

Acquisition and Application of a Tactile Database

Matthias Schöpfer, Helge Ritter and Gunther Heidemann

Abstract—We present a database of 2D pressure profile timeseries as a testbed for tactile object and surface recognition. The tactile database captures the surfaces of household and toy objects by moving a 2D pressure sensor mounted to an industrial robot arm around the objects using real-time trajectory calculation. Thus, it represents different “views” of the objects in a similar way as the well known Columbia Object Image Library (COIL) captures different views of an object by a camera. As a first application, objects in the database are classified using a neural network architecture.

Index Terms—Robot tactile systems, Tactile systems (non-biological), Pattern Recognition, Image databases, Neural Networks

I. INTRODUCTION

Tactile sensing is a key discipline for exploration and grasping with autonomous robot hands. Humans can grasp and manipulate objects mostly without looking, guided only by haptics. But to date, there is no technical equivalent to human skin, in spite of research on different sensor designs [1], [2], [3], [4], [5], [6], [7]. So we still lack the “CCD-Chip” for tactile sensing.

But the hardware problem is only part of the story. Even though suitable pressure sensors gradually become available, tactile recognition is still a rare research subject. Why is that so? One reason is that recognition of shape and surface structure *without the help of the visual modality* is a challenging pattern recognition task. However, the same is true e.g. for much better researched field of computer vision.

In the opinion of the authors, a major obstacle in the way towards tactile pattern recognition is the complete lack of any standardized databases for realistic benchmarking of new algorithms. While in computer vision research databases such as the Columbia Object Image Library (COIL) [8] or VisTex [9] are widely established benchmarks, there is no such database for tactile sensing, because the gathering of tactile data is much more difficult than the acquisition of pictures, as will become clear in the following sections.

In this paper, we present a database of tactile pressure profile timeseries which can be regarded as the “tactile equivalent” to COIL [10], [8]. COIL is a collection of images from 100 household and toy objects, where each object is represented by 72 different views, taken at angles 0, 5, 10 ... degrees while the object rotates on a turntable. The database thus allows testing recognition algorithms

with various training and test sets for evaluation of view sensitivity and generalization.

How can a “tactile COIL” be designed? First, we will have to decide on the sensor. If we want to acquire tactile data in a way similar to human sensing, a 2D sensor as a “fingertip” is required that is moved smoothly over the surface of an object. Basically, there are two ways to do this: The fingertip can either slide over the object, or “roll” without sliding. Since the real time control of sliding is highly difficult, we will roll the sensor over the object. Second, it has to be defined what we mean by a “view” of the object. While a view of an object is defined by the camera position relative to the object for image acquisition, there are much more degrees of freedom for the active acquisition of tactile data. Not only can the object be presented to the sensor in different positions and poses, but also the trajectory by which the sensor rolls over the object surface can vary. Therefore, the representation of each object in the database is characterized by two different sets of parameters: Object pose parameters and trajectory parameters.

In this paper, we present a complete setup of a robot mounted tactile sensor, which is used to record haptic data (i.e. tactile and kinesthetic) from a set of 16 different small household and toy objects. Further, we describe a first application of the database: A neural recognition architecture which combines feature extraction by a local PCA approach with subsequent classification is used to analyze the high-dimensional tactile data for common patterns. We show, that classification of the object postures is possible and analyse the ability to generalize over different features. The reason for picking this application is not so much the demonstration of a certain use or practical application, but rather to show that the gathered data are valid, feasible and usable.

A. Related Work

The most active area in tactile sensing still seems to be the design and construction of new sensors [11]. Motoo, Arai and Yamada recently proposed a novel piezoelectric fingertip sensor [12]. Kim et. al. present a low cost 3 component tactile device array [13]. The soft tissue structure of the human finger for texture sensing is imitated in a design by Mukaibo et. al. [14]. Murakami and Hasegawa use a rather conventional approach as they utilize a 6 DOF force torque sensor with a soft fingertip to detect edges and their direction [15]. All of these papers have in common, that they include some more or less extensive experimental part. Fewer papers or articles cope with algorithms on tactile data [11]. Platt, Fagg and Gruben propose a control basis for force-based interaction [16]. Several papers deal with the

This work was funded by the Deutsche Forschungsgesellschaft (DFG)
Helge Ritter and Matthias Schöpfer are with Faculty of Technology,
Neuroinformatics Group, Bielefeld University, 33615 Bielefeld, Germany.
{helge, mschoepf}@techfak.uni-bielefeld.de
Gunther Heidemann is with Intelligent Systems Group, Stuttgart Univer-
sity, D-70569 Stuttgart, Germany. .ais@vis.uni-stuttgart.de



Fig. 1. On the left the sensor panel with RS-232 and power cables. An ordinary glass as seen on picture produces a pressure profile as can be seen on the right

shape reconstruction from tactile images [17], [18], [19]. Zhang and Chen use tactile sensing to control a robot trajectory [20], while Melchiori detects the slippage of objects via tactile sensing [21]. Once again, every new algorithm working on tactile data has to prove its performance in an experiment. While the developers of novel sensors will have to prove the performance of the sensor in experiments, the developers of algorithms on tactile data could easily resort to a tactile database. This would not only be a great timesaver, but would also allow the comparison and optimization of algorithms. The pendant to our approach in Computer Vision would be, as mentioned, the Columbia Object Image Library (COIL) [8] and VisTex [9].

II. DATABASE DEVELOPMENT

A. Hardware

The sensor (Fig. 1) is a low-cost tactile sensor with a resolution of 16×16 Texels. It has a frame rate of about 12 Hz, which was artificially throttled to 10 Hz to ensure an uninterrupted data flow. The spatial resolution is 6 mm, but due to the sensor design, there are no measurement gaps. The measurement range of a single Texel ranges from 4 to 120 kPa, with dynamics of 12 bit.

To archive high quality and reproducible data sets, the sensor was mounted on a Unimation 6-DOF PUMA 200 robot arm (see Fig. 2). The robot is controlled under RCCL, the Robot Control C Library [22]. Its abilities for real time trajectory control made the programming part possible. The Robot control was performed on a Sun Ultra 5 running Solaris 2.6.

B. Acquisition

The goal was to gather as much data per object as possible, so every possible stable posture of an object was taken into account. The basic sensing procedure consists of rolling the sensor over the object in such a way, that a certain contact and normal force is always maintained (compare Fig. 3). While the trajectory is straight in the x plane of the robots end effector, the sensor is rotated around the y

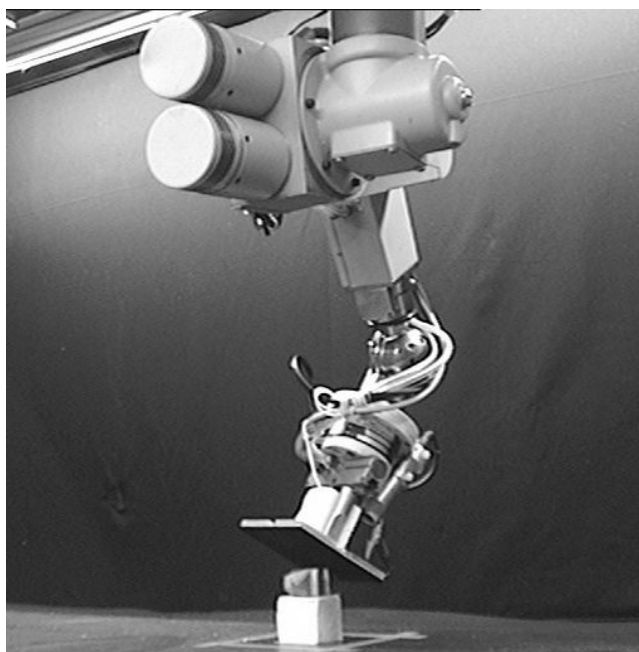


Fig. 2. The PUMA 200 robot with tactile sensor mounted while probing an object.

axis (starting at -30° and stopping at 30°) and the z values varied dynamically depending on the structure of the object. The axis of rotation varied depending on the tactile data, but was parallel to the y axis of the robots end effector.

The first point of contact in the direction of the rotation (rotating about y means we consider the point with the highest x value) determines the center of rotation. The algorithm also does some adaption to too low or too high pressures, as the correct center of rotation is erroneous due to the sparse resolution of the sensor.

This procedure is repeated for four different roll angles (RA) ($0^\circ, 45^\circ, 90^\circ, 135^\circ$). It is assumed, that probing the object from an angle of 180° would not yield any different results than those from the run at 0° , besides jittering effects and minimal translation. Therefore, the procedure was instead repeated at different starting points with translational offsets (TO) of (0mm,0mm), (3mm,0mm), (0mm,3mm) and (3mm,3mm). This ensures that the gathered data reflects jitter and translational effects beyond the spatial resolution of the sensor.

To simulate the rotation of the object without having to move the object manually, a starting offset rotation (OR) around the z axis of $0^\circ..80^\circ$ with a stepping of 10° was applied.

All objects were smaller in size than the sensor area. Otherwise we would have needed a much more complex algorithm to explore the object, which would have also had an impact on the data, because the datasets would have become quite diverse from one another.

C. Database structure

To assure the compatibility to different database management systems and for an optimal performance, the database was implemented in ANSI SQL. The management system used was MySQL.

The database (Fig. 5) consists of three tables. The object_tbl holds general information about the objects that were probed. Every pose of an object is treated as a unique object in the sense of this database (and therefore has a unique id). Besides a description of the object, also a picture of the object was added, to avoid later confusion. The tactile images, that were taken at a rate of 10 Hz (due to hardware limitations), were kept in the tactile_data.tbl. Besides the tactile frame, which was for space optimization compressed, a pointer to the probed object, a sensor and a system timestamp and the kind of run, i.e. rotational and translational offsets and probing angle, were also saved.

The kinesthetic data from the robot arm was sampled at 100 Hz. Since the robot control and the retrieval of the sensor data was done by a single CPU Sun Ultra 5 Workstation, no real concurrency could occur. A timestamp was taken, so the tactile frames could be associated to a single, temporally close, kinesthetic frame. To keep the amount of data manageable, only this frame was saved to the database. Since the trajectory of the robot was only updated at the speed of the tactile sensor, the kinesthetic frames between two tactile frames would not have yielded any crucial information.

The kinesthetic frames were stored in the t6_trsf_table. They consist of a pointer to the probed object, the id of the run, a timestamp and the 4×3 T_6 -Tool-Matrix, i.e. the homogeneous transformation from the robots base coordinate system to its tool coordinate system.

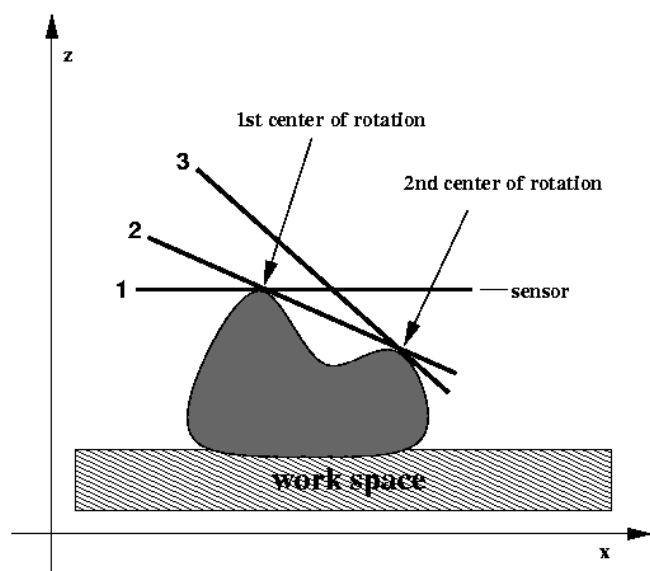


Fig. 3. The basic sensor trajectory algorithm displayed in a side view. Adjusting the center of rotations to the objects shape ensures good coverage of surface features while keeping the applied pressure in decent ranges.

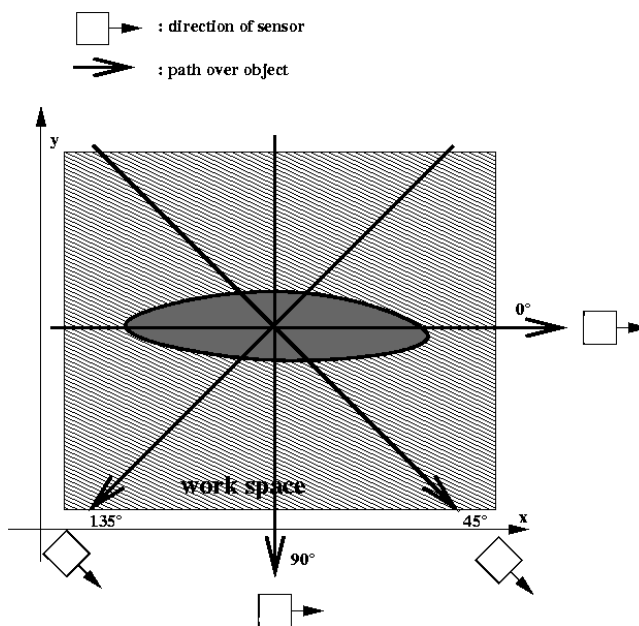


Fig. 4. Schematic view of the probing process. A top view of the sensor paths at different angles is shown.

D. Database summary

A total of 16 different objects (pencil sharpener, toy cars, etc.), all in size and shape ranging between 1 cm and 8 cm, were used for the database. These objects provided 44 different stable positions, which were probed. One basic probing took about 220 seconds, due to the low frequency of the tactile sensor. The database currently holds a total of over 14 million haptic frames, which sums up to about 5.8 GB of compressed data.

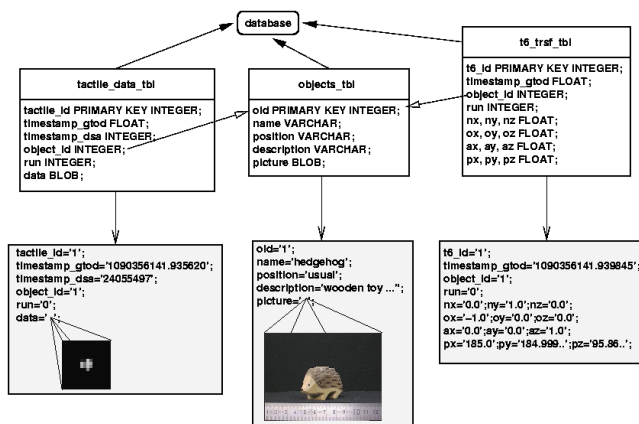


Fig. 5. Schematic view of the structure of the database. The database consists of three tables: object_tbl keeps general informations about the object, tactile_data.tbl is used to store all tactile images while t6_trsf.tbl keeps all kinesthetic data from the robots internal joint encoders. The last two tables keep references on the object they belong to.

III. EXPERIMENTAL APPLICATION

The applications for a tactile database are versatile: the off-line work on algorithms on tactile data (e.g. slip detection or shape reconstruction), classification of tactile and haptic features as well as the feature extraction on tactile datasets and maybe even explorative strategies on the dataset. As a first application on this tactile database, an object (or rather object and posture) recognition with an universal classification architecture, which main appliance is the field of computer vision [23], [24], is tried. The approach is motivated in [25], for the current paper, we will give only a brief introduction. The goal of the classification task is not only to prove that classification on tactile data is possible but also to get an idea of the complexity of the dataset. In particular we are interested in the generalization properties for different types of parameters.

The architecture is a three stage processing scheme employing three different types of Neural Networks. It can be trained to map the raw sensor input, i.e., a time series of 2D pressure profiles, onto discrete classes (object plus pose). Since the “gap” between the raw signal and the symbolic level is particularly large for haptic data, the neural architecture has to be trained in three passes: The first layer is trained unsupervised, it performs a vector quantization of the raw data to find its main clusters. The subsequently trained second layer implements a local principal component analysis (local PCA). Thus, the adapted first and second layer realize a feature extraction from the sensor signal. By contrast, the third layer is trained supervised to the classification of these features. The main advantage of this architecture is its suitability for very high dimensional and complex data, as has been shown in several earlier applications both of the isolated stages [26] and as a whole [23], [24].

The tactile datasets, which also can be seen as time series of different length, are cut to a common length. This is no issue, because the recording of the time series starts *before* any contact is made. So the frames, that are being omitted are all blank frames. Since we are looking at a time series, consecutive frames tend to be very similar. That is why a 22:1 temporal sub-sampling could be justified to reduce the dimensionality of the input vector. Still the dimensionality remained of high order: 16×16 (image size) \times 100 (frames) = 25600.

A. Overview of the VPL Architecture

Rather than, as the case may be with the use of world knowledge, “designing” a feature extraction from the tactile data, a three-stage neural classifier is applied (Fig. 6). The strength of this architecture lies in its good performance with high-dimensional input as well as the automatic feature extraction. The latter is based on local PCA [27], [28]. The actual classification is then performed by several neural networks.

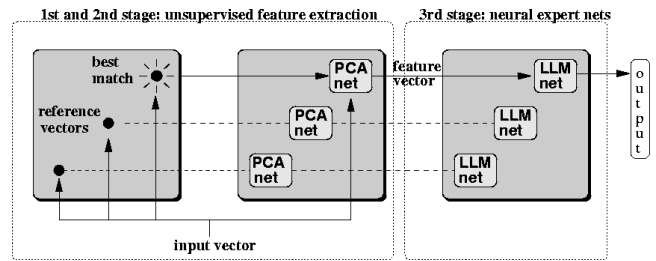


Fig. 6. The three stage VPL classifier. The first two stages embed a vector quantization and a principal component analysis to extract features unsupervised. The last stage performs the classification through neural expert nets in a supervised way.

From a conceptual point of view, the classification system maps an input vector $x \in \mathbb{R}^n$ on the output vector $\vec{y} \in \mathbb{R}^m$: $\vec{x} \rightarrow \vec{y}$. To use the system for classification, choose $m = \#_of_classes$. Using an m -dimensional output vector is useful to avoid the introduction from artificial neighborhoods of classes. To train the system, use pairs $(\vec{x}_i^{Tr}, \vec{y}_i^{Tr})$, where $(\vec{y}_i^{Tr})_j = \delta_{jc}$ for $j = 1..m$ and c being the correct class number. After the training is complete, the system will yield to an input \vec{x} the class c' via $c' = \arg \max_j (\vec{y}(\vec{x}))_j$.

The following sections briefly describe the specific stages of the VPL-architecture.

B. Stage 1: Partitioning

In the first stage, the input space is partitioned using vector quantization (VQ) (see e.g. [29]). The number of reference vectors N_{VQ} , which are positioned in the input space by *Activity Equalization VQ* (AEV) [26] is one parameter which has influence on the performance of the classifier. The use of the AEV algorithm, which, in short, avoids the problem of codeword under-utilization in sparsely filled input space [30], i.e. the problem that in a high dimensional space many reference vectors are likely to remain outliers, render the system very well behaved in this parameter. During training, the input space is completely partitioned before the training of stage 2 is performed.

During classification, in the first stage the best matching reference vector $\vec{r}_{n^*(\vec{x})}$ for an input \vec{x} which minimizes $\|\vec{x} - \vec{r}_i\|, i = 1 \dots N_{VQ}$ is selected. The path for the further processing is chosen in this step.

C. Stage 2: Local PCA

The second stage performs a PCA for each of the training samples \vec{x} . A separate PCA is done for each Voronoi tessellation cell of the reference vectors \vec{r}_i , which means, N_{VQ} PCAs are computed. The principal components (PCs) are calculated with a single layer feed forward neural network using the training rule proposed by Sanger [31]. This iterative method is preferred over direct computation of the eigenvectors, which is at least impractical, if not impossible, for a large dimensionality. After the training is done, the weight vectors $\vec{w}_{ij} \in \mathbb{R}^d, i = 1 \dots N_{VQ}, j = 1 \dots N_{PC}$ of the neural nets stand for the local PCs. After the training, an input vector \vec{x} is projected on the weights $\vec{w}_{n^*(\vec{x}),j}$ of net

number $n^*(\vec{x})$: $\vec{x} \rightarrow \vec{p}_{n^*(\vec{x})}(\vec{x})$. The dimension reduction is now performed, in that in the following only the first N_{PC} dimensions will be considered, i.e. $\vec{p} \in \mathbf{R}^{N_{PC}}$.

After the first two stages, the unsupervised feature extraction is complete.

D. Stage 3: Classification

In this stage, supervised learning is involved. The feature vector, which is extracted by the first two stages is now fed into neural classifiers of type *Local Linear Map* (LLM), which are related to self-organizing maps [32] and the GRBF-approach [33]. For details on LLMs, see [34]. In short, the LLM-Nets approximate nonlinear functions through N_{LLM} different linear mappings.

Each local PCA has its own local expert net associated to it. To obtain the final output for \vec{x} , after finding the best match reference vector number $n^*(\vec{x})$ and the projection $\vec{p}_{n^*(\vec{x})}(\vec{x})$, the LLM number $n^*(\vec{x})$ maps $\vec{p}_{n^*(\vec{x})}(\vec{x}) \rightarrow \vec{y}$.

E. Discussion of the VPL architecture

In contrast to normal PCA [35], which is the optimal linear method to cover the variance of a data distribution, local PCA can be seen as a nonlinear extension. This allows a much better approximation of nonlinear data and clustered distributions [27], [28]. Since Local PCA provides efficient and unsupervised feature extraction, it is used in this architecture. It has been successfully applied in a number of applications for pattern recognition [36], [37], [38] and dimension reduction [39].

Besides the presented approach to calculate the local PCA representation, other methods produce improved results through repeated iterations of clustering and PCA to minimize the least square reconstruction error [39]. Even further advanced is the probabilistic method of [40]. But the here used decoupled approach has the advantage of being computational efficient, which is of more importance in this application. Moreover, the reconstruction accuracy can be improved by increasing N_{VQ} .

With the separation into isolated VQ and PCA, there is also no need for extensive parameter tuning. Both algorithms are well known and can be treated as “black boxes”. The remaining parameters are N_{VQ} , N_{PC} and N_{LLM} , which have an impact on the system performance. If they are chosen to small, it may result in poor classification results. But as previous work has shown [24], the finding of suitable parameters is an easy task, as the classification performance rises smoothly up to a saturation point with all three parameters. Overfitting phenomena, as known from different neural approaches, were not observed.

F. Results

Several experiments with different training and test datasets were conducted. Each VPL was only used for testing, if it reached a correct classification rate of over 98% on the training dataset. The results were averaged over 10 random partitionings of training/test data. The results are listed in table I.

TABLE I
CLASSIFICATION RESULTS

training set	test set	rate on test set
random Roll Angle (RA) & Offset Rotation (OR) pair, Translational Offset (TO)=(0mm,0mm)	RA & OR as in training, TO=(3mm,0mm)	29.5%
random RA & OR pair, TO=(0mm,0mm)	RA & TO as in training, random OR!=training	69.9%
random RA & OR pair, TO={(0mm,0mm), (3mm,0mm), (0mm,3mm)}	RA & OR as in training, TO=(3mm,3mm)	97.2%
OR = {x, x + 20°}, random RA & TO	RA & TO as in training, OR={x + 10°}	84.1%
OR={x, x + 10°, x + 30°, x + 40°}, random RA & TO	RA & TO as in training, OR={x + 20°}	89.2%
OR={x, x + 10°, x + 30°, x + 40°}, random RA, TO={(0mm,0mm), (3mm,0mm), (0mm,3mm)}	RA as in training, OR={x + 20°}, TO=(3mm,3mm)	91.9%
random OR & TO, RA=0°	OR & TO as in training, RA=45°	6.0%

G. Discussion of the Classification Results

First of all, it should be stated that the classification results are quite satisfying, taking the coarse resolution of the sensor in account. But we are not yet a real challenge to the human tactile sense, which is said to work at a about 96% to 99% correct classification [41].

The system seems to be sensitive to translation, if it is not explicitly trained on it. The location of contact points appear to yield much information. If translation is explicitly trained, the results are with over 97 % extremely good. In contrast to translation, different offset rotations generalize at least somewhat. And once again, if explicitly trained on different offset rotations, the results improve. It is also possible, to train the system simultaneous on both, on offset rotations and on translation. As one could expect, there is no generalization for the roll angle.

IV. CONCLUSIONS AND FUTURE PROSPECTS

In this work, we have presented the construction of an extensive haptic database. A robot mounted tactile sensor was used to gather tactile and kinesthetic data on 16 different objects and 44 different positions. An algorithm for the detailed probing of the objects was presented. The database can be used to evaluate different algorithms on haptic data, as it provides a common set for comparison between different approaches. Scientist are freed from the time consuming and unpleasant acquisition of suitable datasets for further research.

As a first application, the classification of tactile data with the VPL-architecture was used. The results of this appliance enable the drawing of several conclusions about the nature of the data.

For the future, the evaluation of different algorithms, which might then take advantage of the kinesthetic data, as well as algorithms, who are (explicitly) aware of the fact,

that the gathered data is a time series sounds promising. Also, explorative algorithms could be investigated on the grid data provided by the database.

ACKNOWLEDGMENT

This work was funded by the Deutsche Forschungsgemeinschaft (DFG).

REFERENCES

- [1] H. R. Nicholls, "Tactile Sensor Designs," in *Advanced tactile sensing for robotics*, H. R. Nicholls, Ed. PO Box 128, Farrer Road, Singapore 9128: World Scientific Publishing Co. Pte. Ltd., 1992, ch. 2, pp. 13–48.
- [2] D. Johnston, P. Zhang, J. Hollerbach, and S. Jacobsen, "A Full Tactile Sensing Suite for Dextrous Robot Hands and Use in Contact Force Control," in *Proc. IEEE Int'l Conf. Robotics and Automation ICRA 1996*. IEEE, 1996, pp. 3222–3227.
- [3] J. Jockusch, J. Walter, and H. Ritter, "A Tactile Sensor System for a Three-Fingered Robot Manipulator," in *Proc. IEEE Int'l Conf. Robotics and Automation ICRA 1997*. IEEE, 1997, pp. 3080–3086.
- [4] P. Schmidt, E. Maël, and R. P. Würtz, "A novel sensor for dynamic tactile information," 2004, in Revision, To appear in *Robotics and Autonomous Systems*.
- [5] D. Taddeucci, C. Laschi, R. Lazzarini, R. Magni, P. Dario, and A. Starita, "An Approach to Integrated Tactile Perception," in *Proc. IEEE Int'l Conf. Robotics and Automation ICRA 1997*. IEEE, 1997, pp. 3100–3105.
- [6] K. Kamiyama, H. Kajimoto, N. Kawakami, and S. Tachi, "Evaluation of a Vision-based Tactile Sensor," in *Proc. IEEE Int'l Conf. Robotics and Automation ICRA 2004*. IEEE, 2004, pp. 1542–1547.
- [7] A. E. Schultz, J. H. Solomon, M. A. Peshkin, and M. J. Hartmann, "Multifunctional whisker arrays for distance detection, terrain mapping, and object feature extraction," in *Proc. IEEE Int'l Conf. on Robotics and Automation, Barcelona, Spain, April 2005*, 2005, pp. 2599–2604.
- [8] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia Object Image Library: COIL-100," Dept. Computer Science, Columbia Univ., Tech. Rep. CUCS-006-96, 1996.
- [9] "Vision Texture Database (VisTex) by R. Picard, C. Graczyk, S. Mann, J. Wachman, L. Picard and L. Campbell," 1995, copyright 1995 by the Massachusetts Institute of Technology. See <http://www-white.media.mit.edu/vismod/imagery/VisionTexture/vistex.html>.
- [10] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia Object Image Library (COIL-20)," Dept. Computer Science, Columbia Univ. New York, N.Y. 10027, Tech. Rep. CUCUS-006-96, 1996.
- [11] M. H. Lee, "Tactile sensing: New directions, new challenges," *International Journal of Robotics Research*, no. Vol. 19, No. 7, 2000.
- [12] K. Motoo, F. Arai, and Y. Yamada, "Novel force sensor using vibrating piezoelectric element," in *Proc. IEEE Int'l Conf. on Robotics and Automation, Barcelona, Spain, April 2005*, 2005, pp. 2593–2598.
- [13] J.-H. Kim, J.-I. Lee, H.-J. Lee, Y.-K. Park, M.-S. Kim, and D.-I. Kang, "Design of flexible tactile sensor based on three-component force and its fabrication," in *Proc. IEEE Int'l Conf. on Robotics and Automation, Barcelona, Spain, April 2005*, 2005, pp. 2589–2592.
- [14] Y. Mukaibo, H. Shirado, M. Konyo, and T. Maeno, "Development of a texture sensor emulating the tissue structure and perceptual mechanism of human fingers," in *Proc. IEEE Int'l Conf. on Robotics and Automation, Barcelona, Spain, April 2005*, 2005, pp. 2576–2581.
- [15] K. Murakami and T. Hasegawa, "Tactile sensing of edge direction of an object with a soft fingertip contact," in *Proc. IEEE Int'l Conf. on Robotics and Automation, Barcelona, Spain, April 2005*, 2005, pp. 2582–2588.
- [16] R. P. Jr., A. H. Fagg, and R. A. Grupen, "Manipulation Gaits: Sequences of Grasp Control Tasks," in *Proc. IEEE Int'l Conf. Robotics and Automation ICRA 2004*. IEEE, 2004, pp. 801–806.
- [17] M. Moll and M. A. Erdmann, "Dynamic shape reconstruction using tactile sensors," in *ICRA-02*, 2002, pp. 1636–1641.
- [18] N. Chen, R. Rink, and H. Zhang, "Local object shape from tactile sensing," 1996. [Online]. Available: citeseer.nj.nec.com/chen96local.html
- [19] M. Skubic and R. A. Volz, "Identifying single ended contact formations from force sensor patterns," *IEEE Transactions on Robotics and Automation*, no. Vol.16, No. 5, 2000.
- [20] H. Zhang and N. N. Chen, "Control of contact via tactile sensing," *IEEE Transactions on Robotics and Automation*, no. Vol.16, No. 5, 2000.
- [21] C. Melchiorri, "Slip Detection and Control Using Tactile and Force Sensors," *IEEE/ASME Trans. on Mechatronics*, vol. 5, no. 3, pp. 235–243, 2000.
- [22] J. Lloyd, M. Parker, and R. McClain, "Extending the RCCL Programming Environment to Multiple Robots and Processors," in *Proc. IEEE Conference on Robotics and Automation*. Philadelphia, PA: IEEE, 1988, pp. 465–469.
- [23] G. Heidemann and H. Ritter, "Visual Checking of Grasping Positions of a Three-Fingered Robot Hand," in *Proc. ICANN 2001*, G. Dorffner, H. Bischof, and K. Hornik, Eds. Springer-Verlag, 2001, pp. 891–898.
- [24] ———, "Learning to Recognise Objects and Situations to Control a Robot End-Effector," *KI Künstliche Intelligenz, special issue on Vision, Learning, Robotics*, vol. 2, pp. 24–29, 2003.
- [25] G. Heidemann and M. Schöpfer, "Dynamic Tactile Sensing for Object Identification," in *Proc. IEEE Int'l Conf. Robotics and Automation ICRA 2004*. New Orleans, USA: IEEE, 2004, pp. 813–818.
- [26] G. Heidemann and H. Ritter, "Efficient Vector Quantization Using the WTA-rule with Activity Equalization," *Neural Processing Letters*, vol. 13, no. 1, pp. 17–30, 2001.
- [27] N. Kambhatla and T. K. Leen, "Fast Non-Linear Dimension Reduction," in *Advances in Neural Information Processing Systems 1993*, J. Cowan, G. Tesauro, and J. Alsppector, Eds., vol. 6. Morgan Kaufmann Publishers, 1994, pp. 152–159.
- [28] C. Bregler and S. M. Omohundro, "Surface Learning with Applications to Lipreading," in *Advances in Neural Information Processing Systems 1993*, J. Cowan, G. Tesauro, and J. Alsppector, Eds., vol. 6. Morgan-Kaufmann Publishers, 1994, pp. 43–50.
- [29] J. Buhmann and H. Kühnel, "Vector quantization with complexity costs," *IEEE Trans. on Information Theory*, vol. 39, no. 4, pp. 1133–1145, 1993.
- [30] S. Grossberg, "Competitive learning: From interactive activation to adaptive resonance," *Cognitive Sci.*, vol. 11, pp. 23–63, 1987.
- [31] T. D. Sanger, "Optimal Unsupervised Learning in a Single-Layer Linear Feedforward Neural Network," *Neural Networks*, vol. 2, pp. 459–473, 1989.
- [32] T. Kohonen, *Self-Organizing Maps*. Springer Verlag, 1995.
- [33] J. Moody and C. Darken, "Learning with localized receptive fields," in *Proc. of the 1988 Connectionist Models Summer School*. San Mateo, CA: Morgan Kaufmann Publishers, 1988, pp. 133–143.
- [34] H. J. Ritter, T. M. Martinetz, and K. J. Schulten, *Neuronale Netze*. München: Addison-Wesley, 1992.
- [35] I. Jolliffe, *Principal Component Analysis*. New York: Springer Verlag, 1986.
- [36] T. Hastie, P. Simard, and E. Sackinger, "Learning Prototype Models For Tangent Distance," in *Advances in Neural Information Processing Systems 1994*, G. Tesauro, D. Touretzky, and T. Leen, Eds., vol. 7. MIT Press, 1995, pp. 999–1006.
- [37] G. E. Hinton, P. Dayan, and M. Revow, "Modelling the Manifolds of Images of Handwritten Digits," *IEEE Trans. on Neural Networks*, vol. 8, no. 1, pp. 65–74, 1997.
- [38] B. Moghaddam and A. Pentland, "Probabilistic Visual Learning for Object Representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 696–710, 1997.
- [39] N. Kambhatla and T. K. Leen, "Dimension reduction by local principal component analysis," *Neural Computation*, vol. 9, no. 7, pp. 1493–1516, 1997.
- [40] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol. 11, no. 2, pp. 443–482, 1999.
- [41] R. L. Klatzky, S. J. Lederman, and V. A. Metzger, "Identifying objects by touch. An "expert system"," *Perception & Psychophysics*, vol. 37, no. 4, pp. 299–302, 1985.