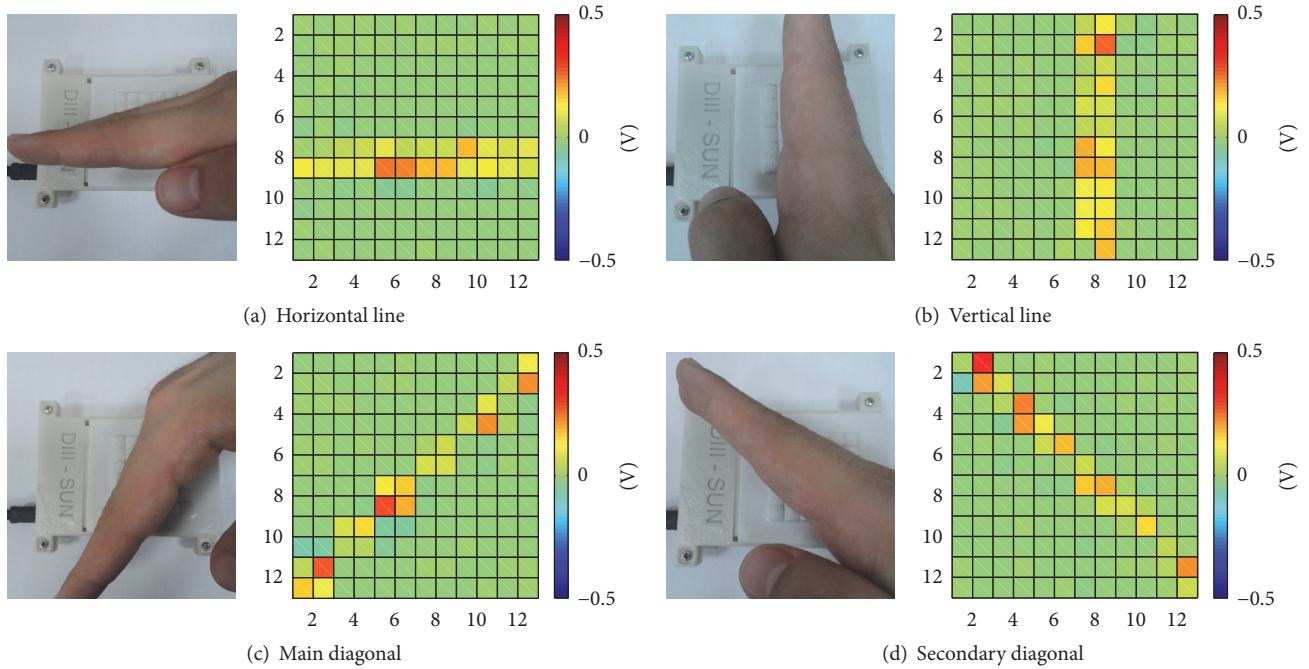


FIGURE 4: Applied gestures and corresponding tactile maps.

achieved by defining an elementary set of tactile gestures (codebook), that is, a set of modalities to touch the sensor patch by a human hand. Only 4 out of the 7 tactile gestures selected to test this kind of algorithm are reported in the left side of Figure 4.

A tactile map corresponds to each tactile gesture that can be represented with a 12×12 matrix constituted by the signals from all the taxels; thus recognition can be performed by resorting to algorithms typically used for image processing applications. In fact, for each time instant, a static representation of the tactile map, that is, an image of 12×12 pixels, can be obtained by properly preprocessing the acquired raw data. In a preelaboration stage an image of Boolean values (“0” and “1”) is obtained by thresholding the sensor voltage signals. Moreover, a bounding box that contains the detected gesture, depicted in the bit-map image as a group of “1” elements, is identified and it is translated in the upper-left corner of the bit-map image. The elaborated gesture can now be used in the recognition process. Let \mathbf{x}_i be the 144×1 vector that contains the 12 columns of the bit-map image corresponding to the i th gesture to be recognized and let \mathbf{y} be the 144×1 vector that contains the columns of the bit-map image corresponding to the acquired tactile image. If n is the total number of the selected gestures, the dot product is calculated as in (2), and the result provides a likelihood measure between the vectors \mathbf{x}_i and \mathbf{y}_j ; that is,

$$s_i = \sum_{j=1}^{144} \mathbf{x}_i \mathbf{y}_j, \quad i = 1, \dots, n. \quad (2)$$

The higher s_i , the closer (in the Hamming sense) the two vectors and the more alike the corresponding gestures. The dot product-based recognition is by far the fastest and easiest

gesture recognition method and it is able to recognize letters and digits. However, this method is not universal, it will often have a problem separating circles and squares, but this is the price for simplicity and speed. In Figure 4 four gestures and the corresponding tactile maps are shown. Gestures like vertical line, horizontal line, line along the main, and secondary diagonal are considered. It is evident how the raw data provide complete information about the contact that occurs on the deformable layer of the sensor.

3.2. Dynamic Gestures Recognition. Two different methods used to recognize dynamic gestures are presented. For the first one, the pressure map obtained reorganizing the 144 tactile sensor signals in a 12×12 matrix has been chosen as recognition feature, while the second one exploits the information about the force contact point in order to recognize more complex gestures. Figure 5 reports a scheme that highlights the training pipeline (right branch) and the recognition pipeline (left branch). The sensor starts acquiring the gesture applied by the user as soon as a contact on the deformable layer is detected. The gesture data are collected until the contact ends. In order to make the recognition process independent of the particular sensor contact area on which the gesture is applied, the data pass through a preelaboration/normalization stage. The preprocessed gesture is, then, compared to each gesture contained in a training set, which is preliminarily collected. The gesture selection is made on the basis of a maximum likelihood criterion. The preelaboration/normalization phase and the error index computation depend on the specific recognition feature used in each implemented method.

For the sake of completeness, the Nearest-Neighbor Algorithm [36] (NNA) used in the recognition methods is

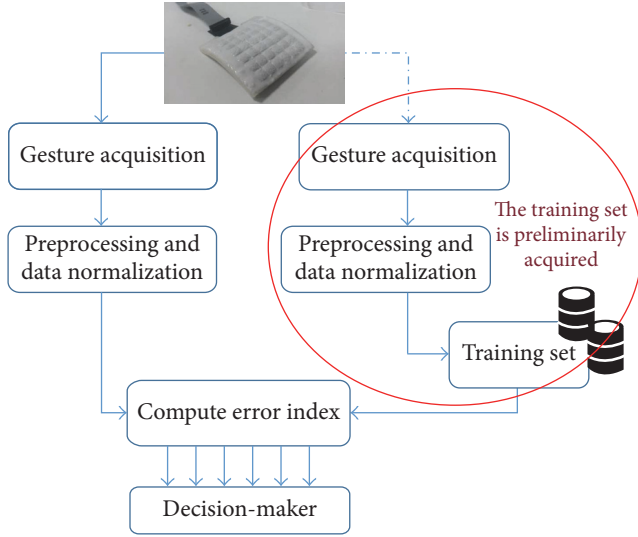


FIGURE 5: Schematization of training and recognition pipelines for haptic gesture recognition.

briefly recalled here. Let us consider a generic interpolation algorithm in the following linear form:

$$f(\mathbf{x}) = \sum_{\mathbf{z} \in \mathbb{Z}^q} f_k \phi(\mathbf{x} - \mathbf{k}), \quad \mathbf{x} = (x_1, x_2, \dots, x_q) \in \mathbb{R}^q, \quad (3)$$

where an interpolated value $f(\mathbf{x})$ at some coordinate \mathbf{x} in a space of dimension q is expressed as a linear combination of the samples f_k evaluated at integer coordinates $\mathbf{k} = (k_1, k_2, \dots, k_q) \in \mathbb{Z}^q$, the value of the function $\phi(\mathbf{x} - \mathbf{k})$ being the interpolation weight. Typical values of the space dimension correspond to bidimensional images (2D), with $q = 2$, and tridimensional volumes (3D), with $q = 3$. In the specific case, when all coordinates of $\mathbf{x} = \mathbf{k}_0$ are integer, the following formulation can be considered:

$$f_{\mathbf{k}_0} = \sum_{\mathbf{z} \in \mathbb{Z}^q} f_k \phi(\mathbf{k}_0 - \mathbf{k}), \quad \mathbf{k}_0 \in \mathbb{Z}^q, \quad (4)$$

which represents a discrete convolution. On the basis of the specific synthesis function ϕ used in the interpolation process, several interpolation algorithms that differ in complexity and accuracy can be identified [37]. The Nearest-Neighbor Algorithm is the simplest interpolation technique from a computational point of view used in image processing for image scaling. The synthesis function associated with it is the simplest, since it is made of a square pulse. For simplicity its expression for a space of dimension $d = 1$ is reported:

$$\phi(x) = \begin{cases} 1, & \text{if } 0 \leq |x| < 0.5 \\ 0, & \text{if } 0.5 \leq |x|. \end{cases} \quad (5)$$

The main interest of this synthesis function is its simplicity, which results in the most efficient of all implementation. In fact, for any coordinate \mathbf{x} where it is desired to compute the value of the interpolated function f , there is only one sample f_k that contributes, no matter how many dimensions q are

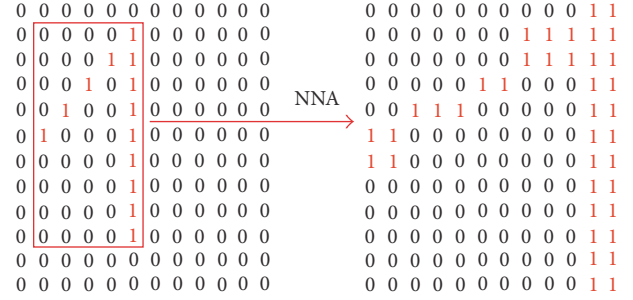


FIGURE 6: Prelaboration/normalization phase adopted for the first recognition method.

involved. The price to pay is a severe loss of quality. The algorithm performs image magnification by pixel replication and image reduction by sparse point sampling, and it derives its primary use as a tool for real-time magnification.

3.2.1. Map-Based Recognition Algorithm. The first recognition algorithm uses the sensor tactile map, suitably adapted and elaborated, as recognition feature. As described in Section 3.1, each tactile gesture corresponds a tactile map that can be represented with a 12×12 matrix constituted by the signals from all the taxels. For each time instant, a static representation of the tactile map, that is, an image of 12×12 pixels, can be obtained by properly processing the acquired raw data and, in a preliminary stage, a binary image is obtained by thresholding the sensor signals. During the gesture acquisition, maps obtained in each time instant are element-wise added (in the binary sense). At the end, a representation, in terms of an image of 12×12 pixels, of the route traced by the user finger on the contact surface of the sensor is available. Since the gesture could be generally traced anywhere on the available sensor contact area, a prelaboration/normalization phase is necessary so that the recognition algorithm can properly work independently of that area. Starting from the map provided at the end of the acquisition phase, a bounding box that contains the detected gesture (see Figure 6), depicted in the bit-map image as a group of “1” elements, is identified. The reduced image, which represents the detected gesture, is rescaled in order to obtain a new image of 12×12 pixels by applying the NNA. The elaborated gesture can now be used in the recognition process. The decision is made by evaluating $n \times m$ error indexes obtained by comparing the elaborated gesture to the n gestures in the codebook, which have been preliminarily acquired for m times, collected in the training set, and then the gesture corresponding to the lowest error index is selected as the recognized gesture. The error indexes are computed according to the Hamming distance (the Hamming distance between two matrices of equal size is the number of positions at which the corresponding elements are different) between the bit-map matrices. The described algorithm is summarized with the pseudocode of Algorithm 1.

3.2.2. Centroid-Based Recognition Algorithm. The method of the previous section is able to recognize in an efficient way

```

Require: 144 tactile sensor signals
Ensure: Recognized gesture
(1) initialization
(2) while TRUE do
(3) TactileMap = extractTactileMap (sensorSignals)
(4) TactileMap+ = TactileMap (element-wise sum)
(5) if sensorNotTouched then
(6)   ClippedTactileMap = getBoundingBox (TactileMap)
(7)   scaledTactileMap = NNA (ClippedTactileMap)
(8)   for each ith gesture in the codebook do
(9)      $H_i = \text{compare}(\text{scaledTactileMap}, \text{codebook}_i)$  (in terms of Hamming distance)
(10)  end for
(11)  makeDecision ( $H_1, \dots, H_n$ )
(12)  clear (TactileMap)
(13) end if
(14) end while

```

ALGORITHM 1: Pseudocode of the map-based recognition algorithm.

different touch gestures, for example, numbers, chars, and geometric primitives. However, the use of a bit-map image as recognition feature does not allow discriminating the direction with which the gesture is made, for example, a line from left to right and vice versa. This second method intends to overcome this disadvantage exploiting the force contact point provided by the sensor, which brings information concerning both the area on which the touch gesture is applied and the direction with which it is traced. By properly processing the sensor raw data, it is possible to estimate the spatial coordinate of the force contact point with respect to a reference frame fixed on the tactile sensor (refer to [24] for more details). Let us define $\mathbf{g}_x(t)$ and $\mathbf{g}_y(t)$ as the vectors that contain the x and y components of the contact point, respectively, whose size depends on the time needed by the user to trace the gesture on the sensor surface. Both $\mathbf{g}_x(t)$, $\mathbf{g}_y(t)$, and the time axis are acquired as soon as a contact is detected on the sensor surface. The couple $(\mathbf{g}_x(t), \mathbf{g}_y(t))$ represents the gesture feature. To simplify the notation, the dependence from the time t has been neglected in the following discussion. The normalization stage foresees two successive feature elaborations. First, the vectors \mathbf{g}_x and \mathbf{g}_y are resampled exploiting the NNA in order to produce a time-independent gesture feature $(\mathbf{g}_x^t, \mathbf{g}_y^t)$. The latter is, then, normalized to obtain a gesture feature independent of the area of the sensor on which the gesture is traced; that is,

$$\bar{\mathbf{g}}_i = \frac{\mathbf{g}_i^t - \min \mathbf{g}_i^t}{\max \mathbf{g}_i^t - \min \mathbf{g}_i^t} \quad \text{with } i = x, y. \quad (6)$$

As in the map-based method, the decision is made by evaluating $n \times m$ error indexes obtained by comparing the elaborated gesture to the n gestures in the codebook, which are preliminarily acquired for m times, collected in the training set, and choosing the gesture corresponding to the lower error index. In this case, the error indexes are calculated as the Euclidean distance between the feature of the acquired

gesture $(\bar{\mathbf{g}}_x, \bar{\mathbf{g}}_y)$ and the $n \times m$ features contained in the codebook $(\hat{\mathbf{g}}_x^j, \hat{\mathbf{g}}_y^j)$ with $j = 1, \dots, n \times m$; that is,

$$E_{d_j} = \frac{1}{2} \left(\sqrt{\sum_{k=1}^l (\bar{\mathbf{g}}_x(k) - \hat{\mathbf{g}}_x^j(k))^2} + \sqrt{\sum_{k=1}^l (\bar{\mathbf{g}}_y(k) - \hat{\mathbf{g}}_y^j(k))^2} \right), \quad (7)$$

where $j = 1, \dots, n \times m$ and l is the number of elements of the vectors $\bar{\mathbf{g}}_x$ and $\bar{\mathbf{g}}_y$ chosen in the resampling phase.

The described algorithm is summarized with the pseudocode of Algorithm 2.

3.3. Assessment of the Recognition Methods. In order to assess the recognition algorithms presented so far, a set of 30 trials for each gesture have been performed by 20 performers and the performance is assessed in terms of the recognition rate, namely, ratio between the number of correctly recognized gestures and the total number of trials.

Table 1 reports a confusion matrix for the dot product-based algorithm used to recognize only the 4 static touch gestures in 4. The table reports the actual gesture on the columns and the recognized gestures on the row. In particular, a recognition rate higher than 80% has been obtained for each gesture. The algorithm is certainly very simple, but it is able to recognize only gestures applied with static contacts on the sensor surface. This characteristic represents a critical disadvantage that limits the touch gestures applicable with a human hand practically to the ones showed in Figure 4.

Table 2 and Figure 7 report the results for the map-based algorithm used in the recognition of the dynamic touch gestures in terms of recognition rate and the set of analyzed gestures, respectively. Let us define the 5 analyzed gestures as a for the diagonal, b for the secondary diagonal, c for the horizontal line, d for the number 1, and e for the number 2.


```

Require: 144 tactile sensor signals
Ensure: Recognized gesture
(1) initialization
(2) while TRUE do
(3) cp = extractContactPoint (sensorSignals) (as described in [24])
(4) if sensorNotTouched then
(5)   resampledCP = NNA (cp)
(6)   normalizedCP = normalize (resampledCP) (as defined in Eq. (6))
(7)   for each ith gesture in the codebook do
(8)      $E_i$  = compare (normalizedCP, codebooki) (as defined in Eq. (7))
(9)   end for
(10)  makeDecision ( $E_1, \dots, E_n$ )
(11)  clear (cp)
(12) end if
(13) end while

```

ALGORITHM 2: Pseudocode of the centroid-based recognition algorithm.

TABLE 1: Confusion matrix of the dot product-based gesture recognition algorithm.

Recognized/traced	Horizontal line	Vertical line	Main diagonal	Secondary diagonal
Horizontal line	86.6%	0.0%	6.7%	6.1%
Vertical line	0.0%	83.3%	8.3%	7.3%
Main diagonal	6.7%	8.3%	85.0%	0.0%
Secondary diagonal	6.7%	8.4%	0.0%	86.6%

TABLE 2: Confusion matrix of the map-based algorithm.

Recognized/traced	a	b	c	d	e
a	95.3%	3.3%	0.0%	2.3%	2.3%
b	3.3%	93.3%	2.7%	6.7%	9.7%
c	0.0%	2.3%	95.3%	5.7%	2.7%
d	1.4%	1.1%	0.7%	85.3%	2.3%
e	0.0%	0.0%	1.3%	0.0%	83.0%

As said previously, the decision-making is not influenced by the direction with which the gesture is traced on the sensor surface, so a diagonal traced from the upper-left corner to the bottom-right corner of the sensor is equivalent to a diagonal traced from the right-bottom corner to the upper-left corner and both are recognized as gesture a. In fact, the performers have executed the tests tracing the gestures into the two directions as shown in Figure 7 by the red arrows but the classifier did not distinguish the direction. Nevertheless, the accuracy can be considered satisfactory with success rates above 80%.

The same analysis has been carried out for the centroid-based algorithm. Table 3 and Figure 8 report the confusion matrix and the analyzed gestures. The codebook is now defined as follows. Let a be the diagonal traced from the upper-left corner to the bottom-right corner, b the diagonal traced from the bottom-right corner to the upper-left corner, c the horizontal line traced from left to right, d the horizontal line traced from right to left, and e the number 1 traced from left to right. It is evident, from the confusion matrix reported in Table 3, that the algorithm is able to correctly discriminate the direction of the gesture. In order to evaluate the

dependency of the algorithm on the specific performer, the standard deviation of the recognition rate has been computed considering the results obtained with the 20 performers for each gesture according to

$$\sigma_h = \left(\frac{1}{N} \sum_{i=1}^N (R_{i,h} - \mu_h)^2 \right)^{1/2}, \quad h = a, b, c, d, e, \quad (8)$$

where N is the number of performers, $R_{i,h}$ is the average recognition rate achieved by the i th performer for the h th gesture, and μ_h is the average recognition rate of h th gesture achieved by all performers computed as

$$\mu_h = \frac{1}{N} \sum_{i=1}^N R_{i,h}, \quad h = a, b, c, d, e. \quad (9)$$

The results are reported in Figure 9. The centroid-based algorithm shows a higher recognition rate for both simple and complex gestures, that is, diagonals and numbers, and it is proven to be more independent of the codebook preliminarily acquired. Moreover, given that the discrete nature of the features involved in the recognition process, that is,

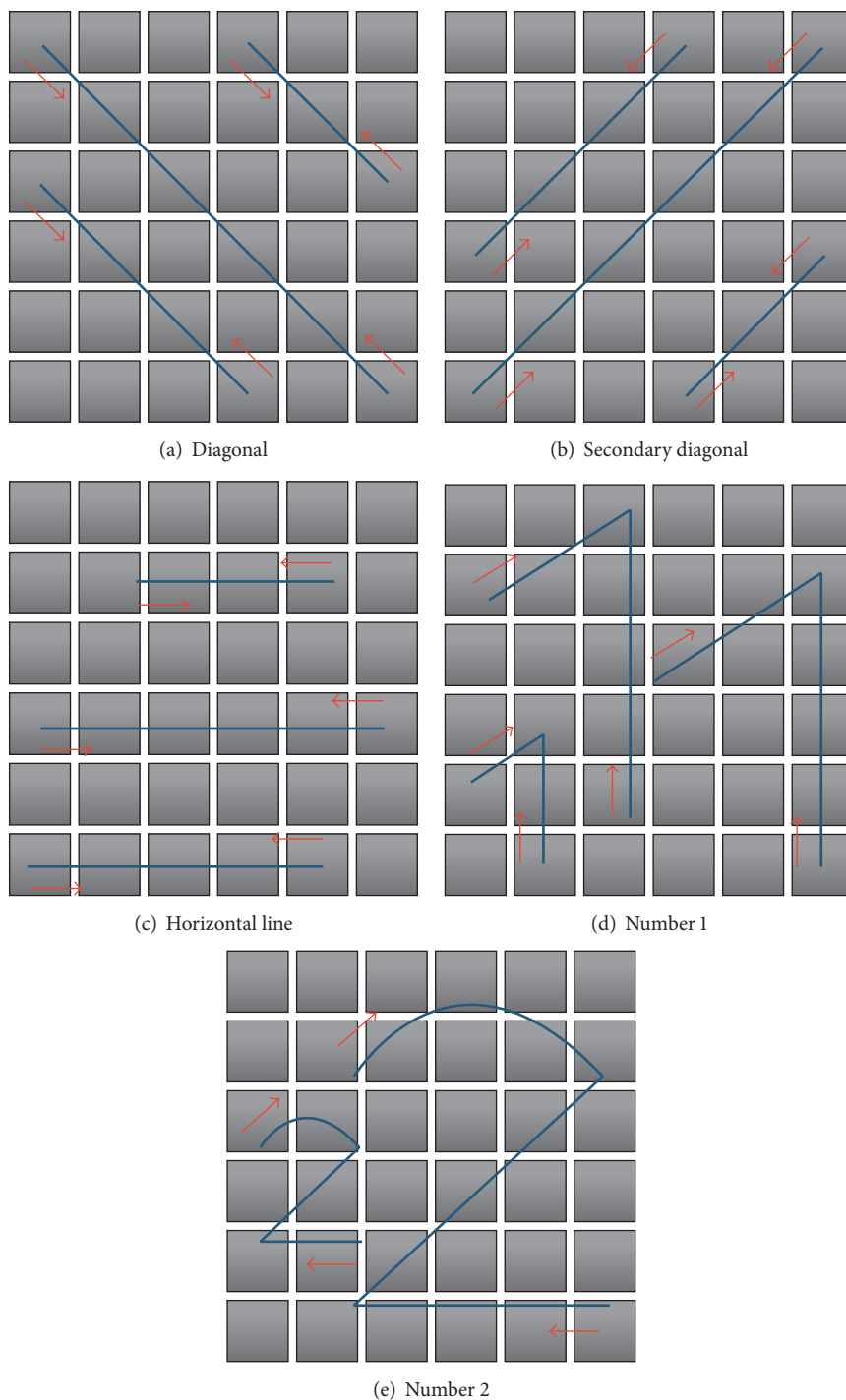


FIGURE 7: Gestures analyzed for the map-based recognition algorithm.

TABLE 3: Confusion matrix of the centroid-based algorithm.

Recognized/traced	a	b	c	d	e
a	100.0%	0.0%	11.3%	0.0%	2.3%
b	0.0%	100.0%	0.0%	11.3%	0.3%
c	0.0%	0.0%	86.0%	0.0%	2.7%
d	0.0%	0.0%	0.0%	88.7%	0.0%
e	0.0%	0.0%	2.7%	0.0%	94.7%

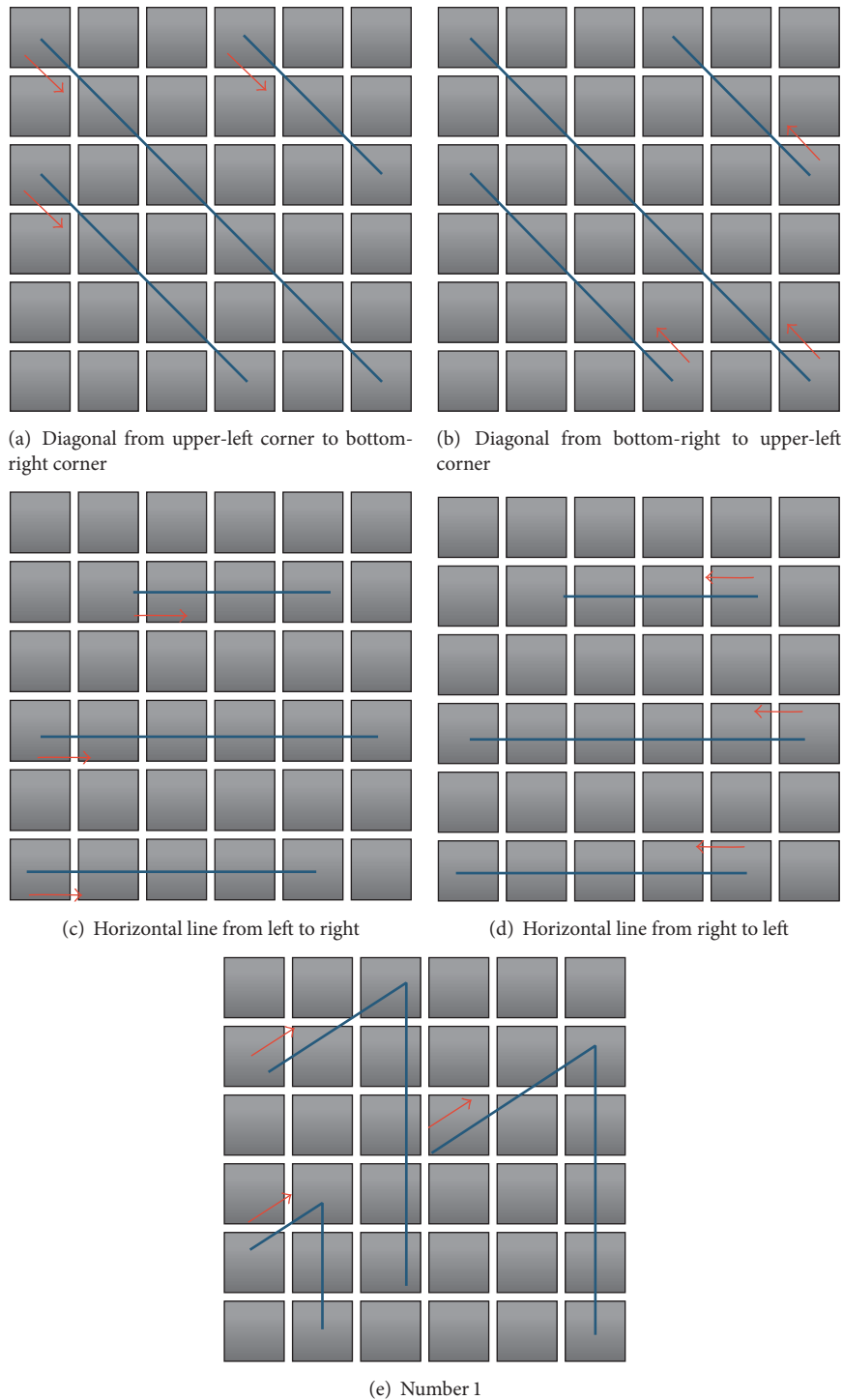


FIGURE 8: Gestures analyzed for the centroid-based recognition algorithm.

the bit-map image, and the coordinates of the contact point depends on the spatial resolution of the sensor, gestures such as horizontal lines, in some cases, are badly recognized for the difficulty to trace a really straight line. Finally, the low values of the standard deviations compared to the high value of the average recognition rates demonstrate that almost all algorithms are fairly independent of the performers.

The centroid-based method is totally independent of the performer for the diagonal gestures that are recognized with a 100% recognition rate. This feature is quite important since it allows the algorithms to be used effectively without any special training of the user.

The CPU time needed to execute the two algorithms is very low due to their simplicity and efficiency. It has been

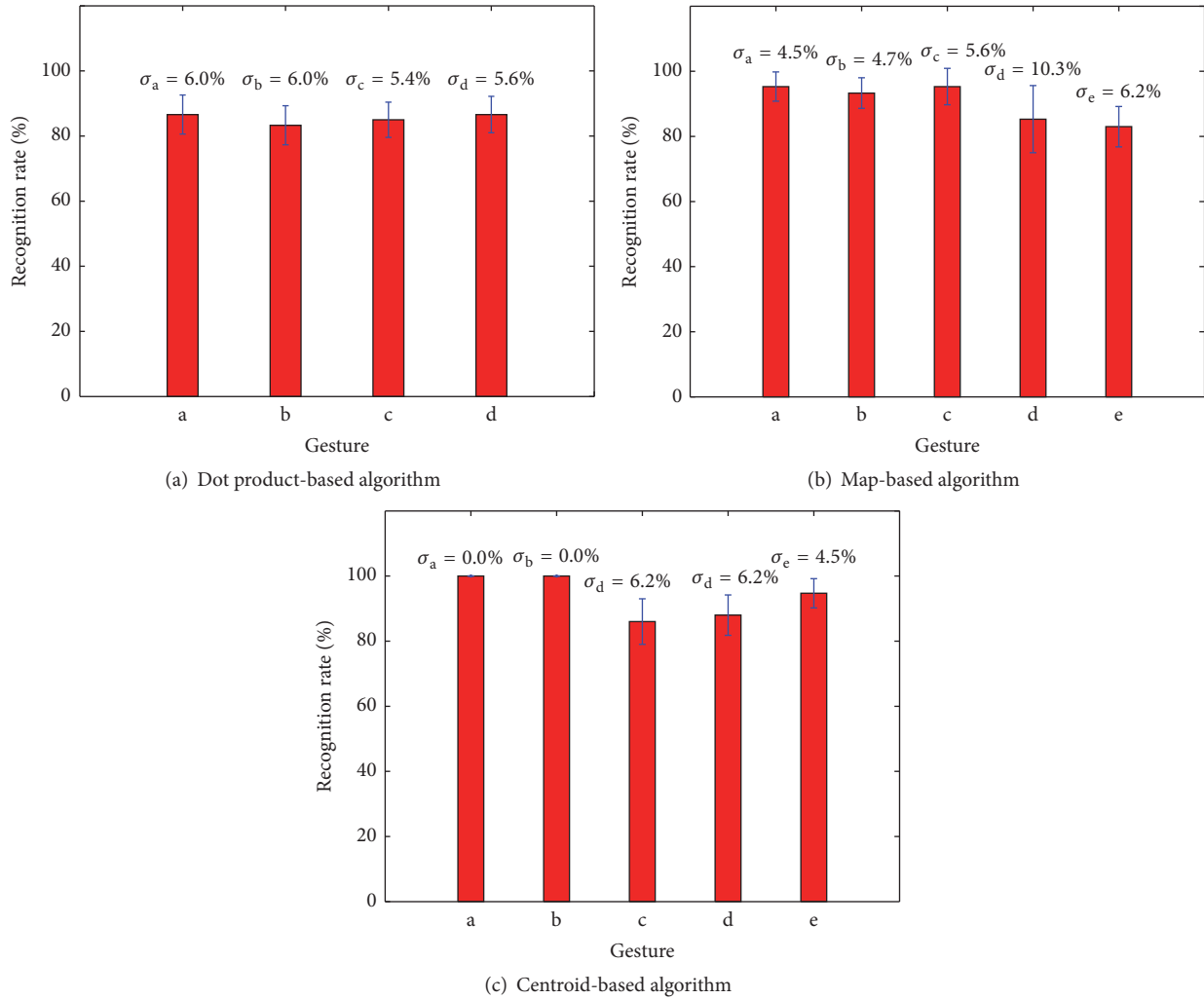


FIGURE 9: Assessment of algorithm sensitivity to the performer: average and standard deviation of gesture recognition rates for the various performers (see (8) and (9)).

evaluated implementing the two algorithms in C++ under a Linux system with an i7 processor @2.8 GHz and it is about 0.05 ms and 0.08 ms for the map-based algorithm and for the centroid-based algorithm, respectively.

4. Conclusions

This paper has presented a conformable distributed tactile sensor able to measure 3D force vectors in multiple contact points and to provide a distributed tactile map. The authors show how the sensor data can be used to recognize haptic gestures using readapted simple recognition techniques. The recognition algorithms discussed in this work result to be fairly independent of the performers and this feature represents a good starting point for further experiments in which the tactile sensor could be tested on a real robot and in complex interactive tasks without any special training of the user. Future work will be devoted to investigate more complex gestures recognition methods that could be also evaluated to handle larger tactile gesture sets, that, for example, could

include the symbols of the tactile languages used for blind-deaf people. The possibility of recognizing the gesture in real time and during the accomplishment of the gesture itself will be also investigated.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partly supported by the European Commission's Seventh Framework Programme (FP7/2007–2013) under Grant Agreement no. 287513 (SAPHARI project).

References

- [1] V. K. Thakur, "Robust hand gesture recognition for human machine interaction system," *Journal of Global Research in Computer Science*, vol. 5, pp. 14–19, 2014.

- [2] R. Azad, B. Azad, N. B. Khalifa, and S. Jamali, "Real-time human-computer interaction based on face and hand gesture recognition," *International Journal in Foundations of Computer Science & Technology*, vol. 4, pp. 37–48, 2014.
- [3] D. Xu, X. Wu, Y. L. Chen, and Y. Xu, "Online dynamic gesture recognition for human robot interaction," *The Journal of Intelligent & Robotic Systems*, vol. 77, pp. 583–596, 2015.
- [4] T. Fujii, J. Lee, and S. Okamoto, "Gesture recognition system for human-robot interaction and its application to robotic service task," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2014.
- [5] Intel RealSense technology, <http://www.intel.eu/content/www/eu/en/architecture-and-technology/realsense-overview.html>.
- [6] R. Stiefelhagen, H. K. Ekenel, C. Fügen et al., "Enabling multimodal human-robot interaction for the Karlsruhe humanoid robot," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 840–851, 2007.
- [7] V. Pitsikalis, A. Katsamanis, S. Theodorakis, and P. Maragos, "Multimodal gesture recognition via multiple hypotheses rescoring," *The Journal of Machine Learning Research (JMLR)*, vol. 16, no. 5, pp. 255–284, 2015.
- [8] L. Lucignano, F. Cutugno, S. Rossi, and A. Finzi, "A dialogue system for multimodal human-robot interaction," in *Proceedings of the 15th ACM International Conference on Multimodal Interaction*, (ICMI '13), pp. 197–204, NY, USA, December 2013.
- [9] R. Stiefelhagen, C. Fügen, P. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel, "Natural human-robot interaction using speech, head pose and gestures," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2422–2427, October 2004.
- [10] L. Muscari, L. Seminara, F. Mastrogiovanni, M. Valle, M. Capurro, and G. Cannata, "Real-time reconstruction of contact shapes for large area robot skin," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '13)*, pp. 2360–2366, German, May 2013.
- [11] E. Baglini, S. Youssefi, F. Mastrogiovanni, and G. Cannata, "A real-time distributed architecture for large-scale tactile sensing," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '14)*, pp. 1663–1669, USA, September 2014.
- [12] H. Iwata and S. Sugano, "Human-robot-contact-state identification based on tactile recognition," *IEEE Transactions on Industrial Electronics*, vol. 52, no. 6, pp. 1468–1477, 2005.
- [13] Z. Ji, F. Amirabdollahian, D. Polani, and K. Dautenhahn, "Histogram based classification of tactile patterns on periodically distributed skin sensors for a humanoid robot," in *Proceedings of the 20th IEEE International Symposium on Robot and Human Interactive Communication, (RO-MAN '11)*, pp. 433–440, USA, August 2011.
- [14] S. Parusel, H. Widmoser, S. Golz et al., http://www.aaaivideos.org/2014/15_hri_planning/.
- [15] R. S. Dahiya, P. Mittendorfer, M. Valle, G. Cheng, and V. J. Lumelsky, "Directions toward effective utilization of tactile skin: a review," *IEEE Sensors Journal*, vol. 13, no. 11, pp. 4121–4138, 2013.
- [16] G. Cannata, M. Maggiali, G. Metta, and G. Sandini, "An embedded artificial skin for humanoid robots," in *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI*, pp. 434–438, kor, August 2008.
- [17] V. Duchaine, N. Lauzier, M. Baril, M. Lacasse, and C. Gosselin, "A flexible robot skin for safe physical human robot interaction," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '09)*, pp. 3676–3681, Kobe, Japan, May 2009.
- [18] X. Lamy, F. Colledani, F. Geffard, Y. Measson, and G. Morel, "Robotic skin structure and performances for industrial robot comanipulation," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '09)*, pp. 427–432, Singapore, July 2009.
- [19] J. Ulmen and M. Cutkosky, "A robust, low-cost and low-noise artificial skin for human-friendly robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '10)*, pp. 4836–4841, USA, May 2010.
- [20] N. Elkmann, M. Fritzsche, and E. Schulenburg, "Tactile sensing for safe physical human-robot interaction," in *Proceedings of the International Conference on Advances in Computer-Human Interactions*, pp. 212–217, Le Gosier, March 2011.
- [21] P. Mittendorfer and G. Cheng, "Humanoid multimodal tactile-sensing modules," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 401–410, 2011.
- [22] P. Saccomandi, E. Schena, C. M. Oddo, L. Zollo, S. Silvestri, and E. Guglielmelli, "Microfabricated tactile sensors for biomedical applications: a review," *Biosensors*, vol. 4, no. 4, pp. 422–448, 2014.
- [23] M. Kaboli, A. Long, and G. Cheng, "Humanoids learn touch modalities identification via multi-modal robotic skin and robust tactile descriptors," *Advanced Robotics*, vol. 29, no. 21, pp. 1411–1425, 2015.
- [24] A. Cirillo, P. Cirillo, G. De Maria, C. Natale, and S. Pirozzi, "An artificial skin based on optoelectronic technology," *Sensors & Actuators: A Physical*, vol. 212, pp. 110–122, 2014.
- [25] A. Cirillo, F. Ficuciello, C. Natale, S. Pirozzi, and L. Villani, "A conformable force/tactile skin for physical human-robot interaction," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 41–48, 2016.
- [26] A. Cirillo, P. Cirillo, G. De Maria, C. Natale, and S. Pirozzi, "Force/Tactile sensors based on optoelectronic technology for manipulation and physical human-robot interaction," pp. 95–131, Springer International Publishing, Cham, Germany, 2017.
- [27] G. De Maria, C. Natale, and S. Pirozzi, "Force/tactile sensor for robotic applications," *Sensors and Actuators A: Physical*, vol. 175, pp. 60–72, 2012.
- [28] Y. Ohmura, Y. Kuniyoshi, and A. Nagakubo, "Conformable and scalable tactile sensor skin for a curved surfaces," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '06)*, pp. 1348–1353, USA, May 2006.
- [29] C. Zhu and W. Sheng, "Online hand gesture recognition using neural network based segmentation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '09)*, pp. 2415–2420, USA, October 2009.
- [30] P. Neto, D. Pereira, J. N. Pires, and A. P. Moreira, "Real-time and continuous hand gesture spotting: an approach based on artificial neural networks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '13)*, pp. 178–183, Germany, May 2013.
- [31] N. Nguyen-Duc-Thanh, S. Lee, and D. Kim, "Two-stage hidden markov model in gesture recognition for human robot interaction," *International Journal of Advanced Robotic Systems*, vol. 9, 2012.
- [32] R. Amstutz, O. Amft, B. French, A. Smailagic, D. Siewiorek, and G. Troster, "Performance analysis of an HMM-based gesture recognition using a wristwatch device," in *Proceedings*

of the *International Conference on Computational Science and Engineering (CSE '09)*, vol. 2, pp. 303–309, IEEE, Vancouver, Canada, August 2009.

- [33] M. Meenaakumari and M. Muthulakshmi, “Mems accelerometer based hand gesture recognition,” *International Journal of Advanced Research in Computer Engineering & Technology*, vol. 2, no. 5, pp. 1886–1892, 2013.
- [34] S. Mitra and T. Acharya, “Gesture recognition: a survey,” *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 37, no. 3, pp. 311–324, 2007.
- [35] S. Bow, *Pattern Recognition and Image Preprocessing*, CRC press, 2002.
- [36] G. Wolberg, *Digital Image Warping*, IEEE Computer Society Press, 1994.
- [37] P. Thévenaz, T. Blu, and M. Unser, *Ch. Image Interpolation and Resampling*, Academic Press, Orlando, FL, USA, 2000.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

