Cross-Talk Compensation in Low-Cost Resistive Pressure Matrix Sensors

Steffen Müller, Daniel Seichter, Horst-Michael Gross
Neuroinformatics and Cognitive Robotics Lab
Technische Universität Ilmenau
Ilmenau, Germany
fg-nikr@tu-ilmenau.de

***Abstract*—For socially assistive robots in close contact to people, a tactile sensor can be useful for gathering feedback and inputs in the form of touch gestures. In this paper, we concentrate on low-cost textile pressure matrix sensors since they are easy to manufacture and due to their flexibility can be adopted to the curved shape of a robot's outer cover. Due to the matrix principle for reading out, the setup suffers from artifacts when it comes to activation of multiple sensor elements. We present a machine learning approach for preprocessing the raw measurements from the pressure sensitive array in order to get reliable pressure patterns which can be used for gesture classification later on. By means of that, an expensive hardware solution for capturing the pressure values can be avoided.**

***Index Terms*—textile pressure sensor, machine learning, matrix of resistive sensors**

I. Introduction

Tactile interaction is a useful medium for socially assistive robots. In our research projects SERROGA [1] and SYMPARTNER [2], we developed service robots for elderly people living alone. Fig. 1 shows these robots with their pressure sensitive matrix sensors used for recognizing simple haptic gestures, like fondling, petting, tickling, stroking, or slaping. Such tactile inputs can be used as intuitive feedback for adaptation and personalization of the robot's interaction behavior, or as direct commands, e.g. for stopping the autonomous navigation when the user touches and pushes the robot. Details on these two applications and the classification of the gestures are given in [3] and [4]. The recognition of social touch gestures on artificial creatures is widely used [5], [6], while the practical realization is often straight forward using low-cost hardware like sensors made from conductive fabric and piezoresitive materials. There is a broad community dealing with classification and recognition of such social touch gestures with own contests, like the Touch Challenge [7].

In [8] the force signals from the robot's surface are not only used for stopping the navigation but are also interpreted directly as pushing forces used to control a motor-assisted manual positioning. Although, the mentioned approach uses capacitive touch signals, application of force or pressure sensing would practically improve that behavior, if the measurements are reliable and free of artifacts. Therefore, we have developed a preprocessing of the raw sensor readings to enable a resistive matrix sensor to be used for reliable multi-touch. This improves applications that need accurate force patterns from low-cost textile resistive sensors.

In contrast to textile low-cost sensors, there are very sophisticated approaches for realizing sensitive robotic skin [9], but these are not in the focus of this paper. A survey on artificial skin and tactile sensing for socially interactive robots can be found in [10].

This paper instead is focused on textile array sensors which come along with a specific problem when it comes to multiple activation points. Due to parallel pathways in the rows and columns of a matrix of pressure sensitive resistors, the activation of one sensor cell also influences other cells of the same column and row leading to artifacts, especially in multi-touch situations.

The main contribution of this paper is a tailored machine learning-based approach, which is used to solve the otherwise not efficiently solvable mathematical model for computing the pressure values (resp. resistances) in a matrix of variable resistors given the measurements of currents into the rows and columns. This allows for the application of a very basic circuit for reading out the resistance values compared to other hardware solutions.

Fig. 1. Robots equipped with textile pressure sensor arrays for recognition of social touch gestures. left: Robot Max presented in [1] with touch sensitive patch of fur at the head, right: Sympartner robot [2] with touch sensor at the back for recognizing tactile feedback from a sitting or standing interaction partner.

in: IEEE International Conference on Mechatronics (ICM), Ilmenau, Germany, pp. 232-237, IEEE 2019

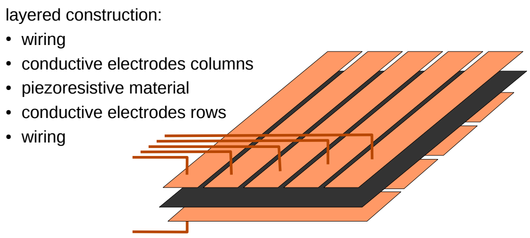


Fig. 2. Principle of layered pressure array sensors

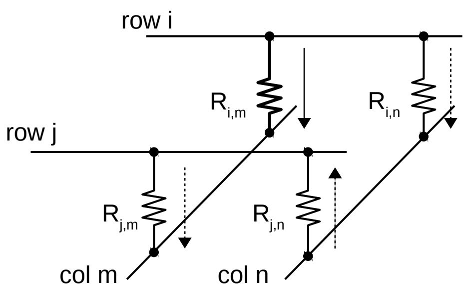


Fig. 3. Parallel resistances (dashed arrow pathway) when reading out resistance $R_{i,m}$. These cause cross-talk effects if not handled explicitely in hardware or software.

II. Related Work

The need for sensing the spatial distribution of pressure over a sensor surface has led to various hardware solutions. In this paper, resistive matrix sensors are addressed, since they are very easy and inexpensive to build. Besides other approaches, which use resistive textile material for interpolation of single contact points by means of the idea of a voltage divider [11], the most popular method for building touch sensitive fabric sensors is a layered architecture resembling a matrix of pressure sensitive cells. Fig. 2 shows the general structure of such sensors. The usage of a flexible textile material allows to overcome the shape restrictions of commercially available sensors and, thus, enables designers to cover non-planar and even flexible surfaces [12].

There are many examples for arrays of resistive sensors [13], each coming along with the fundamental problem of cross-talk when the matrix elements are to be read out, which typically is done sequentially one after another. When one column and one row are connected to a circuit that reads the resistance, the other sensor elements form parallel loops of resistors (see Fig. 3), preventing the direct measurement of an individual cell's resistance. Typically, the range of the taxels' (tactile elements') resistance is huge and non linear starting at $> 10\ k\Omega$ in uncompressed state and reaching values below 100 Ω under pressure. This makes it possible to ignore the cross-talk effects for certain applications, like gesture recognition [3] if most of the cells are in the uncompressed state. However, the problem is more critical, if the exact shape of wide spread activation patterns matters.

In order to reduce the activation of parallel pathways, there exist several approaches introducing a spacer layer in between the electrodes and the piezoresistive layer. Büscher et al. [14] used this for the realization of a data glove. Markham and Brewer [15] added a nylon grid with 3mm spacing to prevent multiple activations in their matrix. Unfortunately, such a spacer only increases the threshold for activation (making uncompressed resistors reach ∞), which significantly reduces the sensitivity and cannot prevent the cross-talk effect if large areas of the sensor are activated.

D'Alessio [16] discussed sources of cross-talk and measurement errors in piezoresistive sensor arrays in detail. He also mentioned lateral current in the piezoresistive layer introducing additional parallel pathways. In order to handle the cross-talk in hardware, there are mainly two approaches. One option involves more complicated circuits, which apply a feedback voltage to the remaining rows and columns when one certain pair is measured. This aims for reducing the current in the parallel pathways to zero such that the measured current only depends on the cell considered. Depending on the ratio of the resistances in the matrix, the feedback current needs to be comparatively high, which brings that approach to its limits when the array is large or the resistances of the taxels are low. The second popular solution is to switch all inactive rows and columns to ground [17]. This also has its limitations regarding the current that has to be provided by the respective drivers and also increases complexity of the sensing circuit.

In contrast to these hardware solutions for handling the parallel resistive pathways, we suggest to compensate them in a computational way. This is possible if there are measurements of the total resistances for all row and column combinations, which can be read out with a simple multiplexing solution. One precondition for our approach is that the pressure pattern is stationary for the duration of one read out cycle, which is almost true for touch gestures and a sampling rate of about 360 cycles per second.

III. Sensor Hardware

In this paper a stand-alone sensor matrix similar to the one mounted at our Sympartner robot (see Fig. 1) is used.

A. Pressure Sensitive Matrix

The matrix consists of 16×16 cells that are formed by 12mm wide stripes of silver coated EMF shielding material as row and column electrodes. The electrodes are placed on a carrier fabric at a spacing of 6mm. These relatively large gaps are intended to reduce the lateral current in the active layer. The piezoresistive material is Eeontex™ with a surface resistance of 20 $k\Omega/cm$.

B. Electronics

As already claimed, we did not use any complex feedback or grounding drivers. Fig. 4 shows the actual setup, which is realized with an ATMEGA328P micro controller and two analog 16 channel multiplexers (74HD4067). The resistive matrix forms a voltage divider with a pull down resistor R_{ref} while the voltage is read at an analog digital converter

in: IEEE International Conference on Mechatronics (ICM), Ilmenau, Germany, pp. 232-237, IEEE 2019

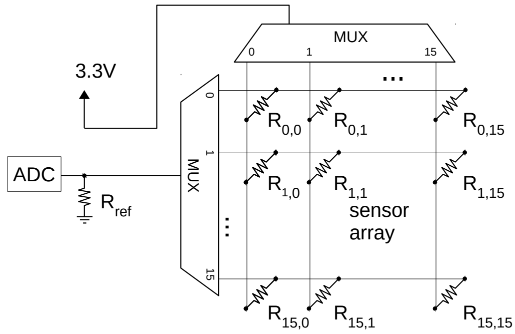


Fig. 4. Electronics for read-out of the resistor matrix consists of two multiplexers and an analog digital converter.

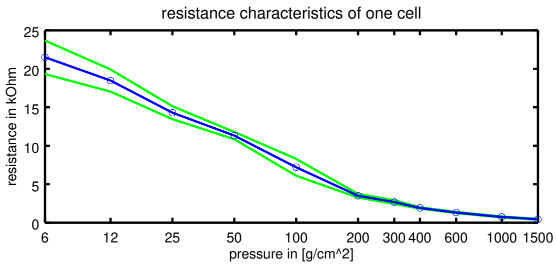


Fig. 5. Resistance of one individual cell depending on the pressure applied; The green curves show the standard deviation of ten measurements.

(ADC) input of the micro controller (10bit resolution). The readout cycle makes use of the maximum ADC speed that is about 100 kHz. Therefore, the resulting cycle rate for a complete read-out of the matrix is 390 Hz. This high sampling rate is used for recursive low pass filtering of the raw measurements over time in order to reduce sampling noise. Finally, the resulting matrices of ADC values are sent to the PC asynchronously via USB at a rate of up to 100 Hz. The actual processing of the systematic cross-talk errors and the conversion into pressure values is done by an external PC. (see Section V)

IV. Properties of the Textile Sensor Array

A. Characteristics

In order to compute the actual pressure from a voltage read at the ADC, the characteristics of one individual matrix element need to be known. For that reason, we analyzed the sensor setup and measured the resistance while a cell was loaded with a series of defined weights. The experiment was conducted 10 times. Fig. 5 shows the resulting curve of resistance over pressure in a logarithmic scale. In contrast to a version with a spacer material, the sensitivity of that construction is excellent in the low pressure regions that are typical for social interaction gestures.

In addition to the resistors inside the sensor array, there are resistances in the external measuring circuit as well. First, there is the reference resistor R_{ref} which has a known value of 1.78 $k\Omega$ in our case. Furthermore, the multiplexers generate an additional parasitic resistance in the measurement branch, which should be called R_{par} here. By connecting a known resistor instead of the sensor array and by measuring the resulting ADC voltage, the value of R_{par} could be determined indirectly. In our case it was 326 Ω.

Given these characteristics, it should be possible to reconstruct the weight/pressure distribution on the resistive matrix by connecting each of the cells once to the ADC and measuring the resulting voltage.

Unfortunately, there are several types of measurement errors related to the given setup. First, there is the high variance of the sensor readings, especially in the low pressure region, which may be induced by the loose stacking of the layers, moving around in between the pressure events. This noise is non-deterministic and can hardly be corrected in software. Furthermore, there are systematic sources of measurement errors mainly resulting in cross-talk effects between the cells. One source are lateral currents in the continuous piezoresistive layer. By activating an individual cell and measuring its neighbors, we found that this influence is relatively low. Only 5 thousandth of the voltage is coupled to the direct neighbors. Therefore, this effect can be neglected in the following. An inexpensive solution to prevent from such lateral currents is the subdivision of the active layers into individual sensor cells.

The more relevant effect is related to alternative pathways in the matrix (see Fig. 3). An activated cell $R_{j,n}$ causes the reduction of resistance in the parallel branches of other cells (dotted lines in Fig. 3) and therefore, the resistance values of measurements in the same row and column decrease too. In crossing points of such rows and columns this effect accumulates, leading to "ghost" activations. Most state-of-the-art approaches for readout hardware solutions address these cross-talk activations.

B. Analytical Forward Model

It is possible to compute the currents and, thus, the voltages at the ADC for a given matrix of resistors. For one combination of row and column that are connected to the input supply by the multiplexers, there are $16 \times 16 + 1$ unknown current values (through each of the resistors and the input current). On the other hand, we have $16 \times 16 + 1$ given parameters (resistors + supply voltage) in order to build up a linear equation system by means of Kirchhoff's circuit laws.

Fig. 6 shows an example with $n = 3$ rows and columns. The resulting linear equation system eq. (1) contains one line resulting from Kirchhoff's loop rule for the direct path through $R_{0,0}$ followed by $(n-1)^2$ lines for the indirect pathways through the resistors of the matrix except for the connected row 0 and column 0. R_{ext} represents the external resistors $R_{ref} + R_{par}$ in Fig. 6. The remaining five lines result from Kirchhoff's nodal rule for each of the column and row electrodes. There is one equation more than actually needed, thus, one of the nodal rules can be left out. Therefore, the resulting system contains $n^2 + 1$ equations. The linear equation system for the 16×16 array can be constructed in the same manner yielding 257 equations.

Solving this system provides the input current i_{in} for the matrix, and by means of the reference resistor R_{ref} the ADC

in: IEEE International Conference on Mechatronics (ICM), Ilmenau, Germany, pp. 232-237, IEEE 2019

voltage for one particular cell can be computed by $U_{adc} = i_{in} \cdot R_{ref}$. In order to get the full matrix of measured ADC values, this equation system has to be solved 16×16 times.

For correcting the measurement errors in a captured matrix of ADC voltages, the inverse of that operation is needed. For the known ADC voltages, the resistor values $R_{i,j}$ in the matrix and with these the actual pressure are the unknown variables. Unfortunately, it is not that simple to invert the system. One ends up with a heavily coupled non-linear equation system, that is not solvable in real-time anymore.

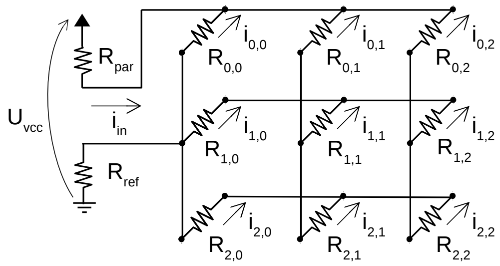


Fig. 6. Reduced example matrix with only 3×3 resistors. Row 0 and column 0 are selected in this case. R_{ref} is the reference resistor generating the ADC voltage and R_{par} models parasitic resistances in the multiplexers.

$$R = \begin{bmatrix} R_{ext} & R_{0,0} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ R_{ext} & 0 & R_{0,1} & 0 & R_{1,0} & -R_{1,1} & 0 & 0 & 0 & 0 \\ R_{ext} & 0 & R_{0,1} & 0 & 0 & 0 & -R_{1,2} & R_{2,0} & 0 & 0 \\ R_{ext} & 0 & 0 & R_{0,2} & 0 & 0 & R_{1,2} & 0 & -R_{2,1} & 0 \\ R_{ext} & 0 & 0 & 0 & 0 & 0 & R_{1,2} & 0 & R_{2,1} & -R_{2,2} \\ -1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ -1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$I = \begin{bmatrix} i_{in} \\ i_{0,0} \\ i_{0,1} \\ i_{0,2} \\ i_{1,0} \\ i_{1,1} \\ i_{1,2} \\ i_{2,0} \\ i_{2,1} \\ i_{2,2} \end{bmatrix} \quad U = \begin{bmatrix} U_{vcc} \\ U_{vcc} \\ U_{vcc} \\ U_{vcc} \\ U_{vcc} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad R \cdot I = U \tag{1}$$

V. Signal Processing and Cross-talk Compensation

The idea for correcting the cross-talk measurement errors in software makes use of the analytical model and the pressure resistance characteristics measured before. Since the inversion is not possible analytically, we use the general function approximation capabilities of neural networks for representing the inverse operation of the analytical forward model. The actual processing of the measured ADC voltage values consists of two essential steps. First step is the computation of the individual cells' resistance values from the given ADC voltages. During this step, the correction of the cross-talk effects needs to be considered. After that, the resistances can be used to get weight/pressure values using the characteristics of the cells measured in section IV.

Using a naïve approach ignoring the alternative pathways at all, one can directly computes $R_{x,y}$. For that, first, the current $i_{in} = U_{adc}/R_{ref}$ can be determined. Then the voltage over R_{par} results from $U_{par} = R_{par} \cdot i_{in}$, and $R_{x,y}$ arises from $R_{x,y} = (U_{vcc} - U_{adc} - U_{par})/i_{in}$. This naïve solution leads to systematic overestimation of the pressure values as the resistances measured are lower than the individual cells resistances. Fig. 8(c) and Fig. 9(c) show results for given input patterns. The "ghost" activation due to the cross-talk in the matrix is clearly visible (e.g. additional blobs in Fig. 9 row 3).

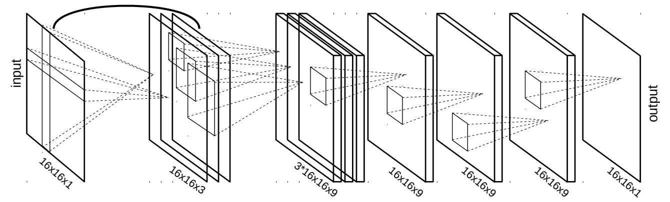


Fig. 7. The two convolutions in the first stage use 1×16 and 16×1 kernels and are intended to compute row and column sums. Both results are scaled up to the original 16×16 input size and are concatenated with the input. In the second stage, multiple parallel dilated convolutions with 3×3 kernels and dilation rates ranging from 1 to 3 are used for embedding context information [18]. In the final stage, multiple subsequent convolutions with 3×3 kernels perform the actual weight/pressure computation. All layers, except the first two, use zero padding to keep the size of the output the same as the input. All neurons are ReLUs [19].

Our proposed approach for pressure computation uses a convolutional neural network (CNN), which is trained for a mapping directly from ADC voltage matrices to pressure/weight matrices. By means of that, the two steps of correcting the measured resistance values and look-up the corresponding pressure value in the characteristics are done at once.

A. Network Architecture for Approximation of the Inverse Model

The network used is a convolutional feed forward network with the architecture given in Fig. 7. The input is the matrix of ADC voltages scaled to the range of [0,1]. The output represents the normalized weight/pressure values. The kernels in the first layer are intended to compute the row respectively the column sums and provide these for further computation to all the subsequent layers. This is motivated by the fact that one strong activation influences the whole column and row. The network has 4.544 trainable parameters, which is few compared to typical CNNs used for image processing. The small network also makes it possible to evaluate one matrix in $2.2ms$ on a i7-3720QM CPU, which is fast enough for processing the raw ADC matrices at the 100 Hz frame rate.

B. Training

The network has been trained on synthetic data generated using the analytical forward model from section IV. For that purpose, random pressure patterns were drawn in three different ways. First, by randomly selecting multiple rectangular regions in the input matrix, in which a Gaussian blob was added including a bit of noise. The first row of Fig. 8 shows an example of such patterns. In a second version, the rectangles

in: IEEE International Conference on Mechatronics (ICM), Ilmenau, Germany, pp. 232-237, IEEE 2019

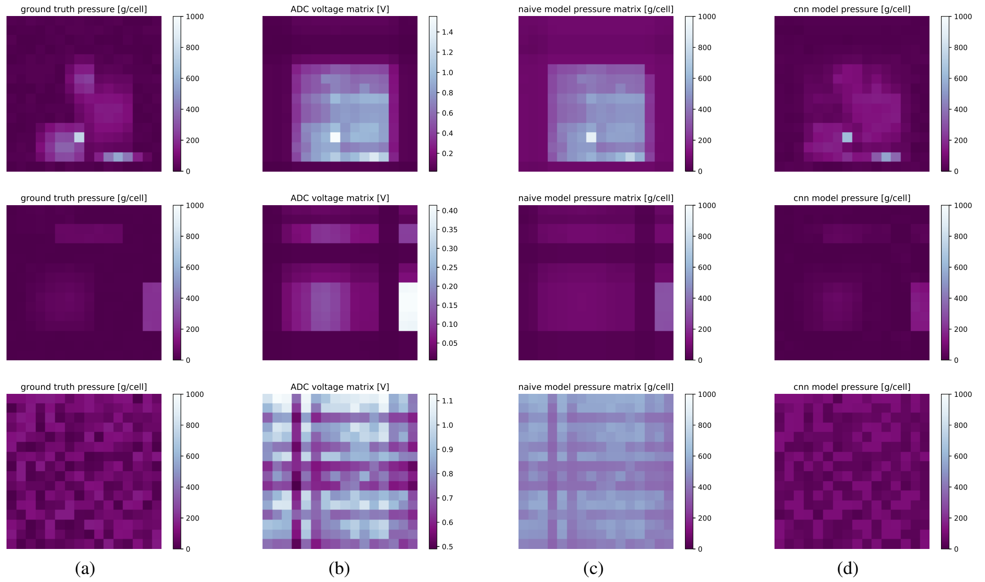


Fig. 8. Examples of synthetic datasets. (a) Ground truth pressure distribution, (b) The ADC voltage matrix resulting from the forward model applied to (a) and input for deriving the pressure matrix. (c) Result using the naïve approximation ignoring parallel pathways in the matrix, (d) Results of the CNN approach.

were filled with a constant value and some noise was added. This leads to more separated activation blobs and is shown in the second row of Fig. 8. The last version for generating training data was simply drawing random matrices and scaling them to the desired range. The sampled patterns are pressure distributions, that get transformed into resistance patterns by means of a lookup in the characteristic curve (Fig. 5) in order to apply the forward model. Solving the equation systems finally yields the ADC voltage matrices, which are inputs for the network training. A total of 100,000 pairs of ADC voltage and weight/pressure patterns were generated. The network was trained with stochastic gradient descent and mean squared error as loss function. The best epoch was selected based on the error on a validation set containing another 30.000 pairs.

VI. Experimental Results

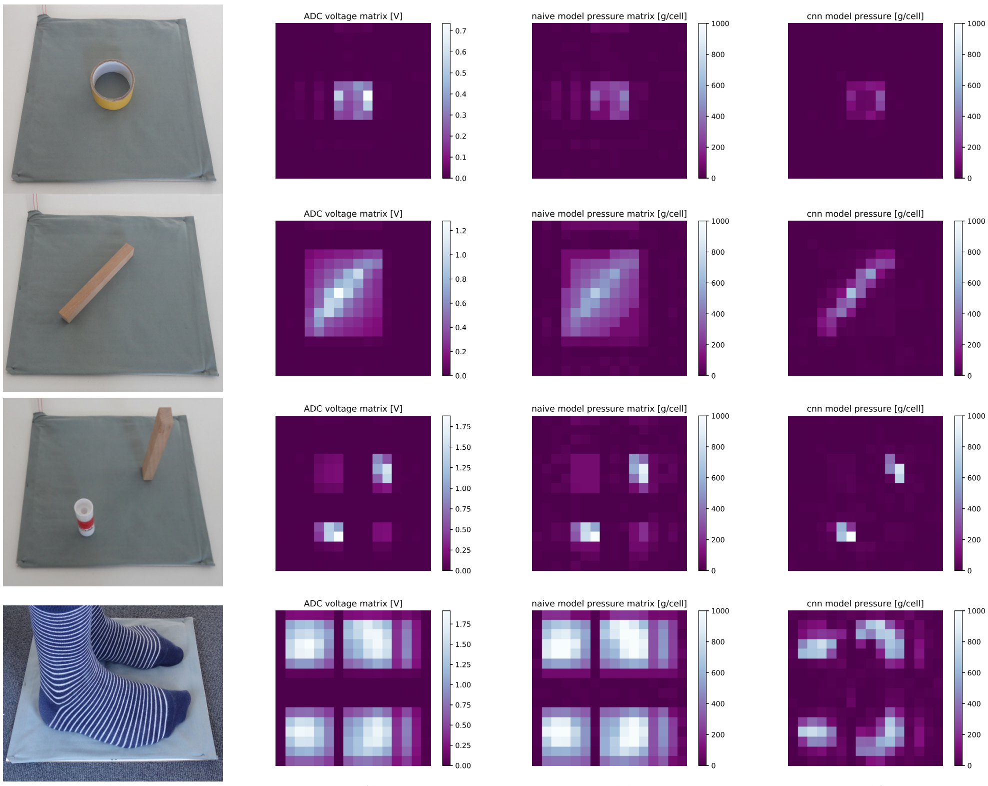
Using the trained network, two experiments were conducted to perform a quantitative and qualitative evaluation. First, the network outputs on an additional synthetic test dataset were compared to the ground truth patterns of pressure. The test patterns were similar but not equal to the training set and had pressure values up to $1\ kg/cell$. This cell wise comparison for the prediction of the convolutional network yield an average deviation of $11\ g$, which is pretty reasonable. Unfortunately, for the whole 16×16 sensor these deviations sum up to about $2.8\ kg$. In contrast to that, the naïve method in average has deviations of $150\ g$ in one cell and, therefore, a completely invalid sum for the whole array. Besides the exact values for each cell, the benefit of the proposed approach lies in the reconstruction of the actual shape of the input patterns, which involves suppression of wrong activations due to cross-talk. The second experiment, in which the network was applied to real data captured with the sensor array, illustrates this. The CNN is able to reconstruct the shape of the pressure patterns in a reasonable way, as shown in Fig. 9(d) while the naïve method (column c) fails to reconstruct the correct input shapes.

Regarding absolute pressure values in the real data experiment, the ground truth is missing. The actual resistive activation of the sensor depends on the exact shape and pressure distribution inside the cells which is unknown. Additionally, undefined amounts of an objects weight lies on the insensitive area in between the cells. For that reason, we used a fixed soft shape as interface to the sensor and increased the weight successively. This additional experiment showed that the results of the processed values scale proportionally with the weight, but are at a nearly constant factor of 1.42 greater than the exact load of the test weights. The standard deviation of that scaling factor is about 10% for different shapes and sizes of the interfacing object and, therefore, it is possible to calibrate the sensor by scaling the network outputs with the empirically determined factor.

VII. Conclusion

In this paper, we presented a machine learning approach for correcting the cross-talk effects of a resistive array sensor. The approach makes use of an analytically solvable forward model of the current distribution in the resistor matrix and aims to approximate its inverse using a CNN. Trained on sythetic data, the network is able to directly map an ADC voltage matrix to the correspondig pressure/weight matrix. Compared to the naïve method ignoring parallel pathways in the sensor array at all, the network shows great improvements, especially with respect to recovering the actual shape of the pressure patterns. Our approach shows that it is possible to implement a resistive sensor without sophisticated hardware for reading the resistance values. Future work needs to show if the improved pressure patterns gained with the proposed method will increase the classification accuracy for gesture recognition as well.

in: IEEE International Conference on Mechatronics (ICM), Ilmenau, Germany, pp. 232-237, IEEE 2019



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

Fig. 9. Examples of real pressure patterns captured with the 16x16 sensor array. (a) These objects have been pushed down on the sensor with increasing weights in order to generate the input patterns, (b) The ADC voltage matrix measured with a mid range weight, (c) Result using the naïve approximation ignoring parallel pathways in the matrix, (d) Results of the CNN approach showing improved and better fitting shapes of activation.

REFERENCES

[1] H.-M. Gross, St. Mueller, Ch. Schroeter, M. Volkhardt, A. Scheidig, K. Debes, K. Richter, and N. Doering, "Robot companion for domestic health assistance: Implementation, test and case study under everyday conditions in private apartments," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 5992–5999.

[2] H.-M. Gross, A. Scheidig, St. Mueller, B. Schuetz, Ch. Fricke, and S. Meyer, "Sympartner: Living with a Sympathetic Robot Companion at Home - Findings from a 20-Weeks Field Trial with 20 Seniors" to appear in proc. of *IEEE International Conference on Robotics and Automation (ICRA)* 2019.

[3] St. Mueller and H. M. Gross, "Making a socially assistive robot companion touch sensitive," in *Int. Conf. on Haptics and Touch Enabled Computer Applications (Eurohaptics)*. Springer, 2018, pp. 476–488.

[4] St. Mueller, Ch. Schroeter, and H.-M. Gross, "Smart fur tactile sensor for a socially assistive mobile robot," in *Int. Conf. on Intelligent Robotics and Applications (ICIRA)*, ser. LNAI, vol. 9246, no. II. Springer, 2015, pp. 49–60.

[5] L. Cang, P. Bucci, and K. E. MacLean, "Cuddlebits: friendly, low-cost furballs that respond to touch," in *Proc. of International Conference on Multimodal Interaction*. ACM, 2015, pp. 365–366.

[6] X. L. Cang, P. Bucci, A. Strang, J. Allen, K. MacLean, and H. Liu, "Different strokes and different folks: Economical dynamic surface sensing and affect-related touch recognition," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM, 2015, pp. 147–154.

[7] M. M. Jung, X. L. Cang, M. Poel, and K. E. MacLean, "Touch challenge'15: Recognizing social touch gestures," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM, 2015, pp. 387–390.

[8] St. Mueller, Ch. Schroeter, and H.-M. Gross, "Low-cost whole-body touch interaction for manual motion control of a mobile service robot," in *Int. Conf. on Social Robotics (ICSR)*, ser. LNCS, vol. 8239. Springer, 2013, pp. 229–238.

[9] J. Kim, M. Lee, H. J. Shim, R. Ghaffari, H. R. Cho, D. Son, Y. H. Jung, M. Soh, C. Choi, *et al.*, "Stretchable silicon nanoribbon electronics for skin prosthesis," *Nature communications*, vol. 5, p. 5747, 2014.

[10] D. Silvera-Tawil, D. Rye, and M. Velonaki, "Artificial skin and tactile sensing for socially interactive robots: A review," *Robotics and Autonomous Systems*, vol. 63, pp. 230–243, 2015.

[11] A. Schmeder and A. Freed, "Support vector machine learning for gesture signal estimation with a piezo-resistive fabric touch surface." in *NIME*, 2010, pp. 244–249.

[12] Z. Pan, H. Cui, and Z. Zhu, "A flexible full-body tactile sensor of low cost and minimal connections," in *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, vol. 3. IEEE, 2003, pp. 2368–2373.

[13] L. Osborn, H. Nguyen, J. Betthauser, R. Kaliki, and N. Thakor, "Biologically inspired multi-layered synthetic skin for tactile feedback in prosthetic limbs," in *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*. IEEE, 2016, pp. 4622–4625.

[14] G. Büscher, R. Kõiva, C. Schürmann, R. Haschke, and H. J. Ritter, "Tactile dataglove with fabric-based sensors," in *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*. IEEE, 2012, pp. 204–209.

[15] H. C. Markham and B. R. Brewer, "Development of a skin for intuitive interaction with an assistive robot," in *Engineering in Medicine and Biology Society*, 2009. EMBC 2009. IEEE, 2009, pp. 5969–5972.

[16] T. D'Alessio, "Measurement errors in the scanning of piezoresistive sensors arrays," *Sensors and Actuators A: Physical*, vol. 72, no. 1, pp. 71–76, 1999.

[17] A. Freed, "Novel and forgotten current-steering techniques for resistive multitouch, duotouch, and polytouch position sensing with pressure." in *NIME*, 2009, pp. 230–235.

[18] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[19] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

in: IEEE International Conference on Mechatronics (ICM), Ilmenau, Germany, pp. 232-237, IEEE 2019