

Towards Dimensionality Reduction for Smart Home Sensor Data

Bassam Mokbel, Alexander Schulz *

CITEC centre of excellence, Bielefeld University, Germany

{bmokbel|aschulz}@techfak.uni-bielefeld.de

Preprint of the publication [11], as provided by the authors.

Abstract

In this paper, we investigate in how far nonlinear *dimensionality reduction* (DR) techniques can be utilized to tackle particular challenges of sensor data from smart home environments. Smart homes often contain a large number of sensors of various types, providing output in real time, which results in a sequence of high-dimensional, heterogeneous data vectors. We propose that DR techniques can provide a truthful low-dimensional representation (i.e. a compression) of this kind of data, together with a corresponding reconstruction (i.e. decompression). This yields an automatic fusion of uncoordinated raw sensor signals, as well as an economical storage format, with a certain robustness against sensor failure. In proof-of-concept experiments, we present first empirical results to test our approach based on real-world data.

1 Introduction

Motivation

In recent years, the topic of *smart homes* gained major attention in the electronics and electrical appliances industry. The term describes environments, in which a collection of different sensors and individual devices are utilized for home automation, for assisted living in healthcare situations, or other interactive home scenarios to provide convenience and intelligent assistance in everyday life. In this context, efficient data mining and machine learning algorithms play a prominent role, in order to adapt these systems to the inhabitants and their environment, based on given sensor data. For example, smart devices may learn the daily habits of a user, and infer rules to detect the corresponding living situations automatically.

Data in smart home environments are typically characterized by temporal streams of heterogeneous, high-dimensional sensor readings as well as preprocessed sensor evaluations, e.g. temperature sensors, motion sensors, or cameras providing face detection and the location of inhabitants. In this context, we expect the following challenges for machine learning:

*Funding from DFG under grant number HA2719/7-1 and by the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277) at Bielefeld University is gratefully acknowledged.

- (I) *High-dimensional data* Given the increasing ubiquity and precision of sensors, together with the limited memory capacity in typical home automation devices, it becomes challenging to analyze the output of all available sensors at once, and to store a sufficient amount of historical data in order to apprehend an adequate temporal context for analysis tasks.
- (II) *Heterogeneous data* At the same time, the large variety in the output data from different sensor types typically requires hand-tuning and expert knowledge to set up a successful machine learning system for the given data. Thereby, raw sensor readings are preprocessed and meaningfully represented in accordance with other sensor output. It would be beneficial to avoid costly human effort for such problem-specific adaptations.
- (III) *Uncoordinated sensors* Similarly, it can be expected that the available collection of sensors and sensor modalities is very diverse in general, and may even be flexible over time: home environments are different in shape and size, so the technical configuration will also vary, while sensors may dynamically change or possibly fail. Therefore, a hand-crafted organization of sensors to gain higher-level abstract data representations seems no longer appropriate. Rather, machine learning techniques will have to deal with loose collections of sensors, which provide individual local views of the environment and may yield missing values.

Nonlinear dimensionality reduction

Regarding these three challenges, the aspect of ‘high-dimensional data’ can be addressed directly by *dimensionality reduction* (DR) techniques: With a large collection of sensors available in a smart home, it is expectable that several signals will be correlated, and the intrinsic dimensionality is actually lower than the number of sensor outputs. Therefore, DR is a promising tool to establish a low-dimensional representation from the original sensor data, which is easier to handle in terms of storage space and complexity for data analysis and machine learning algorithms.

DR has been an emergent research topic over the last decade, with successful applications in a variety of scientific and industrial fields, especially for nonlinear DR methods. Recent advances include novel acceleration techniques for big data sets [12, 4], as well as principled evaluation and parameterization approaches [7, 6]. Hence, there are many DR algorithms readily available, see [8, 13] for overviews. Some existing work in the literature demonstrates the general benefit of DR also in the context of sensor data: for example, linear DR [14], and sparse coding techniques [1] were used to achieve compressed representations from sensor signals.

However, given the heterogeneous characteristics of sensor data in smart homes, as described in our challenge (II), we can expect that correlations are usually not linear. Therefore, linear DR may fail to capture the relevant information in the data, unless problem-specific preprocessing steps (including nonlinear transformations) are established by human experts. Hence, we believe that nonlinear DR techniques would be more appropriate, which has not yet been sufficiently investigated in current literature.

There exist successful applications of manifold-based nonlinear DR techniques like the *self-organizing map* (SOM) [2] or *Isomap* [5] for spatio-temporal sensor data. However, those approaches realize a rather restrictive embedding: while the SOM depends heavily on the a-priori choice of a lattice structure for the embedded points, Isomap requires an appropriate neighborhood graph in the original data space. This makes both methods less flexible w.r.t. unseen data. Regarding challenges (II) and (III), we believe that the generalization ability of a DR embedding is a key ingredient for treating smart home data, and should be investigated further. Modern high-performance nonlinear DR techniques have proven to be successful in terms of generalization

for biological data, see [3]. Therefore, these approaches seem like a viable alternative to tackle the challenges of smart home data. In this contribution, we present first empirical results by testing the t-SNE [13] method in particular, together with a recent extension by a kernel mapping approach [4], using a small data set of raw sensor signals for proof-of-concept experiments.

2 Dimensionality reduction for smart home data

The goal of DR is to find low-dimensional (low-D) vectors $\{\mathbf{y}_1, \dots, \mathbf{y}_N\} = Y \subset \mathbb{R}^L$, which resemble the structure of the original high-dimensional (high-D) data set $\{\mathbf{x}_1, \dots, \mathbf{x}_N\} = X \subset \mathbb{R}^H$. The dimensionality L of the embedding space \mathbb{R}^L is defined a priori by the user, and is often chosen as 2 or 3 for the benefit of visualizing the embedded points in a scatter plot. Many nonlinear DR techniques optimize the low-D vector locations such that their neighborhoods resemble the neighborhood structure in the original data w.r.t. certain criteria of distance preservation. If the data’s intrinsic dimensionality is higher than L , not all pairwise distances between the data can be represented accurately in the embedding, and a certain loss of information is inevitable. To assess how truthful a given embedding represents the original high-D data, there exist independent quality criteria to evaluate the reliability of their low-D counterparts, see e.g. [9, 10].

Typically, nonlinear DR methods emphasize the preservation of local neighborhoods in favor over a faithful reproduction of larger distances. In this work, we use the well-established nonlinear DR technique *t-distributed stochastic neighbor embedding* (t-SNE) [13], which aims to minimize the discrepancy between neighborhood probability distributions in the original, and the embedding space, as measured by the Kullback-Leibler divergence. The approximate size of the considered neighborhood can be controlled explicitly via the *perplexity* parameter in t-SNE. One limitation of t-SNE (and most other nonlinear DR techniques) is that the embedding Y is obtained by an optimization procedure based on the given data X , but there is no functional form available for the mapping. In case of time series as they occur in smart homes, this has certain downsides: On the one hand, we require an explicit function to add unseen data (e.g. incoming sensor readings) seamlessly to the existing low-D representation of previous data, i.e. an embedding function of the form $f: \mathbb{R}^H \rightarrow \mathbb{R}^L$ with $\mathbf{x}_i \mapsto f(\mathbf{x}_i) = \mathbf{y}_i$ for all $i = \{1, \dots, N\}$. On the other hand, we may see the embedding as a compressed data representation, which allows us to store and handle all \mathbf{y}_i as a compact alternative to \mathbf{x}_i . Hence, we also demand a corresponding ‘decompression’ function, by which we can reproduce original sensor signals from their low-dimensional counterparts, i.e. an inverse mapping $g: \mathbb{R}^L \rightarrow \mathbb{R}^H$ which approximately recovers high-D points $\tilde{\mathbf{x}}_i \in \mathbb{R}^H$ with $\mathbf{y}_i \mapsto g(\mathbf{y}_i) = \tilde{\mathbf{x}}_i \approx \mathbf{x}_i$ for all $i = \{1, \dots, N\}$.

To achieve these desired properties, we refer to a recent extension called *kernel t-SNE* [4]. This method uses the original high-D points together with their low-D counterparts embedded by the regular t-SNE, and establishes a mapping between the two via regression. Therefore, a generalized linear mapping is assumed, with parameters α and an appropriate basis function, in our case a normalized Gaussian kernel of bandwidth σ , see [4] for details. This addresses both of our requirements. On the one hand, we can train the kernel mapping to establish an embedding function $f(\mathbf{x}) = \mathbf{y}$ for unseen data \mathbf{x} . On the other hand, we can obtain an inverse function $g(\mathbf{y}_i) = \tilde{\mathbf{x}}_i \approx \mathbf{x}_i$ for reconstructing high-D points from low-D inputs, by using the same learning scheme in the opposite manner: the kernel mapping is then trained for given low-D vectors with their high-D counterparts as regression targets.

3 Experimental evaluation

Application background & technical setting

In our experiments, we use real world sensor data from “*The Cognitive Service Robotics Apartment as Ambient Host*” project (CSRA) at the CITEC research facility in Bielefeld, Germany. The CSRA project involves an intelligent apartment for research purposes, where multiple sensors constantly monitor various conditions of the environment, as well as human interactions in each room. The apartment consist of a living room with an adjacent kitchen area, a small entrance hallway, and a fitness room, each holding appropriate furniture and home appliances. Details can be found on the project’s website¹.

We are addressing two modalities of sensor readings, both producing multi-dimensional data streams in real-time. With a rather small dimensionality and limited complexity of this data, we are not covering all three challenges from Section 1 to the full extent with our experiments. However, we believe that the data represents all three aspects to some degree, and our results may serve as a first proof-of-concept.

In the first modality, we use 3-dimensional person tracking data from three different overhead camera systems: These depth-image cameras are installed in the ceiling at the entrance, the kitchen, and the living room. Each camera preprocesses the depth image to yield a person location hypothesis in its local 3D coordinate system, whenever a person is detected in the camera’s field of view. Since the fields of view of adjacent cameras are overlapping, a person hypothesis may be present in several local coordinate systems at a time, in case a user is moving through the overlapping areas. Note, that these depth cameras are low-cost devices similar to the Microsoft Kinect model, which seems realistic for person tracking in future home automation systems. Additionally, the fact that the cameras yield separate local tracking results (rather than unified spatio-temporal data in a global coordinate system) is in line with our premise of ‘uncoordinated sensors’ from Section 1. From all three cameras with individual 3D coordinate systems, we gain 9 real-valued data dimensions in each time step.

The second modality comes from a SensFloor[®] tactile floor mat, which is installed on the kitchen ground. It yields continuous pressure levels for 8 individual segments which are evenly partitioned in a clockwise fashion. Like in the case of a single camera, we can expect changing values only when a person is moving in the kitchen area. With these 8 pressure sensors, and the 9-dimensional data from person tracking, we are addressing 17 real-valued data dimensions in total. Since the distribution and correlation of values in the pressure sensors are much different from the camera tracking output, our scenario involves the challenge of ‘heterogeneous data’.

In the following, we will address three distinct research problems in the context of smart home data, which relate to the challenges in Section 1. For each problem, we will present experimental results from our described setting. The data for our experiments consist of two sequences, in which a single person walks through the apartment in an exploratory manner. The duration of *Sequence A* is 86 seconds, while *Sequence B* lasts 50 seconds. In both sequences, the person enters the field of view of every camera at least once, and is walking over the tactile floor panels briefly.

Each sequence is a time series $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ of 17-dimensional vectors $\mathbf{x}_i \in \mathbb{R}^{H=17}$. Every vector represents the sensor values provided within a 50 millisecond time frame. We applied only general, fairly simple preprocessing steps, which are not specific to the given scenario or sensor characteristics: If a sensor provides missing values (e.g. when a camera does not detect any person), we continue to write the last known values in the subsequent vectors for each following

¹<http://cit-ec.de/en/content/cognitive-service-robotics-apartment-ambient-host>

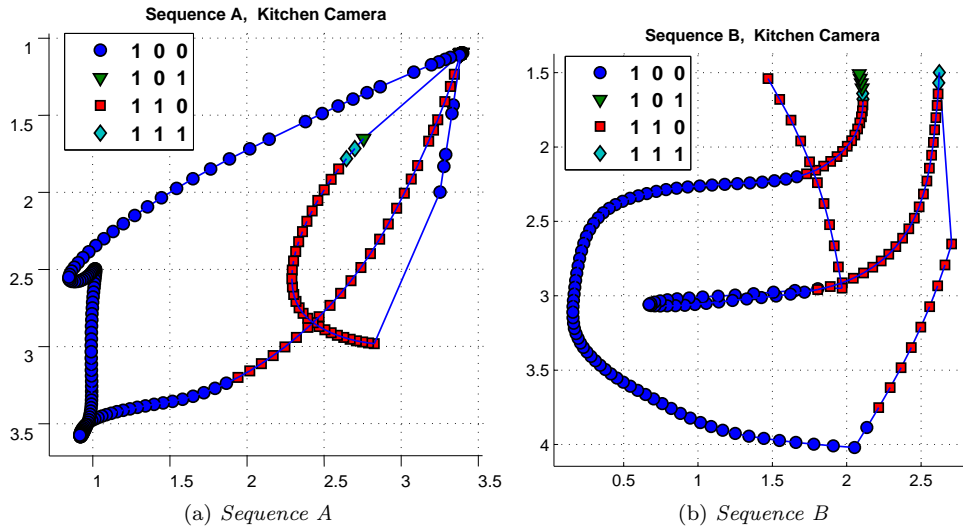


Figure 1: Top-down view of the person tracking coordinates from the kitchen camera. The symbols indicate which of the three cameras is tracking the person at that time point.

time step, until the sensor delivers values again. Next, every vector, where no change to its predecessor is observed, was removed from the sequence, in order to avoid many zero distances between distinct data points. To avoid strong discontinuities between subsequent vectors, we additionally smoothed² the time series by setting each vector entry $[\mathbf{x}_i]_d$ to the mean value of corresponding entries in a window of 10 predecessors and successors. (This translates to 1 second in the original time series, if no steps were removed from the sequence.) Lastly, we used a z-score transformation for the whole sequence to balance the numeric representation of all dimensions.

As an example, the person tracking results from the kitchen camera are displayed in a top-down³ plot in Figure 1. A line connects the points in the order of their temporal progression. The symbolic labels indicate whether each of the three cameras is tracking the person at that time: the binary code signifies the tracking status of the first camera (in the kitchen), the second (in the living room), and the third (in the entrance area), e.g. “1 0 1” means that the kitchen camera *is* tracking the person, while the camera in the living room is *not* tracking, and the entrance camera *is* tracking as well⁴.

Problem 1 – low-dimensional representation

Our first question is, whether we can obtain a reliable low-dimensional representation of the time series data, without a problem-specific preprocessing of the given sensor signals. Specifically, can we observe a rather smooth temporal trajectory in the embedding, despite the given heteroge-

²Note that this common preprocessing step is not problem-specific. However, it is necessary, since discontinuities in the data can lead to clustered low-D representations (in our case due to the pressure sensors in particular). This would yield an insufficient basis for training robust kernel mappings.

³Note that the tracking data is in fact 3D, however, the depth values follow a standard pattern of decreasing with the distance to the camera’s center, and are thus omitted in this top-down perspective.

⁴Since we display the particular field of view of the kitchen camera in Figure 1, the first bit of the points’ labels is always 1 in these plots.

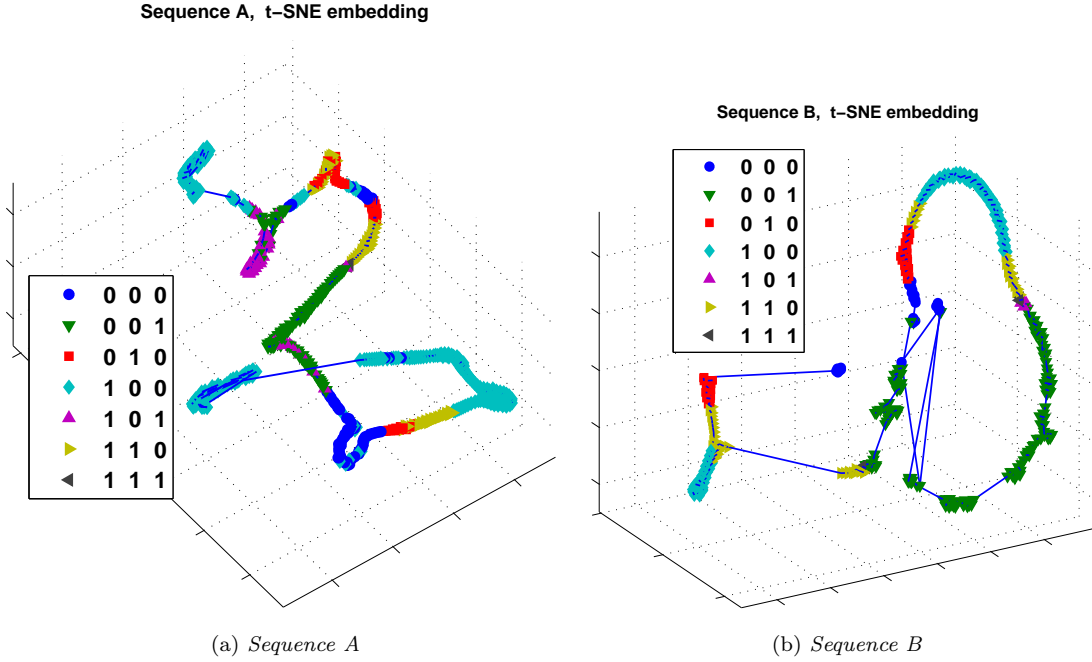


Figure 2: 3D embedding obtained by the t-SNE method for both 17-dimensional sequences. The colored symbols indicate which of the three cameras is tracking the person at that time. The temporal progression of the sequence is mostly captured by a smooth trajectory in 3D.

neous multi-dimensional sensor data? This relates mainly to challenge (II) from Section 1.

We applied the t-SNE method with a target dimensionality of $L = 3$, for both given sequences A and B. Our assumption is that a 3-dimensional embedding can represent the original data rather truthfully, since only a single person is moving through the spatio-temporal sensor system. Since t-SNE relies on a non-convex optimization starting from a random initialization, we ran the method several times and evaluated the reliability of every embedding in terms of the quality criterion Q_{NX} , as proposed in [9]. Additionally, we evaluated the quality index $Q_{NX}^{x_i}$ for every single point, as proposed in [10], which yields the ratio of preserved neighbors in a point's k -neighborhood. We chose the best respective 3D embedding in terms of this evaluation; the result for each sequence is presented in Figure 2.

In both embeddings, we can see that the time series is represented as a rather smooth trajectory, with only a few positions where the sequence is torn apart. The average quality $Q_{NX}^{x_i}$ over all points is 91.6% for A and 91.9% for B, in a $k = 10$ neighborhood. This substantiates our positive visual assessment. From these results, we can conclude, that DR can actually produce a fairly accurate 3D embedding of the 17-dimensional original sensor data, even without hand-crafted preprocessing steps. The low-D representation yields a homogeneous representation of heterogeneous sensor signals.

Problem 2 – explicit parametric mapping & inverse

Is it possible to obtain an explicit embedding function, which generalizes to unseen data, as well as an inverse mapping, which allows for an approximate reconstruction of the original sensor signals from the compressed low-dimensional representation? This relates mainly to challenge (I) from Section 1, since it would enable a compression and reconstruction of high-D sensor signals for more efficient storage and handling, without too much loss of information.

With the kernel t-SNE method, we now train a kernel mapping f based on the embeddings in Figure 2. To measure the discrepancy between the original embedding and the result of our trained mapping function, we can simply calculate the mean squared error (MSE) between both sets of vectors. We evaluate the generalization ability by averaging the training and test errors for random splits of the data over a 5-fold cross-validation with 10 repeats. After meta-parameters have been adjusted to the given data, an acceptable average error for both embedded sequences was observed, although with rather high standard deviations (std.): For Sequence A, we achieved a mean MSE of 9.5 (0.5 std.) on the training, and 31.1 (11.2 std.) on the test set. For Sequence B, the errors were generally lower with a mean MSE of 1.4 (1.2 std.) on the training, and 10.4 (2.0 std.) on the test set. From the experiment, we observed that the kernel mapping was often not able to capture the few discontinuous parts of the embedded trajectory. The other parts were reproduced more truthfully by the learned mapping. However, the training behavior in general tended to overfit. This may be due to the fact that the input data contains many degrees of freedom which are mapped to considerably less variables in the output.

Instead, training an inverse mapping function g was much easier in the experiments. We again used a repeated 5-fold cross-validation, which resulted in very low errors in general. The mean MSE for Sequence A was 0.007 (0.001 std.) on the training, and 0.017 (0.005 std.) on the test set. For Sequence B, we achieved 0.037 (0.006 std.) MSE on the training, and 0.741 (0.883 std.) on the test set. We can exemplify these very good results by visualizing the respective reconstructions of the person tracking data from Figure 1: In the Figure 3, we show the original person tracking from the kitchen camera, together with its reconstruction via the inverse mapping function for both sequences. In each case, the original trajectory is resembled rather smoothly, which verifies our quantitative evaluation showing very high performance on the test set.

We can conclude that training an accurate forward mapping function $f(\mathbf{x}_i)$ is harder than training the inverse mapping $g(\mathbf{y}_i)$. Therefore, the idea of using a DR technique for compressing sensor signals can be a viable option to handle smart home data efficiently, since we are able to reconstruct the approximate original signals via a trained kernel mapping $g(\mathbf{y}_i)$. However, the idea to use a kernel mapping $f(\mathbf{x})$ to extend the DR embedding to unseen data seems less robust with the techniques we applied, and needs further refinement.

Problem 3 – robustness against sensor failure

Can we utilize the learned embedding $f(\mathbf{x})$ and reconstruction function $g(\mathbf{y}_i)$ to fill missing sensor readings with plausible values? This relates to challenge (III) from Section 1, since it would signify that the compressed representation is robust in a dynamically changing technical environment.

Therefore, we chose an arbitrary 30% time window in Sequence B, and manipulated the original data by filling one dimension with erroneous values. We replaced the y-coordinate of the entrance camera in that time frame with values that simulate a failing sensor: small Gaussian noise around the average output of the y-coordinate over the given sequence. Figure 4a shows the manipulated camera data. We then used the previously trained mapping f to embed these

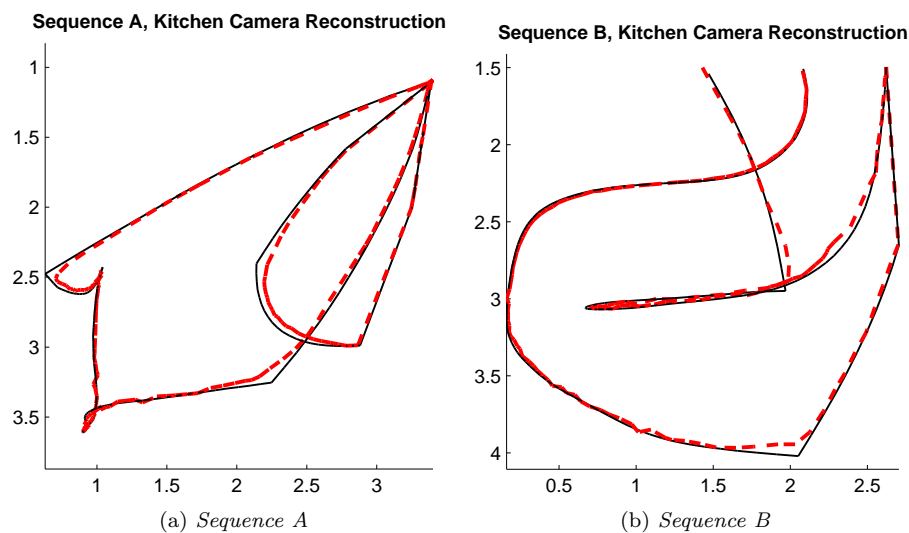


Figure 3: The original person tracking data from the kitchen camera (the thin black line), together with the reconstruction (the red dashed line) based on a 3D embedding of the respective 17-dimensional sequence.

erroneous data vectors in 3D, and applied the trained inverse function g to reconstruct the original sensor signals. The result showed the interesting effect of reconstructing the original trajectory nearly perfectly, see Figure 4b. Hence, the trained kernel mappings seem robust against local failures of single sensors, and can help to fill missing values in dynamic sensor environments.

Conclusion

Our preliminary results show, that modern nonlinear DR techniques offer promising features to tackle the challenges of smart home sensor data. Trained kernel mappings are able to capture the characteristics of our example data and did generalize well. Our ongoing research will include a comparative study with other DR methods (e.g. sparse coding).

References

- [1] R. Calderbank, S. Jafarpour, and R. Schapire. Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain. Technical report, 2009.
- [2] E. Catterall, K. Van Laerhoven, and M. Strohbach. Self-organization in ad hoc sensor networks: An empirical study. In *Proceedings of the Eighth International Conference on Artificial Life, ICAL 2003*, pages 260–263, Cambridge, MA, USA, 2003. MIT Press.
- [3] A. Gisbrecht, B. Hammer, B. Mokbel, and A. Sczyrba. Nonlinear dimensionality reduction for cluster identification in metagenomic samples. In *IV 2013*, pages 174–179, 2013.
- [4] A. Gisbrecht, A. Schulz, and B. Hammer. Parametric nonlinear dimensionality reduction using kernel t-sne. *Neurocomputing*, 147:71–82, 2015.
- [5] O. C. Jenkins and M. J. Matarić. A spatio-temporal extension to isomap nonlinear dimension reduction. In *Proceedings of the Twenty-first International Conference on Machine Learning, ICML '04*, pages 56–, New York, NY, USA, 2004. ACM.

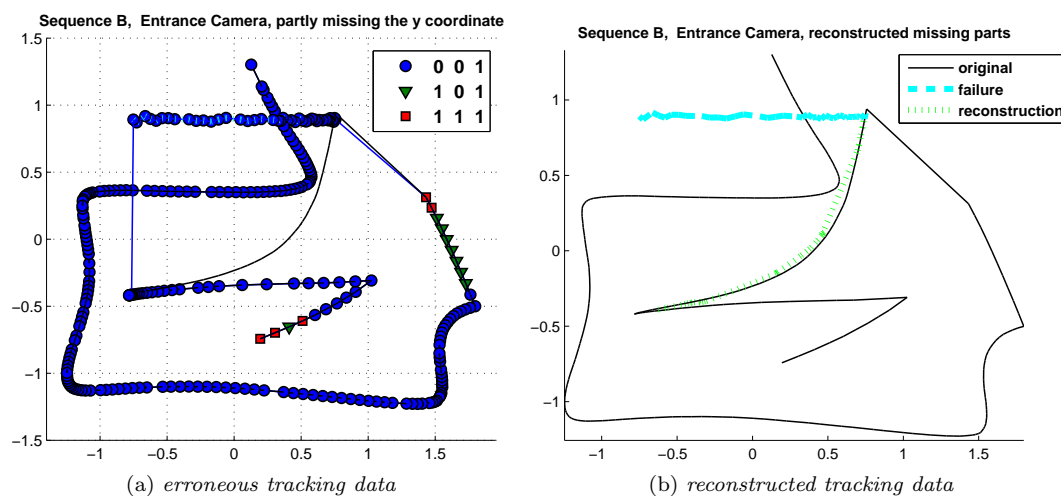


Figure 4: Top-down plots of the person tracking data from the entrance camera in Sequence B, where parts of the y-coordinate were replaced by its mean value to simulate a sensor failure. On the left, the manipulated input data is shown, where the noisy/failing part is highlighted in cyan and the original trajectory is the black line. On the right, the kernel mapping was used to successfully reconstruct the original tracking data, where the reconstruction is drawn in green.

- [6] J. A. Lee, D. H. Peluffo-Ordóñez, and M. Verleysen. Multiscale stochastic neighbor embedding: Towards parameter-free dimensionality reduction. In *ESANN 2014*, 2014.
- [7] J. A. Lee, E. Renard, G. Bernard, P. Dupont, and M. Verleysen. Type 1 and 2 mixtures of Kullback-Leibler divergences as cost functions in dimensionality reduction based on similarity preservation. *Neurocomputing*, 112:92–108, July 2013.
- [8] J. A. Lee and M. Verleysen. *Nonlinear dimensionality reduction*. Springer, 2007.
- [9] J. A. Lee and M. Verleysen. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomput.*, 72(7-9):1431–1443, 2009.
- [10] B. Mokbel, W. Lueks, A. Gisbrecht, and B. Hammer. Visualizing the quality of dimensionality reduction. *Neurocomputing*, 112:109–123, 2013.
- [11] B. Mokbel and A. Schulz. Towards dimensionality reduction for smart home sensor data. In T. Villmann and F.-M. Schleif, editors, *Machine Learning Reports 03/2015 – Proceedings of the Workshop New Challenges in Neural Computation (NC²)*, number MLR-03-2015 in Machine Learning Reports, pages 41–48, 2015. ISSN:1865-3960.
- [12] L. van der Maaten. Accelerating t-SNE using tree-based algorithms. *Journal of Machine Learning Research*, 15:3221–3245, 2014.
- [13] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, November 2008.
- [14] Y. Zhu, E. Song, J. Zhou, and Z. You. Optimal dimensionality reduction of sensor data in multisensor estimation fusion. *Signal Processing, IEEE Transactions on*, 53(5):1631–1639, May 2005.