Automatic configuration and calibration of modular sensing floors

Roberto Vezzani, Martino Lombardi, Rita Cucchiara
Softech-ICT - University of Modena and Reggio Emilia
Via P. Vivarelli, 2 - 41125 Modena - Italy
http://imagelab.ing.unimore.it

Abstract

Sensing floors are becoming an emerging solution for many privacy-compliant and large area surveillance systems. Many research and even commercial technologies have been proposed in the last years. Similarly to distributed camera networks, the problem of calibration is crucial, specially when installed in wide areas. This paper addresses the general problem of automatic calibration and configuration of modular and scalable sensing floors. Working on training data only, the system automatically finds the spatial placement of each sensor module and estimates threshold parameters needed for people detection. Tests on several training sequences captured with a commercial sensing floor are provided to validate the method.

1. Introduction

According with the "Internet-of-Everything" paradigm, the technological and cultural evolution of the last few years has led to an increasing presence around us of sensing devices. The environment where we live is increasingly scattered with sensors able to improve the human-environment interaction and to provide distributed surveillance and monitoring. Among different types of sensors, in this paper we will focus on the exploitation of sensing floors. Their applications are manifold in several fields, including both public and private environments. For example, smart buildings can include sensing floors to detect the presence of people and to automatically switch on/off the lighting or the heating systems. In the e-health field, these devices can be used to detect dangerous situations such as an elder falling or getting out of his/her bed [8]. Furthermore, sensing floors can be used for people counting or to monitor crowd movements during public events, exhibitions, and so on.

In comparison with other traditional technologies such as video cameras, sensing floors provide less information but have two undoubted advantages. First, they are completely privacy compliant. In fact, it is not feasible to recognize and identify users from their data only. Thus, the installation of these sensors is also allowed in private places, such as toilets or bedrooms. Second, sensing floors data are not affected from occlusions, a typical issue of visual camera systems.

In this work, we discuss the very common and general problem of sensor calibration in wide areas. Regardless of the technology, the whole floor should be covered with several modules, which need to be opportunely calibrated. The ease of installation and configuration are essential features for the diffusion of sensing floor systems, specially when the number of installed modules grows. However, most of the solutions proposed in the past have two drawbacks. First, an individual calibration phase is required for each sensing element, because its response depends both on intrinsic characteristics and on installation issues. For example, the presence of ceramic tiles above the sensors, the morphology and the planarity of the layer under the sensors may affect the response level and range of each element. Secondly, a manual configuration of the setup should be provided to correctly specify the location of each module within the whole sensing area. More specifically, the spatial transformations which convert coordinates from the local coordinate systems of each module to a global and common coordinate system are needed. The same problem arises from the installation of many cameras [10].

In this paper, we propose an automatic method to calibrate and discover the layout of sensing units solely based on acquired data.

2. Related work

Some prototypes of sensing floors have been designed and proposed in the past and most of them are composed of two separate layers: an electronic device and a hard (*i.e.* wood, ceramic) or soft (*i.e.* carpet, PVC) coverage surface. The electronic layer captures the pressure exerted on the floor, while the above tiles help to preserve the sensor integrity as well as acting as a lens which distributes punctual pressures over a larger space. For example, modules of matrix switch resistors placed under a carpet coverage were proposed in [5] and in [6] for gait recognition and human

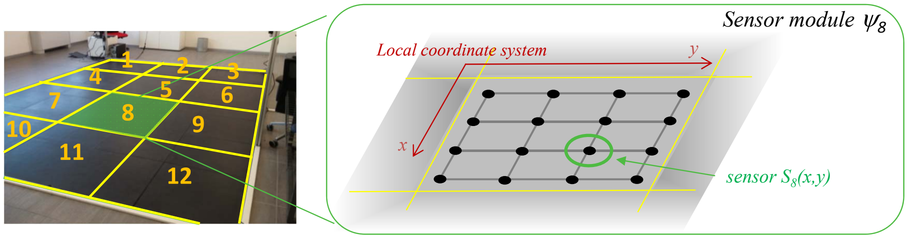

Figure 1. An example of a sensing floor composed by 12 modules. Each module is made of a grid of sensors $S_m(x, y)$.

tracking applications, respectively.

Law *et al* [2] presented a sensing floor based on a square array of individual wooden tiles (30.5 × 30.5 cm), each of them equipped with four Force-Sensing Resistors (FSR).

The *SensFloor* system was proposed in [9] for human tracking and identification. It is composed by a grid of textile modules (50 × 50 cm) beneath a ceramic tiles coverage. Each module is equipped with eight surrounding triangular capacitive sensor pads and an embedded system-on-chip device to perform capacitance measurements and the necessary signal filtering.

The *Future Care Floor* described in [3] is built using wooden tiles (600 × 600 × 40 mm) equipped with a piezoelectric sensor and a micro-controller for data acquisition and processing.

In collaboration with an Italian ceramic company[1], we have recently developed a commercial solution, called *FlorimAge*. The system follows the guidelines presented in [4]. The modular solution is composed by small standalone devices (1 × 1 m) that can be placed side by side to cover wide areas. Each module is assembled as a stack of three layers: the first one can be placed over a pre-existing floor and it is a flexible electronic circuit. The second layer is made of a conductive polymer. The top layer is a coverage of ceramic tiles (600 mm by 600 mm wide) which are thin enough (4.5 mm) to allow the sensing elements below them to capture the presence of walking people.

3. Sensing Floor Data and Problem Definition

Let Ψ be the whole data captured from a generic sensing floor. If the sensing elements are spatially arranged on a grid, the captured data Ψ can be stored as a 3D matrix $\Psi = \{\psi(x, y, t)\}, x \in [1 \dots W], y \in [1 \dots H], t \in [1 \dots T]$. W and H define the sizes of the whole floor grid and T is the number of temporal samples.

Each element $\psi(x, y, t)$ is the digital value sampled at time t from the sensor $S(x, y)$. The spatial neighborhood relations among sensors are preserved in the matrix Ψ. Due to anomalies of the coat below the floor or irregular placement of the tiles above it, each matrix element $\psi(x, y, t)$ may have a proper output range different from the others. A linear transformation should be included in the capturing step to normalize the sensor response, as in the following equation:

$$\hat{\psi}(x, y, t) = \alpha_{x,y} \cdot \psi(x, y, t) + \beta_{x,y}. \tag{1}$$

The parameters α and β of the linear transformation are specific for each sensor and they should be provided or learned during a sensor calibration step. Once the linear transformation is applied, a global threshold could be selected and used to binarize the floor data. In particular, the binary foreground matrix F can be obtained as:

$$F(x, y, t) = \begin{cases} 1 & \text{if } \hat{\psi}(x, y, t) > Th \\ 0 & \text{otherwise} \end{cases} . \tag{2}$$

The threshold Th depends on the specific application constraints and it is related to noise level and to the required floor sensitivity. For instance, people detection and surveillance algorithms need that the foreground matrix contains a '1' when a person is stepping on the corresponding sensor. In this case, Th is a sort of lower bound of a person weight.

In real implementations, the whole data Ψ are collected by multiple sensor modules working in parallel for scalability issues. The modules can be freely disposed on the floor to cover large areas. However, since a grid distribution of the modules leads to an easier integration and processing of the data, we adopt this grid assumption. Therefore, Ψ is obtained by mosaicing the sub-matrices $\Psi_m, m \in [1 \dots M]$ sampled from the M modules.

A schema of the processing system is reported in Figure 2. Each sensor $S_m(x, y)$ delivers a digital value $\psi_m(x, y, t)$ every sample instant t. The index m refers to the hardware module which contains the sensor, while the coordinates x, y indicate the position of the sensor within the module using the local reference system (see Figure 1).

A linear transformation is applied to the captured values to normalize their range. As above mentioned, the parameters of the transformation are different for each sensor and are defined during the sensor calibration step. The hardware boards responsible of capturing the sensor data are able to package their own values into data blocks Ψ_m, but the complete floor data Ψ should be reassembled by mosaicing the blocks using the provided floor layout before further high-level processing tasks.

[1] http://www.slim4plus.it/en/floor-sensor-system/

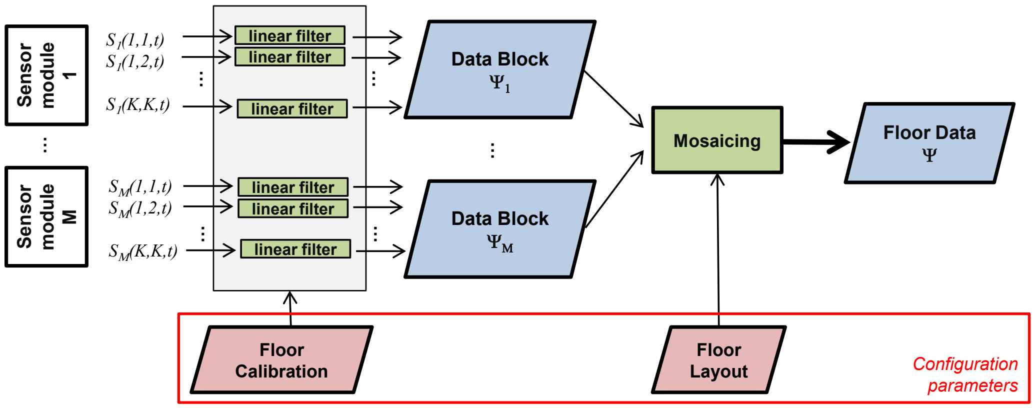

Figure 2. Low level processing of sensor data. The output value normalization and the module mosaicing step are used to generate the whole floor data Ψ.

In this paper, we propose an automatic system to estimate the *Sensor Calibration* and the *Floor Layout* configuration parameters (red parallelograms in Figure 2) directly from training data.

4. Sensor calibration

The calibration step is carried out at sensor level. No spatial relations are exploited. Due to several factors, each sensor may provide values within different output ranges. Figure 3 shows the temporal response of a sample sensor. The graph on the top plots the temporal sequence of sensor responses. During the sequence, a person walked over the sensor about ten times, generating the same number of positive peaks. When the person raises his feet from the floor, the corresponding sensor provides negative peaks, since the sensor coverage is subjected to a temporary swing behavior. Similar responses are also generated if the coverage is somehow unsteady or if a person is walking near to the sensor. For the rest of the time, the sensor output is related to the dead load and is perturbed by a Gaussian measurement noise.

We dealt the sensor calibration as a classification problem. Given a new sample from a sensor, we want to classify it as belonging to one of the following classes: *negative-peak*, *background*, and *positive-peak*. A mixture of three Gaussian distributions is fitted to the data captured from each sensor as a training sequence. The parameters obtained using the Expectation Maximization (EM) algorithm are stored and used as input for a Bayesian classifier, which also allows a binarization of the floor data into background-foreground masks (positive peaks are treated as foreground points, and the others as background). In addition, the parameters of the mixture are exploited to compute the linear transformation parameters α and β of Eq. 1.

Given a data sequence $\{\psi(x,y,t)\}_{t=1...T}$, the EM fitting algorithm returns the 3D mean and variance vectors $\mu = [\mu_1\ \mu_2\ \mu_3]$ and $\sigma = [\sigma_1\ \sigma_2\ \sigma_3]$, corresponding to the three components depicted in the bottom graph of Figure 3:

$$\begin{array}{l}\hat{\psi}(x,y,t) = \alpha_{x,y} \cdot \psi(x,y,t) + \beta_{x,y} = \frac{\psi(x,y,t) - L_1}{L_2 - L_1} \\ L_1 = \mu_2 - \sqrt{\sigma_2}, \quad L_2 = \mu_3 + 2 \cdot \sqrt{\sigma_3}\end{array} \tag{3}$$

where $[L_1, L_2]$ is the normalized range of each sensor output. The indexes x, y have been omitted for the sake of readability. In Figure 3.top the mean of the Gaussian component related to foreground values is drawn in green, while the automatic threshold used by the classifier to distinguish between background and foreground points is drawn in red. The Threshold Th is set as the intersection value between the background and the foreground Gaussian curves (see Figure 3.bottom).

5. Layout discovery

The layout discovery algorithm is based on the work of Pomeranz *et al.* [7]. They proposed a system to automati-

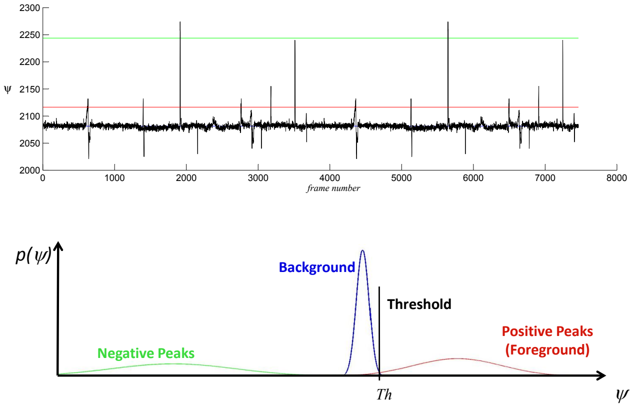

Figure 3. Plot of a real sequence of data captured from a single sensor (top). The EM fitting algorithm automatically recovers the three Gaussian distributions shown in the sketch (bottom).

cally solve a jigsaw puzzle. After splitting an image into a scrambled set of tiles, the algorithm find the correct position occupied by each in the source image. The method trusts on the similarity at the borders of neighbor tiles. Thus, key elements of the approach are a suitable compatibility measure between couples of tiles and a greedy optimization algorithm to reduce the computational complexity. In fact, the brute force approach is not allowed due to the factorial complexity of the problem with respect to the number of tiles.

Similarly to the jigsaw puzzle, we need to discover the correct position of each portion of the floor (floor tile). To this aim, we can exploit the similarity of the temporal data sequences captured by the sensors placed near to the edges of the tiles. Given two random floor tiles $\Psi_i, \Psi_j \in \Psi$, we need a metric to measure their compatibility given a spatial relation $R \in \{l, r, u, d\}$ (i.e., left, right, up, down) between them. In other words, we need a suitable compatibility metric which indicates how likely the tile Ψ_j is placed on the left/right/up/down side of the reference tile Ψ_i, based on a training data.

If the spatial distance between two neighbor sensors is low enough, the correspondent temporal series of the captured data will be highly correlated. This correlation is enhanced thanks to the ceramic tiles, which spatially smooth the sensor response. Let $C(\Psi_i, \Psi_j, R)$ be a compatibility metric estimated between the two tiles, given the spatial relation R. The compatibility of two tiles can be computed as a function of their dissimilarity [7]:

$$C(\Psi_i, \Psi_j, R) \propto \exp\left(-\frac{D(\Psi_i, \Psi_j, R)}{quartile(i, R)}\right), \qquad (4)$$

where $quartile(i, R)$ is the quartile of the dissimilarity between all other parts in relation R to part Ψ_i.

In case of images, the dissimilarity $D(\cdot)$ of two tiles can be measured by summing up the color differences of the pixels along the parts' abutting boundaries [1]. Since floor tiles are temporal sequences, their dissimilarity metric could take into account temporal as well as spatial data missing correlations along the parts' abutting boundaries.

5.1. Dissimilarity metrics

The first *Dissimilarity metric* D_1 for floor data can be defined as following, where the right (*'r'*) relation is used:

$$D_1(\Psi_i, \Psi_j, r) = \sum_{t=1}^{T}\sum_{x=1}^{K} d\left(\psi_i(x, K, t), \psi_j(x, 1, t)\right), \quad (5)$$

where $d(\cdot, \cdot)$ is a distance function between two sensor values. The Euclidean distance has been used to this aim in our system, but more complex may improve the results in some particular cases [7].

Figure 4 shows the subset of sensors (colored in green) used in the computation of the Dissimilarity metric in the case of *right* relation.

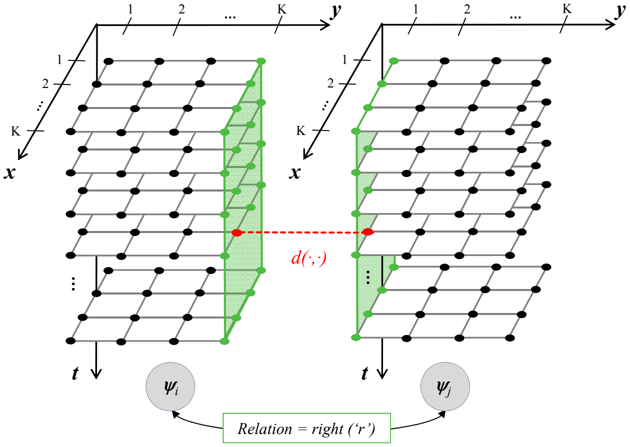

Figure 4. Computation of the Dissimilarity metric in the case of *right* relation. The temporal series of each sensor colored in green are taken into account.

The Dissimilarity metric D_1 does not take into account some useful information available thanks to the automatic calibration step described in Section 4. Thus, we propose and test other two dissimilarity metrics. Exploiting the linear filtering, Equation 5 can be modified to compute the distances between the normalized sensor values:

$$D_2(\Psi_i, \Psi_j, r) = \sum_{t=1}^{T}\sum_{x=1}^{K} d\left(\hat{\psi_i}(x, K, t), \hat{\psi_j}(x, 1, t)\right). \quad (6)$$

Furthermore, the automatic calibration step indicates the best threshold values to classify each floor data element into background/foreground.

If available, the binarized values of Equation 2 allow to define a suitable Dissimilarity metric D_3:

$$D_3(\Psi_i, \Psi_j, r) = 1 - \frac{CoOcc(F_i, F_j, r)}{Occ(F_i, F_j, r) + \epsilon}, \qquad (7)$$

where $CoOcc$ and Occ are the number of *CoOccurrences* and *Occurrences* of foreground pixels at the selected borders of Ψ_i and Ψ_j and are computes as:

$$\begin{aligned} CoOcc(F_i, F_j, r) &= \textstyle\sum_{t,x} F_i(x, K, t) \wedge F_j(x, 1, t) \\ Occ(F_i, F_j, r) &= \textstyle\sum_{t,x} F_i(x, K, t) \vee F_j(x, 1, t), \end{aligned} \qquad (8)$$

where the sum limits are omitted for the sake of readability. The ϵ term as been included in Equation 7 to avoid division by zero.

The dissimilarity D_3 is related to the number of times two neighbor sensors are contemporaneously excited (i.e.,when they are synchronously stepped by a person), but it does not handle temporal causalities (i.e., when they are stepped by a person one immediately after the other). Common correlation and convolution filters are able to discover uni-directional relations and are time consuming. The alternative we propose is based on binary morphological operations. In particular, we have applied a binary dilation of the foreground multidimensional matrices F using a cubic structuring element of size 3. In addition to temporal causalities, the dilation operation allows to spatially enlarge the neighborhood of each sensor during the computation of the number of *CoOccurrences* and *Occurrences*. In the experimental section, we call D_4 the Dissimilarity metric computed as in Equation 7 but on the dilated foreground matrices.

6. Experimental evaluation

The automatic configuration and calibration system has been tested on different sequences captured from a FlorimAge setup installed in our laboratory. The floor is composed by 12 independent modules of 8x8 sensors each, placed as in Figure 1. The whole floor data is thus a grid of 32x24 sensors. To increase the number and difficulty of the tests, we have also split each modules into many finer modules. In particular, we tested the system with $K = 2,\ 4$ and 8, generating $M = 12,\ 48$ and 192 blocks, respectively. The blocks are randomly scrambled before the tests, keeping their original position as ground truth.

We have collected 14 sequences, characterized by different durations, people weight, count and motion patterns. These sequences were collected emulating real conditions. For example, to simulate a calibration phase during a sensing floor installation, we collected 2 sequences with a person following a preset pattern to completely scan all the sensor units under the floor. To simulate a crowd situation, 9 sequences with walking/standing people have been acquired, involving 1 to 6 individuals randomly moving on the sensing floor. The individuals were also free to leave the sensorized area. Finally, to emulate the use of the system in points of passage such as corridors or gangways, 3 sequences have been collected with people walking according to a predefined unidirectional flow. Table 1 reports a summary of the adopted dataset.

Performance evaluation metrics: the *Direct Comparison* (DC) is the most important metric for the evaluation of the proposed system. Given an output layout assignment, the DC value indicates the fraction of blocks correctly placed over the total number of blocks. A perfect match leads to a DC value equal to 1. As stated in [1], this score is not complete and can be misleading. DC is equal to zero even if the layout has been shifted by one row or column only. In these cases, the *Neighbor Comparison* (NC) metric provides a better performance evaluation. NC is computed as the fraction of mutual relations correctly found between couples of blocks.

Table 1. Dataset summary

# seq.	Duration[1] [mm:ss]	Duration[1] frames	People count [2]	Description
1	06:13	3731	1	Complete scan of the floor by a single person
2	05:05	3051	1	
3	02:23	1436	2	Unidirectional flow of people
4-5	01:00	600	1	
6	00:23	236	1	Random walk of a single person
7-11	01:17	776	1	
12-13	01:50	1200	2	Random walk of groups of people
14	02:15	1356	6	

[1] For aggregate rows, duration and frames are average values.
[2] Number of people simultaneously on the floor

Figure 5 shows the DC and NC scores obtained on each sequence using the four proposed metrics D_1-D_4. The correct layout is always discovered using the original block size ($K = 8$) and exploiting the foreground metrics D_3 or D_4. The other metrics reach good results only on sequences 1 and 2, which contain a lot of handoffs between tiles. Using smaller block sizes, the system performance decreases. However, if the training sequence is complete enough (sequences 1 and 2), the proposed system is still able to find the correct tile placement. Average performances are reported in the last bin of each graph. The metric D_4 has provided the best average result in all the cases.

Since the sensor calibration and layout discovery need to be computed only once, no time constraints are usually imposed in real application. On average, the solver took less than one second to provide a solution with $K = 8$, while about 100 seconds to place the 192 tiles obtained with $K = 2$. The code has been written using Matlab and without including specific code optimization.

7. Conclusions

We have proposed an automatic system to configure a wide area sensing floor. In particular, the framework finds the correct spatial placement of sub-modules using a training sequence instead of needing a manual configuration. In addition, a sensor-wise calibration and automatic threshold finding procedure is applied to binarize the sensor data into the common background-foreground masks used in surveillance applications. The provided tests on a real setup confirm the effectiveness of the method. In the future, we aim at removing the assumption of grid distribution of the modules, to allow an automatic localization of a free distribution of sub-modules on the floor.

Acknowledgments

This work was partially supported by Florim Ceramiche S.p.A. (Italy). The experimental tests have been carried out within the project "La Citta' Educante" (CTN01_00034_393801) of the National Technological

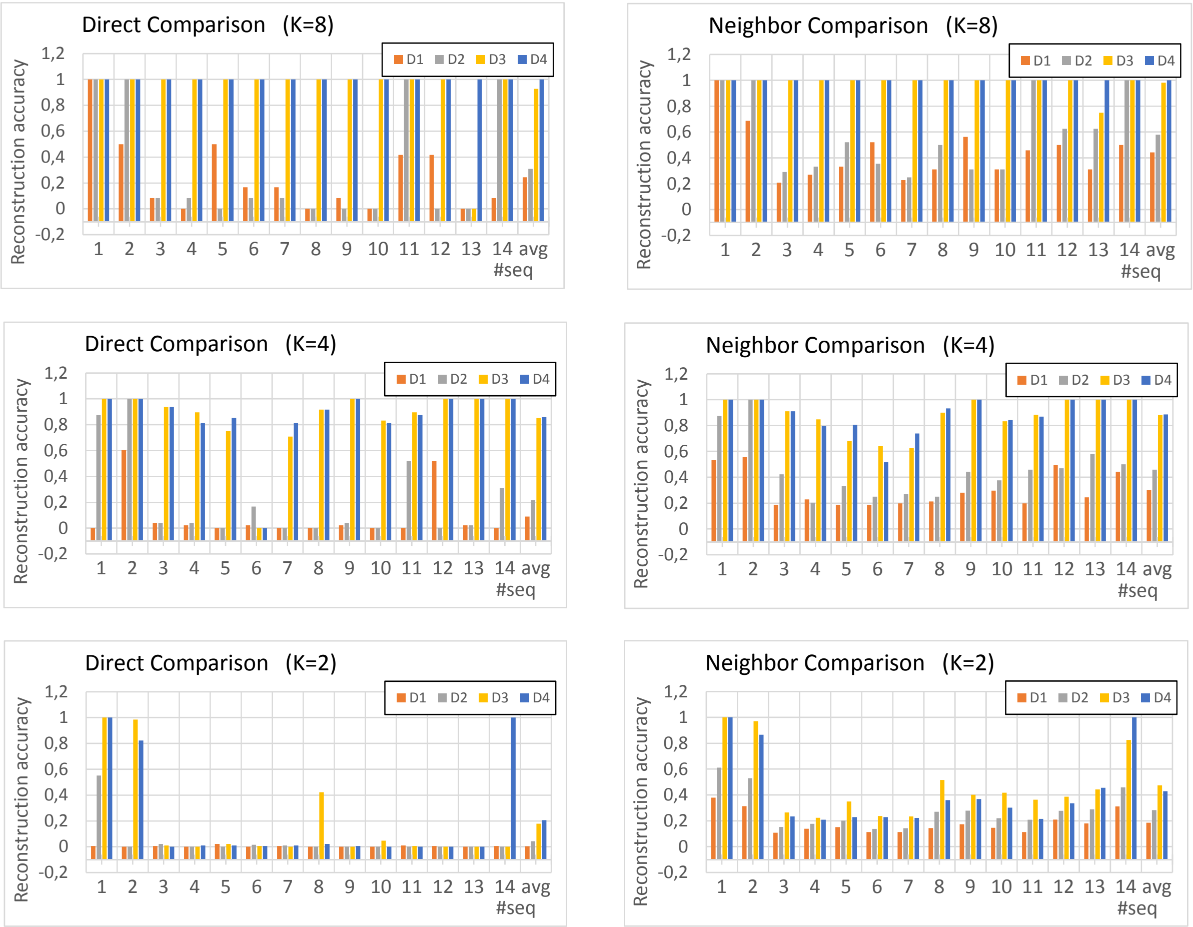

Figure 5. Performance evaluation of the proposed system in terms of DC and NC metrics for each video sequence and block size K.

Cluster on Smart Communities funded by the Italian Ministry of Education, University and Research (MIUR).

References

[1] T. S. Cho, S. Avidan, and W. Freeman. A probabilistic image jigsaw puzzle solver. In *Proc. of IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, pages 183–190, June 2010. 4, 5

[2] A. Law, B. Peck, Y. Visell, P. Kry, and J. Cooperstock. A multi-modal floor-space for experiencing material deformation underfoot in virtual reality. In *Proc. of IEEE Int'l Workshop on Haptic Audio visual Environments and Games*, pages 126–131, Oct. 2008. 2

[3] P. Leusmann, C. Mollering, L. Klack, K. Kasugai, M. Ziefle, and B. Rumpe. Your floor knows where you are: Sensing and acquisition of movement data. In *Proc. of Int'l Conf. on Mobile Data Management*, pages 61–66, June 2011. 2

[4] M. Lombardi, A. Pieracci, P. Santinelli, R. Vezzani, and R. Cucchiara. Sensing floors for privacy-compliant surveillance of wide areas. In *Proc. of IEEE Conf. on Advanced Video and Signal-Based Surveillance*, Kraków, Poland, Aug. 2013. 2

[5] L. Middleton, A. Buss, A. Bazin, and M. Nixon. A floor sensor system for gait recognition. In *IEEE Workshop on Automatic Identification Advanced Technologies*, pages 171–176, 2005. 1

[6] T. Murakita, T. Ikeda, and H. Ishiguro. Human tracking using floor sensors based on the markov chain monte carlo method. In *Proc. of Int'l Conf. on Pattern Recognition*, volume 4, pages 917–920 Vol.4, Aug. 2004. 1

[7] D. Pomeranz, M. Shemesh, and O. Ben-Shahar. A fully automated greedy square jigsaw puzzle solver. In *Proc. of IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, pages 9–16, June 2011. 3, 4

[8] H. Rimminen, J. Lindstrom, M. Linnavuo, and R. Sepponen. Detection of falls among the elderly by a floor sensor using the electric near field. *IEEE Transactions on Information Technology in Biomedicine*, 14(6):1475–1476, Nov. 2010. 1

[9] M. Sousa, A. Techmer, A. Steinhage, C. Lauterbach, and P. Lukowicz. Human tracking and identification using a sensitive floor and wearable accelerometers. In *Proc. of IEEE Int'l Conf. on Pervasive Computing and Communications*, pages 166–171, Mar. 2013. 2

[10] X. Wang. Intelligent multi-camera video surveillance: A review. *Pattern Recognition Letters*, 34(1):3 – 19, 2013. 1