

University of Innsbruck

Institute of Computer Science
Intelligent and Interactive Systems

Evaluation of Tactile Sensors

Peter Kiechle

M.Sc. Thesis

Supervisor: Justus Piater
22nd October 2015
Revised version

Abstract

The Schunk SDH-2 dexterous robotic hand, in use at IIS, is equipped with tactile sensor arrays at the insides of the fingers. This project explores their practical use for the measurement of contact forces (calibration), the detection of the presence of a grasped object, and the recognition of object surface types (or even object identities) via tactile patterns using classification and regression techniques.

In particular, a framework to process and record tactile sensor data as well as temperature and hand kinematics has been developed. Based on this preliminary work, the effectiveness of various spatial and temporal signal conditioning techniques has been examined. As a consequence of mixed results, a calibration model using nonlinear least-squares regression has been realized to relate measured pressure profiles and temperatures to the applied force. As a side effect, spatio-temporal smoothing techniques and ordinary least-squares regression were combined to implement a robust, temperature-aware, high sensitivity mode for the SDH-2.

Tactile sensors can be used to minimize the required contact force in object manipulation tasks. To support this strategy, translational and rotational slip-detection algorithms have been investigated. Along the way, the possibility of surface classification based on frequency analysis was ruled out. The resulting framework can be used to effectively prevent grasped objects from slipping while keeping the applied force as low as possible.

In addition, a classical approach to forward kinematics for individual tactile sensor cells was realized. This information can be used to estimate the size and compressibility of a grasped object. Complementary to these grasp characteristics, two-dimensional shape features based on translation and rotation invariant discrete Chebyshev moments have been developed. Combining these methods, it was demonstrated that a wide range of different grasp-object configurations can reliably be classified using linear discriminant analysis and multi-class support vector machines.

Acknowledgements

I would like to express my gratitude to Univ.-Prof. Justus Piater, Ph.D. for giving me the opportunity to work on an interesting topic of robotic research. I appreciate not only his expertise and suggestions but also his great support and patience.

I also want to thank the members of the examination board, Univ.-Prof. Dr. habil. Matthias Harders and Emre Uğur, Ph.D..

This thesis would not have been possible without Simon Haller and Cornelia Vidovic. I would like to thank them for their constant helpfulness and immediate reactions.

Thanks to all the members of the IIS Team for providing a positive and constructive working environment, especially those who gave me valuable input in the lab. I cannot resist but to name them all, in no particular order: Antonio Rodríguez-Sánchez, Sándor Szedmák, Mirela Popa, Sebastian Stabinger, Simon Hangl, Alexander Rietzler, Hanchen Xiong, Senka Krivić, Dadhichi Shukla, Safoura Rezapour Lakani, Özgür Erkent, Philipp Zech, Heiko Hahn, Damien Teney and Thomas Hoyoux.

Big thanks also to my brother Martin for his commitment in proof-reading. Last but not least, I have to thank my parents Elisabeth and Willi Kiechle for the support they provided me through my entire life. I will be forever grateful for their efforts.

Contents

Abstract	i
Acknowledgements	iii
Contents	v
List of Figures	vii
List of Tables	xiii
Declaration	xv
1 Introduction	1
1.1 SCHUNK Dexterous Hand 2.0	2
1.2 Tactile sensors	2
1.3 Applications	5
1.4 Working principles	6
1.5 Kelvin-Voigt Model	8
2 Signal Processing / Filtering	11
2.1 Temporal Filtering	11
2.1.1 Convolution filters	12
2.1.2 Kalman filter	12
2.2 Savitzky-Golay filter	15
2.3 Spatial Filtering	16
2.4 Ghosting	17
3 Calibration	19
3.1 Model fitting	22
3.2 Temperature	27
3.3 Pressure calibration set-up	28
3.4 High sensitivity mode	32
4 Forward kinematics	37
5 Slip detection	51
5.1 Frequency Analysis	53
5.2 Translation	54
5.3 Rotation	60
5.3.1 Image moments	61

5.3.2	Principal axis method	62
6	Object classification	69
6.1	Introduction	69
6.2	Overview	70
6.3	Object diameter	71
6.4	Compressibility	72
6.5	Standard deviation of intensity values	73
6.6	Chebyshev moments	73
6.6.1	Geometric moments and orthogonality	75
6.6.2	Chebyshev polynomials	77
6.6.3	Chebyshev moments	78
6.6.4	Computational aspects	81
6.6.5	Translational invariance	82
6.6.6	Rotational invariance	83
6.6.7	Experimental results	86
6.7	Dataset - Acquisition and Preprocessing	90
6.8	Classification	95
6.8.1	Feature scaling	95
6.8.2	Support Vector Machines (SVM)	95
6.8.3	Feature selection	99
6.8.4	Linear Discriminant Analysis (LDA)	101
6.8.5	Model selection and performance measurement	103
6.9	Results and Discussion	104
7	Developed Software	109
8	Conclusions	113
Bibliography		115

List of Figures

1.1	SCHUNK Dexterous Hand 2.0 with tactile sensor matrices by Weiss Robotics.	1
1.2	Predefined grasp configurations. Convenient combinations are composed of a coupled rotational axis angle of 0°, 60° and 90° and parallel or tilted fingertips. © SCHUNK	2
1.3	Size comparison: Finger with proximal DSA 9205	3
1.4	Artist's interpretation of Penfield's iconographic somatosensory homunculus.	4
1.5	Working principle of a resistive tactile sensor.	6
1.6	Viscoelastic Kelvin-Voigt model.	8
1.7	Dynamic effects explained by the Kelvin-Voigt Model.	10
2.1	Signal smoothing in real time. Data represents the average sensor value during a typical grasp. Running mean and Gaussian smoothing are both implemented as a convolution filter with a sliding window size of 5 data points. The depicted Kalman filter is not based on the model derived in section 2.1.2 since it does not produce smooth estimates at all - even with EM-tuned parameters. Instead, a simple model that always predicts the current position is used.	15
2.2	First and second derivative of the Gaussian smoothed signal are computed using the Savitzky-Golay filter. Cubic polynomials are fitted to a sliding window of size $N = 5$. Even though the original signal is already smoothed, the amplified noise turns out to be too large to robustly analyze the sensor measurements.	16
2.3	Spatial filters applied to a profile with unequally distributed pressure and fuzzy outline. See text for details.	17
2.4	Ghosting effect in high sensitivity mode. Spatial, temporal and spatio-temporal filters are examined for their ability to remove noise. All taxels are masked, meaning they show their original value if they were not removed by the applied filter. See text for details.	18
3.1	Pressure profile for illustration purposes: A flat stamp, covering the entire width of the sensor matrix, is centrically loaded with a weight of 1000 g. Although the pressure is distributed uniformly on the contact area, the resulting sensor readings show noticeable deviations. The values of four exemplary sensor cells as well as the (spatially filtered) matrix mean are plotted against time.	20
3.2	Border effect and irregularities of the tactile sensor. A point contact of constant force is moved along the y-axis of a distal sensor matrix. Dashed lines represent taxel centers. Data origin: Karakatsani (2011)	21
3.3	Step response of a PT2-element with linear drift.	24
3.4	Exemplary curve fits and observed effects.	27

3.5	Long-term measurement of the rising temperatures under static load. The power consumption of the grasp is $24 \text{ V} \times 0.4 \text{ A} = 9.6 \text{ W}$. The sensor matrices are not covered by the grasped object, which in addition, is a poor heat conductor.	28
3.6	Pressure calibration set-up.	29
3.7	Pressure calibration for different temperature ranges. For reasons of clarity, only the fitted function $a(t)$ is drawn. There is an approximately linear relationship between the applied pressure and the resulting mean sensor value as well as the slope m of the drift.	30
3.8	Pressure calibration in \mathbb{R}^3	31
3.9	Pressure calibration: simplified projection	32
3.10	Sensitivity settings from 0.1 to 1.0. The estimated values of K are indeed related by the given factor. Weight: 700 g, square stamp: 1.1025 cm^2 , threshold: 100 . .	33
3.11	The ground noise of the transducers differs from matrix to matrix. In addition, individual taxels might deviate from the mean.	34
3.12	Ground noise calibration: The matrix thresholds can be set according to the upper bounds of the estimated prediction intervals in order to achieve high sensitivity while maintaining the usual frame rates. The series of measurements consists of 1297 tactile sensor- and temperature readings, taken every 3 seconds. Data is then divided into 20 equally sized bins to overcome the low temperature bias of the measurements, cf. 3.5. The reciprocal standard deviations of the bins serve as a factor in the weighted least-squares regression. See text for details.	35
4.1	Schematic of a SDH-2 finger's kinematic chain along with the coordinate frame assignment at zero-position ($\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6 = 0^\circ$). The transformation matrices ${}_1^0\mathbf{T}$, ${}_2^1\mathbf{T}$, and ${}_3^2\mathbf{T}$ correspond to origins O_1 , O_2 and O_3 and are computed according to the Denavit-Hartenberg tables. The offsets to the hand's origin as well as the final transformation to the coordinate frame of the proximal and distal tactile sensor are handled separately.	39
4.2	Side view of the SDH-2 including essential dimensions for the description of the kinematic chain. Measurement units in mm. Derived from assembled CAD models provided by SCHUNK	41
4.3	Top view of the SDH-2. The finger layout is based on an equilateral triangle. Measurement units in mm. Derived from assembled CAD models provided by SCHUNK	42
4.4	Isometric projection of the SDH-2 summarizing the kinematic chain from the original coordinate system at the gripper's base to the fingertips.	44
4.5	Technical drawing of the proximal digital sensor array DSA 9205 showing the position of individual taxels of size 3.4×3.4 mm. Modified version of Weiss Robotics GmbH & Co. KG (2012a)	46
4.6	Illustration of the transformation from origin O_2 to the coordinate frame of the <i>proximal</i> tactile sensor matrix.	46
4.7	Technical drawing of the distal digital sensor array DSA 9210 showing the position of individual taxels of size 3.4×3.4 mm. The curved surface requires additional effort to determine the exact coordinates. Modified version of Weiss Robotics GmbH & Co. KG (2012b)	48
4.8	Illustration of the transformation from origin O_3 to the coordinate frame of the <i>distal</i> tactile sensor matrix.	48

5.1	Point-contact with friction cone. The simple Coulomb friction model is in the equilibrium state with $ F_t \leq \mu F_n $. The configuration is depicted in the transitional phase before slip occurs.	51
5.2	Short-time Fourier transform of a single taxel during a slip event. The window size of the FFT is 32 samples with 31 samples overlap and a Hann function is applied to reduce aliasing. Since the sample rate could only be increased to about 75 Hz by masking all but two taxels of the sensor matrix, the highest occurring frequencies in the signal are ~ 37.5 Hz. Experiments confirmed that these frequencies are not enough to distinguish between rough and smooth object surfaces.	54
5.3	Translational slip-detection workflow as described by Alcazar and Barajas (2012). The method finds the difference of the convolutions of two pairs of tactile sensor matrices. The first convolution matrix \mathbf{C}_{n-1} could also be the result of a convolution of a reference matrix with itself. In that case only two consecutive frames are needed to compute a slip vector, but at the costs of an additional convolution per time step. Note that two different scales are used for the purpose of visualization to highlight the maximum values of the profiles and the convolution.	56
5.4	Normalized cross-correlation between matrices \mathbf{T}_{n-1} and \mathbf{T}_n of figure 5.3: The position of the maximum cross correlation peak relative to the matrix center is directly related to the translation between the matrices. To achieve sub-pixel precision, the area around the peak can either be interpolated or curve fitted or both. Depicted are the 2D 3-point method and a 7×7 neighborhood around the interpolated peak.	58
5.5	Principal Component Analysis vs. second order Image Moments. The length of a semi-axis is determined by the corresponding eigenvalues. In the image moment approach, the center of the ellipse is equivalent to the center of gravity while its position in the PCA case is determined by the mean coordinate of all active cells.	63
5.6	Snap-shots of a sliding motion. The images of the slipping teapot roughly correspond to the depicted tactile sensor matrices in the second row. Since the orientation angle is clamped to $[0, 180[$, the cumulative slip angle is determined by tracking the angle across the method's discontinuities. Note how the eccentricity and roundness measure complement each other.	65
5.7	Translational and rotational slip-detection: Alternative representations of the same slip trajectories depicted in figure 5.6. Alcazar is the center of mass tracking of the correlation peak, NCC 1 refers to the normalized cross-correlation using interpolation and Gaussian-weighted paraboloid curve fitting while NCC 2 is the result of the 2D 3-point closed-form solution of a Gaussian function. PC stands for the phase-correlation approach. See table 5.1 for details. An objective method to evaluate the accuracy is missing. But to the human eye, the center of mass tracking clearly has some bias while the relatively smooth NCC 2 variant seems to capture the true movement. NCC 1 and PC however show more details.	66
6.1	Classification workflow	70
6.2	Visualization of the minimal bounding sphere, a proxy for the grasped object's size. Forward kinematics are used to obtain a three-dimensional point cloud of all active taxels.	71

6.3	Differing compressibility of equal sized objects. The minimal bounding sphere of all active taxels is calculated for each tactile sensor frame. The difference in diameter at the beginning and at the resting position of a grasp serves as a measure of the objects compressibility. In addition, this feature is independent of the actual grasping speed and the dynamics of the compressed object. At first glance, the rigid hockey ball is compressed by about 2 mm. But in reality, the displacement is purely caused by the indentation of the tactile sensor's rubber foam.	72
6.4	Objects similar in color and size pose a problem to conventional sensors such as cameras or 3D scanners. Apart from more advanced 2D and 3D features, the standard deviation of the taxel's intensity values is a simple but effective indicator of the objects overall consistency.	73
6.5	Motivating example of advanced shape descriptors: Two very similar plastic balls may lead to completely different pressure profiles. A complex shape, for instance resulting from a ball that buckles under pressure, imposes certain requirements on the feature descriptors.	74
6.6	Shape description techniques, Yang et al. (2012).	74
6.7	Chebyshev polynomials of the first kind in the interval [-1,1]. Extremal points are located at $x_k = \cos\left(\frac{\pi k}{n}\right)$ for $k = 0, 1, \dots, n$. Maxima and minima have values +1 and -1 and occur at even and odd values of k . The roots are at $x_k = \cos\left(\frac{(2k-1)\pi}{2n}\right)$ for $k = 1, 2, \dots, n$ and may serve as nodes in polynomial interpolation as the nonuniform grid minimizes Runge's phenomenon.	78
6.8	The basis functions of the two-dimensional discrete Chebyshev moments up to order $p, q = 5$ with $N = 256$. Positive amplitudes appear brighter, negative amplitudes are shown darker.	79
6.9	Sampling in radial-polar coordinates	84
6.10	Reconstruction from Chebyshev moments of order $p', q' = 4$ i.e. $p, q = 8$: The left image of each pair represents the original sensor matrix (synthetic data) while the right frame is the reconstruction. Note the ringing artifacts at sharp edges due to Gibbs phenomenon.	88
6.11	Synthetic ring-shaped pressure profile reconstructed from rotational invariants of different moment orders. The last image is a perfect reconstruction.	89
6.12	Comparison of resulting translational and rotational moment invariants. The invariants of the shifted image are virtually the same. The rotated version was generated with bilinear interpolation. Considering the large discretization error, the invariants are still somewhat close to the original ones.	89
6.13	Learned grasp-object configurations	91
6.13	Learned grasp-object configurations (continued)	92
6.13	Learned grasp-object configurations (continued)	93
6.14	Scatterplot matrix of the first four standardized features. The object diameter is obviously a discriminatory feature with very little variation. The behavior of the examined compressible objects under deformation is complex. The plastic bottle for example is more flexible in the middle section than near the strengthened top or bottom. The applied force and thus the deformation also depends on the relative position of the object between the fingers due to the available torques and leverages. The resulting compressibility measures therefore show high variance. Note that the standard deviations of the intensity values are highly correlated.	94

6.15	Relevance of individual features: The absolute values of the weight vectors of all binary SVM classifiers are taken into account. The assigned weights are therefore computed according to $\sum_{i=1}^{630} \mathbf{w}_i $ since there are $N(N-1)/2 = 630$ different hyperplanes for $N = 36$ classes. Note the disproportionate relevance of the diameter feature as well as the symmetry with respect to the sensor matrices. In addition, higher-order Chebyshev moments seem to be superior to lower-order ones.	99
6.16	Recursive feature elimination with cross-validation accuracy. The most important feature alone, the diameter, already reaches good accuracy and the score keeps increasing for about 10 extra features. But the cross-validation accuracy (see subsection 6.8.5) decreases for every additional feature that is taken into account.	100
6.17	Illustration of a dimensionality reduction using the first three features of three very similar objects. PCA projects the data to the orthogonal coordinate system that maximizes the preserved variance, thus rendering the aerosol can and the glass bottle indistinguishable from each other. LDA, by contrast, takes the class labels into account and finds a projection that maximizes separability while minimizing the within-class variability.	101
6.18	Illustration of classification methods.	103
6.19	SVM hyperparameter optimization using a grid search strategy: Average cross-validation accuracy for different parameter combinations. More precisely, the randomized procedure is run 10 times with varying seeds for smoother results. The last column refers to the original feature vectors without preceding LDA. Note that after applying the LDA, the actual choice of the SVM's regularization parameter C is almost arbitrary within a large range of values.	104
6.20	Cumulative confusion matrix of a randomized and stratified cross-validation.	105
7.1	Screenshot of the framework's graphical user interface. If the SDH-2 is connected, the application allows the execution of predefined grasp primitives while recording pressure profiles, joint angles and temperatures. All data is visualized in real-time. The time series of the recorded pressure profiles can then be analyzed off-line. The implemented filters, slip-detection and classification can be accessed from the GUI for demonstration purposes.	110
7.2	Real-time visualization of the slip-detection.	111

List of Tables

3.1	Pressure calibration set-up. Contact area: 1.1025 cm^2	30
4.1	Denavit-Hartenberg parameters of finger 0	40
4.2	Denavit-Hartenberg parameters of finger 1	40
4.3	Denavit-Hartenberg parameters of finger 2	40
5.1	Benchmark of the examined methods applied to a sequence of 10000 matrices randomly sampled from a much shorter recorded slip profile. The frame rate is extrapolated to 6 sensor matrices assuming zero overhead and is hence very optimistic. The computationally expensive convolutions, correlations, interpolations and curve-fittings rely on the Python-based NumPy, Skimage and SciPy library and were carried out on a 2.1 GHz Dual-Core machine.	60
6.1	Overall cross-validation accuracy for different feature sets.	103
6.2	Summary of the classification performance metrics using a randomized and stratified cross-validation scheme.	107

Declaration

By my own signature I declare that I produced this work as the sole author, working independently, and that I did not use any sources and aids other than those referenced in the text. All passages borrowed from external sources, verbatim or by content, are explicitly identified as such.

Signed: Date:

Chapter 1

Introduction



Figure 1.1: SCHUNK Dexterous Hand 2.0 with tactile sensor matrices by Weiss Robotics.

Schunk GmbH & Co. KG (2008)

Tactile perception plays an important role in everyday life and contributes to our understanding of the world, from spatial orientation to fine motor skills and even emotional aspects. Tactile feedback from the fingertips for example, is crucial for the dexterous manipulation of objects as a sudden loss of sensation, caused e.g. by local anaesthesia or frostbite, demonstrates. It is difficult to give a good definition of tactile perception, (from Latin tangere, “to touch”) the sense of touch. In contrast to haptic perception (from Greek háptō, “I touch” or haptós, “palpable”) in the sense of active exploration, tactile sensing refers to the passive reception of location and intensity, shape and texture, movement and direction, as well as temporal variation of mechanical stimuli.

Unlike traditional industrial robots that operate in secured and well-defined workspaces, autonomous service robots are expected to reach their goals in unstructured environments. Such systems are therefore faced with various uncertainties resulting from noisy, cluttered and dynamically changing conditions and imperfect sensor data. Following the principles of the embodiment thesis, current robotic research focuses on topics like grasping and manipulation of unknown objects or the challenges of human-robotic interaction. The incorporation of tactile sensors is a promising approach to the development of more robust algorithms. This thesis aims to investigate possible methods and techniques.

1.1 SCHUNK Dexterous Hand 2.0

The SCHUNK Dexterous Hand 2.0 (SDH-2), already introduced in 2008, is the first multi-fingered gripper designed with industrial needs in mind. The servo-electric hand consists of a body, housing the FPGA controller, and three identical fingers mounted on top of it. It can grasp a wide range of objects since each finger possesses two independent degrees of freedom, allowing force and form closure. Two rotational joints in the hand's base are mechanically coupled via a worm drive resulting in two fingers with opposing rotation angles. The rotational axis of the third finger is fixed. In total, the hand therefore provides 7 degrees of freedom.



Figure 1.2: Predefined grasp configurations. Convenient combinations are composed of a coupled rotational axis angle of 0°, 60° and 90° and parallel or tilted fingertips. © SCHUNK

Important for the interaction in an environment primarily made for humans are the gripper's compact dimensions of about 1.4 times the size of a human hand and the maximal exerted force. The proximal finger joints achieve torques up to 2.1 Nm while the distal joints reach only 1.4 Nm. Both joints can rotate at 210 °/sec. With the library provided by the manufacturer, the hand can be controlled in joint-space using position, velocity and acceleration modes. A couple of predefined grasps, including parallel and centrical configurations, are illustrated in figure 1.2. Given a single open-close ratio, these grasps are executed by the internal controller. Together with an additional pinch grip, the configurations were reimplemented and parametrized in user space to create more flexibility.

1.2 Tactile sensors

The SDH-2 is equipped with 6 digital sensor arrays (DSA) by Weiss Robotics GmbH & Co. KG. The tactile sensor DSA 9205 is installed on the proximal sections of the fingers while the curved fingertips are covered by the DSA 9210 as shown in figure 1.3. The individual touch sensitive cell is referred to as *tixels*, meaning tactile element with reference to pixel for picture element. But other names such as tacter (tactile actuator), tactel or texel can be found in literature. The latter can, however, be confused with texture element.

The proximal sensor matrix consists of 6×14 taxels with a spatial resolution of 3.4 mm while the size of the distal matrix is only 6×13 . In addition, 5 taxels are missing on each side of the narrow finger tips. Special care was taken to seal the sensing matrix hermetically under a cover of silicon rubber with skin-like friction properties. But in practice, wear and tear limits the lifetime of the sensor surface. Shear forces and high temperatures must be avoided. The measurement range is specified to be 250 kPa (0.25 N/mm² or 2.89 N/taxel) and in theory, the sampling rate could be up to 230 Hz. But the connection from the sensor controller DSACON32-M (ARM7) to the hand control unit, a Stratix II FPGA by Altera, limits the frame rate to only about 30 Hz. Further details are discussed when needed in the rest of this thesis.



Figure 1.3: Size comparison: Finger with proximal DSA 9205 and curved distal DSA 9210 tactile sensors.

For a comparison, it is well worth to take a look at the basis of human haptic perception. Apart from the very sensitive hair follicle receptors and free nerve endings which are responsible for the perception of pain, Johansson and Vallbo (1984) distinguished four classic cutaneous mechanoreceptors. They are classified according to their rate of adaption, either slow or fast and the size of their receptive field. Tonic receptors (SA-type) adapt slowly to a stimulus and sustain action potentials over the entire duration while phasic mechanoreceptors (FA-type) react quickly to rapid changes only. The mechanoreceptors of type I with a small receptive field are found more frequently in sensitive areas like the fingertips while type II receptors are especially sensitive to large-scale stimuli.

Merkel disk receptors (SA I) are sensitive to low frequencies around 5 to 15 Hz and are important for sustained touch, i.e. position, pressure and features like edges and shapes.

Meissner's corpuscles (FA I) are most sensitive to vibrations around 50 Hz. They show high sensitivity to light touch since they respond to superficial skin deformation. The nerve endings also play a role in controlling the grasp force since they react to vibrating ridges of the dermal papillae (fingerprints).

Ruffini (bulbous) corpuscles (SA II) are stimulated by skin stretch and are thought to contribute to the kinaesthetic sense, allowing for example the refinement of a grasp.

Pacinian (lamellar) corpuscles (FA II) reach their lowest threshold at 250 Hz and are capable to distinguish between surface textures. They are also responding to rapid and deep touches.

See Mountcastle (2005) for a detailed description of cutaneous receptors.

Since a picture is worth a thousand words, the importance of tactile information for humans is visualized in figure 1.4. As presented in Crichton (1994), Wilder Penfield, a pioneering neurosurgeon, together with Edwin Boldrey electrically stimulated neural tissue while operating on the brain in 1937 and onwards. The treated epilepsy patients, though under local anaesthesia, were awake and could report on their sensations. During these now famous experiments, Penfield created maps of the motor cortex and the somatosensory cortex of the human brain. It is now known that the cortical area responsible for the processing of tactile information is proportional not to the size of body parts, but to the density of cutaneous tactile receptors.



Figure 1.4: Artist's interpretation of Penfield's iconographic somatosensory homunculus. The size of body parts is proportional to the cortical areas devoted to their tactile representation. © Natural History Museum, London

It is worth mentioning that although all types of receptors are already developed in newborns, at least a decade is required to reach adult-level haptic perception. Likewise, the importance of “higher-level processing” becomes apparent when the average sense of touch is compared to the performance of blind persons. Velazquez and Pissaloux (2008) state that Meissner cells in finger tips are on average 0.9 mm apart. But the two-point discrimination threshold, that is the ability to distinguish two simultaneous contacts, is only in the range of 2-3 mm. Compared to these characteristics, the 3.4×3.4 mm resolution of the investigated tactile sensor is not too far off anymore. But in contrast to current robotic hands and grasping algorithms, humans perform slight unconscious movements to increase the understanding of touched surfaces. It is likely that this behavior also improves the perceptual spatial resolution.

Many different types of tactile sensors for robotic purposes have been developed over the last decades. Their measurement principles range from the use of piezoelectric or piezoresistive materials, that either generate voltage or change their resistance under pressure, to capacitive sensors consisting of plate capacitors with a variable distance between the plates. Dahiya and Valle (2008) give a detailed overview of tactile sensors used in robotic applications and Petchartee (2008) compiled an impressive list of various transducer principles published during the last 25 years. Among them are magnetic and inductive, ultrasonic, electro-chemical, thermal, optical and even quantum tunnel effect sensors. In addition, there are completely different approaches such as proximity sensors that do not depend on mechanical interaction at all.

Newer developments often integrate several measurement principles into a single sensor. The SynTouch BioTac for example separately measures forces, vibrations and heat flux. There is also a self-organizing, multimodal artificial skin incorporating accelerometers, normal force-, proximity-, and temperature sensors. Kaboli et al. (2014) show that the combined sensor readings can be used to classify objects according to texture, weight and hardness/softness.

1.3 Applications

Apart from slip-detection and object classification as discussed in the later chapters, various applications of tactile sensors have been proposed. Haase (2011) for example studied reactive grasping skills, that is the usage of tactile information to execute compliant, robust grasps. Reactive grasping is a well known topic in robotics. Hsiao et al. (2010) for example use tactile sensors to guide the end effector around the object if an asymmetric or unexpected contact is made. During grasp planning, they consider such events and prepare fall-backs in case the local adjustment fails.

Another simple form of tactile feedback utilization is demonstrated by Morales et al. (2007). They enhanced predefined grasping primitives with force control such that the robot tries to maximizing the contact surface to minimize local forces.

Dang and Allen (2012) dealt with the problem of blind grasping. Using a soft finger contact model, they performed thousands of virtual grasps using the GraspIt! framework. Primitive tactile features, extracted from the simulator, are then classified according to grasp stability. In a following-up paper, Dang and Allen (2013) demonstrated the practicality of their approach using “experience from local geometries” in real-life applications.

Talking about grasp stability, Bekiroglu et al. (2011) proposed a kernel logistic regression model for conditional grasp success probability, separated for visual pose estimation and tactile sensor readings. Low-dimensional geometric image moments are used to represent tactile features. A more recent paper by Hyttinen et al. (2015) also estimates grasp stability via kernel logistic regression. The stability model is again trained on low-dimensional image moments, the hand joint angles and the ID of the prototype grasp. But in contrast to the previous paper, the method generalizes to novel objects.

An interesting approach to grasp adaptation in the presence of perturbations is presented by Sauser et al. (2012). Using a Gaussian Mixture Model (GMM), they learn the relation between pressure, contact normal direction and joint angles. This model can then be used to predict hand poses given the desired pressure and contact normals. Since the model is learned iteratively by human demonstration and correction as well as self-demonstration, a sparse dataset of valid poses and dense, but less certain configurations is available. This enables the system to quickly react to external perturbations.

It has been shown that artificial skin greatly improves the interaction between robots and humans. Strohmayer (2012) for example developed a scalable and multifunctional whole-body tactile sensor and implemented a collision detection for save interaction.

In the absence of visual information, Meier et al. (2011) examined the feasibility of three-dimensional object probing using the SDH-2. Their haptic approach creates a probabilistic model of the sparse contact point cloud. With the help of the Iterative Closest Point (IPC) algorithm they are able recognize objects using nearest neighbor classification.

Strub et al. (2014) build a learning framework for the haptic exploration of object shapes. Based on the 3D position, orientation and curvature of a pressure profile and Dynamic Neural Fields (DNF) they approach the task with the help of the SDH-2 as well. By solving the sparse *tactile* Simultaneous Localization and Mapping (SLAM) problem, the authors are able to determine the number of object surfaces which can be used to find an appropriate correction for the initial pose estimation.

Finally, it should also be noted that there is related interdisciplinary research at the interface between computer science, engineering and medicine. Walker et al. (2015) for example showed recently that patients with prosthetic hands were able to recover objects from slip much more frequently with vibrating feedback than with force feedback alone. The stimulus was artificial, but appropriate tactile sensors could one day replace the lost sense of touch.

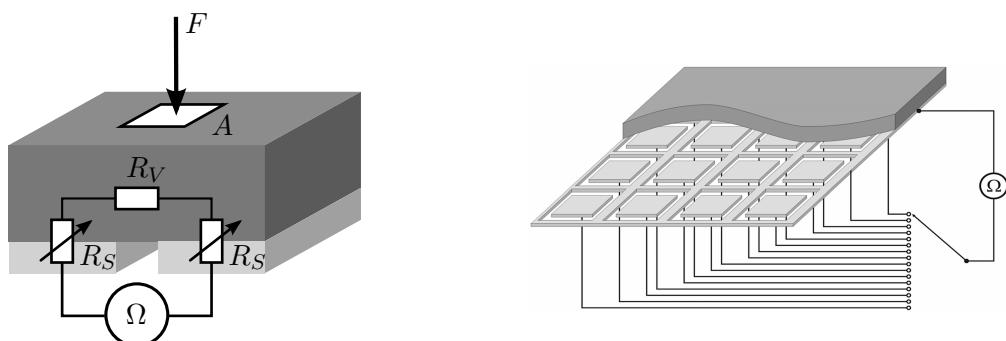
Tactile sensor are available for quite some time now and the low hanging fruits have all been harvested. However it is still an unresolved question how to process and represent tactile data most effectively. And the integration of tactile feedback in robotic grasping remains an active topic of research as well.

1.4 Working principles

The actual working principles of the DSA 9205 and DSA 9210 are not specified in detail, but Dr.-Ing. Karsten Weiß, founder of Weiss Robotics, published a series of papers on the design of an anthropomorphic robotic hand (Weiss and Woern (2004)), the development of tactile sensors (Weiss and Worn (2005a) and Weiss and Worn (2005b)), sensitive skin (Göger et al. (2006)) and the prototype of tactile transducers for a predecessor of the SDH-2 (Weiß et al. (2006)).

The resistive measurement principle based on conductive elastomers is one of the more widespread techniques among tactile sensors for robotic applications. Usually, a row and a column electrode, separated by a conductive elastic material are arranged in a matrix layout. One electrode carries a reference voltage while the other serves as a sliding contact comparable to a potentiometer. As a consequence of this layout, the measured resistance between the reference electrode and a sensor cell electrode is a function of the applied load. Unfortunately, persistent bending stress applied to the upper sensor cell electrode results in material fatigue and reduced lifetime of the tactile sensor. To avoid this problem, Weiß placed both electrodes on the back side of the sensor material mounted on a rigid surface as illustrated in figure 1.5a.

There are two different effects responsible for the observed behavior. For one, the elastic material, silicone rubber in this case, contains conductive nanoparticles, possibly carbon blacks or semiconductive particles. The deformation resulting from the applied load alters the particle density within the material and the electrical volume resistivity R_V decreases. See Ying et al. (2007) for more information on the conductive chain barrier tunneling effect. But the larger portion of the working principle depends on an interface effect between the electrodes and the surface of the rubber foam. When the sensor is unloaded, the rough surface of the foam only shares a few contact points with the electrodes. The surface resistivity R_S is therefore large. But as a consequence of the applied load, the contact area rises and the corresponding resistance drops.



(a) Cross section of a single cell. Note that R_V is assumed to be constant

(b) Multiplexed single sided electrode contacting. Weiss and Worn (2005a)

Figure 1.5: Working principle of a resistive tactile sensor.

In the mentioned matrix layout with row and column electrodes, the resistance of individual taxels can be measured with only one channel per column. But large sensor areas usually result in slow sampling speeds due to capacitive effects of the wiring. The actual arrangement of the single

sided contacting is depicted in figure 1.5b. In this approach, each measurement electrode leads to a multiplexer which is sequentially connected to the resistance transducer. This way, high sampling rates can be achieved while maintaining reasonable spatial resolutions. Weiss Robotics calls this technique *active sampling* of the matrix. This issue shows that there is a trade-off between resolution and sampling speed when designing a tactile sensor. Weiss Robotics for example, also offers related tactile transducers with a spatial resolution of only 4×6 taxels, but sampling rates of up to 800 fps. There is another interesting aspect mentioned in Weiβ's papers: Higher spatial resolution can be achieved with the same amount of electrodes by inter-electrode sampling. The electrodes are therefore grouped and connected to two different multiplexers. The final pressure reading associated to a single taxel is then computed not from a single electrode, but from four surrounding electrodes. That is, the actual taxels are located in-between the electrodes. Weiβ proposed square, triangular and hexagonal electrode layouts, but the profile of a maximally loaded sensor matrix reveals a barely visible checkerboard pattern, indicating the first variant is used. The tactile sensors also have less sensitive taxels at the matrix border. This could partially be explained by this inter-electrode sampling technique since those taxels only have 2 or 3 neighboring electrodes.

Extensive research has been conducted in the field of contact mechanics dating back to Heinrich Hertz in 1886. The relationship between the protective layer of a tactile sensor and a contacting body is quite complex. Consequently, the properties of the surface material in combination with different contact models are subject to ongoing studies. Generally speaking, the reconstruction of object details from pressure profiles is related to deconvolution and an ill-posed problem since varying object surfaces may cause the very same imprints.

Assuming a homogeneous and isotropic material, while ignoring effects like adhesion, an elastic material may be viewed as a linear spring. In the case of rubber however, this assumption is only an approximation since the polymer structure of the material depends on temperature and loading rates. In any case, the deformation of the sensor's silicon rubber is linearly proportional to the applied force according to Hooke's law. This relationship can be derived by examining the deformation ε of a linear elastic body such as a flexible rod under tensile stress σ . According to the well known stress-strain relation

$$\sigma = E\varepsilon$$

where E is the ratio of stress to strain, also known as modulus of elasticity, the applied force to a cross-sectional area A_c is given as

$$F = EA_c\varepsilon.$$

The material's inner deformation energy U can be obtained by integrating over the imaginary rods that constitute the rubber surface.

$$U = \int EA_c\varepsilon d\varepsilon.$$

This quantity must be equal to the potential energy stored in a spring with constant k and deformation x , that is

$$U = \frac{1}{2}kx^2 = \frac{1}{2}xF.$$

It is now possible to define a function $A(F)$ relating the outer load with the deformed area. From figure 1.5a follows that the electrical resistance R of a sensor cell can be expressed as a sum of its elementary resistors

$$R(F) = 2R_S(F) + R_V$$

where R_V is the volume resistance of the sensor material. The electrical interface resistance R_S is a function of the area A of an applied load and the surface resistance of the sensor material R_{SO} , e.g.

$$R_S(F) = \frac{1}{A(F)} \cdot R_{SO}.$$

As mentioned before, the interface effect is much larger than the change in volume resistance. R_V is therefore assumed to be constant.

1.5 Kelvin-Voigt Model

To obtain a deeper understanding of the working principles and associated effects, the actual material properties have to be examined in more detail. A viscoelastic material such as silicon rubber shows certain characteristics like stress-relaxation, creep and hysteresis. There are various models that deal with these properties. Argatov (2013) for example discusses the nonlinear deformational behavior of materials. But according to Özkaya et al. (2012) only two basic elements, namely springs and dashpots arranged in different configurations, are sufficient to analyze viscoelasticity. While the springs are used to account for the elastic solid behavior, dashpots or dampers are used to describe the viscous fluid behavior. One of the simplest models is the Kelvin-Voigt model which consists of a viscous damper and an elastic spring connected in parallel as shown in figure 1.6.

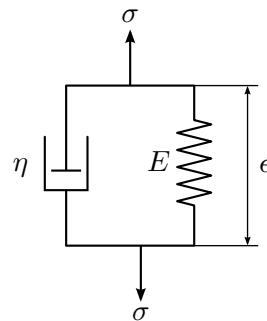


Figure 1.6: Viscoelastic Kelvin-Voigt model.

For the spring we have the already mentioned relationship between stress σ and strain ε

$$\sigma(t) = E\varepsilon(t)$$

with E being the modulus of elasticity. For the dashpot, stress is a function of the strain rate, i.e.

$$\sigma(t) = \eta\dot{\varepsilon}(t)$$

with a coefficient of viscosity η . Since the stress is shared in the parallel configuration, the stress-strain relation in the Kelvin-Voigt model is given as

$$\sigma(t) = E\varepsilon(t) + \eta\dot{\varepsilon}(t). \quad (1.1)$$

The Kelvin-Voigt model is idealized, meaning no known material's behavior can be fully described by it. It is therefore just a basic element for more complex models to build on, but certain time-dependent effects can still be explained. For example, it follows from the differential equation (1.1) that the response of a viscoelastic material does not only depend on the magnitude

of the applied load. It also matters how fast the stress is applied or removed. In other words, the stress-strain relation is not constant but a function of time.

Under sudden constant stress, applied at time t_0 and removed at time t_1 as seen in figure 1.7a, the material shows viscoelastic creep and complete recovery. That is, the material continues to deform even though the stress remains constant over time. After the stress is removed, the material returns to its initial position. The strain $\varepsilon(t)$ between t_0 and t_1 can be determined by finding the homogeneous solution of (1.1) for a constant stress σ_0 or alternatively by the Laplace transform as discussed in chapter 3.1 (Model fitting). To keep it short, starting at t_0 , the strain decays exponentially with

$$\varepsilon(t \leq t_1) = \frac{\sigma_0}{E} \left(1 - e^{-\frac{E}{\eta} t} \right).$$

Note that for a permanent load, the deformation tends to σ_0/E for $t \rightarrow \infty$.

There is another important effect present in silicon rubber. In contrast to idealized elastic materials which store all their energy during deformation, viscoelastic materials lose energy due to internal friction. Figure 1.7b illustrates the stress-strain curve as a result of periodic loading and unloading. The enclosing area of the so called hysteresis loop is proportional to the amount of energy dissipated as heat per cycle. Note that the hysteresis loop depends on the strain rate and reduces to a line for an elastic material.

To give a mathematical explanation of elastic hysteresis, consider the stress response of a periodic strain input i.e. $\varepsilon(t) = \varepsilon_0 \sin(\omega t)$. It follows that

$$\begin{aligned} \sigma(t) &= \varepsilon_0(E \sin(\omega t) + \eta \cos(\omega t)) \\ &= \sigma_0(\omega t + \delta) \end{aligned}$$

where δ is a delay between strain and stress, a phase shift for all intents and purposes. By separating the modulus of elasticity E of the first equation in the complex domain, Brinson and Brinson (2010) demonstrate that this phase lag corresponds to the dissipated energy with $\tan(\delta)$.

Hysteresis is hard to quantify in practice since the loop is usually not an ellipse but an asymmetric nonlinear curve with saturation properties. In addition, it depends on the strain rate, an unknown variable in the grasping process. Sánchez-Durán et al. (2012) have already examined the effects of hysteresis with respect to a tactile sensor based on conductive polymer as well. The authors were able to compensate for hysteresis related errors by a factor of 5 using the generalized Prandtl-Ishlinskii model. Put simply, this method consists of a weighted sum of linearized play operators. The fitted model can be analytically reversed and used for compensation.

Creep and recovery as well as hysteresis are non-negligible effects observed in the tactile sensor's behavior. While the creep can be compensated as discussed in one of the following chapters, it was decided to ignore the latter. This decision can be justified by the fact, that the hysteresis disappears for slow loading-unloading cycles. To give a rough empirical estimate, a dead time of five seconds between grasps should be enough to completely escape the effect.

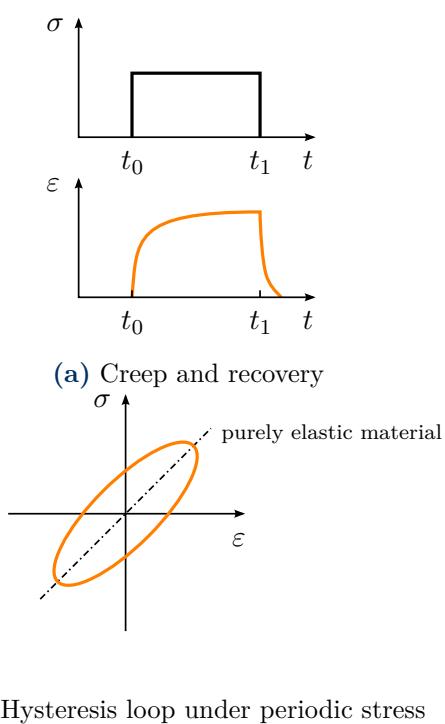


Figure 1.7: Dynamic effects explained by the Kelvin-Voigt Model.

Chapter 2

Signal Processing / Filtering

The spatial resolution of the tactile sensor images in question is several orders of magnitude lower than the resolution of raster images created for example by digital cameras. In addition, the sensor shows highly nonlinear characteristics which are supposedly linearized by the internal signal conditioning electronics. But in reality, the tactile sensor readings are far from being exact pressure measurements. Therefore, depending on the intended usage of the tactile images, the first step of any information extraction process usually involves some kind of preprocessing or filtering. Common operations found in most publications dealing with tactile sensors include upscaling using various interpolation methods, denoising in the spatial as well as in the temporal domain and peak detection. Schmid (2008) for example developed a tactile language to control a robot through an artificial skin. Being interested in the number of contacts and their extent, he first looked for local maxima in a 3×3 taxel neighborhood and then applied a watershed algorithm to the "mountains" in order to segment the different regions. Also working with the SDH-2, Saal et al. (2010) learn the dynamic properties of liquids by shaking bottles. Since they only use the frequency components, they project the time series of all active taxels onto their principal components in order to achieve spatial invariance with respect to the most responsive taxels. Apart from such special applications, noise removal remains the main issue for signal conditioning. Technically, data smoothing is a form of low pass filtering and surely a subject worth a book on its own. Further information on the filters briefly discussed in the following can be found in one of the signal processing bibles Proakis and Manolakis (2007).

2.1 Temporal Filtering

There is a vast number of different smoothing and noise reduction techniques available. They can roughly be classified into frequency and time domain based filters and also differ with respect to the underlying noise models. Another important aspect is their ability to process online streams recursively. Locally weighted regression smoothing (LWR, LOWESS) for example is a non-parametric method that can approximate a wide range of noisy functions. But it operates in batch mode only and inevitably introduces lag. Recursive Bayesian regression on the other hand is updated with every new data point but is computationally expensive. Given the real-time constraints while keeping the computational costs in mind, only a handful of method seem to be practical.

2.1.1 Convolution filters

The simplest form of smoothing is the moving average or running mean filter. Each data point is thereby simply replaced by the average value of neighboring values. In signal processing notation, this amounts to

$$y[i] = \frac{1}{2M+1} \sum_{j=-M}^M y[i+j]$$

where $2M + 1$ is the number of points in the neighborhood of the current signal value $y[i]$. This can be done recursively for each new measurement. But alternatively, the problem is typically formulated as a sliding window approach where the convolution kernel consists of $2M + 1$ values all equal to $1/(2M+1)$. Note that the window size is assumed to be odd with the current measurement positioned in the middle. Reason for that is the reduction of lag. Smith (1999) mentions that many scientists and engineers feel guilty about using the moving average filter due to its simplicity. But in fact it is optimal in the sense of reducing Gaussian white noise while keeping the sharpest step response. This convolution approach to filtering can be used with any type of window function. One of the most popular variants of weighted averaging applies a normalized Gaussian kernel given as

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{x^2}{2\sigma^2}}.$$

The frequency response of a Gaussian filter is also a Gaussian. It is therefore considered an ideal time domain and true low-pass filter. Hence, high frequency artifacts are avoided.

2.1.2 Kalman filter

The Kalman filter, after Rudolf E. Kálmán, is an important and well researched model-based state estimator. In the context of control theory, a state refers to a discrete point in time of a linear dynamic system. In the absence of noise, such a system is completely described by the current state and a linear differential equation allowing the transition to any past or future state. This simplification clearly does not hold in practice. In contrast, the Kalman model assumes that the state transitions as well as the measurements, although linearly related, are corrupted by Gaussian white noise. The state transition model in matrix-vector form is given as

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k$$

where \mathbf{F} is the state transition matrix relating the previous state \mathbf{x}_{k-1} with the current state \mathbf{x}_k . Similarly, \mathbf{B} relates the control-input model with the current input \mathbf{u}_k and \mathbf{w}_k denotes the process noise. The measurement model is given as

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k$$

where the current observation \mathbf{z}_k is related by the current state via the observation matrix \mathbf{H} . In addition, the process and measurement noise are assumed to be independent with

$$\begin{aligned} \mathbf{w}_k &\sim \mathcal{N}(0, \mathbf{Q}_k) \\ \mathbf{v}_k &\sim \mathcal{N}(0, \mathbf{R}_k) \end{aligned}$$

where process noise covariance \mathbf{Q}_k and measurement noise covariance \mathbf{R}_k may change over time.

The actual real-time implementation processes incoming data and estimates the current state by alternating predict and update steps. Similar to recursive Bayesian estimation, the new state

and its error covariance are first predicted *a priori* using the dynamic model. When the next measurement is available, the *posterior* estimates are obtained in a correction step. The details and the derivation of the Kalman filter require excursions to least squares estimation, probability theory and the theory of dynamic systems and is well beyond the scope of this thesis. Reference is therefore made to the standard work Grewal and Andrews (2001).

Following up on section 1.5, the Kelvin-Voigt model consists of a viscous damper and an elastic spring connected in parallel. The tactile sensor therefore resembles a mass-spring-damper system. With reference to figure 1.6, let $y(t)$ be the displacement of the mass m from its resting position at time t . Consequently $\dot{y}(t)$ refers to the velocity of the mass and $\ddot{y}(t)$ to its acceleration. From the basic properties of a Hookean spring with constant E and a viscous damper with coefficient η , the total force applied to the mass over time can be computed as

$$\begin{aligned} F(t) &= F_s(t) + F_d(t) \\ &= -Ey(t) - \eta\dot{y}(t). \end{aligned}$$

With the help of Newton's second law of motion $F(t) = m\ddot{y}(t)$, the second-order ordinary differential equation for the Kelvin-Voigt model is given as

$$m\ddot{y}(t) = F(t) - \eta\dot{y}(t) - Ey(t) \quad (2.1)$$

with initial conditions $y(0) = 0$ and $\dot{y}(0) = 0$. Since the state-space representation is limited to first-order ordinary differential equations, equation (2.1) has to be converted into two first-order differential equations using substitution. Let $x_1 = y$ denote the position, $x_2 = \dot{y}$ the speed and $u = F$ the input variable. Rewritten in standard form, equation (2.1) is given as

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= \frac{1}{m}(-\eta x_2 - Ex_1 + u). \end{aligned}$$

The initial position and speed are now $x_1(0) = 0$ and $x_2(0) = 0$. The second-order linear state-space model in matrix-vector form $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$ is hence given as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{E}{m} & -\frac{\eta}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u.$$

Regarding the position as output variable, i.e. $y = x_1$ we have

$$y = [1 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0] u.$$

It is well known that good estimates for the model parameters m , E and η as well as for the noise covariance matrices \mathbf{Q} and \mathbf{R} are crucial for the performance of the Kalman filter. But at the same time, those parameters are difficult to find. Shyam et al. (2015) for example revisit a variety of different methods to obtain good estimates. But although an Expectation-Maximization (EM) algorithm was used to determine optimal parameters, the result is not convincing. A simple zero-order Kalman filter with a dynamic model that only wants to keep its current position produces smoother estimates than the version with the more elaborated model. In principle, variants like the Extended Kalman Filter (EKF) or the Unscented Kalman Filter (UKF) could deal with nonlinear state transitions. The performance of the EKF was therefore evaluated using the pykalman library implementation, however with little success.

The Kalman filter only represents a particular instance of a Bayesian view on the filtering problem. See Särkkä (2013) for details. Consider the probabilistic state space model with measurement model

$$\mathbf{y}_k \sim p(\mathbf{y}_k | \mathbf{x}_k)$$

and dynamic model

$$\mathbf{x}_k \sim p(\mathbf{x}_k | \mathbf{x}_{k-1}).$$

In general, a fixed-interval smoother recursively takes all measurements up to time T into account and computes the marginal posterior distribution of the state \mathbf{x}_k at time step k , i.e.

$$p(\mathbf{x}_k | \mathbf{y}_{1:T}) \quad \text{for all } k < T.$$

Starting from the last step $p(\mathbf{x}_T | \mathbf{y}_{1:T})$, the Bayesian optimal smoothing equations then consist of a prediction- and backward update step given as

$$\begin{aligned} p(\mathbf{x}_{k+1} | \mathbf{y}_{1:k}) &= \int p(\mathbf{x}_{k+1} | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{y}_{1:k}) d\mathbf{x}_k \\ p(\mathbf{x}_k | \mathbf{y}_{1:T}) &= p(\mathbf{x}_k | \mathbf{y}_{1:k}) \int \frac{p(\mathbf{x}_{k+1} | \mathbf{x}_k) p(\mathbf{x}_{k+1} | \mathbf{y}_{1:T})}{p(\mathbf{x}_{k+1} | \mathbf{y}_{1:k})} d\mathbf{x}_{k+1}. \end{aligned}$$

A possible implementation of this forward-backward approach is the Rauch-Tung-Striebel smoother, also known as Kalman smoother since it typically runs on a set of means and covariances computed by a preceding Kalman filter.

Figure 2.1 sums up the effect of three suitable temporal filters with real-time capabilities.

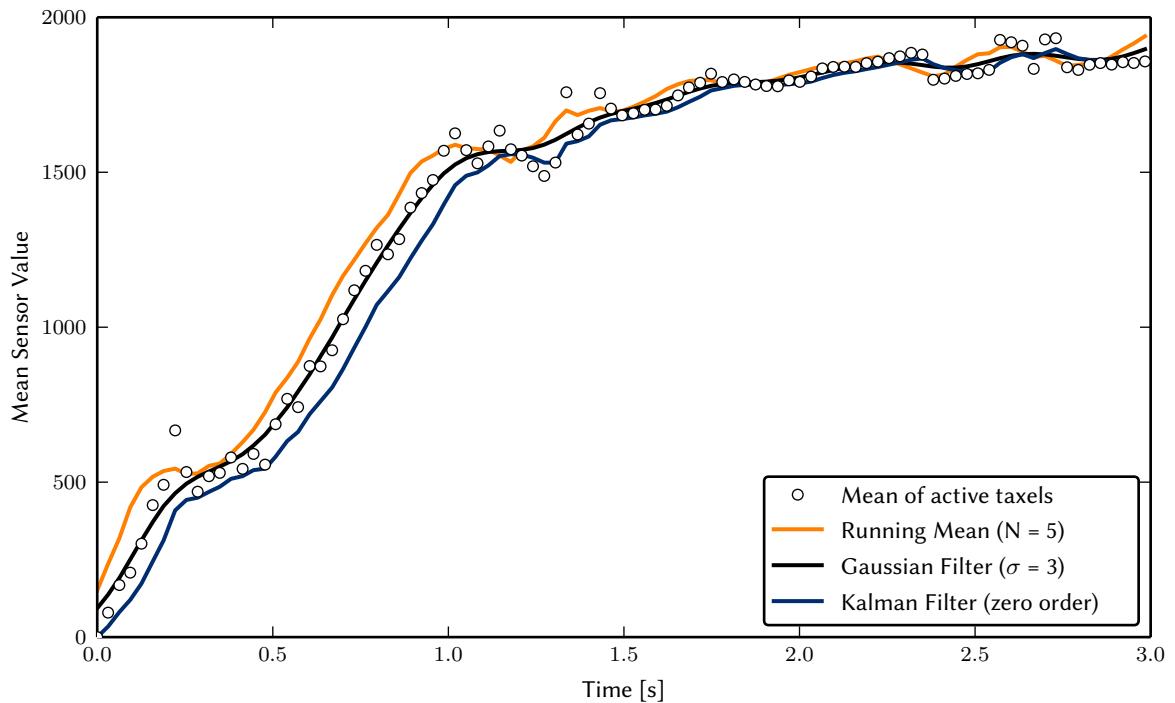


Figure 2.1: Signal smoothing in real time. Data represents the average sensor value during a typical grasp. Running mean and Gaussian smoothing are both implemented as a convolution filter with a sliding window size of 5 data points. The depicted Kalman filter is not based on the model derived in section 2.1.2 since it does not produce smooth estimates at all - even with EM-tuned parameters. Instead, a simple model that always predicts the current position is used.

2.2 Savitzky-Golay filter

In order to extract characteristic values such as extremal values or points of inflection from a signal, one is often faced with the task of differentiating data numerically. But numerical derivatives can amplify noise in the sensor readings. A classical local method is the so called Savitzky-Golay filter, Savitzky and Golay (1964). The main idea behind this method is a least-squares fit of each new data point with a polynomial of order $m < 2k$ over a window of size $2k + 1$, centered at the point. The derivatives are then obtained from the coefficients of the polynomial. In contrast to smoothing splines, the segments are not guaranteed to be continuously differentiable. But the convolution filter is known to preserve the original shape of the signal better than other types of filters and is therefore still a very popular tool, especially in applied science.

The same dataset shown in figure 2.1 is used in figure 2.2 to illustrate the first and second derivatives using a second-order Savitzky-Golay filter. The results show that it is extremely difficult to chose the right parameters for the filtering process of tactile sensor signals with a high dynamic range. The next chapter on calibration therefore discusses regression as an alternative to smoothing techniques.

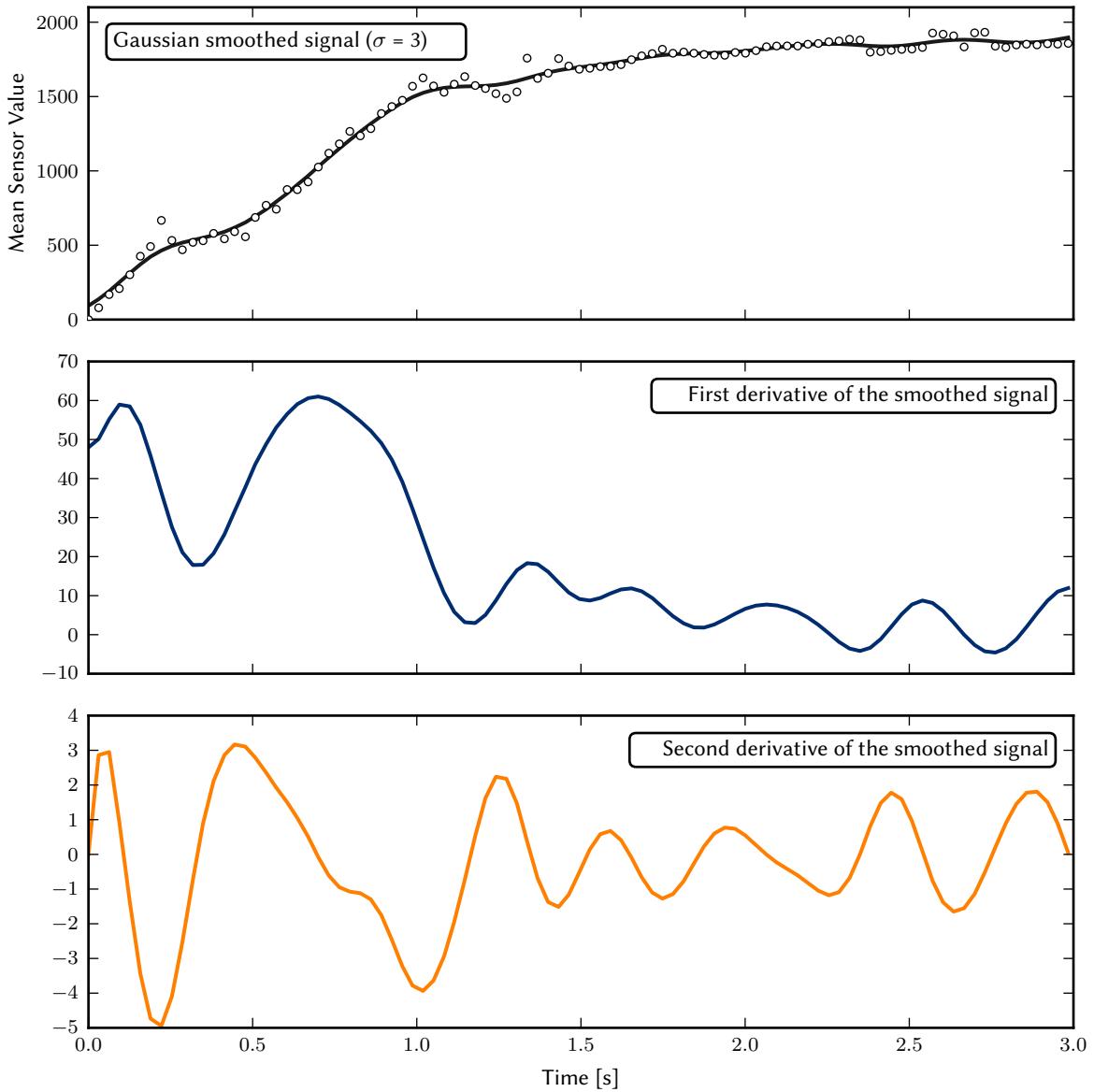


Figure 2.2: First and second derivative of the Gaussian smoothed signal are computed using the Savitzky-Golay filter. Cubic polynomials are fitted to a sliding window of size $N = 5$. Even though the original signal is already smoothed, the amplified noise turns out to be too large to robustly analyze the sensor measurements.

2.3 Spatial Filtering

Obtained pressure profiles often show some kind of taxel defects. More precisely speaking, individual taxels sometimes deliver unexpected but time-invariant readings. As a consequence of the sensor's working principles, the silicon rubber is placed freely moveable on top of the sensor cell. Shearing stress and tension may cause deformation of the flexible material. As a consequence of the buckling, technically unloaded areas might be subject to pressure while the stress on loaded areas is relieved. There are other effects like less sensitive border taxels or unequally distributed pressure as seen in figure 2.3.

Any introductory book on image processing like Jähne (2005) covers the following techniques. Simple convolution filters, as discussed in the previous section, can be easily applied to two-dimensional data. The depicted box- and Gaussian filters directly correspond to their one-dimensional version. Both are especially suitable to minimize the effect of flickering taxels. The median filter on the other hand is a nonlinear filter that replaces the taxel in the center of the kernel with the median, that is the most frequent value, of its neighborhood. While being known to preserve edges much better than the linear convolution kernels, the median filter is able to effectively remove speckle noise. The Non Local Means (NLM) filter is a relatively new algorithm by Buades et al. (2005). In contrast to the neighborhood filters, the value for a taxel is computed as a weighted average of all taxels of the sensor matrix. The weighting function, applied to all compared pairs of taxels, depends on the similarity between the intensity values and their Euclidean distance-weighted surrounding taxels. Since the OpenCV (C++, Python) and NumPy library (Python) can directly access the tactile sensor frame, a large number of filters are available for experimentation. As an example, the Bilateral filter, an edge-preserving Gaussian smoothing filter, was added in a minimum of time.

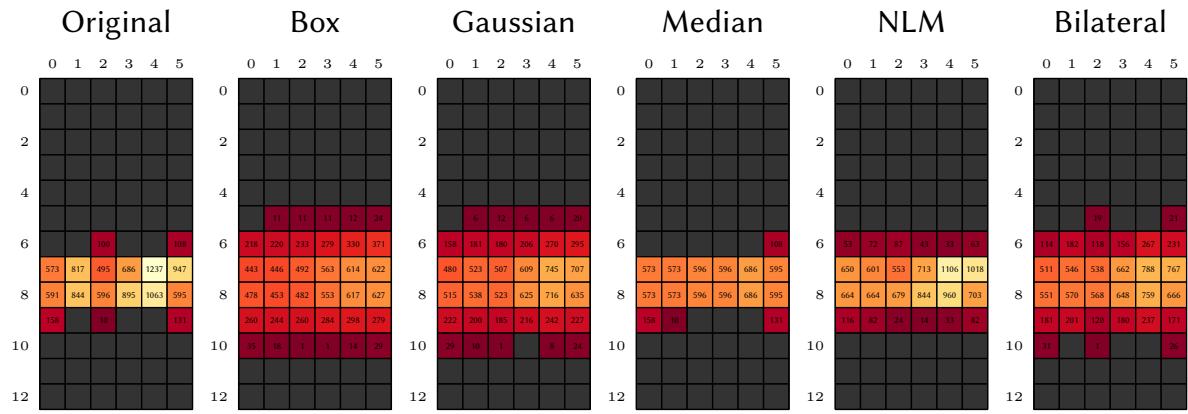


Figure 2.3: Spatial filters applied to a profile with unequally distributed pressure and fuzzy outline. See text for details.

2.4 Ghosting

Another problem is the inherent background noise of the tactile sensors. Under normal circumstances, this noise is not visible due to the default settings of the DSA. But if the sensitivity is increased while the noise suppression thresholds are lowered, unloaded taxels begin to fire sporadically. In the absence of a better name, these randomly appearing signals are referred to as *ghosting* effect in the following.

The intensity values of these flawed taxels are small and they could simply be ignored. But the main problem occurs when a contact should be detected as soon as possible. As mentioned before, some reactive grasping algorithms rely on very sensitive proximity sensors. But all the SDH-2 can offer are resistive tactile arrays, which are, on top of it, dull per default. The therefore implemented *high sensitivity mode* tunes all changeable and temperature dependent parameters in order to make the most of the situation. The robust removal of ghosting is part of that strategy.

Apart from the previously mentioned median filter, another spatial method is well known to remove speckles as well as salt and pepper noise. Morphological filtering is a nonlinear local transform on geometric features, deeply rooted in set theory and topology. But for the purpose

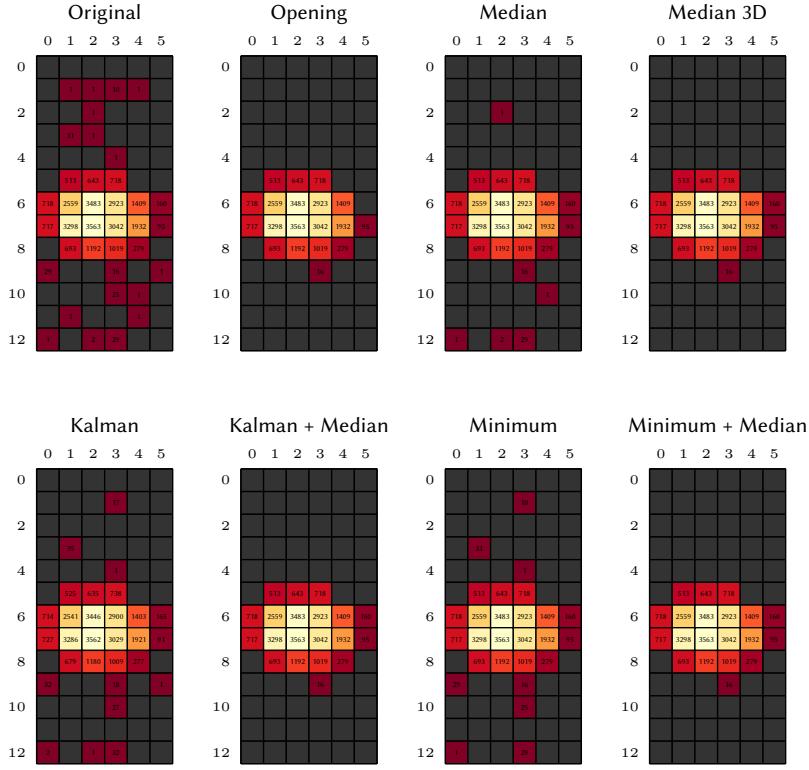


Figure 2.4: Ghosting effect in high sensitivity mode. Spatial, temporal and spatio-temporal filters are examined for their ability to remove noise. All taxels are masked, meaning they show their original value if they were not removed by the applied filter. See text for details.

of noise removal it is enough to consider the *erosion* and *dilation* operator. Morphological erosion sets the taxel in the center of the sliding window, also known as structuring element, to the minimum over all taxels in the neighborhood. Dilation on the other hand, applies the maximum function. In other words, erosion removes islands of taxels smaller than the structural element while dilation fills holes up to the size of the kernel. Applying the erosion filter subsequently followed by dilation is a common operation with a name of its own. The so called *opening* operation first shrinks the shapes and then expands the remaining structures again to their original size.

Some of the mentioned filters do not only remove noise but also alter the intensity of the remaining valid taxels. Therefore a masked version of the original sensor matrices, where only the surviving taxels are punched out, is returned. Figure 2.4 gives an overview of simple spatial, temporal and spatio-temporal filters. The effect of the opening filter is a bit misleading since the main contact is relatively large. Smaller shapes and details are easily removed by this "ruthless" method. The temporal Kalman filter could be used to reduce ghosting as well. It can be individually applied to the signal of all 456 taxels to effectively prevent flickering taxels. In combination with a spatial filter, even more persistent taxels can be removed. But this approach occupies a significant portion of the available processing power. The minimum filter on the other hand is a much simpler method to get rid of flickering taxels. A very short lag of one or two frames is enough to remove this kind of noise. The results in combination with a spatial median filter are quite pleasing. The method of choice, considering the result, simplicity and computational complexity, should be the spatio-temporal median filter. With a kernel size of $3 \times 3 \times 3$, each taxel is replaced by the median value of its 26 neighbors in space and time.

Chapter 3

Calibration

The SDH-2 does not have any internal sensors to measure force or torque. Yet there are countless applications, from parameter optimization to adaptive force control and weight measurement just to name a few, where this information is crucial for the success of a grasp. From the working principles of the installed tactile sensor, described in chapter 1.4, it is clear that the delivered voltage readings correlate with the applied force. The exact details of this relation are examined in the rest of this chapter.

Pressure p is defined as the amount of force acting per unit area, in other words

$$p = \frac{F_n}{A}$$

where F_n is the normal force applied to a surface of area A . The tactile sensor value for a particular taxel on a given sensor matrix is denoted as $t(x, y)$ and the portion of the normal force for that sensor cell is $F_n(x, y)$. It can be estimated as follows

$$F_n(x, y) = \frac{t(x, y)}{\sum_x \sum_y t(x, y)} F_n.$$

Let the number of active taxels be denoted as $N_{active} = |\{(x, y) \mid t(x, y) > 0\}|$. Since an individual taxel covers an area of size $w \times h$, the occupied area of the entire pressure profile is consequently given as $A_{active} = N_{active} \cdot w \cdot h$. Approximating the total pressure applied to the sensor matrix yields

$$p = \frac{F_n}{A_{active}}.$$

The SDHLibrary already offers the possibility to calibrate individual sensor matrices with respect to the applied pressure. First, a pressure calibration and a voltage calibration constant are defined on a per taxel basis

$$c_p = \frac{p}{N_{active}} \quad \text{and} \quad c_v = \frac{\sum_x \sum_y t(x, y)}{N_{active}}.$$

Note that the value of $t(x, y)$ is only proportional to the voltage of a sensor cell and additionally depends on the sensitivity settings and unspecified integrated signal conditioning. Anyway, the pressure applied to an individual taxel is then given as

$$p(x, y) = t(x, y) \frac{c_p}{c_v}.$$

Correspondingly, the magnitude of the equivalent normal force is

$$F_n = \frac{\sum_x \sum_y p(x, y)}{A_{active}}.$$

The question remains if it is reasonable to just sum up the voltages delivered by the tactile sensor or, on the contrary, if the correlation between voltage and pressure is nonlinear.

Haase (2011) already engaged the problem and came to the obvious conclusion that the sensitivity of individual taxels is not constant and depends on a variety of parameters. He identified three factors namely the overall nature of the external load, the number of active and inactive neighboring cells and their deviation from the mean. The force applied to a single sensor cell is then approximated using the third-order polynomial

$$F_n(x, y) \approx q_1 \cdot t(x, y)^3 + q_2 \cdot t(x, y)^2 + q_3 \cdot t(x, y)^3 + q_4$$

where the coefficients q_n depend on the mentioned criteria. For each taxel, the number of neighboring taxels, its deviation from the mean and the maximal occurring sensor value are determined from a couple of calibration frames featuring different pressure distributions. This leads to an overdetermined system of linear equations which can subsequent be solved. According to the author, this method is able to predict the effect of an applied force with an accuracy of $\pm 21.3\%$. This approach is however not considered in following calibration scheme. The reason for that is, that a couple of important effects are neglected in Haase's calibration model. In fact, the stated accuracy for a per taxel calibration seem to be too optimistic considering the effects of temperature, shear forces and declining sensitivity near the border of the sensor matrices. Instead, it is reasonable to give up the idea to obtain absolute pressure readings for individual taxels. After all, relative haptic information such as the rate by which a contact area spreads over the finger and an overall force estimation involving muscular strength, tendons and joints seem to be the basis of human haptic perception.

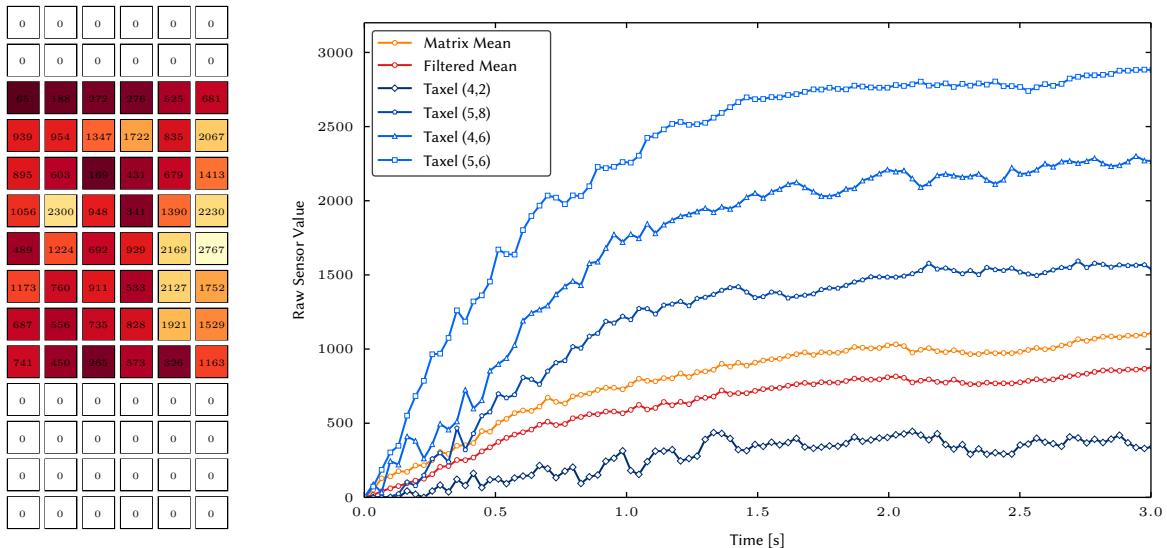


Figure 3.1: Pressure profile for illustration purposes: A flat stamp, covering the entire width of the sensor matrix, is centrally loaded with a weight of 1000 g. Although the pressure is distributed uniformly on the contact area, the resulting sensor readings show noticeable deviations. The values of four exemplary sensor cells as well as the (spatially filtered) matrix mean are plotted against time.

Figure 3.1 gives an impression of the occurring difficulties when dealing with pressure profiles delivered by the DSA. A weight of 1000 g is applied on a flat contact area covering the entire width of the sensor matrix and 2 cm in the vertical direction. The size of a sensor cell is specified to be 3.4×3.4 mm, but the border of the rubber foam adds to the sensor's width of 24.4 mm and is sensitive as well. The used stamp has a height of 20 mm but activates 8 taxel rows, actually spanning a distance of 27.2 mm. The real area is therefore 24.4×20 mm = 48.8 mm² resulting in a pressure of ~ 20.1 kPa. In order to level the effect of a few highly sensitive taxels on the overall pressure estimation, a more robust average is needed. The spatially filtered mean of all active cells gives a much better estimate. In particular, a simple Gaussian filter of size 3×3 and 0.85σ corresponding to one FWHM ¹ was used during the following experiments.

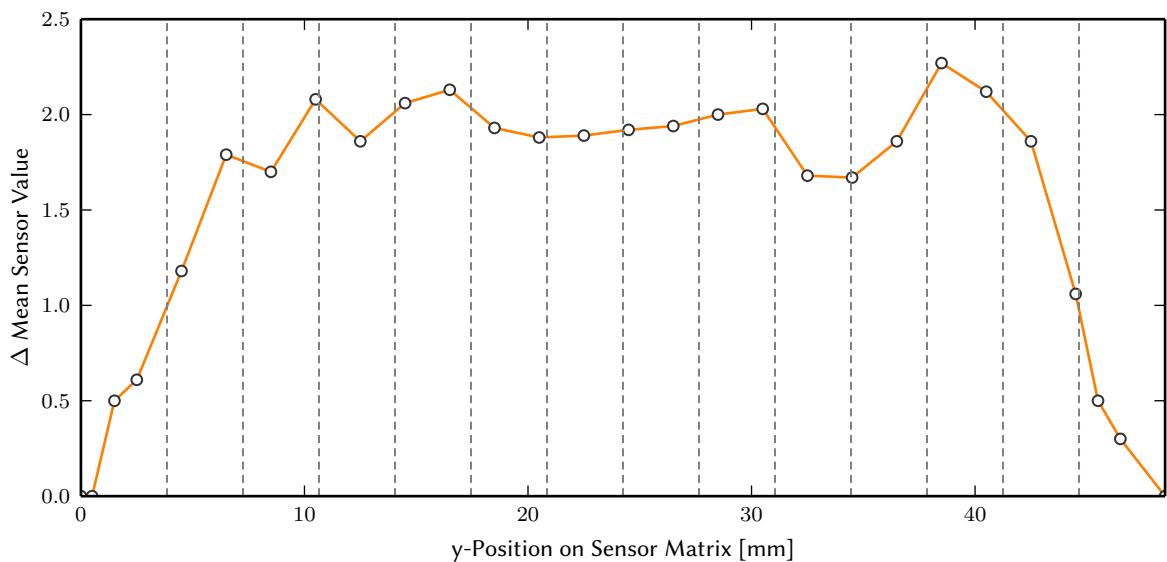


Figure 3.2: Border effect and irregularities of the tactile sensor. A point contact of constant force is moved along the y-axis of a distal sensor matrix. Dashed lines represent taxel centers. Data origin: Karakatsani (2011)

Due to the construction of the tactile sensor, there is a significant difference in the sensitivity near the border of the sensor matrix. The border of the rubber foam is not as elastic as the center since it is supported from below. This effect has already been described and quantified by Karakatsani (2011). He applied a point contact of unspecified size, but constant force, along the y-axis of the distal sensor matrix in intervals of 2 mm. The results can be seen in figure 3.2. In addition, there is a related problem that occurs when the contact area overlaps the surface area of the sensor. A rigid object may rest on the brim without causing an indentation in the middle of the sensor matrix. This problem could only be solved with a larger sensor area or by avoiding the sharp bend at the border. The current design is probably a trade-off to achieve the ingress protection class IP65.

¹ Full Width at Half Maximum: The Gaussian bell curve should reach half of its maximum value at the borders of the filter kernel. It can be shown that $\text{FWHM} = 2\sqrt{2 \ln 2} \sigma \approx 2.355 \sigma$.

3.1 Model fitting

As a result from the previous chapter on filtering, it is clear that the approach to extract characteristic values from the smoothed signal suffers from the relatively high amount of noise. It is also difficult to determine suitable tuning parameters in order to get rid of irregularities, while maintaining a short response time. Fitting an explicit function to a series of data points poses a viable alternative to smoothing technique at the expense of real time capabilities.

It is important to understand the working principles of the resistive tactile sensor in order to make an educated guess about a underlying model and hence a suitable basis for curve fitting. With reference to section 1.4 (Working principles), the tactile sensor's viscoelasticity can be described by the Kelvin-Voigt model. But the recorded pressure profiles reveal that the viscoelastic model has to be extended in order to describe a typical grasping process more accurately. Remember that this system resembles a mass-spring-damper system since it consists of a viscous damper and an elastic spring connected in parallel. As a result from section 1.5 (Kelvin-Voigt Model) and subsection 2.1.2 (Kalman filter) the differential equation for the Kelvin-Voigt model is given as

$$m\ddot{y}(t) = F(t) - \eta\dot{y}(t) - Ey(t)$$

with force F being applied to a mass m , spring constant E and viscous damping coefficient η . The initial conditions for position and velocity are $y(0) = 0$ and $\dot{y}(0) = 0$.

Like with all systems providing a storage for both potential and kinetic energy, its dynamics can be modeled using a second-order proportional time element, or PT_2 -element for short. The PT_2 -element belongs to a class of linear time-invariant transfer functions, well known in electrical engineering and control theory. For an in-depth view, reference is made to the German language standard works Lunze (2012) and Unbehauen (2008). The differential equation of a PT_2 -element in standard form is given as

$$T^2\ddot{y}(t) + 2dT\dot{y}(t) + y = Ku(t) \quad (3.1)$$

where T is a time constant, $d > 0$ is a damping factor and K is frequently referred to as system gain. $y(t)$ and $u(t)$ are of course the output and the control signal. The mass-spring-damper system is therefore a PT_2 system with

$$T^2 = \frac{m}{E} \quad d = \frac{\eta}{E} \quad \text{and} \quad K = \frac{1}{E}.$$

To obtain the step response, that is the output signal of the control element for a unit step input, the differential equation is usually modified in the frequency domain. The standard method involves the Laplace transform, an integral transform that turns linear differential equations into much easier to solve polynomial equations. To give a short overview, the Laplace transform is formally defined as

$$F(s) = \mathcal{L}\{f(t)\} = \int_0^\infty e^{-st} f(t) dt$$

with the complex frequency variable $s = \sigma + i\omega$. The integral only exists if $f(t)$ is locally integrable, i.e. $\int_0^\infty |f(t)| dt < \infty$. Just like with the Fourier transform, the Laplace transform corresponds to a change of basis. But instead of representing the function as a sum of sinusoids, $f(t)$ is expressed as a superposition of moments. Reference to subsection 5.3.1 (Image moments) is made for more information on moments. Since this transform is a very popular tool in engineering, convenient sets of properties as well as correspondence tables of transforms and

their inverse are readily available for various types of functions. With the help of the so called linearity and differentiation properties, the Laplace transform of equation (3.1) becomes

$$(T^2 s^2 + 2dT s + 1)Y(s) = KU(s).$$

A common way to describe the behavior of a system in the frequency domain is the equation

$$Y(s) = G(s)U(s)$$

where $G(s)$ is called the transfer function between control $U(s)$ and system response $Y(s)$. The transfer function of the PT_2 system is therefore given as

$$G(s) = \frac{K}{T^2 s^2 + 2dT s + 1}.$$

Being the improper integral of the Dirac delta function, i.e.

$$\sigma(t) = \int_{-\infty}^t \delta(s) \, ds,$$

there exists no Laplace transform of the step function $\sigma(t)$. But with the help of the antiderivative at the bounds $[0, \infty)$, a modified step function can be defined assuming $\text{Re}(s) > 0$. Knowing that the modified unit step function as a control input $u(t)$ becomes $\frac{1}{s}$ in the Laplace domain, the system's response becomes

$$Y(s) = \frac{K}{T^2 s^2 + 2dT s + 1} \cdot \frac{1}{s}.$$

The Laplace transform is a linear operator. In order to transform $Y(s)$ back to the time domain, the rational function is therefore expanded into partial fractions such that the individual sums can be transformed independently using correspondence tables. But depending on the value of the dampening factor d , there are three different pairs of roots in the complex denominator of $G(s)$, remember $s = \sigma + i\omega$. For the non-oscillating case corresponding to $d > 1$, we have two different real poles

$$\alpha_{1,2} = \frac{1}{T}(-d \pm \sqrt{d^2 - 1}).$$

The corresponding partial fraction decomposition is then

$$\begin{aligned} Y(s) &= \frac{K}{(s - \alpha_1)(s - \alpha_2)s} \\ &= \frac{K}{\alpha_1(\alpha_1 - \alpha_2)} \frac{1}{(s - \alpha_1)} - \frac{K}{\alpha_2(\alpha_1 - \alpha_2)} \frac{1}{(s - \alpha_2)} + \frac{K}{\alpha_1 \alpha_2} \frac{1}{s}. \end{aligned}$$

With the help of the inverse transformation rule

$$\mathcal{L}^{-1}\left\{\frac{k_i}{(s - \alpha_i)}\right\} = k_i e^{\alpha_i t}$$

and the relation between the poles of a system and the corresponding time constants

$$T_i = -\frac{1}{|\alpha_i|},$$

the final step response of a PT_2 -element with a dampening factor $d > 1$ is given as

$$a(t) = K - \frac{K}{T_1 - T_2} (T_1 e^{-\frac{t}{T_1}} - T_2 e^{-\frac{t}{T_2}})$$

where the individual time constants of the underlying systems are given as

$$T_{1,2} = \frac{T}{d \pm \sqrt{d^2 - 1}}.$$

All polymer materials show creep behavior under load and silicon rubber is no exception. As mentioned by Weiss and Worn (2005b), the electrical resistance of a sensor cell becomes continuously lower when a constant force is applied. The authors claim to compensate for the creep effect, but the signal still shows some drift. The proportional time element is therefore modified to contain a linear drift term., i.e.

$$a(t) = K - \frac{K}{T_1 - T_2} (T_1 e^{-\frac{t}{T_1}} - T_2 e^{-\frac{t}{T_2}}) + mt.$$

Figure 3.3 gives an overview of the derived step response $a(t)$.

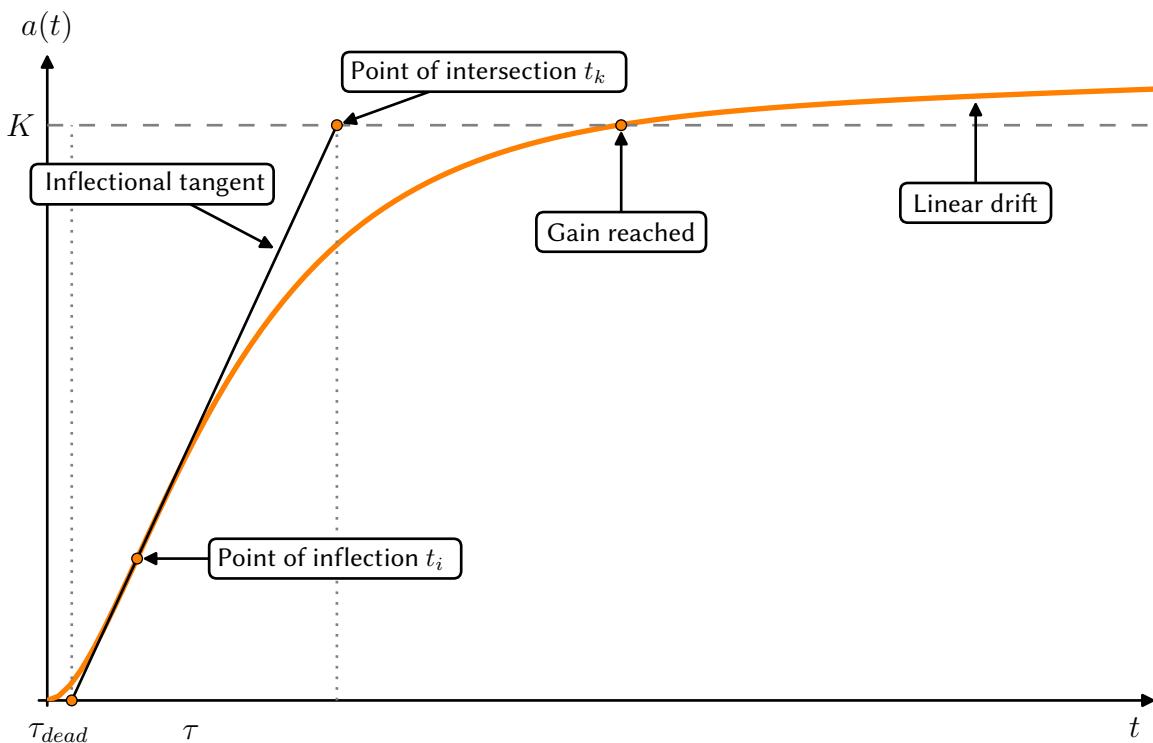


Figure 3.3: Step response of a PT_2 -element with linear drift.

Depending on the application, it might be of interest to know when the signal has reached the steady-state gain or more precisely, when the linear drift takes over the system dynamics. After the modified step response function has been fitted, it is easy to determine a point of reference such that $a(t) = K$. An optional drift compensation for example would operate from that point onwards since the grasping movement has momentarily come to a rest. Other interesting properties of a controlled system are the dead time τ_{dead} and the process lag τ . Control theory tool kits offer rules-of-thumb to estimate the characteristics of a dynamic system in order to infer the control parameters. But since the fitted function $a(t)$ already contains the necessary

parameters, it is easy to extract the characteristic values algebraically. The first and second derivative of the step response of a PT_2 -element with linear drift are therefore given as

$$a'(t) = -\frac{K}{T_1 - T_2}(-e^{-\frac{t}{T_1}} + e^{-\frac{t}{T_2}}) + m$$

and

$$a''(t) = -\frac{K}{T_1 - T_2}\left(\frac{e^{-\frac{t}{T_1}}}{T_1} - \frac{e^{-\frac{t}{T_2}}}{T_2}\right).$$

Setting the second derivative $a''(t)$ to zero yields the point of inflection

$$\begin{aligned} t_i &= -\frac{1}{T_1 - T_2} \ln\left(\frac{T_2}{T_1}\right) T_1 T_2 \\ &= \frac{1}{2} \frac{T \ln\left(\frac{d+\sqrt{d^2-1}}{d-\sqrt{d^2-1}}\right)}{\sqrt{d^2-1}}. \end{aligned}$$

The equation of the inflectional tangent in the point-slope form is then given as

$$a(t) - a(t_i) = a'(t_i)(t - t_i)$$

and thus it follows that the intersection point t_k of the tangent with the gain K can be determined with

$$t_k = \frac{K - a(t_i)}{a'(t_i)} + t_i.$$

The dead time τ_{dead} is given as

$$\tau_{dead} = -\frac{a(t_i)}{a'(t_i)} + t_i.$$

And the slope of the equivalent time tangent is hence

$$m_{\tau_{dead}+\tau} = \frac{a(t_k)}{t_k}.$$

In the context of control theory, the ratio τ/τ_{dead} is important for the controllability of the dynamic system. But for the calibration task it is sufficient to have an estimate of the time constants. They anyway depend on a variety of unknown parameters such as the applied impulse when grasping an object. The series of measurements indicate however that process control would be difficult to achieve especially with time-critical applications in mind. In other words, the tactile sensor is relatively slow compared to the time required by a control loop for corrective actions. It is worth mentioning, that H. Wörn und T. Haase (2011) already developed a motion control concept for the internal controller of the SDH-2. In essence, it is a closed-loop position control with force constraints, designed for reactive grasping skills in combination with a user-friendly interface. The authors thereby encountered a relatively large transient response and attribute the settling time to the low frame rate of the tactile sensors. But in reality, this lag is an inherent property of the measurement principle.

The derived model can now be fit either to the time series of individual taxels or their spatially filtered mean as mentioned previously. This should be carried out in a least squares fashion, but unfortunately the variable t as well as the parameters do not appear linearly in $a(t)$. There is

no obvious way to use substitutions or other algebraic manipulations to linearize the function. Pretending to know K, T, d or m may lead to a transformed formulation, but the parameters obtained by such a linearization could be far away from those of the original problem, since the residual sum of squares of a completely different objective function would be minimized.

There is no other option in sight than to solve the nonlinear least squares problem. The popular Levenberg-Marquardt algorithm is a robust iterative algorithm to do just that. The details are beyond the scope of this thesis, but it is important to start with a good initial guess for the parameters. Otherwise the algorithm tends to get stuck in a local minimum. Fortunately the gain K is close to the maximal value after a short period of time and once the calibration model is determined, it is possible to relate the drift m to K . Time constant and damping factor however always depend on the performed grasp and cannot be known beforehand.

For numerical reasons, the time variable t is scaled by a factor of $s = \max y / \max t$ such that it is in the same range as the sensor values. The obtained estimates of T and m have then to be adapted accordingly. Another trick is the modification of $T_{1,2}$:

$$\frac{T}{d \pm \sqrt{d^2 - 1}} \quad \text{instead of} \quad \frac{T}{d \pm \sqrt{d^2 - 1}}$$

leads to a faster convergence of the algorithm.

Although in principle only four data points are sufficient to fit the curve, the parameter K is thereby usually underestimated and m consequently overestimated. In order to reliably determine the parameters of $a(t)$, the signal should already have reached the gain. A couple of additional data points in the beginning of the drift phase further improve the fit.

Figure 3.4 presents a couple of exemplary curve fits and thereby observed issues. Regarding figures 3.4a and 3.4b it is striking that, although the grasps were performed with the same settings, it takes three times longer for the signal to settle when grasping a foam ball compared to a hockey ball. Figure 3.4c illustrates the phenomenon of flickering taxels. This effect usually appears near the sensor's lower sensitivity limit and is amplified by the fact that only active taxels contribute to the matrix average. As a reminder and assuming there actually are active taxels, the spatially filtered arithmetic mean is defined as

$$\bar{t} = \frac{\sum(G * t)(x, y)}{|\{(x, y) \mid (G * t)(x, y) > 0\}|}. \quad (3.2)$$

The applied Gaussian filter G tends to reduce the flickering, but temporal filtering is inevitable in order to completely avoid it. On the other side of the pressure range, sensor cells go into saturation as shown in 3.4d. Both effects can be countered with appropriate sensitivity settings, but obviously there is a trade-off to be made.

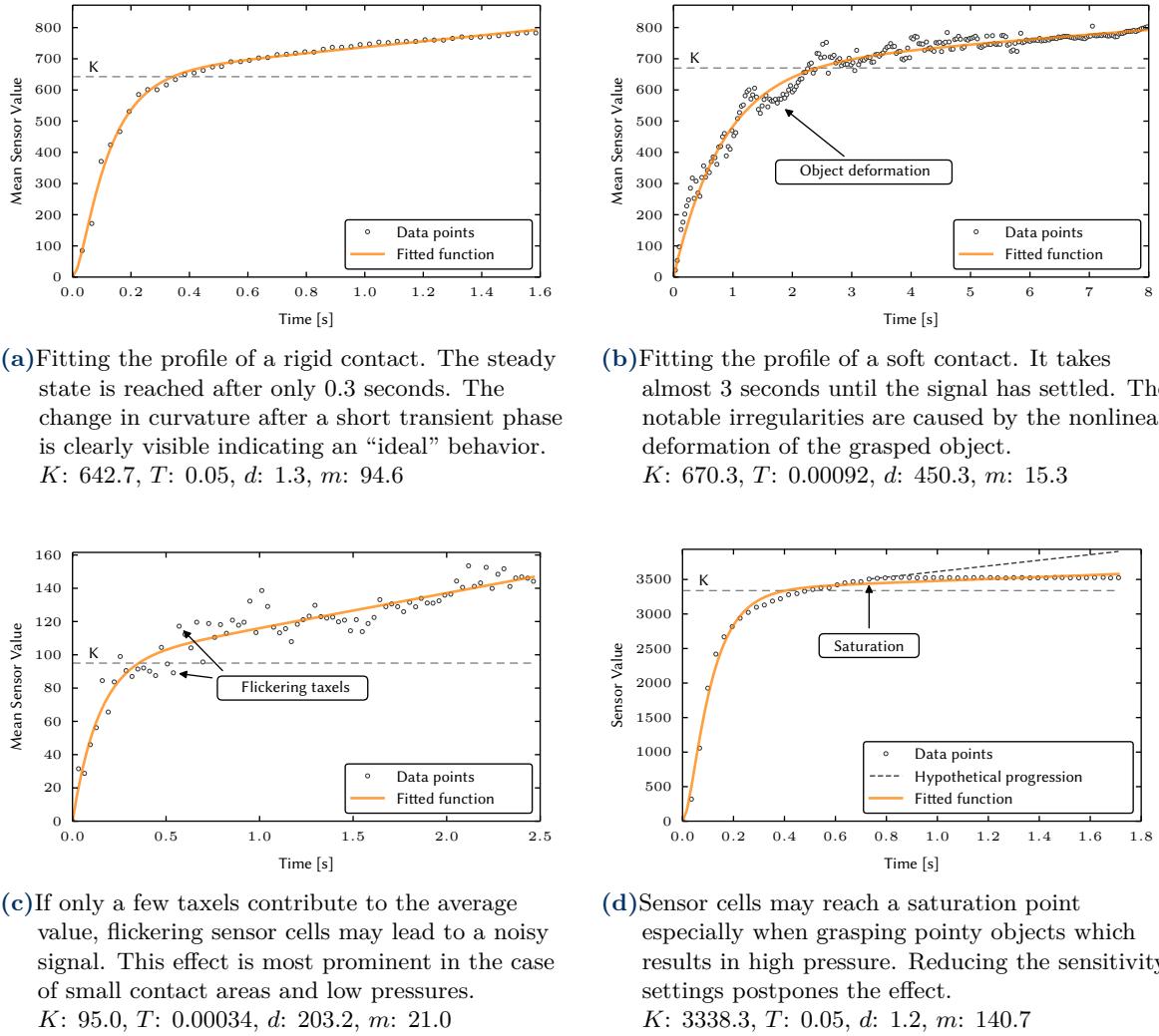


Figure 3.4: Exemplary curve fits and observed effects.

3.2 Temperature

Since the material properties of the rubber foam change with temperature, the sensitivity of the tactile sensors depends on the SDH-2’s degree of heating. The temperature measurement command of the SDHLibrary returns with a 9-element array referring to the controller and driver temperature sensors installed in the SDH-2. The first seven sensors are placed close to the corresponding axes motors 0-6. They serve as proxies for the temperature of the adjacent tactile sensors. The remaining two temperature readings originate from the controller chip of the FPGA and the printed circuit board.

The SDH-2 has a tendency to overheat, especially during long-lasting tasks or if the axis controllers are not switched off in between experiments. It has been reported that the tactile sensors suffer from wear and tear, probably caused by shear forces combined with high temperatures. A simple ventilator to support the passive cooling improves the situation, but it is nonetheless interesting to observe the relationship between temperature and sensor readings.

Figure 3.5 illustrates the rising temperature readings of the SDH-2 under static load. Starting at room temperature, the hand is left alone for an hour while firmly holding position. It is striking that the proximal finger joints are warmer than the distal ones which is plausible since

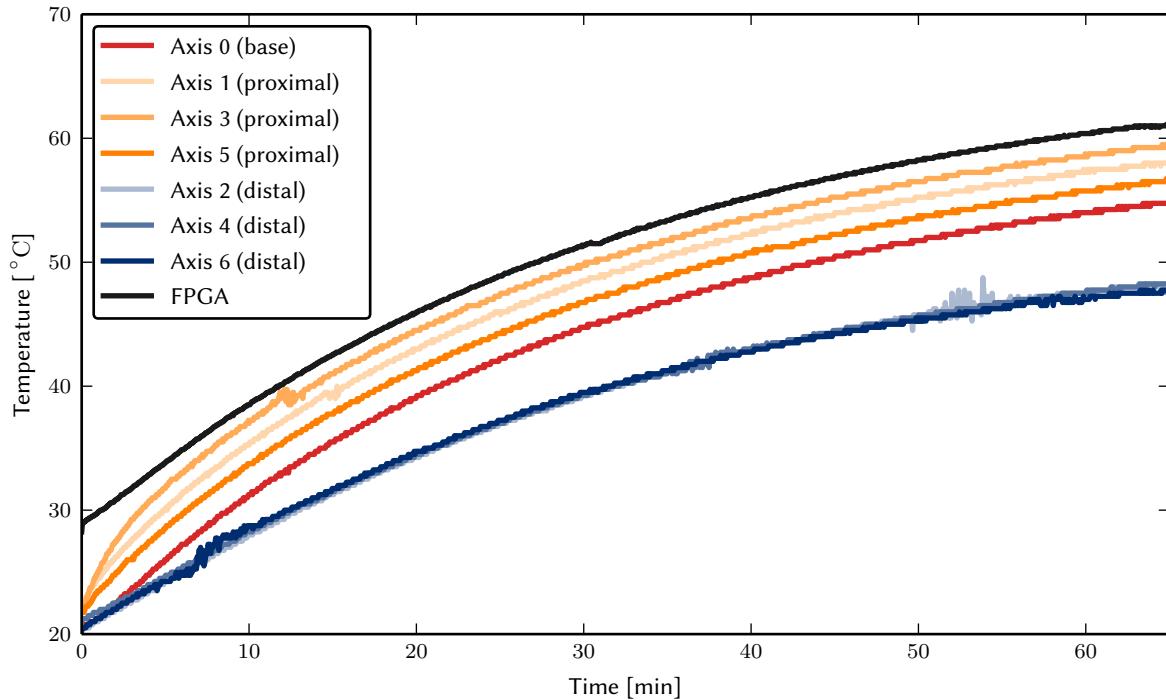


Figure 3.5: Long-term measurement of the rising temperatures under static load. The power consumption of the grasp is $24\text{ V} \times 0.4\text{ A} = 9.6\text{ W}$. The sensor matrices are not covered by the grasped object, which in addition, is a poor heat conductor.

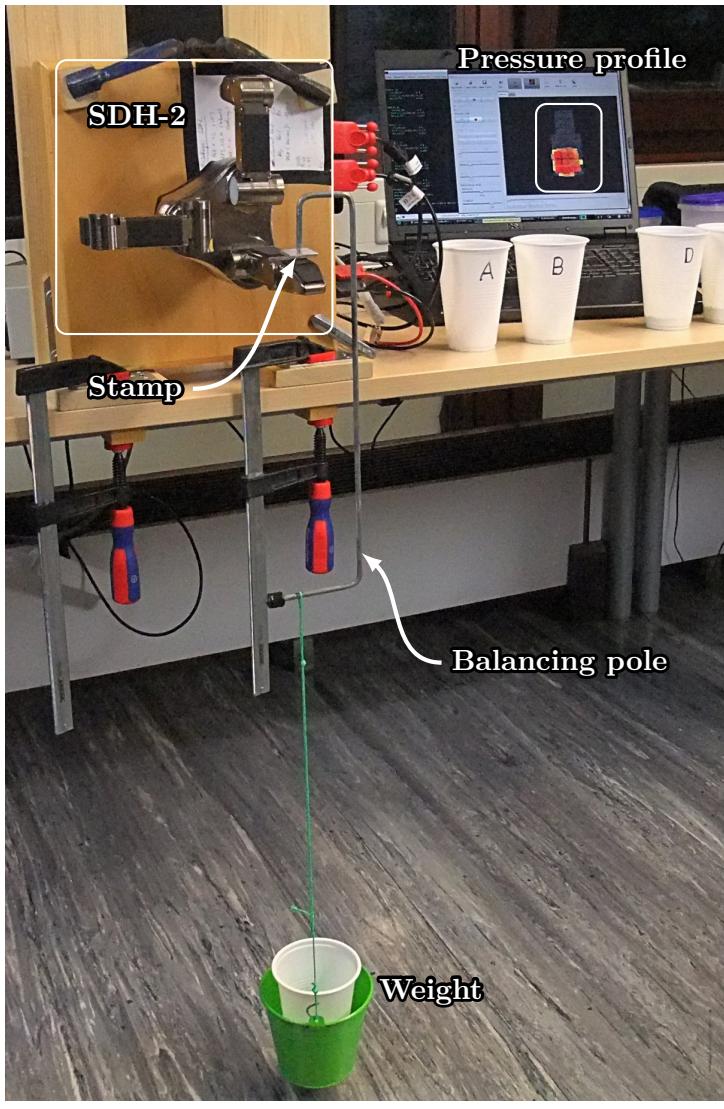
they get heated from both sides. The reason why the proximal temperatures are spread out over a range of about 5 °C while the distal joints show the same temperature is unknown. The performed grasp is by all means symmetrical. For safety reasons, the built-in FPGA seems to stop working at 60 °C. It should be noted that the hand was mounted on its platform during the measurements. The temperature curves are slightly different when the SDH-2 is installed on a robot's arm.

3.3 Pressure calibration set-up

The results obtained in the previous sections are now used to calibrate the signal's gain K and drift m for measured temperature, applied pressure and resulting voltage, i.e. sensor readings.

The pressure calibration set-up is depicted in figure 3.6. While being not as elaborated as calibration rigs found in literature (e.g. Weiss and Worn (2005b)), its design is simple and effective. The SDH-2 is mounted on a platform such that there is sufficient space below the tactile sensor for a weighted pendulum. An exchangeable stamp is placed on the surface of the tactile sensor and a centered indentation in the flat connection piece makes sure that the applied force is uniformly distributed on the stamp via a narrow contact point. Removable cups, filled with an arbitrary amount of lead pellets can be placed in a bucket at the lower end of the pendulum. The fine-tuning of the apparatus is then carried out at the bottom of the balancing pole. One advantage of the separate stamp is that once placed on the sensor matrix, it can remain on the same position throughout the entire experiment while the bucket and the pole are readjusted.

To begin with, experiments with stamps of different sizes were carried out in order to find a suitable trade-off between a preferably large pressure range and the avoidance of saturation.



(a) Calibration rig



(b) Balancing pole mounted on exchangeable stamp



(c) Removable cup filled with lead pellets



(d) Stamps of different size and shapes

Figure 3.6: Pressure calibration set-up.

After a couple of experiments, a square stamp with a side length of 10.5 mm was chosen for a series of measurements with varying weights at different temperatures. The sensitivity of the tactile sensor was set to the maximum value of 1.0 while the matrix threshold was lowered to 100. These settings are further discussed in the next chapter. Since in our case the weight is a normal force acting on a contact area of 1.1025 cm^2 , the applied pressure in the conducted experiment i can be calculated as

$$p_i = \frac{m_i g}{A} = \frac{m_i \cdot 9.81 \text{ m/s}^2}{0.00011025 \text{ m}^2}$$

where the mass m_i is increased by 100 g in each step until a notable amount of taxels reach saturation. The actual weights, including the mass of the bucket (100 g), balancing pole (90 g) and stamp (5 g) as well as the resulting pressures are listed in 3.1. Each measurement lasts for 15 seconds per imprint and is repeated three times.

Table 3.1: Pressure calibration set-up. Contact area: 1.1025 cm^2 .

<i>Applied weight</i>	<i>Resulting pressure</i>
200 g	17796 Pa
300 g	26694 Pa
400 g	35592 Pa
500 g	44490 Pa
600 g	53388 Pa
700 g	62286 Pa
800 g	71184 Pa
900 g	80082 Pa

Given the inherent uncertainties of the measurement principle, the same series of experiments was carried out in three different temperature ranges to simplify the calibration task. Temperatures around 30°C are considered “room temperature”, the most common temperature range $\sim 40^\circ\text{C}$ is denoted as operating temperature and the last experiments take place at high temperatures of $\sim 50^\circ\text{C}$. Figure 3.7 gives an impression of the obtained fitted models.

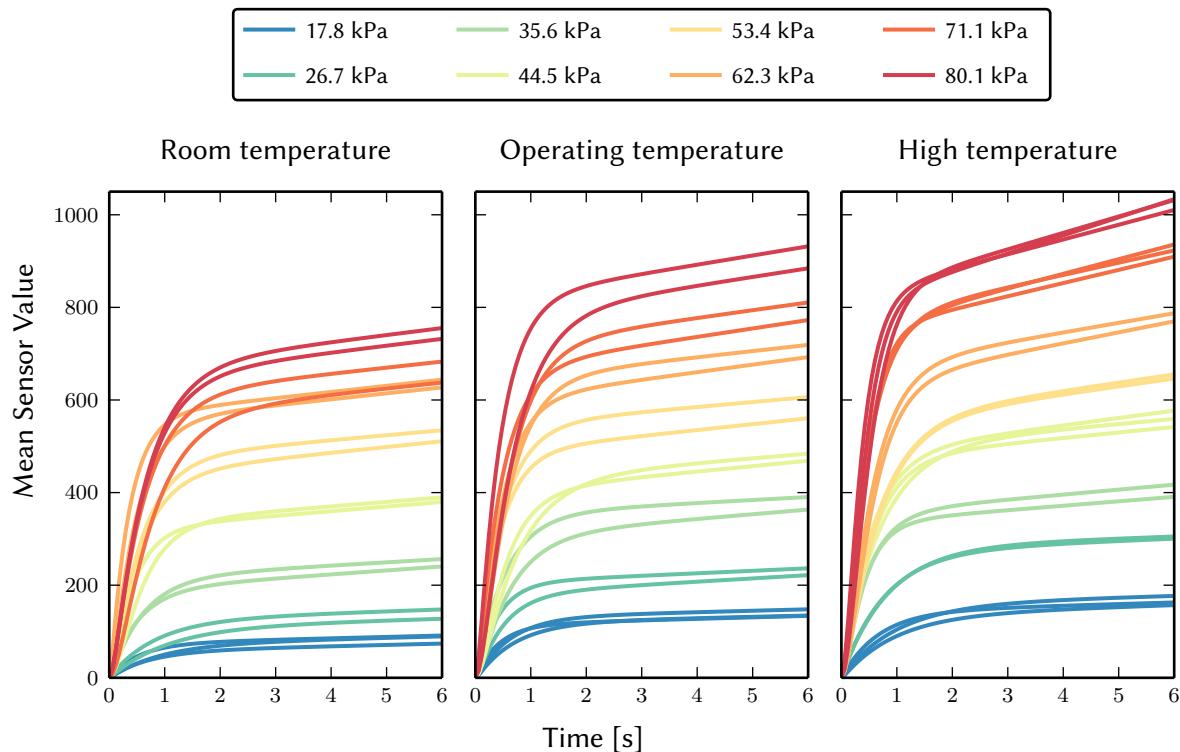
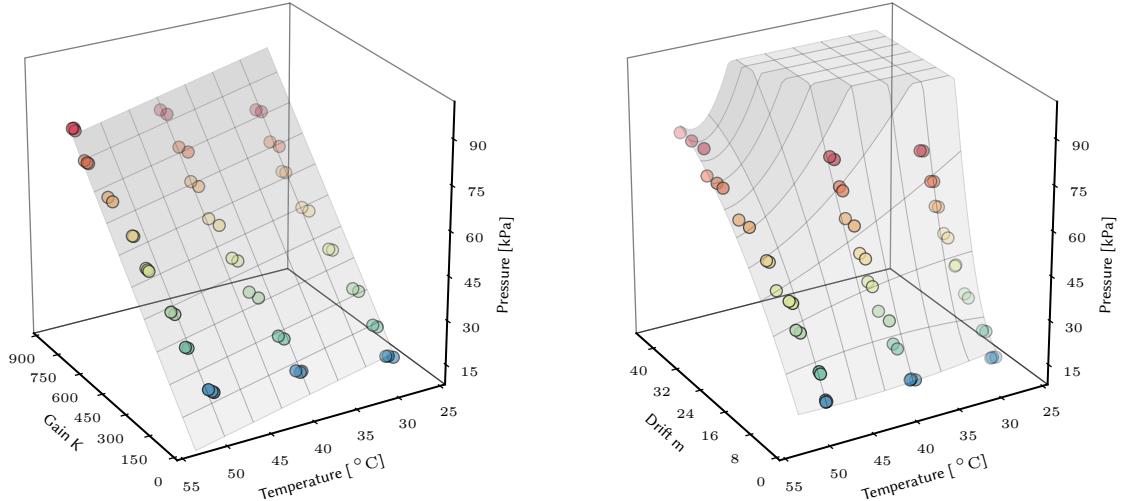


Figure 3.7: Pressure calibration for different temperature ranges. For reasons of clarity, only the fitted function $a(t)$ is drawn. There is an approximately linear relationship between the applied pressure and the resulting mean sensor value as well as the slope m of the drift.



(a) Gain K : A plane constitutes a reasonably good model.

(b) Drift m : A two-dimensional second degree polynomial is a needed to explain the observed measurements.

Figure 3.8: Pressure calibration in \mathbb{R}^3 .

Ideally a calibration process involving temperature, applied pressure and resulting sensor readings aims to find a mapping from \mathbb{R}^3 to \mathbb{R}^2 . Given the number of active taxels, the fitted model and the current temperature, it is possible to give an estimate of the pressure and thus the applied force.

Figure 3.8a illustrates the fitted calibration model for the gain K using a two-dimensional first degree polynomial $P_K(x, y)$, i.e. a plane.

$$\begin{aligned} P_K(x, y) &= (a_0 + a_1x)(b_0 + b_1x) \\ &= c_0 + c_1y + c_2x + c_3xy. \end{aligned}$$

A plain Ordinary Least Squares solution yields for Matrix 1:

$$\begin{aligned} c_0 &= 31.8973801 & c_1 &= -0.6249430 \\ c_2 &= 0.0977683 & c_3 &= -0.00000386 \end{aligned}$$

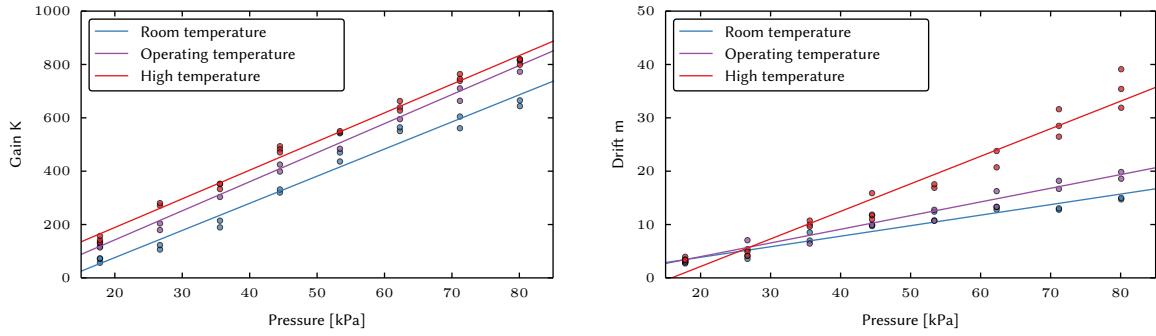
In the case of the sensor drift m , a two-dimensional second degree polynomial $P_m(x, y)$ is necessary to capture the behavior at high temperatures as seen in figure 3.8b.

$$\begin{aligned} P_m(x, y) &= (a_0 + a_1x + a_2x^2)(b_0 + b_1x + b_2x^2) \\ &= c_0 + c_1y + c_2y^2 + c_3x + c_4xy + c_5xy^2 + c_6x^2 + c_7x^2y + c_8x^2y^2. \end{aligned}$$

A least-squares fit for the available measurements is

$$\begin{array}{lll} c_0 = 117.6964412 & c_1 = -4.8357055 & c_2 = 0.0528554 \\ c_3 = -35.3068198 & c_4 = 1.6954670 & c_5 = -0.0184535 \\ c_6 = 2.6927690 & c_7 = -0.1117506 & c_8 = 0.0011429 \end{array}$$

As explained in the introduction of this chapter, there are significant uncertainties in the acquired data of the tactile sensor. A detailed goodness-of-fit analysis is therefore unnecessary and the calibration model remains only a rough estimate.



- (a) The gain K is linearly dependent on pressure and temperature.
(b) The Drift m increases with rising temperatures. Linearity is questionable but applicable in the observed temperature range.

Figure 3.9: Pressure calibration: simplified projection

Since the series of measurements can be roughly binned in three temperature ranges, the calibration surface in \mathbb{R}^3 can be presented in \mathbb{R}^2 for a better understanding. Figures 3.9a and 3.9b show slices along the mentioned temperature ranges.

3.4 High sensitivity mode

Depending on the version of the tactile sensors DSA9205 and DSA9210, indicated by a feature flag, the pressure sensitivity of the matrices can be adjusted by the user. According to the reference manual, the settings tune the amplification factor of the integrated analogue signal conditioning circuit. While high sensitivity settings are very useful in various applications, they do come with a downside. The sensor cells go faster into saturation. The default value is 0.5, but the realization of an adaptive algorithm, presumably using a binary search strategy would be possible, as indicated by figure 3.10. It should be noted that the reconfiguration takes a few hundred milliseconds since the transmission of data frames has to be stopped beforehand. But this way, the dynamic range of the sensor could be greatly improved if needed.

The tactile sensor controller DSACON32-M is connected to the FPGA of the SDH-2 via an internal RS-232 serial connection. The measurement values are 12 bits wide but are transferred in 16 bit words. Generously ignoring header information and checksums, the packet size of the three proximal and three distal sensor matrices amounts in total to $3 \cdot (6 \cdot 14 + 6 \cdot 13) \cdot 16$ bits = 7776 bits. At a maximal transfer rate of 115,200 bits per second, this results in a frame rate of about 15 frames per second. Since most of the taxel values are typically zero, a higher throughput can easily be achieved by a suitable data compression algorithm. The DSACON32-M controller combines the six sensor matrices to a single frame which is then compressed using run-length encoding. On the receiver's side, the data frame is decompressed again by the SDHLlibrary. Given that the matrix dimensions are known, an offset to the matrix data is then calculated. This way, frame rates of about 30 fps can be achieved in the majority of cases. But the inevitable ground noise of the transducer is another important aspect in the data acquisition process. For the functioning of the run-length encoding, it has to be ensured that the signal of an unloaded sensor cell effectively amounts to zero. Therefore a matrix threshold θ_m was introduced to separate the noise level from real excitations.

$$t(x, y) = \begin{cases} t_{raw}(x, y) - \theta_m, & \text{if } t_{raw}(x, y) > \theta_m \\ 0, & \text{otherwise} \end{cases}$$

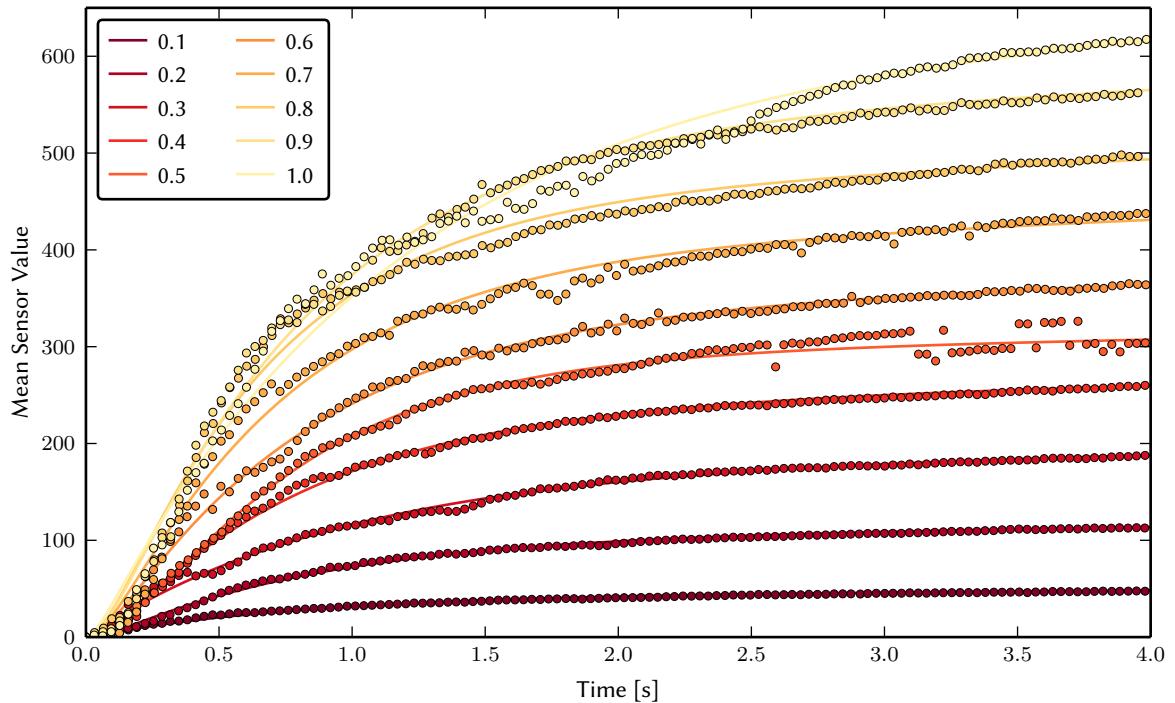


Figure 3.10: Sensitivity settings from 0.1 to 1.0. The estimated values of K are indeed related by the given factor. Weight: 700 g, square stamp: 1.1025 cm^2 , threshold: 100

Figure 3.11 illustrates the ground noise at threshold settings of 0. The resulting frame rate drops to approximately 9 fps due to the failing run-length encoding.

The default value for the threshold is 150 which is quite a robust choice with respect to noise and ghosting effects. But this noise suppression comes of course at the expense of sensitivity. Since the detection of “first contacts” is a crucial aspect of robotic grasping, it is interesting to squeeze every last drop of sensitivity out of the available hardware. The noise level not only depends on the individual matrix, but also on the current temperature.

In an attempt to find the individual matrix thresholds, the same measurements depicted in 3.5 are used to calibrate for temperature and noise level. Figure 3.12 sums up the entire calibration process. The measurement series consists of 1296 tactile sensor and temperature frames taken every three seconds over a period of about 65 minutes. Just like the already examined gain, the ground noise is obviously linearly dependent on the temperature. The temperature sensors have resolution of 0.25°C but since the temperature does not rise linearly within the time series, the number of observations changes for different temperature ranges. In order resolve the subsequent low temperature bias, the measurements are divided into 20 equally sized bins. The data is then fitted using a first order polynomial in a weighted least squares fashion with the weights being the reciprocal standard deviations of the bins. There are quite a few outliers in the dataset, but flawed taxels can be removed using spatial filters. For the practical application it is important to have a good idea of the interval in which future sensor values will fall. For the sake of completeness and to put some order in the terminology of statistics, the notion of a prediction interval is derived in the following. Any introductory text on statistics like chapter two of Bonate (2011) may serve as a reference.

In the case of a simple linear regression model $y_i = \alpha + \beta x_i + \varepsilon_i$ with $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, the expected value is $\hat{y}_i = \hat{\alpha} + \hat{\beta} x_i$. The residual sum of squares (RSS) also known as sum of squared

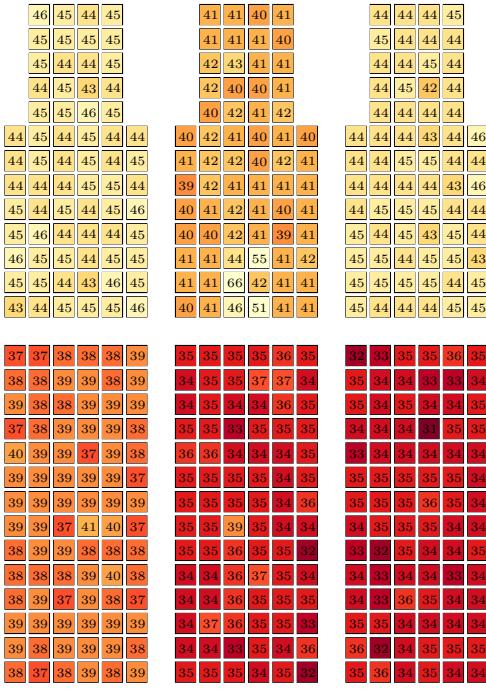


Figure 3.11: The ground noise of the transducers differs from matrix to matrix. In addition, individual taxels might deviate from the mean.

errors of prediction (SSE) is given as

$$\text{RSS} = \sum_{i=1}^n (\varepsilon_i)^2 = \sum_{i=1}^n (y_i - (\hat{\alpha} + \hat{\beta}x_i))^2.$$

In the context of regression, the mean squared error (MSE) for a vector $\hat{\mathbf{y}}$ of n predictions is defined as the residual sum of squares divided by the number of degrees of freedom for the error.

$$\text{MSE} = \frac{\text{RSS}}{n-2}$$

where the degree of freedom amounts to the sample size n minus the number of model parameters slope and intercept. The root-mean-square error (RMSE) of the predicted values $\hat{\mathbf{y}}$ is then given as

$$\text{RMSE} = \sqrt{\text{MSE}}.$$

For an unbiased estimator, the MSE is an estimator of the error variance and thus the RMSE is the square root of the variance, also known as the standard error of the estimate (SE). But for the general case, the variance of a predicted mean value is given as

$$\begin{aligned} \text{Var}(\hat{y}_i) &= \text{Var}(\hat{\alpha} + \hat{\beta}x_i) \\ &= \text{Var}(\hat{\alpha}) + (\text{Var} \hat{\beta}) x_i^2 + 2x_i \text{Cov}(\hat{\alpha}, \hat{\beta}) \\ &= \text{MSE} \cdot \left(\frac{1}{n} + \frac{(x_d - \bar{x})^2}{\sum(x_i - \bar{x})^2} \right) \end{aligned}$$

where the last equation is a consequence of the independence of $\hat{\alpha}$ and $\hat{\beta}$ and the substitution of the residual variance with the MSE. The variance of the predicted distribution of the residuals

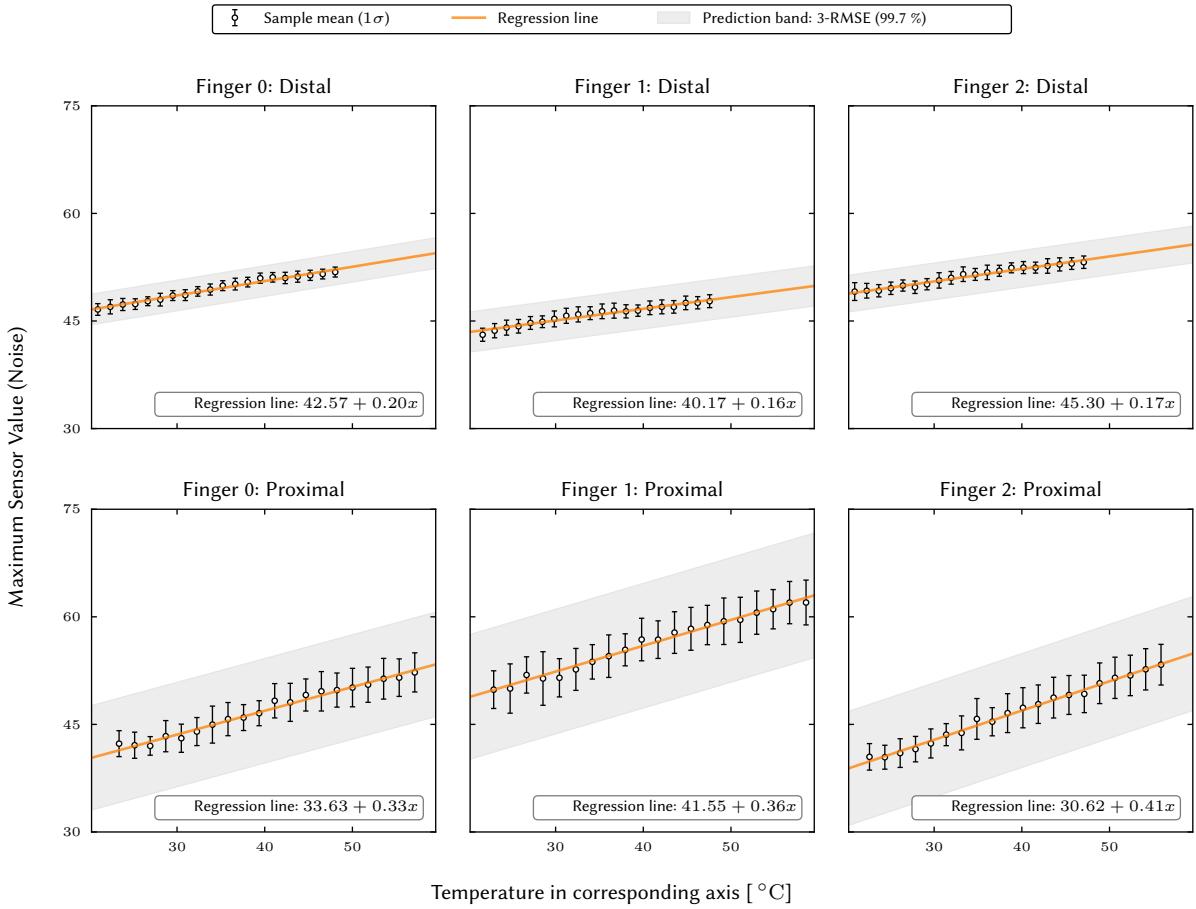


Figure 3.12: Ground noise calibration: The matrix thresholds can be set according to the upper bounds of the estimated prediction intervals in order to achieve high sensitivity while maintaining the usual frame rates. The series of measurements consists of 1297 tactile sensor- and temperature readings, taken every 3 seconds. Data is then divided into 20 equally sized bins to overcome the low temperature bias of the measurements, cf. 3.5. The reciprocal standard deviations of the bins serve as a factor in the weighted least-squares regression. See text for details.

is respectively given as

$$\begin{aligned}
 \text{Var}(y_i - \hat{y}_i) &= \text{Var}\left(y_i - (\hat{\alpha} + \hat{\beta}x_i)\right) \\
 &= \text{Var}(y_i) + \text{Var}(\hat{\alpha} + \hat{\beta}x_i) \\
 &= \text{MSE} \cdot \left(1 + \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum(x_i - \bar{x})^2}\right) \\
 &= \text{SE}(\hat{y}_i).
 \end{aligned}$$

A $(1 - \alpha)$ prediction interval at a given value of x is

$$\hat{y}_i \pm t_{n-2,1-\alpha/2} \cdot \text{SE}(\hat{y}_i).$$

For a large sample size, the prediction interval can be approximated with

$$\hat{y}_i \pm t_{n-2,1-\alpha/2} \cdot \text{MSE}.$$

According to the Student's t-distribution, future temperature measurements x_i therefore fall into the interval $\hat{y}_i \pm 3 \cdot \text{MSE}$ with a probability of 99.7%. An appropriate sensor threshold for matrix m can now be computed according to

$$\theta_m = \hat{\alpha}_m + \hat{\beta}_m x_m + 3 \cdot \text{RMSE}_m.$$

The regression parameters $\hat{\alpha}_m$ and $\hat{\beta}_m$ as well as RMSE_m are saved in an xml-file and could be tweaked manually if wear and tear changes the sensor's behavior. Since the lowering of the thresholds increases ghosting effects, spatial filters as discussed in the previous chapter are a natural add-on to the high sensitivity mode.

Chapter 4

Forward kinematics

So far, the positions of individual taxels were only considered with respect to the two-dimensional coordinate system of their corresponding matrices only. But it is obvious, that the spatial extent of a pressure profile in three dimensions carries important additional information about the grasped object.

The position controller of the SDH-2 is able to deliver joint angle readings at an average data rate of about 110 Hz. Due to bandwidth restrictions of the hand's internal serial communication, this value may vary significantly if additional command or data transmissions are pending. For example the library delivered by SCHUNK offers a function to wait until the movement of a particular axis has finished, and if so for what reason. This is especially useful in the position controller mode where a target position can be specified and subsequently approached. But since the function call returns immediately while continuously querying the axis states until all axes are no longer moving, no joint angle readings can be captured. It was therefore necessary to introduce alternative non-blocking versions of various library functions in order to collect pressure profiles annotated by joint angles or temperature readings for that matter. The absolute angle encoders have a step size smaller than 0.02° and have to be calibrated for exact measurements. But the actual weak points when measuring three-dimensional positions turned out to be the flimsily mounted finger sections and accordingly loose-fitting connections. Although the resulting play is well above the range of the specified precision, the readings are consistent and repeatable and therefore still of great use.

The tactile sensor controller transmits data at a rate of roughly 30 frames per second depending on the number of active sensor cells. Due to the internal run-length encoding of the sensor data, this value varies significantly over time and can lead to timeouts of the hard-coded intervals in the SDHLibrary. This may even result in non-responsive behavior requiring a manual reset of the hand. In any case, the limiting factor when applying forward kinematics to the tactile sensor matrices is the frame rate of the tactile sensor itself rather than the speed of the joint angle encoder.

In order generate a "3D-point cloud" of active (or inactive for that matter) sensor cells, it is necessary to find the dynamic relation of the moving fingers from the hand's base to the tips. Different coordinate frames are therefore attached to the joints and links. The corresponding kinematic equations can subsequently be obtained for a given set of joint angles. It is beneficial to use homogeneous coordinates to describe the transformation from one coordinate system to the next in the so called kinematic chain. See Craig (2005) for further details. There are an arbitrary number of ways to define the direct or forward kinematics of the gripper. Haase (2011) for example implemented his own model of the SDH-2. The classic Denavit-Hartenberg method is commonly used to create an easily comprehensible description of a robot's kinematics using four parameters only. There are proximal and distal variants of the original formulation,

depending on whether the coordinate frame is located at the beginning or the end of each link. A state of the art introduction to robot kinematics and the Denavit-Hartenberg convention can be found for example in Spong et al. (2004) and Cubero (2007). In the distal convention, coordinate frames O_{i-1} and O_i are attached to the joints between a link i . A homogeneous transformation matrix ${}^{i-1}_i\mathbf{T}$ is assigned to it and relates the position of body i with respect to body $i - 1$. The axes of pivot joints are aligned along z_i while the common perpendicular line from z_{i-1} to z_i determines the direction x_i of the next frame. The offsets d_i refer to the distance between x_{i-1} and x_i along z_{i-1} . The joint angle θ_i is measured from x_{i-1} and x_i about z_{i-1} where the direction of rotation is determined according to the right hand rule. The radius of rotation, also known as link length is the distance between z_{i-1} and z_i along x_i and finally the twist angle α_i is measured about x_i such that $z_{i-1} \parallel z_i$. Figure 4.1 illustrates the procedure on the basis of a single finger. Since we are dealing with rigid bodies, each individual homogeneous transformation can be written as the product of four basic transformations.

$${}^{i-1}_i\mathbf{T} = \text{Rot}(z_{i-1}, \theta_i) \text{Trans}(z_{i-1}, d_i) \text{Trans}(x_i, r_i) \text{Rot}(x_i, \alpha_i)$$

$$= \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & r_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In other words, the homogeneous transformation matrix ${}^{i-1}_i\mathbf{T}$, relating the coordinate frame O_i to O_{i-1} , represents a rotation of θ_i degrees about the current z-axis followed by a translation of d_i along the current z-axis, followed by a translation of r_i along the current x-axis, followed by a rotation of α_i degrees about the current x-axis. This sequence of translate and rotate operations can of course be collapsed into a single homogeneous matrix:

$${}^{i-1}_i\mathbf{T} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & r_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & r_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

To summarize, the Denavit-Hartenberg parameters are given as:

d_i (Joint offset) Distance between x_i and x_{i-1} along z_{i-1}

θ_i (Joint angle) Rotation about z_{i-1} such that $x_{i-1} \parallel x_i$ (Right hand rule)

r_i (Link length) Radius of rotation about z_{i-1}

α_i (Twist angle) Rotation about x_i such that $z_{i-1} \parallel z_i$ (Right hand rule)

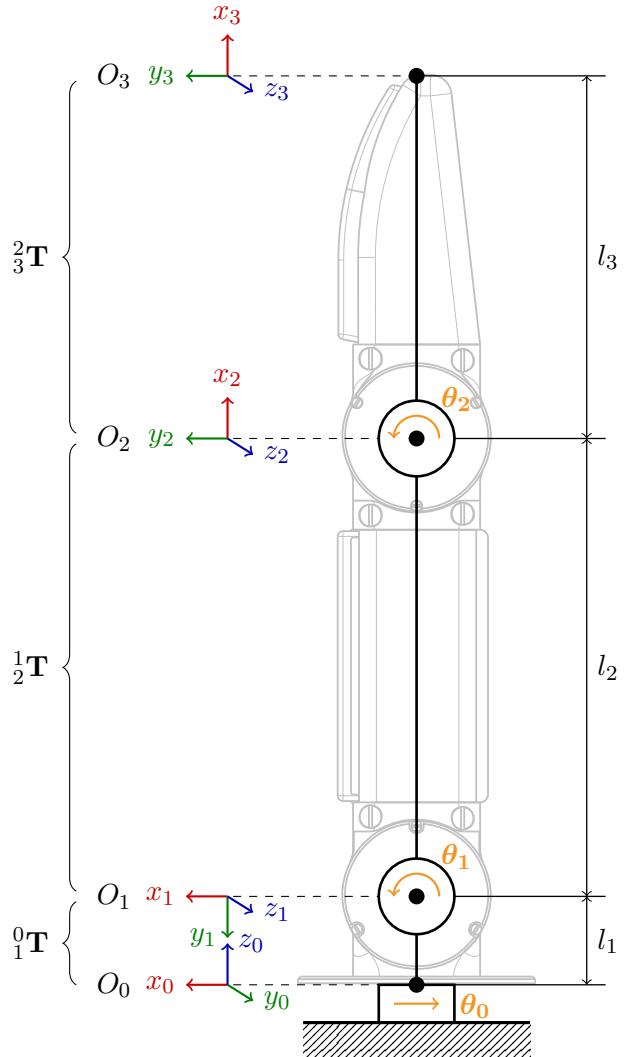


Figure 4.1: Schematic of a SDH-2 finger's kinematic chain along with the coordinate frame assignment at zero-position ($\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6 = 0^\circ$). The transformation matrices ${}^0\mathbf{T}$, ${}^1\mathbf{T}$, and ${}^2\mathbf{T}$ correspond to origins O_1 , O_2 and O_3 and are computed according to the Denavit-Hartenberg tables. The offsets to the hand's origin as well as the final transformation to the coordinate frame of the proximal and distal tactile sensor are handled separately.

The decision was made to model each finger as a separate kinematic chain with the origin at the finger's base. An additional transformation is therefore needed to place the fingers at their final position. Since the three fingers of the SDH-2 are identical in construction, their Denavit-Hartenberg parameters resemble each other. As shown in the parameter tables 4.1, 4.2 and 4.3, the only difference is the rotation angle of the first link. Finger 1 is missing the base joint and is therefore fixed to 180° , and Finger 0 and 2 rotate in opposite directions since they are coupled via a worm drive. It should be noted that the range of ϕ_0 is $[0^\circ, 90^\circ]$ while $\phi_{1..6}$ vary in the range $[-90^\circ, 90^\circ]$.

Table 4.1: Denavit-Hartenberg parameters of finger 0

<i>Link</i>	d_i	θ_i	r_{i-1}	α_{i-1}
1	l_1	$-\phi_0$	0	-90°
2	0 (*)	$(\phi_1 - 90^\circ)$	l_2	0° (*)
3	0 (*)	ϕ_2	l_3	0° (*)

* z-axis is parallel

Table 4.2: Denavit-Hartenberg parameters of finger 1

<i>Link</i>	d_i	θ_i	r_{i-1}	α_{i-1}
1	l_1	180° (**)	0	-90°
2	0 (*)	$(\phi_3 - 90^\circ)$	l_2	0° (*)
3	0 (*)	ϕ_4	l_3	0° (*)

* z-axis is parallel

** stiffened virtual joint

Table 4.3: Denavit-Hartenberg parameters of finger 2

<i>Link</i>	d_i	θ_i	r_{i-1}	α_{i-1}
1	l_1	ϕ_0	0	-90°
2	0 (*)	$(\phi_5 - 90^\circ)$	l_2	0° (*)
3	0 (*)	ϕ_6	l_3	0° (*)

* z-axis is parallel

Following the convention, an ambiguity arises when z_0 and z_1 intersect. It is resolved by determining x_1 by $z_1 \times z_0$. Using the right-hand rule, this corresponds to the thumb representing z_i and the index finger pointing along z_{i-1} . Another ambiguity appears when $z_1 \parallel z_2$, that is $d = 0$ and $\alpha = 0$. In this case, x_2 is determined by the common normal from z_1 to z_2 , or in general z_{i-1} to z_i . The actual values to fill in the transformation matrices of 4.1 based on the parameter tables can be obtained from the technical drawing in figure 4.2 (side view).

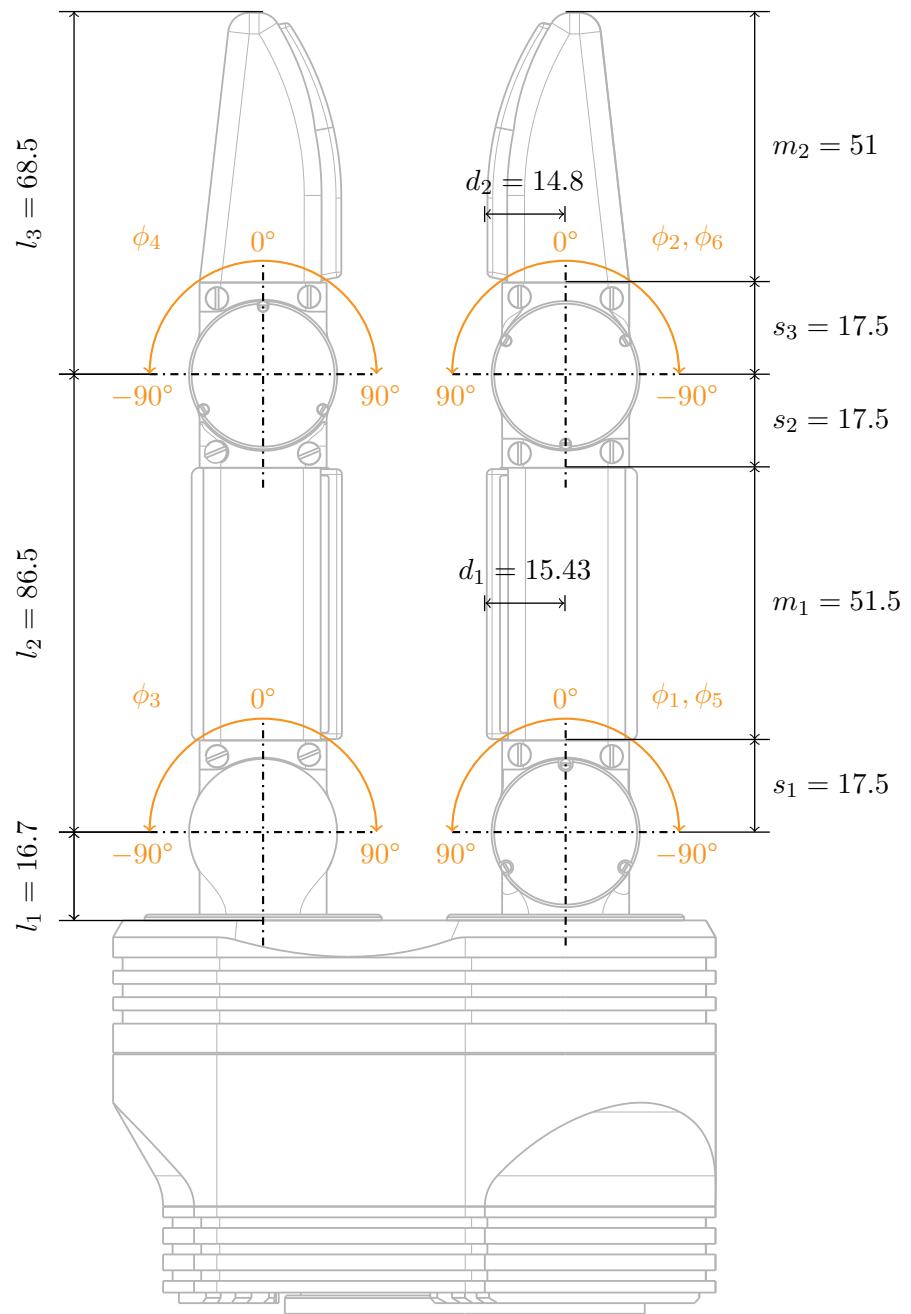


Figure 4.2: Side view of the SDH-2 including essential dimensions for the description of the kinematic chain. Measurement units in mm. Derived from assembled CAD models provided by SCHUNK

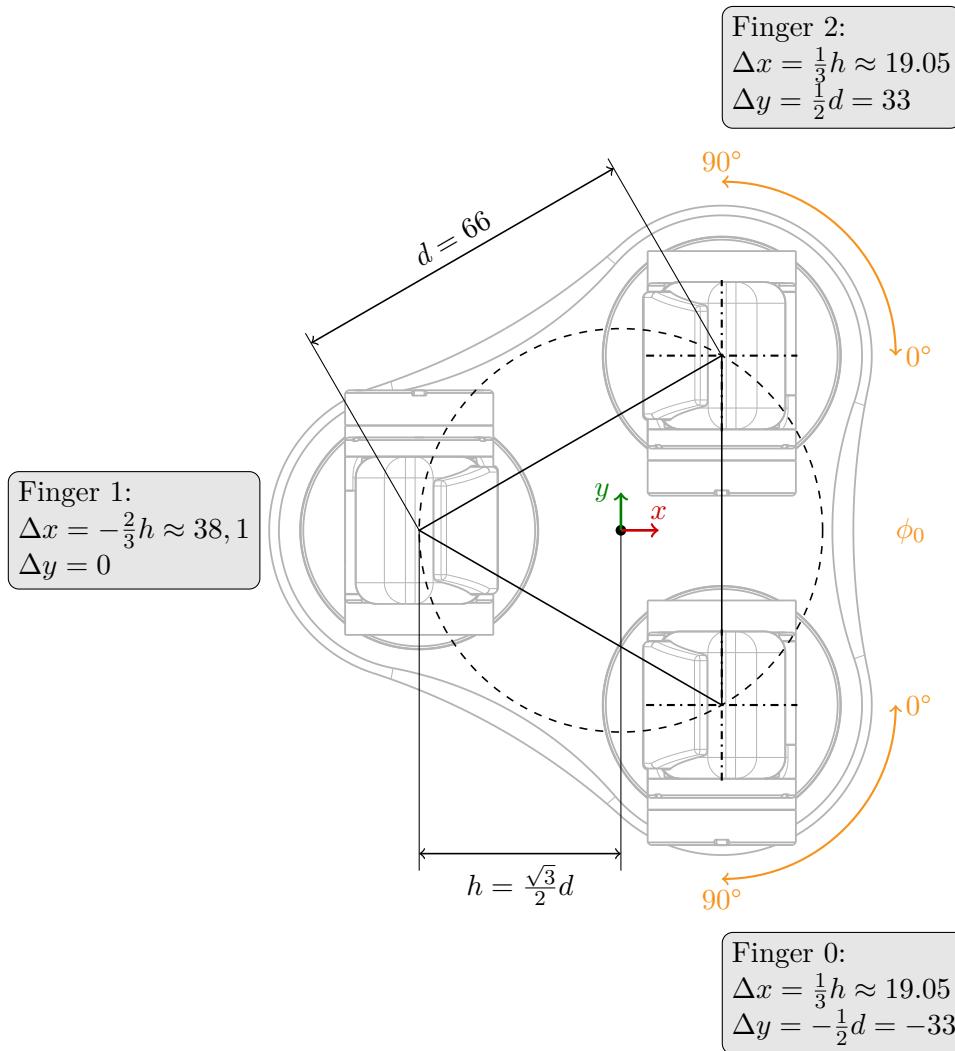


Figure 4.3: Top view of the SDH-2. The finger layout is based on an equilateral triangle. Measurement units in mm. Derived from assembled CAD models provided by SCHUNK

The initial transformation consists of a simple shift without further rotations since the fingers are already orientated in the right way. It can be inferred from the technical drawing in figure 4.3 (top view), derived from the provided CAD model, that the fingers are located at the corners of an equilateral triangle with a side length of $d = 66$ mm. The height of the triangle h is therefore given as

$$h = \frac{\sqrt{3}}{2}d \approx 57.16 \text{ mm} \quad (4.2)$$

For the sake of completeness, the corresponding circumradius is $\frac{\sqrt{3}}{3}d \approx 38.1$ mm. Hence the initial transformation matrices for the individual fingers are given as

$${}_{base}{}^0\mathbf{T}_0 = \begin{bmatrix} -1 & 0 & 0 & \frac{1}{3}h \\ 0 & -1 & 0 & -\frac{1}{2}d \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{base}{}^0\mathbf{T}_1 = \begin{bmatrix} -1 & 0 & 0 & -\frac{2}{3}h \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{base}{}^0\mathbf{T}_2 = \begin{bmatrix} -1 & 0 & 0 & \frac{1}{3}h \\ 0 & -1 & 0 & \frac{1}{2}d \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The isometric projection in figure 4.4 gives an overview of the local coordinate systems of the kinematic chain.

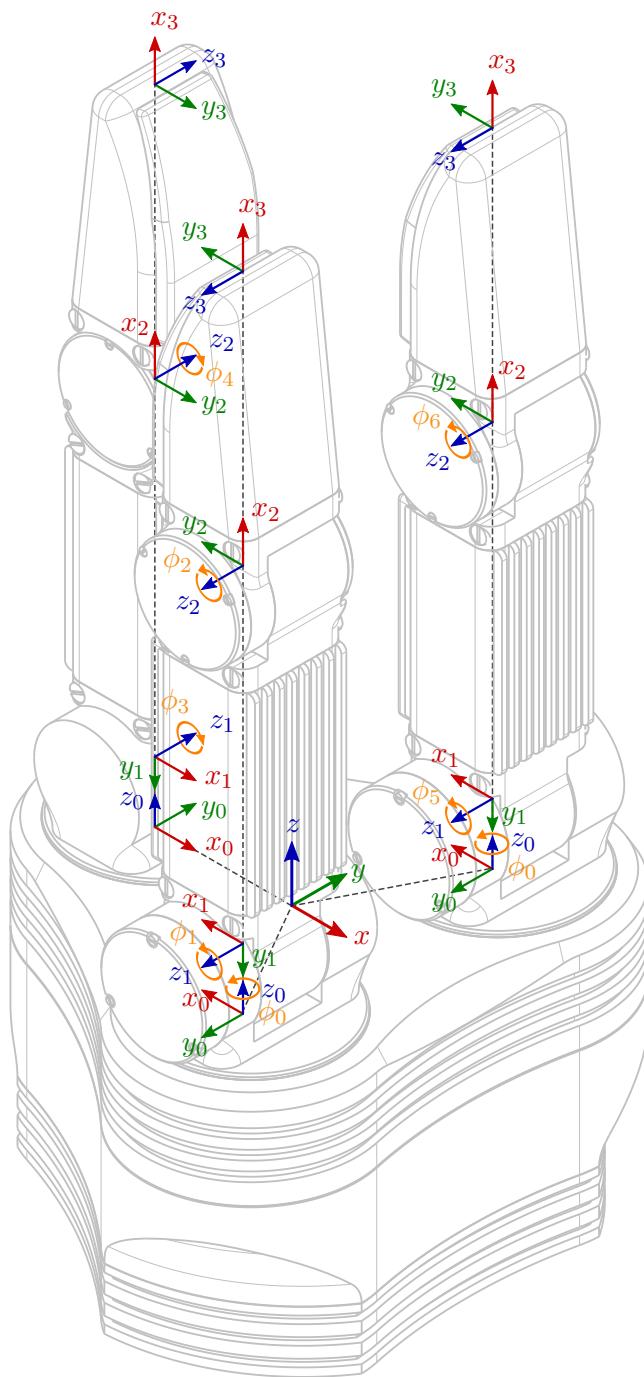


Figure 4.4: Isometric projection of the SDH-2 summarizing the kinematic chain from the original coordinate system at the gripper's base to the fingertips.

What is still missing, is the final static transformation of the actual taxels to the coordinate frame of the corresponding finger link. It is easy to see that only translation and scale operations are required in the case of the proximal sensor matrices. According to the drawing in figure 4.6, the computation can be divided into a transformation within the sensor matrix followed by a translation relative to O_2 . The x-coordinates of the taxels map to the z_2 axis of O_2 while the y-coordinates relate to the x_2 axis. The y_2 value are fixed to the distance of the sensor matrix from the center of the finger. The dimensions were determined using the CAD model as well as the technical drawing of the tactile sensor DSA 9205 in figure 4.5. The following abbreviations are introduced to simplify the description:

$$l_2 = 86.5$$

$$s_1 = 17.5$$

$$d_1 = 15.43$$

$$a = 4.1$$

$$w = 3.4$$

For the transformation of a taxel within the sensor matrix we have

$$sx = w = 3.4$$

$$sy = -w = -3.4$$

$$tx = \frac{w}{2} - \frac{6 \cdot w}{2} = -8.5$$

$$ty = 13 \cdot w = 44.2$$

And for the translation relative to O_2

$$tx_2 = -l_2 + s_1 + a = -64.9$$

$$ty_2 = d_1 = 15.43$$

This results in the transformation matrix from the coordinate system O_2 (x_2, y_2, z_2) to the coordinates (x,y) of the proximal tactile sensor.

$${}_{proximal}^2\mathbf{T_i} = \begin{bmatrix} sy & 0 & 0 & ty + tx_2 \\ 0 & -1 & 0 & ty_2 \\ 0 & 0 & sx & tx \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

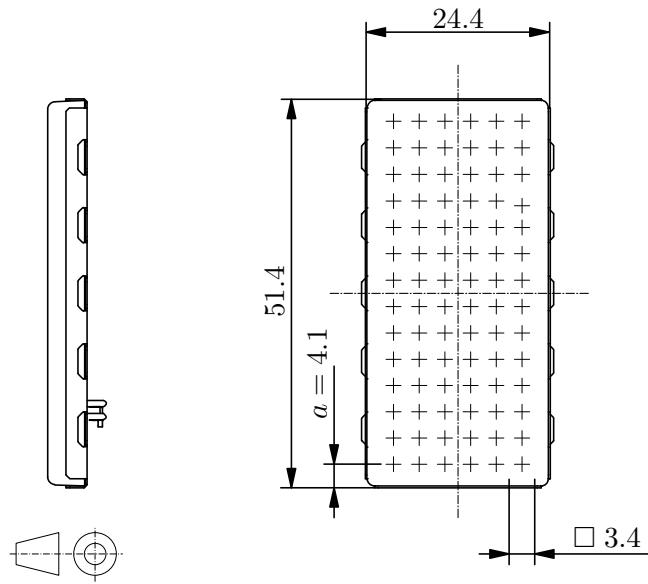


Figure 4.5: Technical drawing of the proximal digital sensor array DSA 9205 showing the position of individual taxels of size 3.4×3.4 mm. Modified version of Weiss Robotics GmbH & Co. KG (2012a)

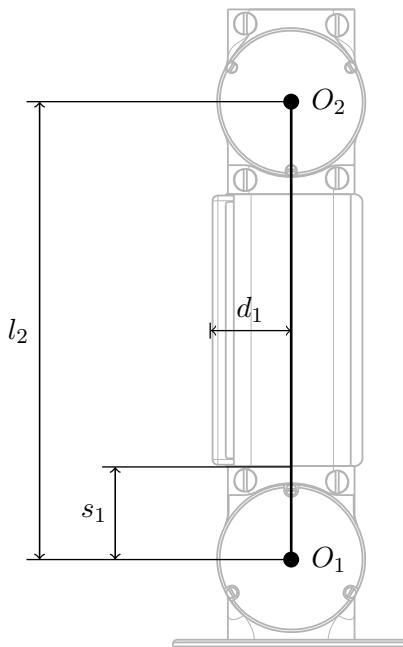


Figure 4.6: Illustration of the transformation from origin O_2 to the coordinate frame of the *proximal* tactile sensor matrix.

The curved surface of the distal sensor can be modeled as the combination of a plane in the lower part followed by a circular sector at the fingertip. See figure 4.7 for a detailed depiction of the tactile sensor DSA 9210.

The two-dimensional coordinate system within the sensor matrix that is used to address the individual taxels stretches along both parts in the y -direction as shown in figure 4.8. For this reason a case distinction is necessary. In the planar segment, a mapping of point $p = (x, y)$ in the sensor space to the axes x_3 , y_3 and z_3 in O_3 is straightforward since both axes scale linear with x_3 and z_3 while the offset to y_3 remains constant. In contrast, the curved segment requires a bit of trigonometry since the y -coordinates corresponds to a circular arc. From the definition of an arc length, given as

$$L = R \cdot \alpha$$

where α is central angle in radians and R is the radius of the circle, it follows that

$$\alpha = \frac{L}{R}.$$

Without considering scale and additional offsets, a mapping of the sensor plane's y -axis in the circular segment to the axes x_3 and y_3 in O_3 can therefore be obtained by

$$t_{x_3} = R \cdot \sin(\alpha) = R \cdot \sin\left(\frac{L}{R}\right)$$

and

$$t_{y_3} = R - R \cdot \cos(\alpha) = R - R \cdot \cos\left(\frac{L}{R}\right).$$

The actual values of the translational offsets and scale factors were determined as follows:

$$\begin{aligned} l_3 &= 68.5 \\ s_3 &= 17.5 \\ d_2 &= 14.8 \\ a &= 4.95 \\ w &= 3.4 \\ R &= 58.0 \end{aligned}$$

For the planar part, that is for values of $y > 8.5$, the transformation within the sensor matrix is

$$\begin{aligned} sx &= w = 3.4 \\ sy &= -w = -3.4 \\ tx &= \frac{w}{2} - \frac{6 \cdot w}{2} = -8.5 \\ ty &= 12 \cdot w = 44.2 \end{aligned}$$

And for the translation relative to O_3

$$\begin{aligned} t_{x_3} &= -l_3 + s_3 + a = -46.05 \\ t_{y_3} &= d_1 = 15.43 \end{aligned}$$

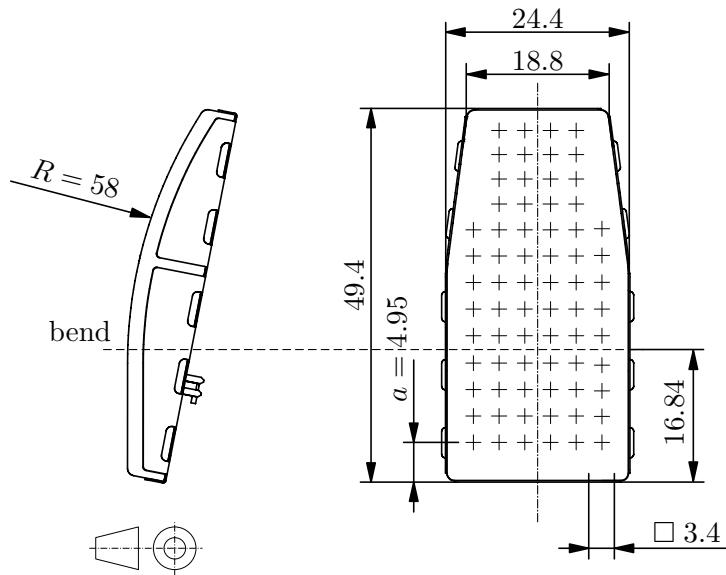


Figure 4.7: Technical drawing of the distal digital sensor array DSA 9210 showing the position of individual taxels of size 3.4×3.4 mm. The curved surface requires additional effort to determine the exact coordinates. Modified version of Weiss Robotics GmbH & Co. KG (2012b)

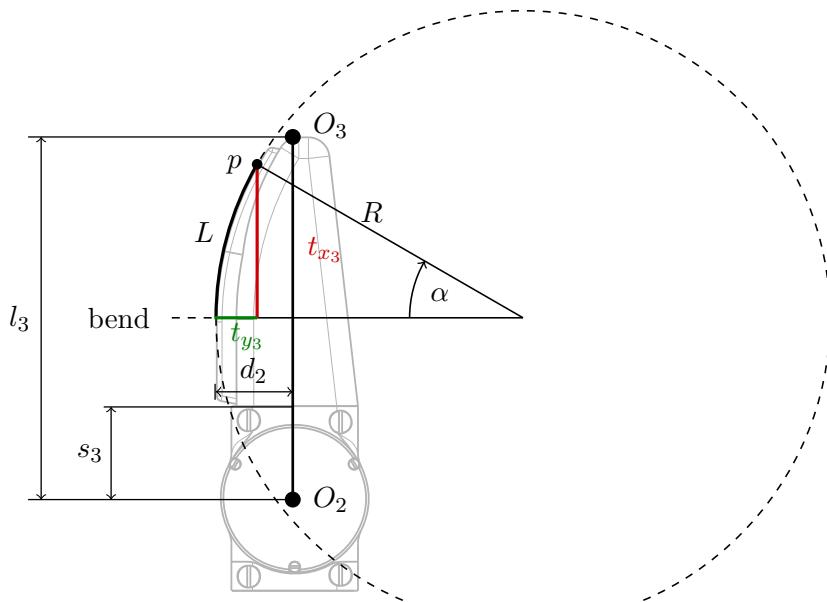


Figure 4.8: Illustration of the transformation from origin O_3 to the coordinate frame of the *distal* tactile sensor matrix.

For the curved part, that is for values of $y \leq 8.5$, the transformation within the sensor matrix is

$$\begin{aligned}s_x &= w = 3.4 \\s_y &= 1.0 \\t_x &= \frac{w}{2} - \frac{6 \cdot w}{2} = -8.5 \\t_y &= R \cdot \sin\left(\frac{(8.5 - y) \cdot w}{R}\right) + (12 - 8.5) \cdot w - y\end{aligned}$$

And for the translation relative to O_3 we have

$$\begin{aligned}t_{x_3} &= -l_3 + s_3 + a = -46.05 \\t_{y_3} &= d_2 - R - R \cdot \cos\left(\frac{(8.5 - y) \cdot w}{R}\right)\end{aligned}$$

This results in the transformation matrix

$${}_{distal}^3\mathbf{T}_i = \begin{bmatrix} sy & 0 & 0 & ty + tx_3 \\ 0 & 1 & 0 & ty_3 \\ 0 & 0 & sx & tx \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

To conclude, the final transformation matrix for the proximal tactile sensor matrices of finger $i \in 0, 1, 2$ is

$${}_{proximal}^{base}\mathbf{T}_i = {}_0^{base}\mathbf{T}_i \ {}_1^0\mathbf{T}_i \ {}_2^1\mathbf{T}_i \ {}_{proximal}^2\mathbf{T}_i$$

For the distal tactile sensor matrix of finger i we have

$${}_{distal}^{base}\mathbf{T}_i = {}_0^{base}\mathbf{T}_i \ {}_1^0\mathbf{T}_i \ {}_2^1\mathbf{T}_i \ {}_3^2\mathbf{T}_i \ {}_{distal}^3\mathbf{T}_i$$

The actual implementation then offers two distinct interfaces to compute the three-dimensional position of either an arbitrary point (x, y) in mm on the sensor plane, with $(0.0, 0.0)$ mapping to the center of the top-left taxel $(0, 0)$, or the discrete taxel centers. In the latter case, the static transformation from O_2 and O_3 to the available taxels are stored in a look-up table to improve the performance.

The Cartesian coordinates (x', y', z') of a taxel in the two-dimensional sensor matrix space (x, y) on the proximal matrix on finger $i \in 0, 1, 2$ is then obtained by

$$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = {}_{proximal}^{base}\mathbf{T}_i \begin{bmatrix} y \\ 0 \\ x \\ 1 \end{bmatrix}$$

And for the distal matrices respectively

$$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = {}_{distal}^{base}\mathbf{T}_i \begin{bmatrix} y \\ 0 \\ x \\ 1 \end{bmatrix}$$

Chapter 5

Slip detection

In order to ensure stable grasps, robust object handling and dexterous manipulation, it is important to predict, detect and prevent incipient slippage. Even though slip conditions can simply be avoided by applying the maximum available force, this approach is not only undesirable from an ergonomic point of view but also completely inapplicable for fragile objects. The obvious solution to this problem is the skillful application of force, just slightly above what is required to hold the grasped object in place. By the way, humans follow the same strategy with the exception that we already have good estimates for the initial force. The detection of incipient slip is a well studied area of robotic grasping with a history of four decades. Howe (1994) gives an overview of early developments and the underlying principles which are still valid today.

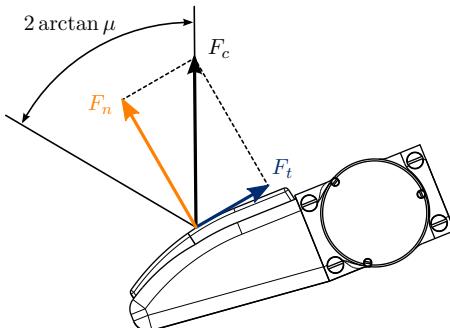


Figure 5.1: Point-contact with friction cone. The simple Coulomb friction model is in the equilibrium state with $|F_t| \leq \mu|F_n|$. The configuration is depicted in the transitional phase before slip occurs.

Before going into detail, the concepts of contact force, friction and slip should be clarified. For more information see Kao et al. (2008). Figure 5.1 illustrates a simple Coulomb friction model with a single friction coefficient μ . At each contact point, the applied force is given as $F_c = F_n + F_t$ where F_n is a normal force and F_t is the tangential force corresponding to friction. The depicted friction cone defines the range of the sum of applied external forces in which the grasped object remains static. The contact begins to slide at $|F_t| = \mu|F_n|$, that is when the applied tangential force overtakes the friction force which is opposed to the direction of motion. This model is sometimes extended with a second friction coefficient to distinguish between static and dynamic friction. Moisio et al. (2012) developed a soft-contact model and evaluated it using the SCHUNK PG70 parallel jaw gripper equipped with the same tactile sensor as the SDH-2. They incorporated the simple, yet powerful, LuGre friction model, which allows the description of static and dynamic phenomena such as stick-slip motion as outlined for example in Astrom and Canudas-de Wit (2008).

A simulation based on a theoretical model for the sliding motion of soft fingertips was developed by Ho and Hirai (2011). The model can be adapted to various types of grippers including anatomically correct human fingertips. By simulating stick-slip transitions and local displacements using finite element analysis, the authors could reproduce several real-world experiments.

The tactile sensor based approaches to slip-detection can roughly be divided into three different categories. Normal force based approaches rely on the Coulomb friction model and report slippage when direction and magnitude of the applied normal force violate the restrictions of the friction cone. However, those methods depend on the knowledge of the friction coefficients between gripper and object. The most common approach to the slip-detection task is frequency based and relies on the subtle vibrations caused by slipping objects. But the used sensors usually have a low spatial resolution and consequently, it is hard to determine the direction of movement. With the development of tactile sensor arrays, position based methods emerged. They have the limitation that the relative motion of the grasped object must result in a moving pressure profile. But this is not always the case. Just imagine an unstructured rod sliding along the sensor surface in the direction of its major axis. In addition, the apparent motion does not necessarily correspond to the real movement as the aperture problem in the context of optical illusions demonstrates.

A typical example of a frequency based special-purpose tactile sensor is mentioned by Teshigawara et al. (2011). With a high sampling rate of several kilohertz and the help of a discrete wavelet transform, which allows the precise location in time, the authors manage to effectively prevent slip using frequency thresholds and a position-based force control. Göger et al. (2009) also work with a high-speed sensor array and experiment with structured sensor covers to imitate human papillary ridges. In order to distinguish between different slip states, they perform a short-time Fourier transform (STFT). The actual features for the nearest neighbor classification are then extracted from the resulting spectrogram following the well known eigenface approach. The method correctly classifies slip conditions but fails to distinguish other vibrations from background noise. Based on conductive elastomers and the resistive principle, Schürmann et al. (2012) developed a tactile sensor with an adequate spatial resolution and high frame rates of up to 1.9 kHz. To show the effectiveness of their high-frequency approach, they perform a Fourier transform on the sensor data and the velocity of the end-effector. A multi-layer neural network with back-propagation is then successfully trained to distinguish between stick and slip conditions of different materials. Related to frequency based slip-detection is the recognition of different surface types as discussed for example by Jamali et al. (2009). The novelty of their approach is the usage of a bio-inspired tactile sensor based on PVDF, a piezoelectric, thermoplastic fluoropolymer. The exact placement of the PVDFs as well as the corresponding strain gauges within the soft finger is not known beforehand. Instead, they successfully train a naive Bayes classifier with the frequency components of ten different surfaces.

To give an example for a position based method: Edussooriya et al. (2008) examined slip conditions with the help of 20 FlexiForce sensors embedded in a glove. They continuously compute a normalized running average for each sensor and determine the speed and direction of the slipping object via the gradients of neighboring sensors.

The work of Melchiorri (2000) is one of the rare cases where rotational slippage is taken into account. The used tactile sensor is only able to measure a single normal force per contact. Translational slip can therefore only be detected if the friction between the gripper and the grasped object is known beforehand. But an additional force/torque sensor in the gripper is used to measure the occurring wrenches when the grasped object rotates around the contact point.

Another application for slip-detection is the automatic estimation of the friction coefficients. This can be done by slowly opening a firm grip until the grasped object begins to slip.

And yet another, quite different approach is based on kinaesthetic sensing, that is the perception of internal forces as opposed to cutaneous or tactile sensing. Vina B et al. (2013) use a Gaussian Process regression to capture the nonlinear relationship between the maximum static friction forces and torques for various grasp configurations. Once the robot has learned a stable range of wrenches for a given grasp configuration, it can infer the allowed movements and thus the grasp affordances. Technically, this is not slip-detection but slip-avoidance in the first place.

There is a reason for this large variety of slip-detection methods: None of the methods mentioned so far works under all possible circumstances. Reinecke et al. (2014) therefore examined three different slip-detection strategies using the multimodal SynTouch BioTac. The authors aim to fuse their approaches in future publications.

5.1 Frequency Analysis

Masataka et al. (2006) tried to reproduce the human grasp force reflex and discovered that distributed stimuli in the 5 and 30 Hz range have the most effect. The detection of a stick-slip condition is therefore largely attributed to a combination of SA I and FA I mechanoreceptors. In order to make use of this frequency based technique, the frame rate of the tactile sensor has to be increased to at least $2 \cdot 30$ Hz according to the sampling theorem. As already mentioned in section 3.4 (High sensitivity mode), the sampling frequency of the tactile sensor controller DSACON32-M is not limited by the attached sensor arrays, but by its connection to the SDH-2. In fact, the distal sensor DSA 9210 could be sampled at 230 fps according to the technical specifications.

The sensor controller manages a static and a dynamic mask for each connected tactile sensor. In the case of the distal matrix, the not editable static mask is used to eliminate the missing taxels at the finger tip. According to the reference manual, the dynamic mask can be used to speed up the sampling of a matrix by unmasking areas of the sensor surface which are not interesting at the moment. In an attempt to increase the sampling rate, the SDHLibrary was therefore extended to support the command set to do just that. As it turns out, the highest possible frame rate to be achieved is ~ 75 fps. Unfortunately, this value drops rapidly with each additional masked taxel and reaches the usual 30 fps with only a dozen of active sensor cells. As a side effect, the frame rate of a typical pressure profile is not constant anymore due to the unpredictable size of the run-length encoded frames. In that case, a non-uniform discrete Fourier transform would be necessary to analyze the signal.

These result severely limits the frequency based slip-detection approach. Instead, the possibility to discriminate between different object surface types, taking only the signal of a single taxel into account, is examined in the following. Since the frequency content of the signal is varying over time, a real-time frequency analysis could be based on the short-time Fourier transform. This variant of the Fourier transform allows tracking the changes over time in the frequency spectrum. The signals are therefore broken up into chunks of overlapping samples, whereby the overlapping segments reduce artifacts at the window boundaries. It must be pointed out that the width of this windowing function is a trade-off between frequency and time resolution, since shorter windows are better localized in time, but provide less frequency information. Figure 5.2 illustrates the magnitudes of the time-frequency spectrum of a slip event.

Unfortunately, all attempts to find distinctive patterns in the spectrogram of objects with different surface properties were unsuccessful. Such a result was to be expected since the previously mentioned publications on frequency based methods were dealing with signals at much higher frequencies. This is no coincidence. In humans, the FA II-type receptors are known to be crucial for the classification of surface texture. They evolved to be most sensitive around 250 Hz.

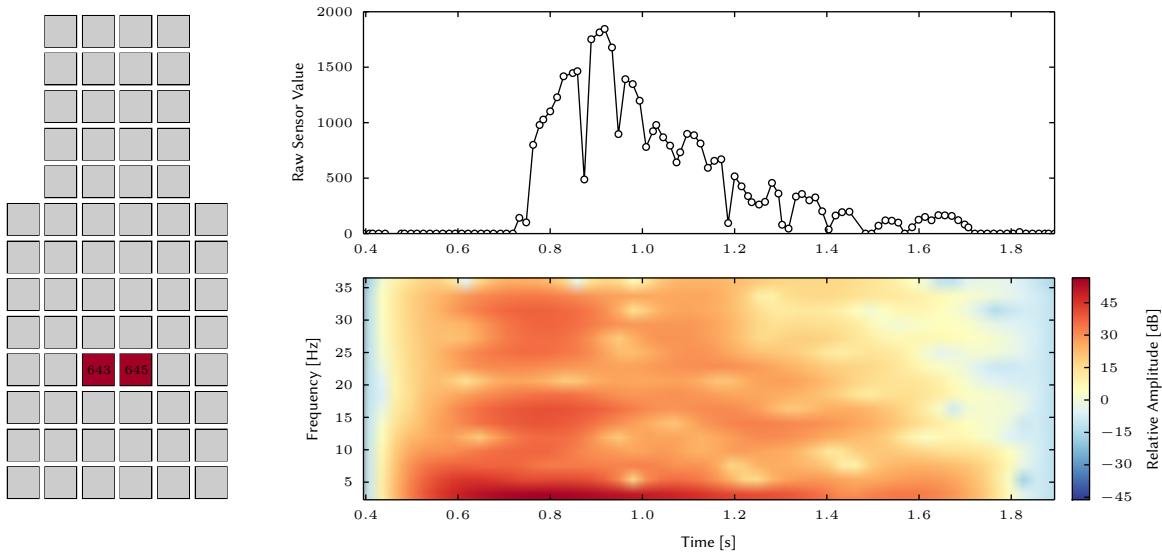


Figure 5.2: Short-time Fourier transform of a single taxel during a slip event. The window size of the FFT is 32 samples with 31 samples overlap and a Hann function is applied to reduce aliasing. Since the sample rate could only be increased to about 75 Hz by masking all but two taxels of the sensor matrix, the highest occurring frequencies in the signal are ~ 37.5 Hz. Experiments confirmed that these frequencies are not enough to distinguish between rough and smooth object surfaces.

5.2 Translation

Image registration and motion estimation are important and very well studied areas of computer vision. Similar fields with uncountable applications include template matching, block motion estimation for video encoding, optical flow estimation and particle tracking velocimetry, just to name a few. Image registration methods can roughly be divided into the following classes according to Reddy and Chatterji (1996).

- ▷ Intensity-based methods
- ▷ Correlation methods
 - ▷ Spatial domain
 - ▷ Frequency domain
- ▷ Optical flow methods
- ▷ Feature-based methods
 - ▷ Low-level features (edges and corners)
 - ▷ High-level features (object parts)
 - ▷ Relations between features

Given the restrictions of the low resolution tactile sensor, all optical flow and image feature tracking methods are inapplicable in practice. Preceding the actual slip-detection, the tactile matrices could of course be resampled to match a higher resolution, but apart from being computationally expensive, the results are not very convincing as a first evaluation using state-of-the-art algorithms showed.

A very simple alternative is the tracking of the center of mass, i.e. the intensity weighted centroid. But this method is quite sensitive to noise and saturation effects. In addition, a shifting of weight is interpreted as a translation even if the pressure profile is not moving at all.

There remain the correlation based methods as discussed in any educational book on signal and image processing, Jähne (2005) for example. Let the taxel intensity function of a sensor matrix \mathbf{T}_n at time step n be denoted as $t_n(x, y)$. At the core of the procedures is the convolution of two usually consecutive matrices \mathbf{T}_{n-1} and \mathbf{T}_n of size $M \times N$. The two-dimensional convolution of the discrete matrices, also known as complex-conjugate multiplication is defined as

$$t_{n-1}(x, y) * t_n(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} t_{n-1}(i, j) t_n(x - i, y - j).$$

This results in the convolution matrix \mathbf{C} of size $(2M - 1) \times (2N - 1)$. Note that in practice, the equivalent indices of summation are $n = 0, \dots, N - 1$ and $m = 0, \dots, M - 1$ due to the periodic replication in all directions beyond the original definition. Similarly, the cross-correlation is defined as

$$t_{n-1}(x, y) \star t_n(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} t_{n-1}(i, j) t_n(x + i, y + j).$$

There is a maximum in the cross-correlation matrix \mathbf{C} for each match in the signals. The elements of \mathbf{C} are therefore a measure of similarity between the two images and the translation of the peak from the origin indicates the shift between them. In the context of slip-detection, this relation can be interpreted as a slip vector between two similar tactile sensor profiles. The methods discussed in the following are often referred to as cross-correlation even if the actual implementation performs a convolution, since both integral transforms can be converted into one another by flipping one of the images in both directions. For symmetrical profiles, a convolution is exactly equivalent to the correlation.

The decision was made to implement a slip-detection algorithm by Alcazar and Barajas (2012). The authors analyzed the performance of their proposed method with the help of known trajectories generated by sliding and rotating an object over the sensor array using a second, high accuracy industrial robot. Their original experiments were carried out using a BarrettHand™ with capacitive RoboTouch™ pressure sensors by Pressure Profile Systems (PPS). Although the spatial resolution as well as the sampling rate of the sensors are slightly lower compared to the ones installed in the SDH-2, see Pressure Profile Systems, Inc. (2010), there is no reason not to transfer the results to other similar digital sensor arrays.

The overall workflow is shown in figure 5.3. First, two index matrices \mathbf{A} and \mathbf{B} of the same size as the convolution matrix and consisting of repeating rows and columns, respectively, are defined as follows:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & \dots & 2N - 1 \\ 1 & 2 & \dots & 2N - 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & \dots & 2N - 1 \end{bmatrix} - N \quad \mathbf{B} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 2 & 2 & \dots & 2 \\ \vdots & \vdots & \ddots & \vdots \\ 2M - 1 & 2M - 1 & \dots & 2M - 1 \end{bmatrix} - M$$

These matrices only have to be created once and can then be reused throughout the whole procedure. But since the height of the proximal and distal sensor differs by one taxel, two versions and a case distinction is needed which is not much of a problem since the actual implementation performs the slip-detection of individual matrices in separate class instances.

In the first step of the slip-detection loop, the convolution matrix \mathbf{C}_{n-1} of the first pair of tactile matrices \mathbf{T}_{n-2} and \mathbf{T}_{n-1} is computed. Note that these matrices do not necessarily have

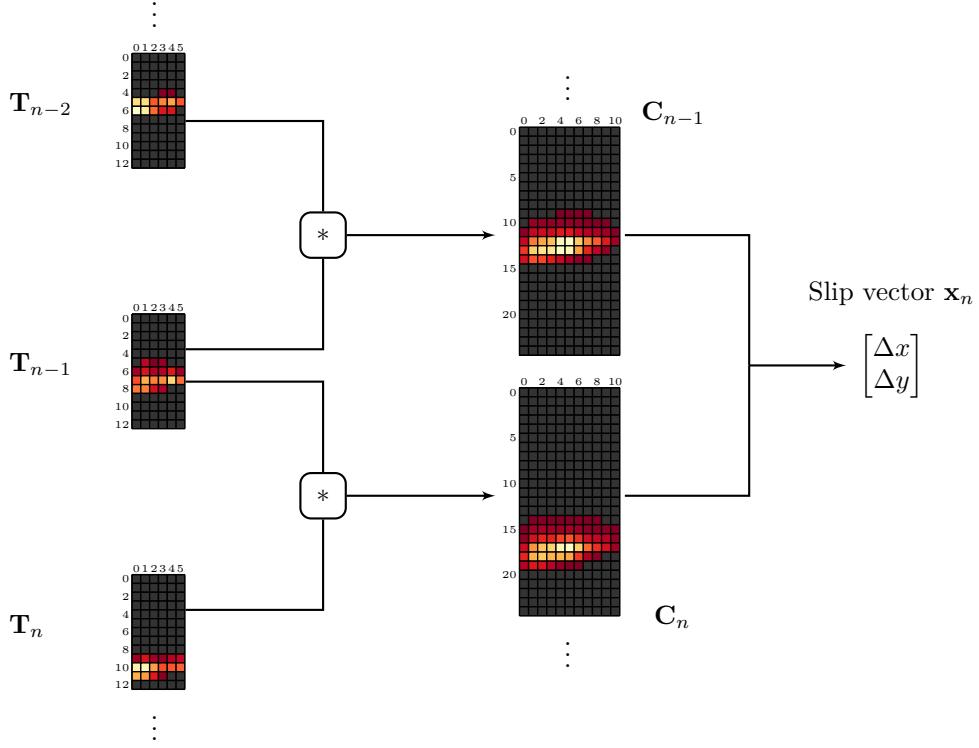


Figure 5.3: Translational slip-detection workflow as described by Alcazar and Barajas (2012). The method finds the difference of the convolutions of two pairs of tactile sensor matrices. The first convolution matrix \mathbf{C}_{n-1} could also be the result of a convolution of a reference matrix with itself. In that case only two consecutive frames are needed to compute a slip vector, but at the costs of an additional convolution per time step. Note that two different scales are used for the purpose of visualization to highlight the maximum values of the profiles and the convolution.

to be consecutive sensor readings. \mathbf{C}_{n-1} could also be the result of a convolution of a single reference frame, for example \mathbf{T}_{n-1} , with itself.

Now a column vector $\boldsymbol{\mu}_c$ consisting of the means of columns of the convolution matrix \mathbf{C}_{n-1} is computed. In contrast, the row vector $\boldsymbol{\mu}_r$ contains all means of the rows of the same convolution matrix. The displacement of the two sensor matrices \mathbf{T}_{n-1} and \mathbf{T}_n is now given as

$$\Delta x_{n-1} = \frac{\mathbf{A} \cdot \boldsymbol{\mu}_c^T}{\sum \boldsymbol{\mu}_c} \quad \text{and} \quad \Delta y_{n-1} = \frac{\boldsymbol{\mu}_r^T \cdot \mathbf{B}}{\sum \boldsymbol{\mu}_r}.$$

As the next tactile sensor matrix arrives, the previous step is repeated, but this time the matrices \mathbf{T}_{n-1} and \mathbf{T}_n are convolved. Again, the column and row means of the resulting convolution matrix \mathbf{C}_n as well as the displacements are computed. The final slip vector is then the difference of the displacements between the two pairs of sensor matrices, i.e.

$$\mathbf{x}_n = \begin{bmatrix} \Delta x_n - \Delta x_{n-1} \\ \Delta y_n - \Delta y_{n-1} \end{bmatrix}.$$

Interestingly enough, the used implementation performs a two-dimensional Discrete Fourier Transform. Signal processing libraries typically make use of the circular convolution theorem that relates convolution in the spatial domain to multiplication in the Fourier domain. The details

are addressed later on, but once the Discrete Fourier Transforms of both sensor matrices are computed, they can be convolved in linear time. Including the subsequent inverse transformation, the computational complexity is $\mathcal{O}(MN \log(MN))$ instead of the naive $\mathcal{O}(M^2N^2)$. OpenCV for example, uses the DFT-based algorithm for images larger than 11×11 pixels for that reason.

Alcazar and Barajas (2012) claim to describe an “optical flow computation [...] using a block-based method which maximizes the normalized cross-correlation”. But their method has nothing to do with optical flow and neither does it maximize the normalized cross-correlation coefficients. In fact, the algorithm is a variant of the center-of-mass tracking approach, but in the convolution domain. In the case of unimodal pressure distributions, the centroids of the convolution matrices should be more robust to noise than the original centroids. Experiments with different slip trajectories of moving pressure profiles confirm this finding.

Finding an image block in another, usually much larger image is a very well researched area of computer vision and the normalized cross correlation (NCC) is one of the most popular template matching algorithms. Typically, a small region of an image I_1 is selected as a template to look for in another image I_2 . A simple metric for the similarity between template and sliding window in the search image at position (u, v) is the sum of absolute differences

$$SAD(u, v) = \sum_{x,y} |I_1(x, y) - I_2(x - u, y - v)|.$$

The NCC is motivated by the fact that intensity and contrast differences between two images may complicate the exact location of the match. The same reasoning holds for pressure profiles and the zero-mean normalized cross-correlation coefficient is therefore defined as

$$NCC(u, v) = \frac{\sum_{x,y} (I_1(x, y) - \bar{I}_1)(I_2(x - u, y - v) - \bar{I}_2)}{\sqrt{\sum_{x,y} (I_1(x, y) - \bar{I}_1)^2 \sum_{x,y} (I_2(x - u, y - v) - \bar{I}_2)^2}}.$$

This similarity measure is obviously expensive to calculate, since the mean and the standard deviation have to be computed for every position of the sliding window. But Lewis (1995) popularized the method with the help of summed area tables. It may have helped though, that the technique was prominently featured in the movie Forest Gump (1994).

The computation is trivial in the case of small tactile sensor matrices, because no sliding window is necessary and the whole normalization can be done before calculating the shift in the correlation space. Assuming the maximum value of the cross correlation is located at (y_0, x_0) , the translation is given as

$$\Delta x = \frac{2M - 2}{2} - x_0 \quad \text{and} \quad \Delta y = \frac{2N - 2}{2} - y_0$$

with M and N still being the height and width of the tactile sensor matrices. Under the given circumstances, this integer position is by far not accurate enough to register small incremental movements. Three more advanced peak detection methods, namely interpolation, 3-point closed form- as well as least-squares curve fitting were tested for that matter. They are reviewed in Debella-Gilo and Käab (2011) and illustrated in figure 5.4.

The first approach involves the two neighboring taxels in the x and y direction. A parabola can be defined which connects the three points $x_0 - 1, x_0, x_0 + 1$ and $y_0 - 1, y_0, y_0 + 1$ respectively. The analytical position of the maximum at $(x_0 + \Delta x, y_0 + \Delta y)$ can then trivially be obtained. But assuming that the dimensions are separable, there also exists a two-dimensional 3-point closed-form solution of a Gaussian fit which is favored by the authors. The sub-pixel offset from

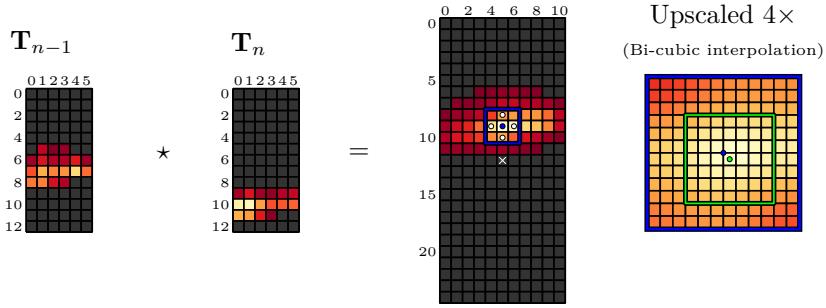


Figure 5.4: Normalized cross-correlation between matrices \mathbf{T}_{n-1} and \mathbf{T}_n of figure 5.3: The position of the maximum cross correlation peak relative to the matrix center is directly related to the translation between the matrices. To achieve sub-pixel precision, the area around the peak can either be interpolated or curve fitted or both. Depicted are the 2D 3-point method and a 7×7 neighborhood around the interpolated peak.

the peak is then given as

$$\Delta x = \frac{\ln(c_{y_0, x_0-1}) - \ln(c_{y_0, x_0+1})}{2 \ln(c_{y_0, x_0+1}) - 4 \ln(c_{y_0, x_0}) + 2 \ln(c_{y_0, x_0-1})}$$

$$\Delta y = \frac{\ln(c_{y_0-1, x_0}) - \ln(c_{y_0+1, x_0})}{2 \ln(c_{y_0+1, x_0}) - 4 \ln(c_{y_0, x_0}) + 2 \ln(c_{y_0-1, x_0})}.$$

Another popular method to locate the correlation peak with sub-pixel precision is the interpolation of the cross-correlation surface. After determining the position of the maximum value, a small image region around the peak is upsampled while applying some sort of interpolation scheme. Bilinear and bicubic polynomial or spline interpolation are often used. Once the smoothed surface is available, a new peak in the rescaled regular grid can be determined. Depending on the used scale factor and the required precision, this sub-pixel location might already be sufficient.

Gleason et al. (1990) makes a case for fitting a paraboloid to the area around the correlation peak. Let the general equation for a two-dimensional parabola, that is a paraboloid, be

$$z(x, y) = ax^2 + by^2 + cxy + dx + ey + f \quad (5.1)$$

The coefficients of $z(x, y)$ can then be found by solving the linear system of equations

$$\begin{bmatrix} x_1^2 & y_1^2 & x_1y_1 & x_1 & y_1 & 1 \\ x_2^2 & y_2^2 & x_2y_2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^2 & y_n^2 & x_ny_n & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} \quad (5.2)$$

where z_n is the value of the image intensity function at (x_n, y_n) . Setting the gradient of $z(x, y)$ to zero leads to the closed-form solution of the location of the paraboloid's maximum, given as

$$\Delta x = \frac{2db - ce}{c^2 - 4ab} \quad \text{and} \quad \Delta y = \frac{2ae - dc}{c^2 - 4ab}.$$

Zhu et al. (2007b) reviewed the method and proposed an additional Gaussian distance weight $e^{-d_i^2/k^2}$ where d_i refers to the euclidean distance of a neighborhood pixel i from the peak. k is chosen arbitrarily.

In template matching applications, accuracy is often traded for speed, since positioning errors do not accumulate when all displacement vectors correspond to the same reference frame at the beginning of an image sequence. But in tasks like slip-detection, the template changes over time and might not be recognizable anymore. It is therefore reasonable to compare similar consecutive images and keep track of the cumulative movement as new images arrive. In this context, the bias of the position estimation plays an important role. Dana (2006) mentions the peak locking effect that causes the position measurement to be biased towards the integer pixel positions.

A last method was therefore taken into consideration. Guizar-Sicairos et al. (2008) describe a variant of the classic phase correlation (PC) first published by Kuglin and Hines (1975). The frequency-domain correlation method exploits two well known properties of the (discrete time) Fourier transform. Let f_1 and f_2 be two images with corresponding Fourier transform $\mathcal{F}\{f_1\}$ and $\mathcal{F}\{f_2\}$, respectively. The already mentioned circular convolution theorem for sequences of finite length is then given as

$$\mathcal{F}\{f_1 * f_2\} = \mathcal{F}\{f_1\} \cdot \mathcal{F}\{f_2\}.$$

This expression can be reformulated as

$$f_1 * f_2 = \mathcal{F}^{-1}\{\mathcal{F}\{f_1\} \cdot \mathcal{F}\{f_2\}\}$$

where \mathcal{F}^{-1} is the inverse transform. The same reasoning applies to correlations with

$$f_1 \star f_2 = \mathcal{F}^{-1}\{(\mathcal{F}\{f_1\})^* \cdot \mathcal{F}\{f_2\}\}.$$

To be precise, this result also holds for the DFT. Given that the 2D FFT and its inverse both have a complexity of $\mathcal{O}(N \log N)$, it becomes clear why the multiplication in the frequency domain is often preferred over a convolution in the spatial domain.

Phase correlation also relies on the Fourier shift theorem which states that two images differing only by a displacement, i.e. $f_2(x, y) = f_1(x - x_0, y - y_0)$, are related by

$$F_2(u, v) = e^{-i2\pi(ux_0+vy_0)} \cdot F_1(u, v)$$

where the uppercase $F(u, v)$ represents the DFT of $f(x, y)$. Intuitively, the frequency content of the images does not change at all. A shift therefore leaves the magnitude unchanged but alters the phase by a constant directly proportional to the displacement. Equivalently, the (normalized) cross-power spectrum is defined as

$$R = e^{i2\pi(ux_0+vy_0)} = \frac{F_1(u, v)F_2^*(u, v)}{|F_1(u, v)F_2^*(u, v)|}.$$

And since the Fourier transform of the Dirac delta function centered at (x_0, y_0) is known to be

$$\mathcal{F}\{\delta(x_0, y_0)\} = e^{i2\pi(ux_0+vy_0)},$$

the translation between the two images is now easily obtained by

$$\delta(x_0, y_0) = \mathcal{F}^{-1}\{R\}.$$

In practice, the phase correlation surface is not a single peak since the translation is usually not a circular shift as required by the circular convolution theorem. Instead, the maximum value

has to be determined just like with the spatial correlation methods. It should be noted that multiplying the images with a window function before computing the FFT reduces edge effects.

The improvement over the traditional PC approach is the upsampling method for the peak detection. Upsampling in the frequency domain can be done by zero-padding the fourier coefficient in combination with an anti-aliasing filter before applying the inverse transform. But upsampling by a large factor, say 100, is expensive and often even impossible since the FFT takes the whole range of the image into account. The author therefore proposed a matrix-multiply IDFT which supports selective upsampling of an image region in order to find the peak with sub-pixel precision given an initial estimate of its location. The original Matlab code was ported to the scikit-image library by Mike Sarahan. The application level code is a one-liner and performs surprisingly well even for larger images.

Table 5.1: Benchmark of the examined methods applied to a sequence of 10000 matrices randomly sampled from a much shorter recorded slip profile. The frame rate is extrapolated to 6 sensor matrices assuming zero overhead and is hence very optimistic. The computationally expensive convolutions, correlations, interpolations and curve-fittings rely on the Python-based NumPy, Skimage and SciPy library and were carried out on a 2.1 GHz Dual-Core machine.

Method	Details	Benchmark	Frame rate
Centroid	Center of mass tracking using vectorization.	2.83 s	590 fps
Alcazar	Centroid of consecutive convolution matrices.	8.15 s	205 fps
NCC 1	A 5×5 neighborhood around the peak is upscaled $4 \times$ using bi-cubic interpolation. A Gaussian-weighted paraboloid is fitted to 11×11 data points in the ordinary least squares sense.	32.33 s	52 fps
NCC 2	Two-dimensional 3-point closed-form Gaussian curve fit.	12.55 s	133 fps
PC	Phase-correlation using selective $32 \times$ upsampling by a matrix-multiply single-step DFT.	18.57 s	90 fps

An analysis of the described algorithms' asymptotic time complexity is omitted since the number of taxels is too small for meaningful performance predictions. And future generations of high resolution tactile sensors will probably demand for other methods such as optical flow estimation. Lacking a proper method to evaluate the accuracy of the slip-detection, only the run-time efficiency was tested in an empirical benchmark. Table 5.1 summarizes the results and figure 5.7 gives an impression of the qualitative differences. Alcazar's method was incorporated into the C++ frame manager and proved to reliably detect slip conditions. The downsides of a centroid based technique are of little significance if the only objective is to find a rough estimate of the slip-vector.

5.3 Rotation

Translational slip is not the only movement that might occur when an object is not grasped firmly enough. Just imagine an elongated object that turns around the point of contact without slipping in any particular direction at all. There are various ways to determine the orientation of an image. To be precise, the actual orientation is not of interest, but the change of orientation is.

After the correlation approach in the previous chapter, a natural extension would be the conversion of the tactile matrices to polar or log-polar coordinates before performing phase correlation. That is, a point (x, y) in Cartesian coordinates is first mapped to (r, θ) where r refers to the radial distance to the center and θ denotes the angle with reference to an axis. A

shift in the correlation matrix in y-direction then indicates rotation. But this mapping introduces errors due to the nonuniform sampling, which is especially a problem for low resolution images as experiments confirmed. Another approach involves the computation of local image intensity gradients, for example using Haar-like features. Their orientation can then be combined, for example using binned histograms (HOG) and tracked over time. But given the small number of contributing taxels, noise will play an important role. Dense optical flow fields would allow the detection of rotation if the movement occurs fast enough. But as mentioned before, this method simply does not perform well on low resolution images. And neither does feature tracking. Orientation-sensitive filters like the two-dimensional Gabor filter are yet another alternative. The convolution of an image with a Gabor filter bank consisting of different angles results in a feature space similar to the simple cell model in the primary visual cortex. But one of the most common ways to compute the orientation of a set of points remains the principal axis method described in the following.

5.3.1 Image moments

Image moments are a well known concept in computer vision. They are widely used in pattern-recognition and shape description applications as they capture characteristic properties of a segmented object. Originating in mechanics where they describe the mass distribution of physical bodies, moments found their way into statistics to characterize the shape of probability density functions. Features like mean, variance, and skewness for instance are commonly referred to as first-, second central- and third standardized moment. Intuitively speaking, moments are a measure of how points deviate from the mean value of a function. Interpreting an image as a spatial random variable gives rise to the notion of images moments. For a two-dimensional probability density function $f(x, y)$, the *geometric moments* of order $(p + q)$ are defined as:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad (5.3)$$

According to Hu (1962), the sequence m_{pq} is uniquely determined by $f(x, y)$ if all nonzero values are in the finite part of the xy-plane only. Hu was also the first to apply the concept of moments to image analysis and introduced additional moments with varying invariance properties. In the case of a discrete image of size $M \times N$ with pixel intensities $f(x, y)$, geometric moments are computed as

$$m_{pq} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} x^p y^q f(x, y)$$

Lower order moments have intuitive interpretations. If the image is normalized such that the sum of all intensity values $m_{00} = 1$, then m_{01} and m_{10} are the mean values. If in addition m_{01} and m_{10} are both zero, then m_{02} and m_{20} are the variances while m_{11} is the corresponding covariance. Thus, the second order moments describe the shape's orientation with respect to the coordinate axes of the image. In analogy to statistics, the skewness is given as $m_{30}/\sqrt{m_{20}^3}$ in x-direction and $m_{03}/\sqrt{m_{02}^3}$ in y-direction. And for the sake of completeness, the ratios m_{40}/m_{20}^2 and m_{04}/m_{02}^2 are known as kurtosis. Assuming elongated object shapes, both third- and forth order moments can help resolve the 180 degree ambiguity concerning orientation. It should be noted that higher order moments are especially sensitive to noise.

In the general case a centroid $\mathbf{x} = (\bar{x}, \bar{y})$ also known as center of mass or gravity can be obtained by dividing the first-order by the zero-order moments, i.e.

$$\bar{x} = \frac{m_{01}}{m_{00}} \quad \text{and} \quad \bar{y} = \frac{m_{10}}{m_{00}}.$$

Based on the centroid, which serves as a reference to the location of the object within the sensor frame, discrete *central moments* can be derived as:

$$\mu_{pq} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

Central moments are invariant with respect to translations since their origin correspond to the centroid. It follows that $\mu_{01} = 0$ and $\mu_{10} = 0$. This is a useful property not only in the context of tactile sensing where the contact points between gripper and object may vary between grasps. Scaling the shape by a common factor $\alpha > 1$ in both dimensions such that $f'(x, y) = \alpha \cdot f(x, y)$ yields

$$\mu'_{pq} = \frac{\mu_{pq}}{\alpha^{p+q+2}}$$

via variable substitution in (5.3). Scale-invariant moments can therefore be constructed by normalizing with the shape's area μ_{00} . *Normalized central moments* are defined as

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{(p+q+2)/2}}$$

which can be interpreted as the scaling of each axis by a factor of $\alpha = 1/\sqrt{\mu_{00}}$. For the purpose of analyzing tactile sensor readings, scale invariancy is not as beneficial as in image registration applications where an object may change its apparent size. Distinguishing between smaller and larger objects of the same type requires shape descriptors that take scale into account. However if objects are to be classified into abstract categories like “round” and “square”, scale should be ignored.

5.3.2 Principal axis method

With the notion of image moments at hand, an explanation of the principal axis method is straightforward. The following derivation is roughly based on a description found in Flusser et al. (2009), a recommendable standard work to moments. Figure 5.5 gives an overview of the method and compares it to the related principal component analysis.

The second order centralized moment matrix, sometimes also called shape dispersion or simply covariance matrix is given as

$$\mathbf{M} = \text{Cov}[f(x, y)] = \begin{bmatrix} \mu'_{20} & \mu'_{11} \\ \mu'_{11} & \mu'_{02} \end{bmatrix}$$

where

$$\mu'_{20} = \frac{\mu_{20}}{\mu_{00}}, \quad \mu'_{02} = \frac{\mu_{02}}{\mu_{00}} \quad \text{and} \quad \mu'_{11} = \frac{\mu_{11}}{\mu_{00}}.$$

As mentioned before, it follows from the normalization that $m_{00} = 1$, $m_{01} = 0$ and $m_{10} = 0$. \mathbf{M} can then be used to normalize for rotation. The covariance measure μ'_{11} is therefore constraint to zero, leading to the diagonalization of \mathbf{M} . This can be done by solving

$$(\mathbf{M} - \lambda_{1,2}\mathbf{I})\mathbf{x} = 0.$$

With Sarrus' rule, the determinant is computed as

$$\det(\mathbf{M} - \lambda_{1,2}\mathbf{I}) = (\mu'_{20} - \lambda_1)(\mu'_{02} - \lambda_2) - \mu'^2_{11}.$$

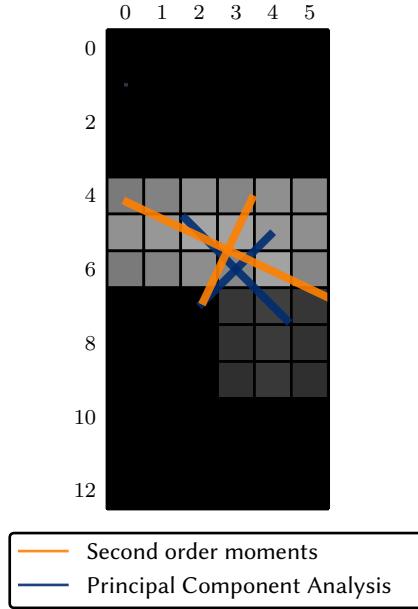


Figure 5.5: Principal Component Analysis vs. second order Image Moments. The length of a semi-axis is determined by the corresponding eigenvalues. In the image moment approach, the center of the ellipse is equivalent to the center of gravity while its position in the PCA case is determined by the mean coordinate of all active cells.

Setting the characteristic polynomial to zero yields the eigenvalues

$$\lambda_{1,2} = \frac{\mu'_{20} + \mu'_{02}}{2} \pm \frac{\sqrt{4\mu'^2_{11} + \mu'^2_{20} - 2\mu'_{20}\mu'_{02} + \mu'^2_{02}}}{2}.$$

The corresponding eigenvectors and thus the orientation of the principal axes can be found by solving the reformulated eigenvalue problem

$$\mathbf{M}\mathbf{x} = \lambda_{1,2}\mathbf{x}, \quad \mathbf{x} \neq \mathbf{0}.$$

One obtains a diagonal matrix with diagonal elements corresponding to the eigenvalues $\lambda_{1,2}$, i.e.

$$\begin{aligned} \mathbf{M}' &= \mathbf{R}^T \mathbf{M} \mathbf{R} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \\ &= \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \mu'_{20} & \mu'_{11} \\ \mu'_{11} & \mu'_{02} \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}. \end{aligned}$$

Note that this decomposition only exists for real symmetric and diagonalizable matrices. By solving the equation with three equations and three unknowns for θ , the angle between the first eigenvector and the x-axis is given as

$$\theta = \frac{1}{2} \arctan \left(\frac{2\mu'_{11}}{\mu'_{20} - \mu'_{02}} \right). \quad (5.4)$$

This expression is only defined if $\mu'_{11} \neq 0$ and $\mu'_{20} \neq \mu'_{02}$. The intuitive interpretation of that restriction is that there simply is no “orientation” for a shape with certain symmetry properties such as a disk with symmetrically distributed intensity values.

The principal axes are related to the notion of the reference ellipse, with the eigenvectors defining the direction and the eigenvalues being proportional to the squares of its semi-axes. It should be noted that the computation of the angle becomes numerically unstable if the corresponding ellipse degenerates to a circle. The orientation however may still be applicable if the skewness is not centered. Suesse and Ditrich (2005) proposed a robust algorithm that takes moments up to the fourth order into account. An implementation showed that their method unfortunately does not perform well for low resolution tactile images. But a grasping algorithm that depends on the orientation angle should be aware of the quality of the measurement. There are several ways to determine the circularity or symmetry of shapes. Some of them are even based on moment invariants. But since the principal axis method produces the best results if the reference ellipse is elongated, a simple measure for elongation will give a good indication for the success of an orientation measurement. A possible shape feature often found in literature is the *eccentricity*, given as

$$e = \sqrt{1 - \left(\frac{\lambda_2}{\lambda_1}\right)^2}.$$

A perfect square shape therefore has an eccentricity of 0 while an elongated rectangle has an eccentricity of 1 in the limit. But a square is only symmetric with a periodicity of π and therefore has an orientation, just four ambiguous ones. As long as such a shape with equal sized eigenvalues rotates slowly enough, the uncertainty can easily be resolved by associating the closest previous angle in a sequence of measurements to it.

A second shape feature is therefore needed to distinguish between a perfectly valid square and an unusable circular object like a disk. The answer is an additional compactness measure that quantifies the roundness of a shape. A circular disk should be perfectly compact while a square should be significantly different. Given again the low resolution of the available tactile image, not all compactness measures found in literature, e.g. Žunić (2011), turned out to perform well. A practical feature is the *roundness* or isoperimetric quotient given as

$$r = \frac{4\pi A}{L^2}$$

with shape area A and perimeter length L . As a side note, the circumference of a circle given the area is $\sqrt{4\pi A}$. The reciprocal value $1/r$ is called compactness, which is somewhat counterintuitive since lower values are considered more compact. In order to compute the area and perimeter, the contour of the shape has to be extracted first. This is done on a pixel level using chain codes and the help of OpenCV. If an object has more than one connected component, only the largest contour is taken into account. The reason why this compactness measure is accompanied by the eccentricity in the first place, is that the contour feature is based on a thresholded binary image. It therefore contains no information on the intensity distribution. But as mentioned before, a circular contour can have an orientation if the shape's intensity values are asymmetric. Figure 5.6 gives an impression of the method with the help of a real-world pressure profile that combines translational and rotational slip.

Now a simple threshold based solution to the original problem can be proposed. According to this scheme, the orientation is only computed if the eccentricity and roundness are within certain ranges. Empirical tests showed that a satisfiable orientation angle can only be determined under the following conditions: Let the eccentricity threshold be $\theta_e = 0.6$ and the roundness threshold be $\theta_r = 0.9$. Then the angle is only reliable if $\neg(e < \theta_e \wedge r > \theta_r)$. Otherwise the shape must be circular and the method is not usable. Both measures complement each other and can be considered to be a robust circularity measure under the given inevitable approximation due to the low resolution.

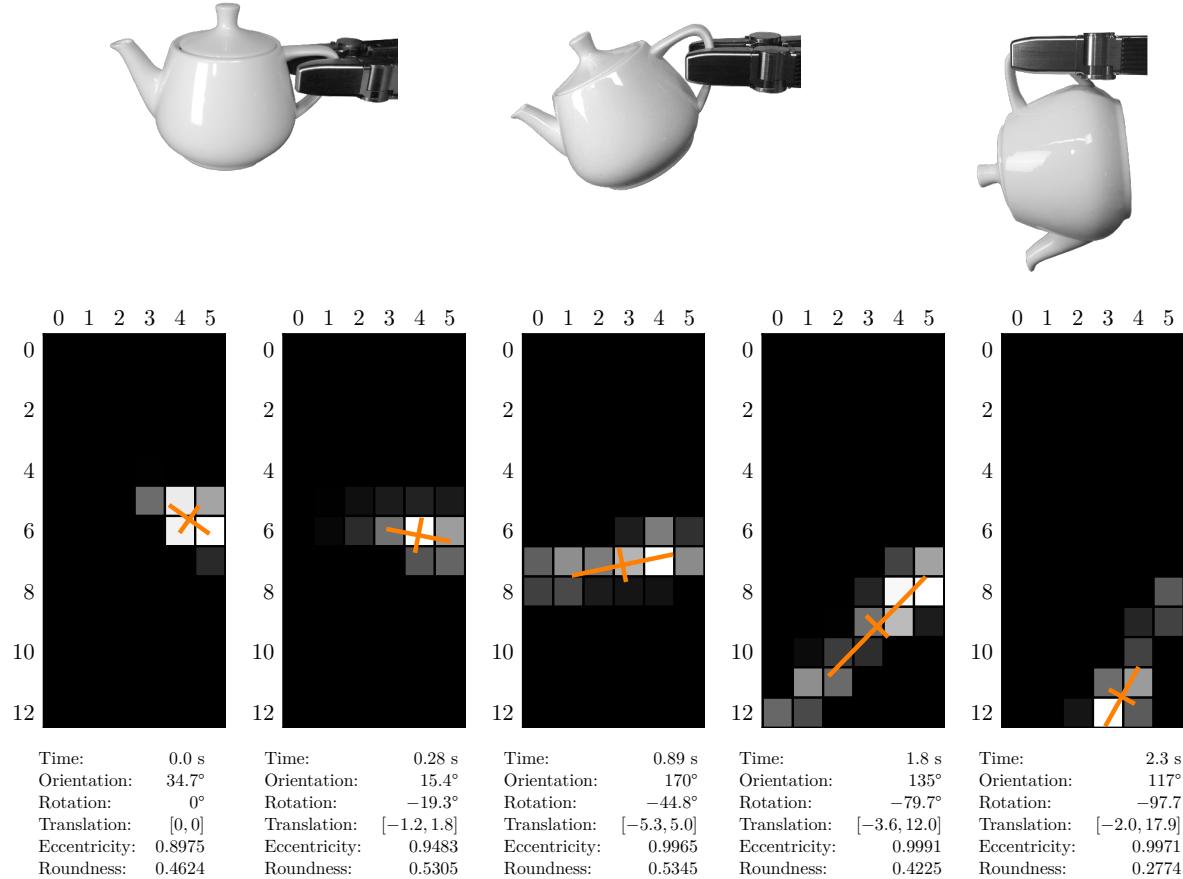


Figure 5.6: Snap-shots of a sliding motion. The images of the slipping teapot roughly correspond to the depicted tactile sensor matrices in the second row. Since the orientation angle is clamped to $[0, 180^\circ]$, the cumulative slip angle is determined by tracking the angle across the method's discontinuities. Note how the eccentricity and roundness measure complement each other.

The actual tracking of the rotation angle is not as trivial as it sounds. After computing the orientation θ in the range $(-\pi/2, \pi/2)$ using (5.4), the angle is clamped to $[0^\circ, 180^\circ]$ in relation to the positive x-axis. With the help of a function that computes the signed difference between two angles while ignoring circular identities, the slip angle between the current orientation $[0^\circ, 180^\circ]$ and the previous angle $[0^\circ, 360^\circ]$ is computed. To be precise, it is computed twice to overcome the 180° ambiguity. After taking the discontinuities $0^\circ \leftrightarrow 360^\circ$ into account, the smallest difference seems to be the more likely one. If the current orientation measure is unreliable, no rotational slip is added to the track. Figure 5.7 sums up the results of the translational and rotational slip detection.

An implemented demo application closes a predefined grasps around an object until the mean forces reach a certain threshold. The slip-detection is then activated just before the robot's arm starts to move. This gives the sensor some time to "level out". If the magnitude of the slip-vector exceeds a certain limit while lifting the object, the grasp is closed a bit further until a second pressure threshold is reached, and so on. It should be noted that there are actually two different thresholds at work. An immediate threshold corresponds for rapid movements between subsequent frames while a second threshold only refers to the displacement to the original position at first contact. As a result, fast as well as slowly moving objects can be detected. This simple

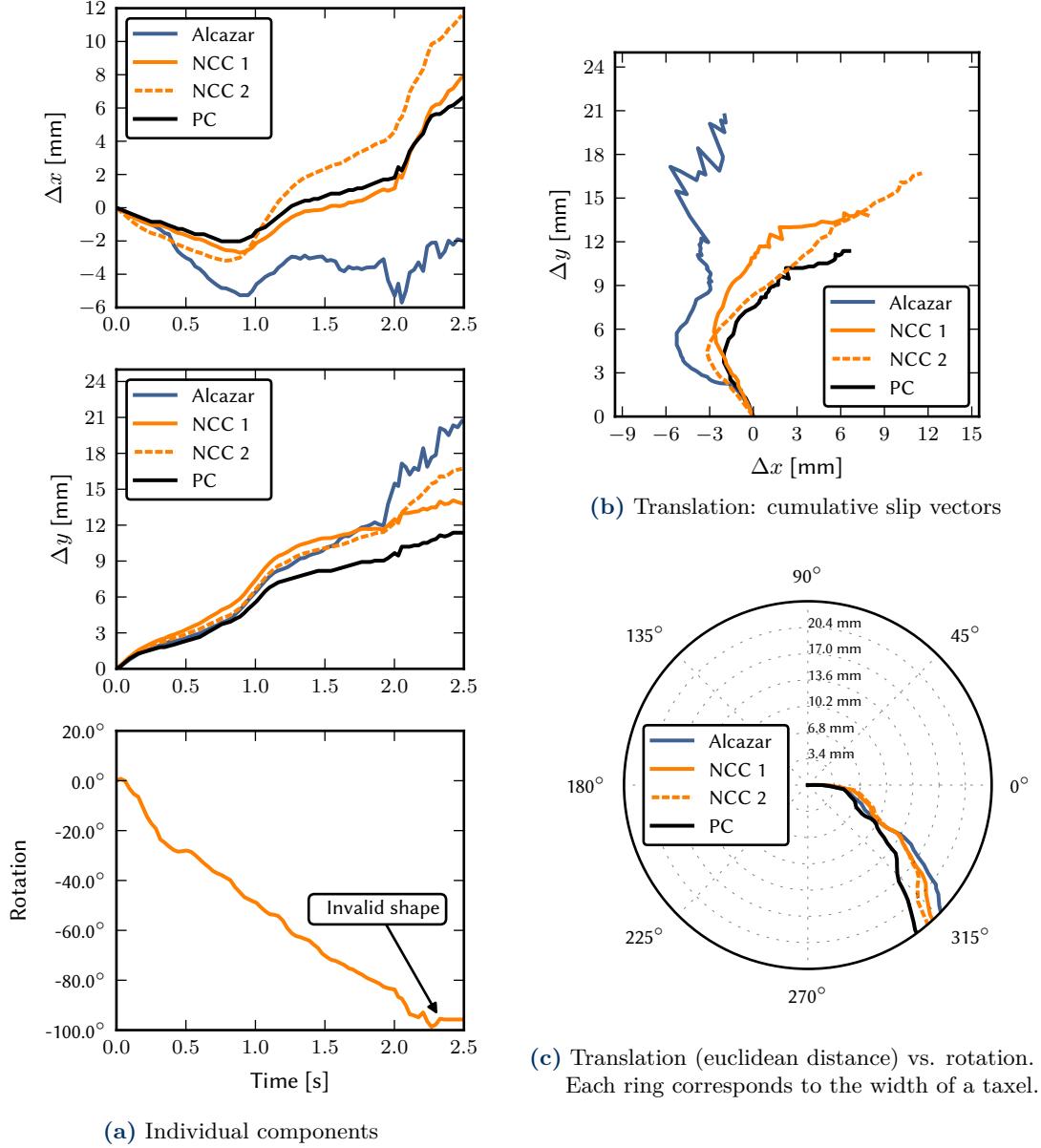


Figure 5.7: Translational and rotational slip-detection: Alternative representations of the same slip trajectories depicted in figure 5.6. Alcazar is the center of mass tracking of the correlation peak, NCC 1 refers to the normalized cross-correlation using interpolation and Gaussian-weighted paraboloid curve fitting while NCC 2 is the result of the 2D 3-point closed-form solution of a Gaussian function. PC stands for the phase-correlation approach. See table 5.1 for details. An objective method to evaluate the accuracy is missing. But to the human eye, the center of mass tracking clearly has some bias while the relatively smooth NCC 2 variant seems to capture the true movement. NCC 1 and PC however show more details.

reactive slip-detection scheme works well until the maximum pressure is reached or the contact is lost. A more advanced algorithm would also try to adapt to the current situation and lower the grasp force again if no slip occurs for a given period of time. The grasp stability unfortunately

does not correlate directly with the measured pressure and continuously closing the hand might even change the grasp for the worse. Slip-detection is therefore by all means no stand-alone solution for stable grasps. It can, however, provide the necessary information to adapt and improve a high-level grasping algorithm in real-time.

The presented methods can be extended to obtain slip vectors in three dimensions as demonstrated by Alcazar and Barajas (2012) as well. Using forward kinematics and the center of mass on the sensor matrices as reference points, a sparse displacement vector field can be created. By adding the slip vectors of all involved matrices, a general direction of the translational movement can be estimated allowing for more complex compensating movements. And if the individual slip vectors sum up to zero, the curl of the vector field can be used to determine the rotational movement of the object. This way, the ambiguous information of the two-dimensional pressure profile about the real movement of the object (cf. aperture problem or barberpole illusion) can be resolved.

Chapter 6

Object classification

6.1 Introduction

The acquisition of accurate 3D sensor data is a crucial prerequisite for robot grasping applications. But all state-of-the-art 3D scanners have to deal with certain limitations, such as difficulties with reflective and transparent objects or cluttered scenes. Tactile sensors on the other hand provide an opportunity to improve the perception of an already touched object, which can then be used to locally optimize the grasp or perform a re-grasp if necessary. Object classification is another, almost natural application of tactile sensors. Consequently, several publications have already addressed this topic with a variety of different feature extraction and classification techniques.

Schopfer et al. (2009) for example conducted a study on 51 simple statistical features of tactile data such as min, max and mean values or moments of second and third order in different spatial resolutions. Their decision tree classification of a large dataset showed that the entropy was distributed over a large set of features. A result, that could be confirmed in this thesis.

As part of a series on papers on blind grasping, already mentioned in chapter 1.3 (Applications), Dang et al. (2011) tried to classify stable grasp-object configurations. They simply combined all available taxels as well as joint angles into a 169-dimensional feature vector and used a SVM classifier with a RBF-kernel to train and predict the stability of robotic grasps. In subsequent publications they introduced more suitable features, presumably to avoid the curse of dimensionality.

Drimus et al. (2011) build a sensor based on piezoresistive rubber and applied it to a classification task. Only two features, namely the mean sensor value and its standard deviation are extracted. But in contrast to most other approaches, the features are computed for every tactile image captured during the grasp. They use the Dynamic Time Warping (DTW) algorithm to extract a scalar similarity measure between two time series. A k -nearest neighbors (k -NN) classification is then applied to the pairwise distances between the sensor readings. Despite of its simplicity, the method is capable to distinguish between rigid and deformable objects solely by comparing the extension of the contact area over time.

Navarro et al. (2012) presented a Bag-of-Words (BoW) approach to object classification which combines tactile images as well as joint configurations. In particular, they propose an unsupervised clustering method instead of the usual feature extraction process. After normalization, a Principal Component Analysis (PCA) is applied to the tactile images for dimensionality reduction. A Self-Organizing Map (SOM) is then used to further reduce the dimensionality while clustering the data in two dimensions. Combined with the likewise clustered joint configurations, they form the final bag of keypoints on which an Artificial Neural Network (ANN) classifier operates. According to the authors, the resulting object recognition framework is quite flexible and performs well on different robotic hands.

An earlier example of the same unsupervised feature extraction idea was presented by Schneider et al. (2009). They used the k -means clustering algorithm to create a codebook for the tactile images and a bag-of-words approach for object identification. Once the histogram of codewords has been obtained for all object categories, new objects are recognized using a Naive Bayes classifier. In particular, distributions over the vocabulary are updated with each additional grasp until the expected information gain does not improve any more.

Pezzementi et al. (2011) went a step further and improved the codebook generation by using rotation invariant Fourier moments instead of raw tactile images. And instead of k -means clustering they used Gaussian mixture models (GMMs). In addition, they proposed local and global exploration strategies to work with unknown object poses.

6.2 Overview

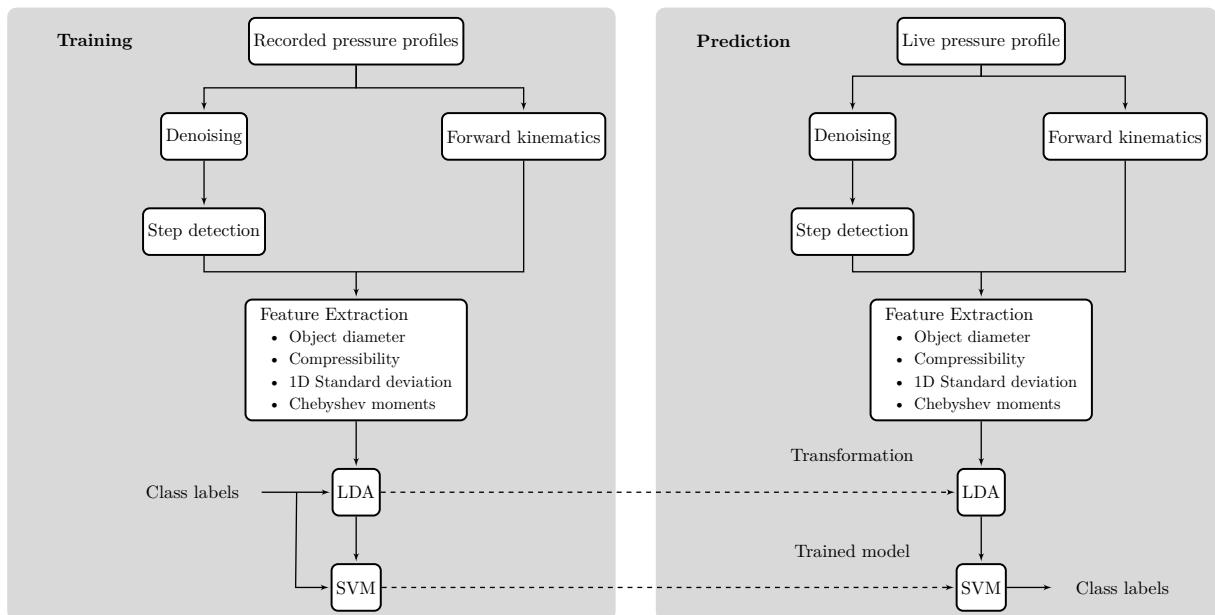


Figure 6.1: Classification workflow

Figure 6.1 gives an overview of a working object-grasp classification procedure using SCHUNK's Dexterous Hand 2.0 with tactile sensors by Weiss Robotics. During the training phase various similar grasps of the same object are performed and recorded. For the sake of simplicity, all grasps are executed with a pinch grip using two fingers only which restricts the tactile feedback to the sensor matrices at the finger tip. By applying a Denavit-Hartenberg-Transformation, the 3D position of each active sensor cell is then computed from the recorded joint angles. Since the tactile readings are noisy, especially while capturing in high sensitivity mode, a spatial digital filter is applied in order to smooth the signal. A median filter that replaces each value with the median of neighboring entries performed best in experiments with a variety of denoising algorithms. In the following step, the actual grasps have to be identified and separated from each other. Begin and end frames are therefore determined using a simple step detection algorithm. The concatenated feature vector consists 54 elements of four different types of features which are further discussed in the following sections. Experiments revealed that the individual features are not equally important. But since they are also not independent from each other, a classical feature selection approach does not work well on the available data. Instead, a Linear Discriminant

Analysis (LDA), a dimensionality reduction techniques related to the Principal Component Analysis (PCA) that maximizes the separation between multiple classes is performed as an intermediate step. A grid-search suggests that the best results are achieved by reducing the 54 dimensions down to 14. A one-vs-one multiclass approach using Support Vector Machines (SVM) is then used to execute the final classification task. The trained classifier, that is the support vectors as well as the performed LDA transformation, is retained for the prediction phase.

6.3 Object diameter

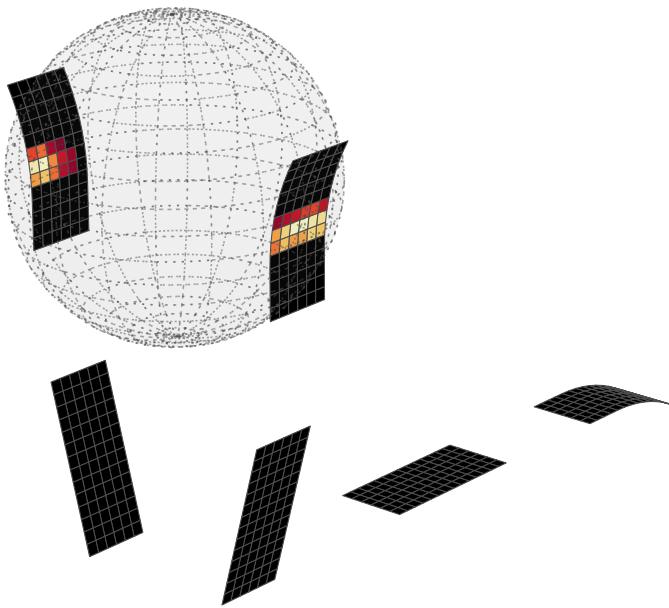


Figure 6.2: Visualization of the minimal bounding sphere, a proxy for the grasped object's size. Forward kinematics are used to obtain a three-dimensional point cloud of all active taxels.

The size of a grasped object is intuitively an important property in the context of classification. Based on the hand's dimensions, the actual forward kinematics can be computed according to the Denavit-Hartenberg-Transformation as discussed in chapter 4. This results in a 3D point cloud of all active sensor cells as seen in figure 6.2. Various distance metrics are conceivable, but those based on the convex hull of the taxel point cloud are very sensitive to the actual grasp-object configuration. Slight changes in the position of the point contacts on the tactile sensors may lead to completely different geometries. In contrast, the radius of the minimal bounding sphere, also known as enclosing ball, has proven to be a versatile metric for the point cloud's extent. Finding the smallest enclosing ball of a set of points is a classical problem of computational geometry. See Fischer (2005) for an in-depth view of the topic. Several algorithms exist for the nontrivial task. The used implementation by Gärtner (1999) relies on the simplex method for linear programming and runs in expected linear time. It is particularly suitable for lower dimensional spaces, like the one of the intended purpose, and is part of the well known CGAL project but also available as a single C++ header file. Training a linear Support Vector Machine

without a preceding dimensionality reduction shows that the corresponding object size feature has the largest assigned weight to it - a clear indication of its discriminatory power.

6.4 Compressibility

The SDH-2 is strong enough to actually compress soft objects. It is therefore possible to measure the object's rigidity by comparing the objects diameter at first contact and after the grasping movement has come to rest. Figure 6.3 illustrates the change in a miniball's diameter during the course of a grasp.

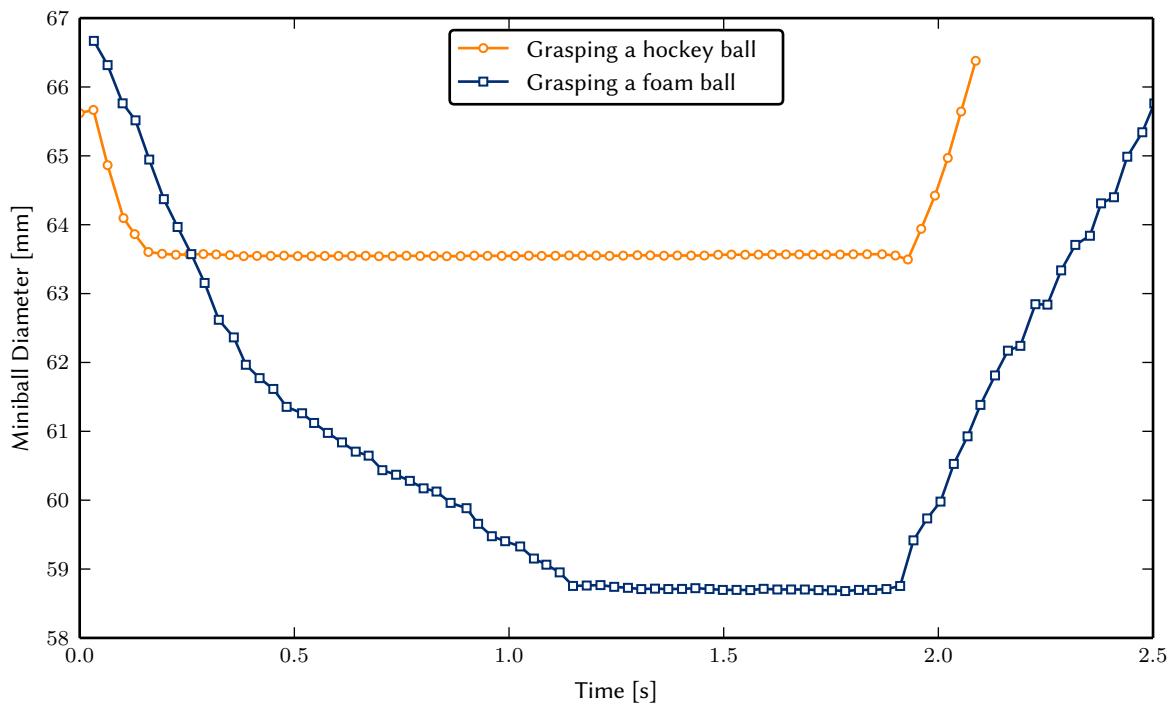


Figure 6.3: Differing compressibility of equal sized objects. The minimal bounding sphere of all active taxels is calculated for each tactile sensor frame. The difference in diameter at the beginning and at the resting position of a grasp serves as a measure of the objects compressibility. In addition, this feature is independent of the actual grasping speed and the dynamics of the compressed object. At first glance, the rigid hockey ball is compressed by about 2 mm. But in reality, the displacement is purely caused by the indentation of the tactile sensor's rubber foam.

It turns out in practice, that the object's initial size cannot be measured accurately enough if the grasp applies torque. In that case, the contact surfaces between the object and both sensor matrices are tilted, which may lead to an overestimation of the real diameter. This asymmetry disappears when all forces reach an equilibrium state. In order to get more robust object size features, the pressure profile's centroids of the end position frame is used to recalculate the diameter during each step of the grasp. The new and hopefully more robust minimal and maximal diameter values are then used to estimate the compressibility of an object, simply by looking at the absolute difference.

6.5 Standard deviation of intensity values

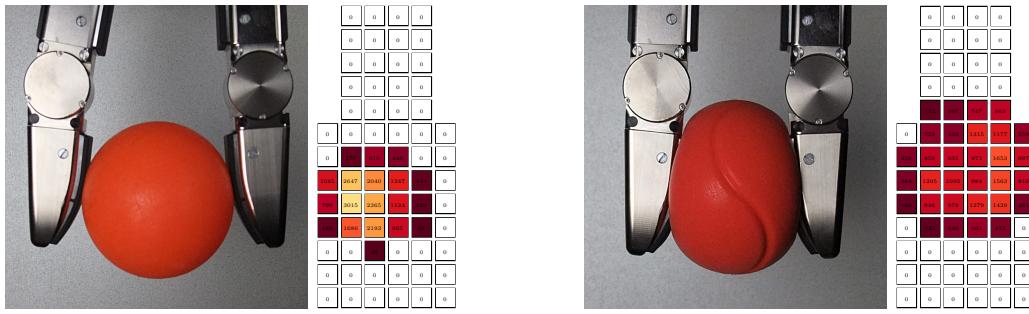
The manipulation of a grasped object might reveal distinctive features such as the already mentioned compressibility. In addition, pressure profiles of rigid bodies differ significantly from those of soft objects. The impression on the tactile sensor matrix appears smoothed out and lacks sharp edges as illustrated in figure 6.4. Therefore, the standard deviation of the intensity values of all active sensor cells represents an easy way to capture this effect, i.e.

$$\sigma = \sqrt{\frac{1}{N} \sum_{x=1}^w \sum_{y=1}^h (t(x, y) - \mu)^2} \quad \text{for } (x, y) > 0$$

with

$$\mu = \frac{1}{N} \sum_{x=1}^w \sum_{y=1}^h t(x, y)$$

where $N = |\{(x, y) \mid t(x, y) > 0\}|$ denotes the number of active taxels on a matrix of size $w \times h$. Note that this feature differs from the second central, two-dimensional image moment.



(a) Pressure profile of a rigid hockey ball

(b) Pressure profile of a foam ball

Figure 6.4: Objects similar in color and size pose a problem to conventional sensors such as cameras or 3D scanners. Apart from more advanced 2D and 3D features, the standard deviation of the taxel's intensity values is a simple but effective indicator of the objects overall consistency.

6.6 Chebyshev moments

Pattern recognition is the basis for object recognition. As motivated by figure 6.5, simple features such as the standard deviation of intensity values are not sufficient to distinguish between more complex pressure profiles. The extraction of image features is a well researched area of computer vision and image retrieval. Detailed surveys and reviews can for example be found in Zhang and Lu (2004) and Tuytelaars and Mikolajczyk (2008). Yang et al. (2012) classify existing approaches to shape-based feature extraction as illustrated in figure 6.6.

Since the spatial resolution of the tactile sensor matrices in use is only 6 by 14 taxels, the number of available feature extraction methods is limited. Without upscaling the tactile information, contour and shape features are as unusable as image feature detectors such as the well known scale-invariant feature transform (SIFT).

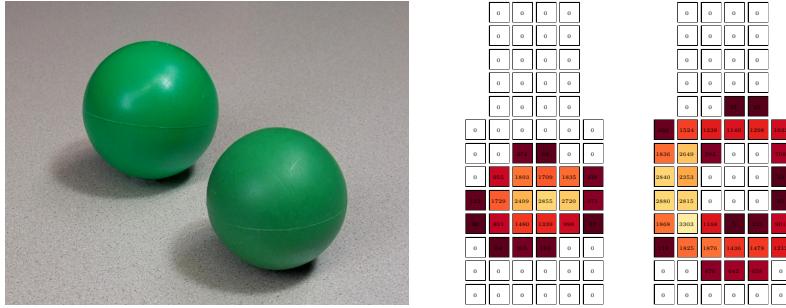


Figure 6.5: Motivating example of advanced shape descriptors: Two very similar plastic balls may lead to completely different pressure profiles. A complex shape, for instance resulting from a ball that buckles under pressure, imposes certain requirements on the feature descriptors.

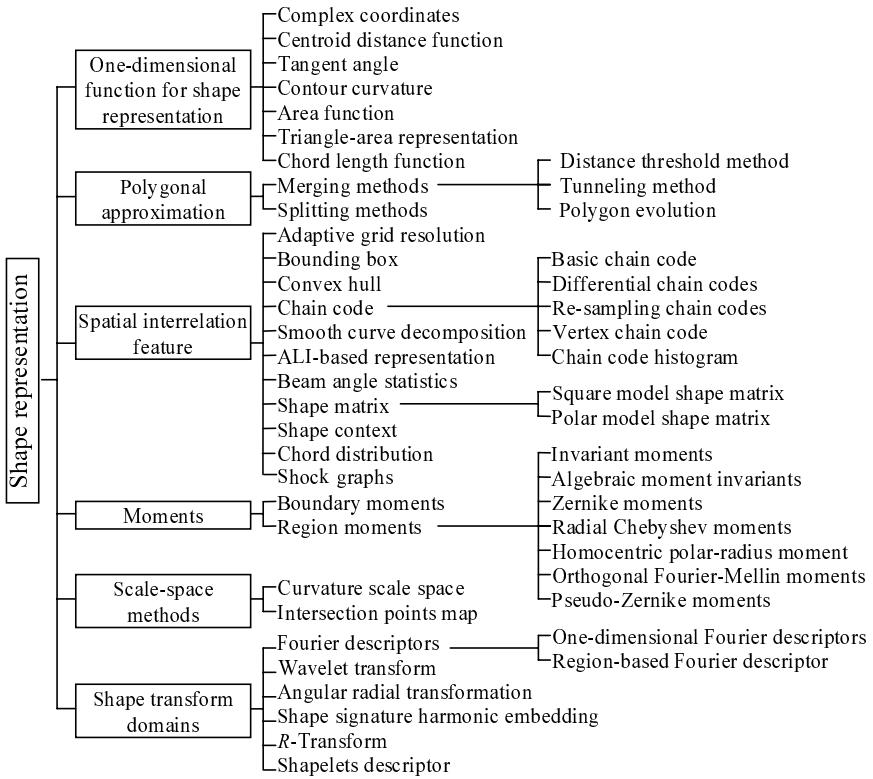


Figure 6.6: Shape description techniques, Yang et al. (2012).

In addition, certain invariance properties are also crucial for the classification success. For example, the extracted features should be invariant to the actual position of the contact area. Rotating the object inside the hand (or vice versa) should not influence the outcome. But similar pressure profiles of different sizes do indicate different objects. To sum it up, the required set of features has to be translation- and rotation invariant but not scale invariant.

One of the most simple approaches to achieve these requirements involves a preceding normalization step. The pressure profile would then simply be translated and rotated such that the shape is centered with standard orientation. A simple taxel based nearest neighbor classification complements the method. Traditional approaches following the eigenface method would also apply a dimensionality reduction via a principal component analysis. Early experiments

showed, however, that the normalization step introduces a large amount of error due to the low resolution of the tactile information. Upscaling before normalizing reduces the effect but at the same time defeats the purpose of dimensionality reduction.

Very recently, Luo et al. (2015) presented a Tactile-SIFT descriptor. This promising approach could not be considered in this thesis but it is worth mentioning their key results. Based on the observation that the high dimensionality of the original SIFT features, see Lowe (2004), tends to overfit the lower dimensional tactile images, several modifications are proposed: First, the scale-space pyramid and key-point localization steps are completely eliminated. The descriptor dimensionality is then reduced from 4×4 orientation histograms of 8 directions each in a 4×4 taxel neighborhood to a single orientation histogram consisting of 8 directions in a 3×3 taxel environment. The tactile image is then segmented into 3 overlapping sub-patches and a orientation histogram is extracted from each sub-patch. Finally they combine the feature vectors to a dictionary in a Bag-of-Words classification approach.

Apart from the Tactile-SIFT descriptor, the most promising features found in literature seem to be moment based. The technique is therefore examined in the following.

6.6.1 Geometric moments and orthogonality

The next topic requires the introduction of some basic concepts from approximation theory. Given a continuous real-valued function f defined on the closed interval $[a, b]$, the n th degree polynomial

$$p_n(x) = \sum_{k=0}^n a_k x^k$$

that minimizes the approximation error

$$\int_a^b [f(x) - p_n(x)]^2 dx$$

is called least squares approximating polynomial. As laid out for example in Burden and Faires (2010), the problem of finding the real coefficients a_k is greatly simplified when dealing with *orthogonal* basis sets. A set of polynomials $\{p_0, p_1, \dots, p_n\}$ satisfies the orthogonality condition over the interval $a \leq x \leq b$ with respect to the weight function $w(x)$ with

$$\int_a^b p_m(x) p_n(x) w(x) dx = \delta_{m,n} c_n \quad (6.1)$$

where $\delta_{m,n}$ is the Kronecker delta given as

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

If in addition their norm is 1, i.e. $c_n = 1$, the pair of polynomials is said to be orthonormal. The weight can be used to assign different levels of importance to certain portions of the interval $[a, b]$. In the discrete case, weighted orthogonality is achieved respectively with

$$\sum_{x=0}^{N-1} p_m(x) p_n(x) = \rho(n, N) \delta_{m,n}, \quad 0 \leq m, n \leq N-1 \quad (6.2)$$

where $\rho(n, N)$ is called the squared norm of the polynomial set $\{p_n\}$.

In the orthogonal case, the error function of the least squares approximation is

$$\int_a^b w(x) \left[f(x) - \sum_{k=0}^n a_k p_k(x) \right]^2 dx$$

and the coefficients of the discrete approximation are easily determined by

$$a_k = \frac{1}{\rho(n, N)} \sum_{x=0}^{N-1} w(x) f(x) p_k(x).$$

As already mentioned in subsection 5.3.1 (Image moments), the geometric moments of an image of size $M \times N$ with intensity values $f(x, y)$ are given as

$$m_{pq} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} x^p y^q f(x, y)$$

and can be interpreted as the projection of $f(x, y)$ to the monomial $x^p y^q$. In theory, it follows from the definition of a vector space that all spanning sets are equivalent, since arbitrary vectors in the space can be uniquely expressed as a linear combination of the basis vectors. This statement does not hold in practice, as many authors working with geometric moments had to recognize. Flusser et al. (2009) summarize specific problems that occur when standard powers are used as the moments basis set.

- ▷ The large variation in the dynamic range of values leads to a loss of precision due to arithmetic overflow and underflow.
- ▷ Correlating moments lead to information redundancy and require moments of higher order.
- ▷ But higher order moments are more sensitive to image noise.
- ▷ Image reconstruction, though rarely used, is complicated.

Teague (1980) realized the shortcomings of traditional moments and proposed the use of orthogonal moments in pattern and object recognition tasks. He therefore replaced the monomial basis set $\{x^p y^q\}$ with Zernike polynomials and algebraically constructed translation, rotation and scaling (TRS) invariants. Subsequent publications, e.g. Teh and Chin (1988), Kim et al. (2000), Zhang and Lu (2001) and Li et al. (2009), focused on Legendre and Pseudo-Zernike polynomials and associated computational aspects.

Two-dimensional polynomials are usually constructed as a product of two one-dimensional polynomials. the general form of orthogonal moments defined on a rectangle is therefore given as

$$M_{pq} = \nu_p \nu_q \iint_{\Omega} p_p(x) p_q(y) f(x, y) dx dy, \quad p, q \in \mathbb{N}_0 \quad (6.3)$$

where the domain of the image intensity function $f(x, y)$ is scaled to the area of orthogonality Ω . ν_p and ν_q are normalization constants.

From (6.3), one can immediately conclude that continuous orthogonal polynomials, such as the ones mentioned above, have two major drawbacks when applied to image processing. For one, the discretization of continuous integrals introduces numerical errors, especially when a simple summation scheme is used. But in addition, the approximation error is amplified by the fact that the image coordinate space has to be transformed to match the area of orthogonality of the polynomials in use.

6.6.2 Chebyshev polynomials

Mukundan et al. (2000) recognized the problems and proposed Chebyshev moments in order to resolve both problems at the same time. His series of publications R. Mukundan (2001), Mukundan et al. (2001) and Mukundan (2004a) form the basis of the following derivation. The well-known Chebyshev polynomials of the first kind are typically defined via the recurrence relation:

$$\begin{aligned} t_0(x) &= 1 \\ t_1(x) &= x \\ t_n(x) &= 2x t_{n-1}(x) - t_{n-2}(x). \end{aligned} \quad (6.4)$$

To avoid confusion, a lowercase t_n is used to denote the polynomial of order n in contrast to thereon based moments. The polynomials are named after Pafnuty Chebyshev (1821-1894), also transliterated, among others, as Tchebychev or Tschebyschow (hence the letter t) who made wide-ranging contributions to mathematics. They are a special case of the Jacobi and Gegenbauer polynomials and can be expressed in many different ways. For a better understanding and because they appear in various numerical analysis applications, a short overview on Chebyshev polynomials is given below. Their relation to trigonometric functions is for example demonstrated by the identity

$$t_n(\cos \theta) = \cos(n\theta), \quad x \in [-1, 1], n \in \mathbb{N}_0.$$

Substituting $x = \cos \theta$ leads to

$$t_n(x) = \cos(n \arccos x).$$

The already introduced triple recurrence relation (6.4) can be derived with the help of the trigonometric addition formulas

$$\cos(\alpha \pm \beta) = \cos \alpha \cos \beta \mp \sin \alpha \sin \beta.$$

Chebyshev polynomials also arise as the solution to the Chebyshev differential equation

$$(1 - x^2) \frac{d^2y}{dx^2} - x \frac{dy}{dx} + p^2 y = 0.$$

The sequence of polynomials t_n is orthogonal on the interval $[-1, 1]$ with respect to the weight function

$$w(x) = \frac{1}{\sqrt{1 - x^2}}$$

such that

$$\int_{-1}^1 p_m(x) p_n(x) \frac{1}{\sqrt{1 - x^2}} dx = \begin{cases} 0 & \text{if } m \neq n \\ \pi & \text{if } m = n = 0 \\ \pi/2 & \text{if } m = n \neq 0 \end{cases}$$

Figure 6.7 gives an impression of the first six polynomials on their interval of orthogonality.

There are n distinct roots of the Chebyshev polynomials $t_n(x)$ in the interval $[-1, 1]$. They are located at

$$x_k = \cos \left(\frac{2k-1}{2n} \pi \right), \quad k = 1, 2, \dots, n$$

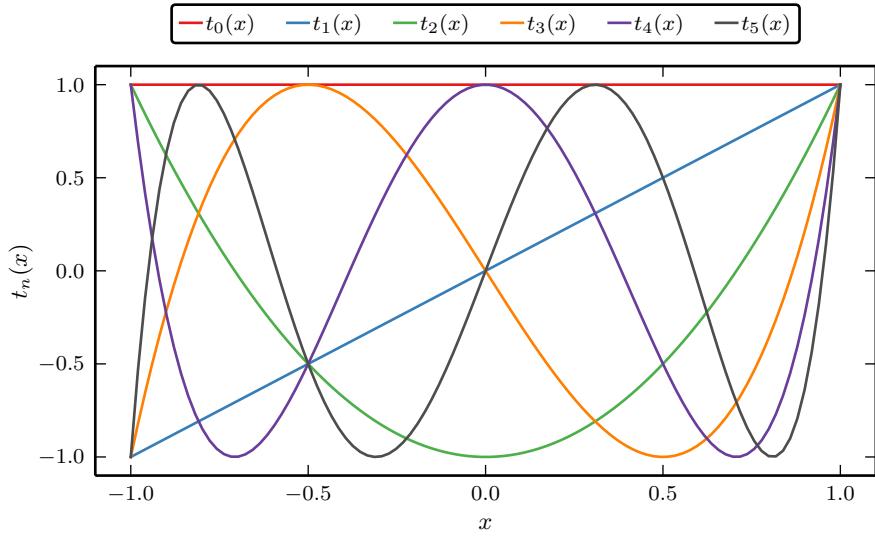


Figure 6.7: Chebyshev polynomials of the first kind in the interval $[-1, 1]$. Extremal points are located at $x_k = \cos\left(\frac{\pi k}{n}\right)$ for $k = 0, 1, \dots, n$. Maxima and minima have values $+1$ and -1 and occur at even and odd values of k . The roots are at $x_k = \cos\left(\frac{(2k-1)\pi}{2n}\right)$ for $k = 1, 2, \dots, n$ and may serve as nodes in polynomial interpolation as the nonuniform grid minimizes Runge's phenomenon.

and are of particular importance in numerical analysis, but beyond the scope of this thesis. With reference to equation (6.2), the Chebyshev polynomials $t_n(x)$ with $n \leq N$ additionally satisfy the *discrete* orthogonality condition on the grid of the roots of $t_N(x)$.

$$\sum_{x=0}^{N-1} p_m(x_k) p_n(x_k) = \begin{cases} 0 & \text{if } m \neq n \\ N & \text{if } m = n = 0 \\ N/2 & \text{if } m = n \neq 0 \end{cases}$$

where x_k are the roots of $t_N(x)$. Since the roots x_k can be transformed to an arbitrary interval $[a, b]$ with

$$x'_k = \frac{(a+b)}{2} + \frac{(b-a)}{2} x_k, \quad k = 1, 2, \dots, n$$

the polynomials exactly satisfy the orthogonality relation in the discrete domain of the image coordinate space. The need for numerical approximations is thereby completely eliminated, which leads to far better results compared to moments based on other orthogonal polynomials.

6.6.3 Chebyshev moments

Two-dimensional Chebyshev moments are defined by

$$T_{pq} = \frac{1}{\rho(p, N)\rho(q, N)} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} t_p(x) t_q(y) f(x, y) \quad (6.5)$$

where the discrete Chebyshev polynomials $t_n(x; N)$ are modified such that their interval of orthogonality is $[0, N - 1]$. The modified recurrence relation of the discrete polynomials is

$$(n+1)t_{n+1}(x) - (2n+1)(2x - N + 1)t_n(x) + n(N^2 - n^2)t_{n-1}(x) = 0 \quad (6.6)$$

with initial conditions

$$t_0(x) = 1 \quad \text{and} \quad t_1(x) = 2x - N + 1.$$

A closed-form expression often found in literature is

$$t_n(x; N) = n! \sum_{k=0}^n (-1)^{n-k} \binom{N-1-k}{n-k} \binom{n+k}{n} \binom{x}{k} \quad (6.7)$$

with $\binom{n}{k} = \frac{n!}{k!(n-k)!}$. The occurrence of the binomial coefficients suggests a connection to hypergeometric functions. We shall return to this point shortly. In the rest of this thesis $t_n(x)$ and $t_n(x; N)$ are used synonymously and usually refer to the discrete version. Figure 6.8 illustrates two-dimensional discrete Chebyshev polynomials.

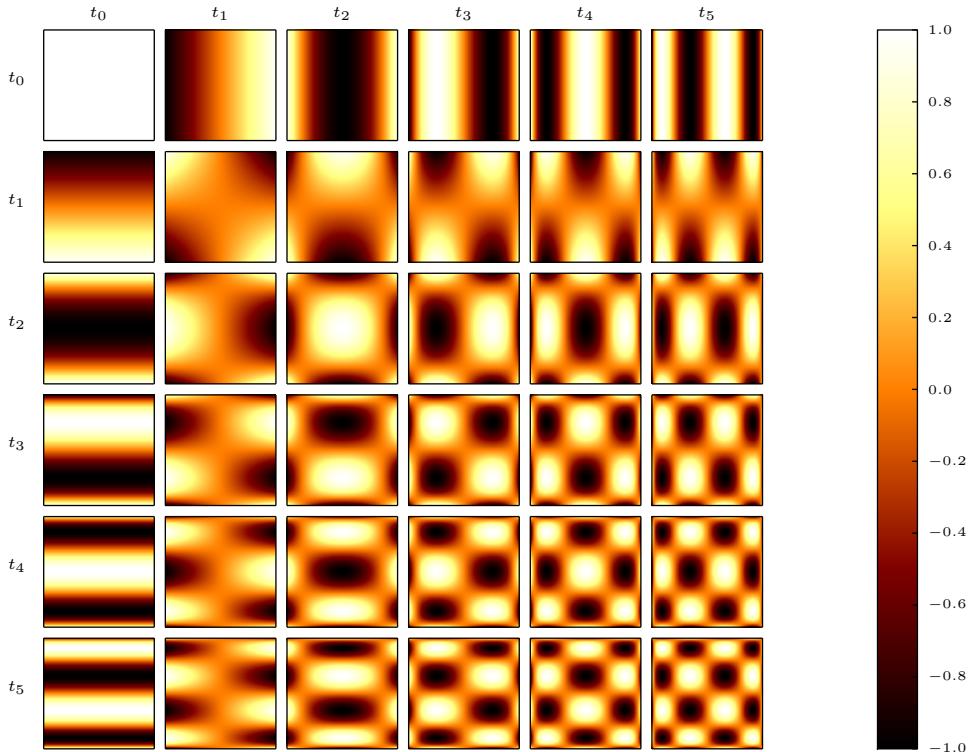


Figure 6.8: The basis functions of the two-dimensional discrete Chebyshev moments up to order $p, q = 5$ with $N = 256$. Positive amplitudes appear brighter, negative amplitudes are shown darker.

The recurrence formula (6.6) and the closed form solution (6.7) are not suitable to derive moment invariants. In order to accomplish that goal, a short excursion to combinatorics is inevitable. Using the notation put forward by Knuth (1992), the n th falling factorial power of x is defined as

$$x^n = x(x-1)(x-2)\dots(x-n+1) = \frac{x!}{(x-n)!}, \quad x \in \mathbb{R}, n \in \mathbb{N}$$

while the rising factorial is

$$x^n = x(x+1)(x+2)\dots(x+n-1) = \frac{(x+n-1)!}{(x-1)!}, \quad x \in \mathbb{R}, n \in \mathbb{N}.$$

Both polynomials are related by $x^{\bar{n}} = (-1)^n(-x)^{\bar{n}}$. To avoid confusion, it should be noted that in the theory of special functions the Pochhammer symbol $(x)_n$ refers to the rising factorial. In combinatorics by contrast, $(x)_n$ is used as a notation for the falling factorial while $(x)^n$ denotes the rising factorial.

Expanding the polynomials yields two other well known sequences, the Stirling numbers:

$$x^n = \sum_{k=0}^n S_1(n, k) x^k \quad (6.8)$$

and

$$x^n = \sum_{k=0}^n S_2(n, k) x^k.$$

This relation to the power series is important to make use of the translation invariants of geometric moments as explained further in a later section. For positive integers n, k with $n \geq k$, the *signed* Stirling numbers of the first kind are given by the recurrence relation

$$\begin{aligned} S_1(n, n) &= 1 \\ S_1(n, 0) &= S_1(0, k) = 0 \\ S_1(n, k) &= S_1(n - 1, k - 1) - (n - 1)S_1(n - 1, k). \end{aligned}$$

Since ambiguous notations are commonly found in literature, caution should be exercised with regard to the *unsigned* Stirling numbers of the first kind given by

$$|S_1(n, k)| = (-1)^{n-k} S_1(n, k). \quad (6.9)$$

The Stirling numbers of the second kind are defined by

$$\begin{aligned} S_2(0, 0) &= 1 \\ S_2(n, 0) &= S_2(0, k) = 0 \\ S_2(n, k) &= S_2(n - 1, k - 1) + k \cdot S_2(n - 1, k). \end{aligned}$$

Orthogonal polynomials can also be defined in terms of generalized hypergeometric functions as laid out in Koekoek and Swarttouw (1998).

$${}_pF_q(a_1, \dots, a_p; b_1, \dots, b_q; z) = \sum_{n=0}^{\infty} \frac{\prod_{k=1}^p a_k^{\bar{n}}}{\prod_{k=1}^q b_k^{\bar{n}}} \frac{z^n}{n!}.$$

In particular, one way of expressing the classic discrete Chebyshev polynomials of the first kind is

$$\begin{aligned} t_n(x) &= (1 - N)^{\bar{n}} {}_3F_2(-n, -x, 1 + n; 1, 1 - N; 1) \\ &= (1 - N)^{\bar{n}} \sum_{k=0}^n \frac{(-n)^{\bar{k}} (-x)^{\bar{k}} (1 + n)^{\bar{k}}}{(k!)^2 (1 - N)^{\bar{k}}} \quad n, x = 0, 1, \dots, N - 1. \end{aligned} \quad (6.10)$$

The orthogonality property on the closed interval $[-1, 1]$ is achieved with:

$$\begin{aligned}
 \rho(n, N) &= \sum_{x=0}^{N-1} \{t_n(x)\}^2 \\
 &= \frac{N(N^2 - 1^2)(N^2 - 2^2) \dots (N^2 - n^2)}{2n + 1} \\
 &= (2n)! \binom{N+n}{2n+1} \\
 &= \frac{(N+n)!}{(2n+1)(N-n-1)!}, \quad n = 0, 1, \dots, N-1.
 \end{aligned} \tag{6.11}$$

As demonstrated by Mukundan (2003), moments computed that way suffer from numerical instabilities as the squared norm $\rho(n, N)$ tends to zero for large values of N . Scaling the polynomials by a factor $\beta(n, N)$ which is independent of x , i.e.

$$\tilde{t}_n(x) = \frac{t_n(x)}{\beta(n, N)}$$

resolves this issue. The squared norm of the scaled polynomials is thus

$$\tilde{\rho}(n, N) = \frac{\rho(n, N)}{\beta(n, N)^2}.$$

Since the magnitudes of $t_n(x)$ grow with a rate of N^n , a suitable scale factor is $\beta(n, N) = N^n$. This leads to smaller variations of the dynamic range of values for different moment orders and therefore reduce numerical errors. Another possible choice is $\beta(n, N) = \sqrt{\rho(p, N)}$ resulting in orthonormal polynomials with $\tilde{\rho}(n, N) = 1$.

6.6.4 Computational aspects

One problem concerning the practical use of Chebyshev moments in real-time applications is the high computational cost. A first naive implementation showed that despite the tactile sensor's low resolution, further optimization was necessary to meet the demands of feature descriptors. Shu et al. (2010) achieved considerable speedup using the symmetry property $t_n(N-1-x) = (-1)^n t_n(x)$ and the so called intensity slice representation. However, this method is not applicable to the intended purpose. A lookup-table approach following Zhang et al. (2010) is therefore used instead. Given equations (6.8), (6.9) and (6.10) the orthonormal Chebyshev polynomials of order p can be rewritten as

$$\tilde{t}_p(x) = \sum_{k=0}^p c_{pk} x^k \tag{6.12}$$

with

$$c_{pk} = \frac{1}{\beta(p, N)} \sum_{r=k}^p S_1(r, k) \frac{(-1)^{p+r} (p+r)! (N-r-1)!}{(p-r)! (r!)^2 (N-r-1)!}.$$

The elements c_{pk} form a lower triangular matrix $\mathbf{C} = (c_{pk})$ of size $N \times N$. Accordingly, the inverse matrix is denoted by $\mathbf{D} = \mathbf{C}^{-1}$. Let the column vector $\mathbf{u} = (t_0(x), \dots, t_{N-1}(x))^T$ and

row vector $\mathbf{v} = (x^0, \dots, x^{N-1})$. Then from equation (6.12) follows $\mathbf{u} = \mathbf{C}\mathbf{v}$ and consequently $\mathbf{v} = \mathbf{C}^{-1}\mathbf{u} = \mathbf{D}\mathbf{u}$. And thus:

$$x^p = \sum_{k=0}^p d_{pk} t_k(x) \quad (6.13)$$

where

$$d_{pk} = \beta(k, N) \sum_{m=k}^p S_2(k, m) \frac{(2k+1)(m!)^2(N-k-1)!}{(m+k+1)!(m-k)!(N-m-1)!}.$$

To summarize, given the coefficients in \mathbf{C} the discrete Chebyshev Moments can be computed as follows:

$$\begin{aligned} T_{pq} &= \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \tilde{t}_p(x) \tilde{t}_q(y) f(x, y) \\ &= \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \sum_{m=0}^p \sum_{n=0}^q c_{pm} c_{qn} f(x, y). \end{aligned} \quad (6.14)$$

Since the size of the tactile sensor matrices remains constant, the resulting coefficients, namely the elements of matrix \mathbf{C} only need to be computed once. They can then be stored in a lookup table which greatly improves the performance in real time applications.

6.6.5 Translational invariance

Translation invariance can be achieved by image normalization methods which inevitably introduces approximation errors. A more efficient approach using Zernike Moments, another orthogonal set of polynomials, is known for quite some time, see Chong et al. (2003). As mentioned before, translation invariant Chebyshev moments can be derived algebraically from the corresponding Geometric Moments as well. Following Zhu et al. (2007a) the Central Chebyshev Moments are given by

$$T'_{pq} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \tilde{t}_p(x - \bar{x}) \tilde{t}_q(y - \bar{y}) f(x, y) \quad (6.15)$$

where \bar{x} and \bar{y} are the image centroids. Since equation (6.15) is again expensive to evaluate, an alternative formulation based on the previous steps is derived in the following. The basic idea here is to express the translation invariant Chebyshev Moments as a linear combination of the original moments. First, equation (6.12) is rewritten using the binomial theorem:

$$\begin{aligned} \tilde{t}_p(x - \bar{x}) &= \sum_{m=0}^p c_{pm} (x - \bar{x})^m \\ &= \sum_{m=0}^p \sum_{s=0}^m \binom{m}{s} c_{pm} x^s (-\bar{x})^{m-s}. \end{aligned}$$

Using (6.13) yields

$$\tilde{t}_p(x - \bar{x}) = \sum_{m=0}^p \sum_{s=0}^m \sum_{i=0}^s \binom{m}{s} c_{pm} d_{si} (-\bar{x})^{m-s} t_i(x).$$

Substituting the result back in (6.15) leads to the final translation invariant formulation

$$T'_{pq} = \sum_{m=0}^p \sum_{n=0}^q \sum_{s=0}^m \sum_{t=0}^n \sum_{i=0}^s \sum_{j=0}^t \binom{m}{s} \binom{n}{t} c_{pm} c_{qn} d_{si} d_{tj} (-\bar{x})^{m-s} (-\bar{y})^{n-t} T_{pq} \quad (6.16)$$

where T_{pq} are the original discrete Chebyshev Moments defined earlier in (6.14).

6.6.6 Rotational invariance

It is often desirable to work with feature descriptors that are independent of the objects position and orientation in the scene. In addition to invariance with respect to translation, rotational invariance is therefore an important property in many object recognition tasks. Consequently, a vast number of different techniques to construct moment invariances has been proposed over the past decades. Well known methods include image normalization using the principal axis method as discussed in section 5.3.2 and normalization constraints based on complex moments where the rotation is transformed into a shift. Influenced by the classic similarity invariants by Hu (1962), which were derived using the theory of algebraic invariants and later be proven to be neither independent nor complete, Zernike, Pseudo-Zernike and orthogonal Fourier-Mellin moments were investigated just to name a few. See Pawlak (2006) for details. In this regard Flusser and Suk (2006) found a generalization based on complex moments to derive complete sets of moment invariants of any order.

Continuous orthogonal polynomials defined on the unit circle, such as Zernike moments, are subject to a coordinate space transformation when applied to the discrete domain of an image. The approximation of the continuous moment integrals introduces additional errors. As a logical consequence, Radial Chebyshev Moments (RCM) were introduced by Mukundan (2004b) to combine the advantages of a polar coordinate formulation with the discretization properties of Chebyshev moments. Several variants were proposed, e.g. Mukundan et al. (2005), Bin Xiao (2010) and Li et al. (2010). Conducted experiments show that RCMs are more robust in the presence of noise and have better recognition capabilities than other moment invariants. Following the original definition, Radial Chebyshev Moments of order p and repetition q of an image of size $N \times N$ are given as

$$S_{pq} = \frac{1}{n\rho(p, m)} \sum_{r=0}^{m-1} \sum_{k=0}^{n-1} t_p(r) e^{-i \frac{2\pi q k}{n}} f(r, \theta_k).$$

The radial parameter $r \in [0, \frac{N}{2}]$ varies in $m = \frac{N}{2}$ steps whereas the angle $\theta \in [0, 2\pi]$ is discretized in n segments equal to the number of taxels along the circumference of the circle. The normalization constant is thus not 2π but n . Hence we have

$$\theta_k = \frac{2\pi k}{n}, \quad k = 0, 1, \dots, n-1.$$

Figure 6.9 illustrates the mapping $(x, y) \mapsto (r, \theta)$.

The coordinates of the taxels are then computed using the equations

$$x = \frac{rN}{2(m-1)} \cos\left(\frac{2\pi k}{n}\right) + \frac{N}{2}$$

and

$$y = \frac{rN}{2(m-1)} \sin\left(\frac{2\pi k}{n}\right) + \frac{N}{2}.$$

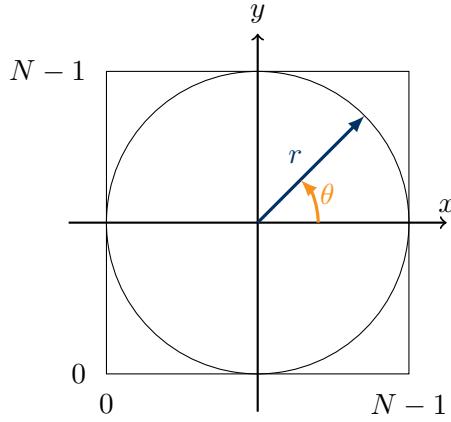


Figure 6.9: Sampling in radial-polar coordinates

As mentioned before, constructing rotational invariants is straightforward since the moments of an image rotated by an angle α about the center undergo the transformation

$$S'_{pq} = S_{pq} e^{-q\alpha}.$$

It is obvious that moments that are independent of α must be invariant with respect to rotation. The RCM invariants are therefore

$$\eta'_p = S_{p0}$$

$$\eta'_{pq} = |S_{pq}|^2.$$

With regard to computational complexity, it is worth mentioning that RCMs are separable, that is they can be split into independent radial and circular moments. Following that approach, the radial moments of order p are can be written in the form

$$Q_p(k) = \frac{1}{\rho(p, m)} \sum_{r=0}^{m-1} t_p(r) f(r, \theta_k)$$

while the circular moments with repetition q are given as

$$S_{pq}^{(c)} = \frac{1}{n} \sum_{k=0}^{n-1} \cos\left(\frac{2\pi q k}{n}\right) Q_p(k)$$

and

$$S_{pq}^{(s)} = \frac{1}{n} \sum_{k=0}^{n-1} \sin\left(\frac{2\pi q k}{n}\right) Q_p(k)$$

such that the real-valued components can be combined to the complex moments

$$S_{pq} = S_{pq}^{(c)} - i S_{pq}^{(s)}.$$

The Radial Chebyshev Moments are defined on a circle, but since the area of the entire tactile sensor is of interest, the rectangular matrices have to be squared by padding out the smaller sides. This also makes sense when working with moments defined on a rectangle to simplify

the formulation. But due to the fact that the original image has to be inscribed in the circle in order to sample from all taxels, the sensor matrix is additionally padded with $\lfloor \sqrt{2m} - m \rfloor$ taxels. Numerical experiments showed that the errors introduced by the mapping $(x, y) \mapsto (r, \theta)$ are too significant in the case of low resolution images, such as the tactile sensor frames in question. Reason for that is, that the sampling is not uniform with respect to the covered taxel area. While taxels in the image center are sampled too often, taxels further away are often missed. This problem could be tackled with nonlinear coordinate mappings, but this only makes sense in case of higher resolution images. An alternative approach is therefore presented in the following. The general idea is to find an algebraic relationship between the original image $f(x, y)$ and the rotated image $g(x, y)$ in terms of Chebyshev Moments only. Given a vector $\mathbf{x} = (x, y)^T$ a linear transformation $T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ can be expressed in matrix notation as $\mathbf{x}' = \mathbf{Ax}$ or

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

with $\det(\mathbf{A}) \neq 0$. The orthonormal Chebyshev Moments of the transformed image $g(x, y)$ are therefore

$$T_{pq}^{(g)} = |\det(\mathbf{A})| \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \tilde{t}_p(a_{11}x + a_{12}y) \tilde{t}_q(a_{21}x + a_{22}y) f(x, y). \quad (6.17)$$

Note that the absolute value of the determinant serves as a scale factor by which the area is multiplied under the applied linear transformation. With the help of equation (6.12) and the binomial theorem we obtain

$$\tilde{t}_p(a_{11}x + a_{12}y) = \sum_{m=0}^p \sum_{s=0}^m \binom{m}{s} c_{pm} a_{11}^s a_{12}^{m-s} x^s y^{m-s}$$

and

$$\tilde{t}_q(a_{21}x + a_{22}y) = \sum_{n=0}^q \sum_{t=0}^n \binom{n}{t} c_{qn} a_{21}^t a_{22}^{n-t} x^t y^{n-t}.$$

Substituting the resulting polynomials of the transformed image back in (6.17) yields

$$\begin{aligned} T_{pq}^{(g)} = & |\det(\mathbf{A})| \sum_{m=0}^p \sum_{n=0}^q \sum_{s=0}^m \sum_{t=0}^n \sum_{i=0}^{s+t} \sum_{j=0}^{m+n-s-t} \binom{m}{s} \binom{n}{t} \\ & \cdot c_{pm} c_{qn} a_{11}^s a_{12}^{m-s} a_{21}^t a_{22}^{n-t} x^{s+t} y^{m+n-s-t} f(x, y). \end{aligned}$$

Using (6.13) to get rid of the powers

$$x^{s+t} = \sum_{i=0}^{s+t} d_{s+t,i} \tilde{t}_i(x)$$

and

$$y^{m+n-s-t} = \sum_{j=0}^{m+n-s-t} d_{m+n-s-t,j} \tilde{t}_j(y)$$

results in

$$T_{pq}^{(g)} = |\det(\mathbf{A})| \sum_{m=0}^p \sum_{n=0}^q \sum_{s=0}^m \sum_{t=0}^n \sum_{i=0}^{s+t} \sum_{j=0}^{m+n-s-t} \binom{m}{s} \binom{n}{t} \\ \cdot a_{11}^s a_{12}^{m-s} a_{21}^t a_{22}^{n-t} c_{pm} c_{qn} d_{s+t,i} d_{m+n-s-t,j} T_{ij}^{(f)}.$$

Every rotation of the axes about the origin is a linear transformation. Without loss of generality, replacing matrix \mathbf{A} in $\mathbf{x}' = \mathbf{Ax}$ with the rotation matrix

$$\mathbf{R} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$$

rotates the vector \mathbf{x} by an angle θ clockwise about the origin. Since the image's area is preserved, the determinant of a rotation matrix is always ± 1 depending on the direction of rotation. Hence $|\det(\mathbf{R})|$ can be omitted when applying the transformation $g(x, y) = f(x \cos \theta + y \sin \theta, -x \sin \theta + y \cos \theta)$. By evaluating the second order moments $T_{11}^{(g)}$, $T_{20}^{(g)}$ and $T_{02}^{(g)}$ one can find the relation

$$\tan 2\theta_g = \frac{u T_{11}^{(g)} - v T_{00}^{(g)}}{T_{20}^{(g)} - T_{02}^{(g)}}$$

where

$$u = \frac{2c_{22}c_{00}}{(c_{11})^2} \quad \text{and} \quad v = \frac{2c_{22}(c_{10})^2}{c_{00}(c_{11})^2}.$$

Assuming $\theta = \theta_f - \theta_g$, the rotation invariant Chebyshev Moments can be derived as follows using trigonometric identities:

$$T''_{pq} = \sum_{m=0}^p \sum_{n=0}^q \sum_{s=0}^m \sum_{t=0}^n \sum_{i=0}^{s+t} \sum_{j=0}^{m+n-s-t} \binom{m}{s} \binom{n}{t} (-1)^t \\ \cdot (\cos \theta)^{n+s-t} (\sin \theta)^{m+t-s} c_{pm} c_{qn} d_{s+t,i} d_{m+n-s-t,j} T_{ij} \quad (6.18)$$

where

$$\theta = \frac{1}{2} \arctan \left(\frac{u T_{11} - v T_{00}}{T_{20} - T_{02}} \right).$$

It is clear that moments of the original image can be substituted in (6.18) by the Central Chebyshev Moments of (6.16), i.e. $T_{ij} := T'_{ij}$. In this way, rotation and translation invariant Chebyshev Moments are obtained.

6.6.7 Experimental results

From (6.18) follows, that original moments of order $2p$ (or $2q$) are needed to compute the rotation invariant moments of order p (or q). The highest possible invariant moment order of a padded tactile sensor matrix of size 14×14 is therefore $p', q' = 6$ since the original moment orders p, q are limited to $(N - 1) = 13$, i.e.

$$p'_{max} = 2p_{max} \leq N - 1.$$

The existence of a reverse transformation follows from the well-known theorem on orthogonal functions. Combining equation (6.2) and (6.5) yields:

$$\tilde{f}(x, y) = \sum_{p=0}^{p_{max}} \sum_{q=0}^{q_{max}} T_{pq} \tilde{t}_p(x) \tilde{t}_q(y).$$

Figure 6.10 gives an overall impression of the feature descriptor's quality while figure 6.11 illustrates the effects of increasing moment orders. Finally image 6.12 shows that the computed moments are indeed invariant with respect to translation and rotation.

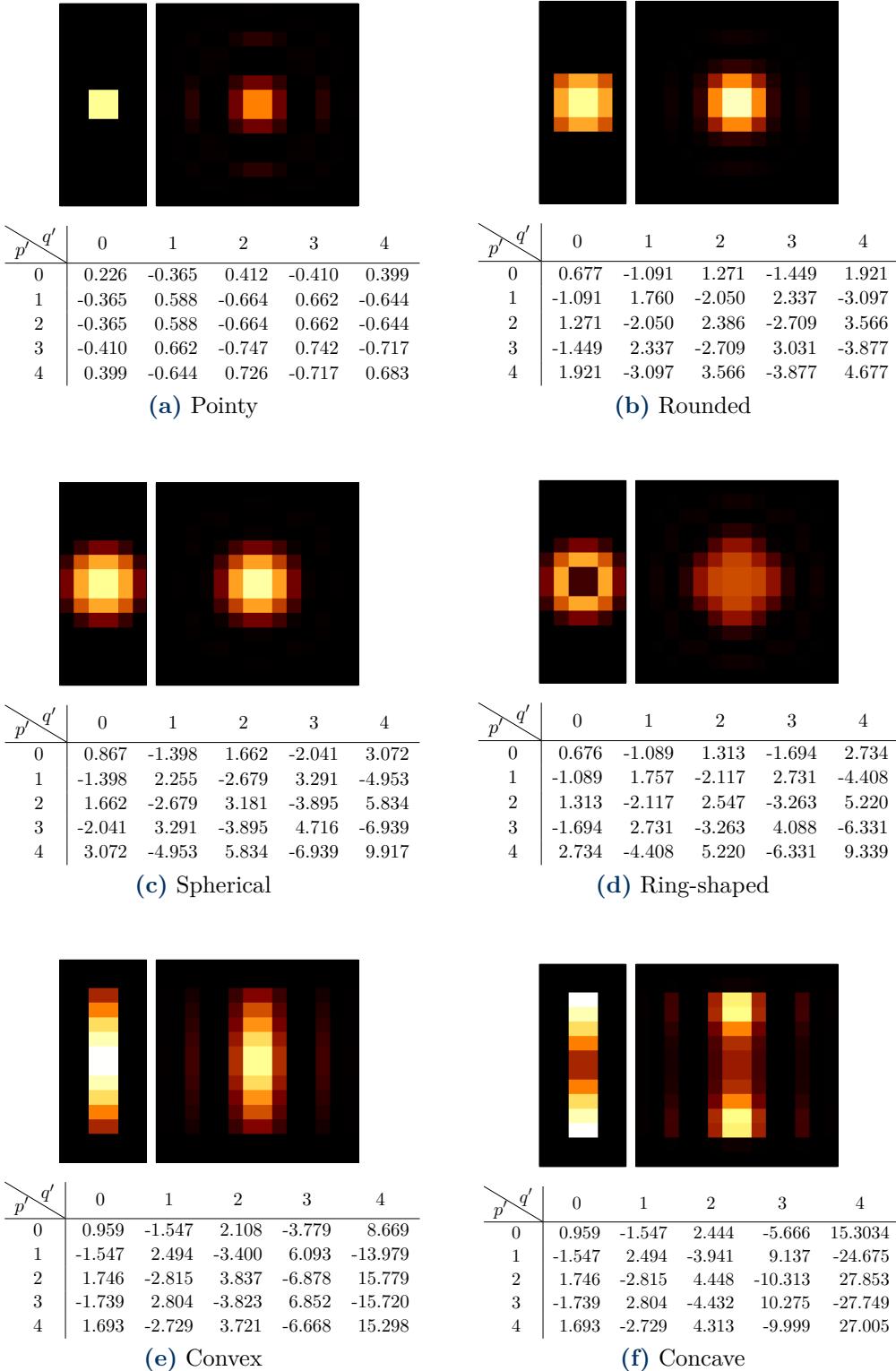


Figure 6.10: Reconstruction from Chebyshev moments of order $p', q' = 4$ i.e. $p, q = 8$: The left image of each pair represents the original sensor matrix (synthetic data) while the right frame is the reconstruction. Note the ringing artifacts at sharp edges due to Gibbs phenomenon.

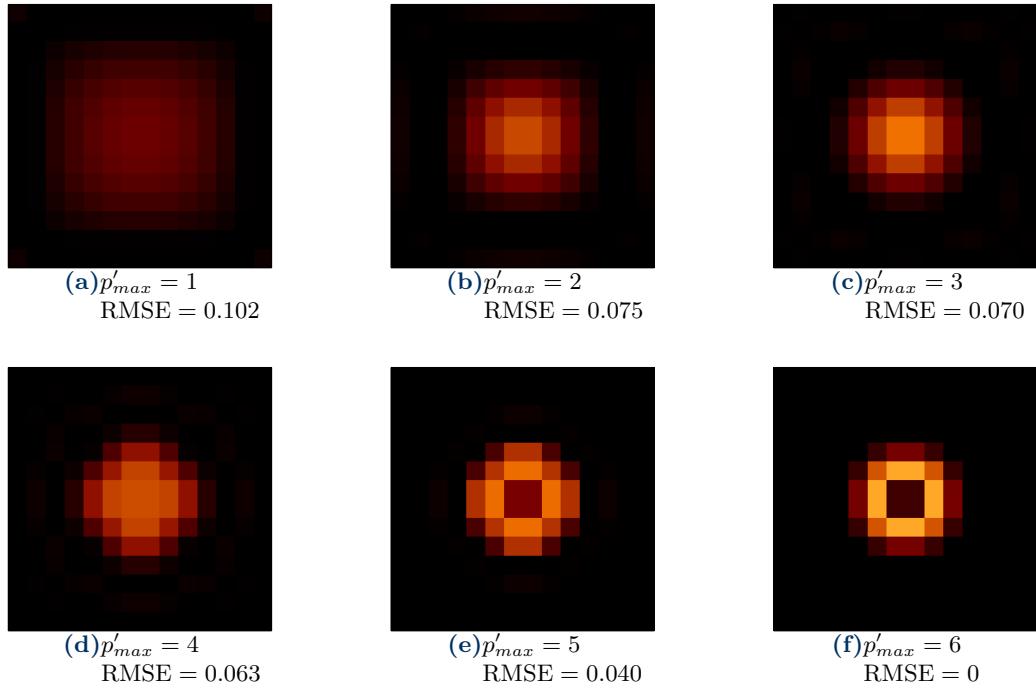


Figure 6.11: Synthetic ring-shaped pressure profile reconstructed from rotational invariants of different moment orders. The last image is a perfect reconstruction.

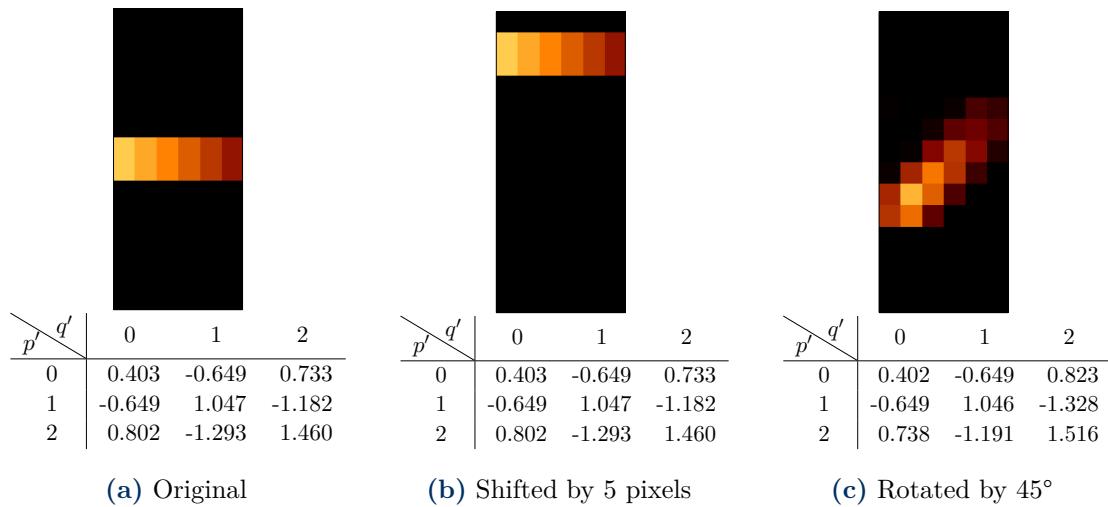


Figure 6.12: Comparison of resulting translational and rotational moment invariants. The invariants of the shifted image are virtually the same. The rotated version was generated with bilinear interpolation. Considering the large discretization error, the invariants are still somewhat close to the original ones.

6.7 Dataset - Acquisition and Preprocessing

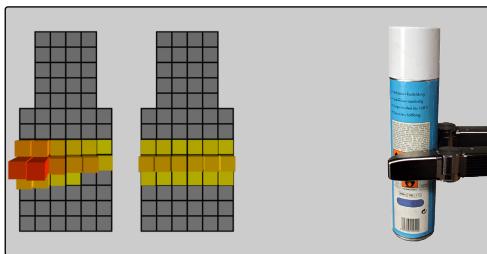
All experiments were performed under constant sensitivity settings of 0.5 and a threshold of 100. The grasps were performed with an angular velocity of 10 degrees per second using the reactive grasping scheme and a limit of 1500. That is, during the execution of the predefined grasp, the maximal raw sensor value is continuously compared to the given limit. The closing motion is then halted as soon as the limit is exceeded. In this way, the influence of the applied absolute pressures is minimized. To clarify, that does not mean that the maximum sensor value is always ~ 1500 . At a constant velocity of $10^\circ/\text{sec}$ in joint space, the centers of both matrices approach each other at rates between ~ 10 and $\sim 30 \text{ mm/s}$ depending on the current distance. At a tactile sensor frame rate of 30 fps, the fingertips easily travel for another millimeter until the movement comes to a rest. Consequently, the resulting maximal values might be well above the limit depending on the contact area.

As an alternative, the obtained pressure profiles could also be scaled to a common range, say $[0, 1]$ based on the corresponding maximum values. But this approach is less favorable due to the sensitivity threshold and saturation as well as the effects of deformable objects.

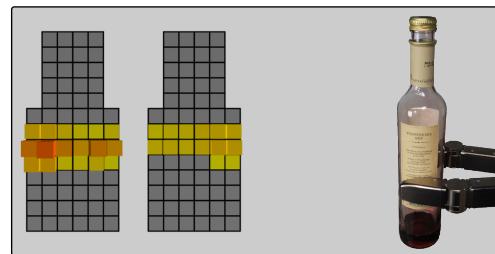
The 12 bit encoding of the tactile sensor suggests a dynamic range of 4096 different pressure readings. But it is clear from all available experiments, that the maximal sensor value for matrix 1 is 3554 but only 2493 for matrix 5. Nonetheless, each sensor matrix is scaled to $[0, 1]$ using the theoretical maximum value of 4096 before computing the Chebyshev moments. These unique intensity properties add to the distinctiveness of the features since the classification scheme is not invariant with respect to the spatial arrangement of the sensor matrices.

Figure 6.13 illustrates the 36 acquired grasp-object configurations. Each class consists of several grasps with varying position and orientation and the number of instances, between 14 and 36, is listed in parentheses. The resulting pressure profiles were manually inspected and separated into small chunks containing only a single grasp to ensure compliant features. But in addition, there is a routine to automatically detect the beginning and end frames of individual grasps within a series of measurements. A drawback of that step-detection approach is that it relies on several “perfect grasps” in a row, a requirement that cannot be met in practice. A more advanced capturing routine would offer a real-time quality assignment or undo option.

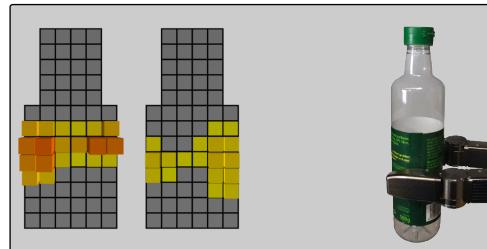
The depicted scatter plot matrix in figure 6.14 gives an impression of the first four features. Plotted data is standardized as discussed in the next section on feature scaling. Axis limits are uniform. Further information on the extracted feature vectors can be found in the appendix of this thesis.



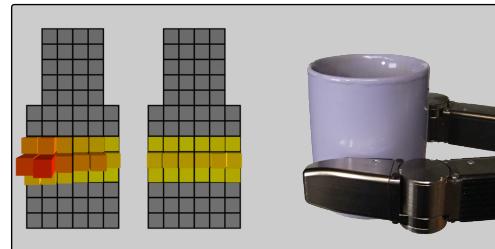
Aerosol can (23)



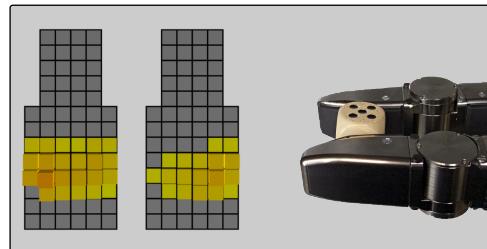
Bottle glass (17)



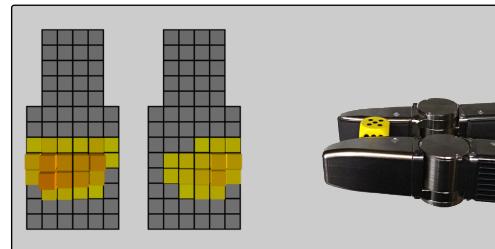
Bottle plastic vinegar (15)



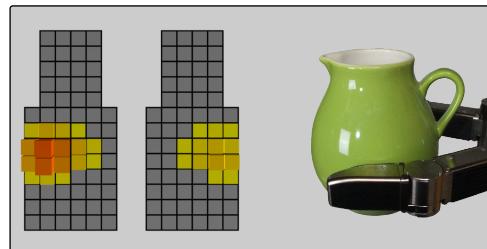
Coffee mug violet (35)



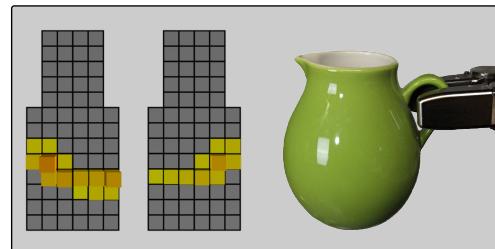
Dice large (30)



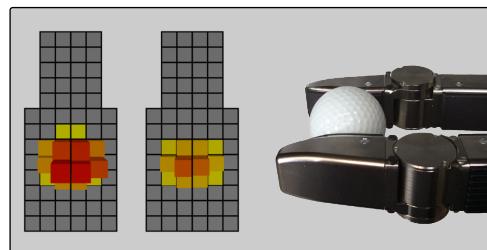
Dice small (22)



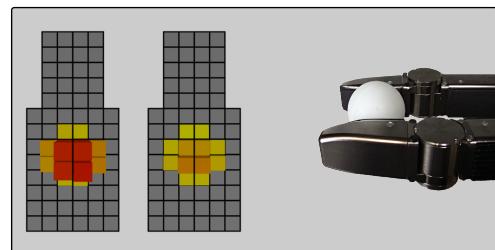
Flower vase body (22)



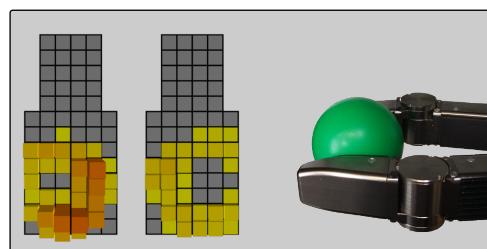
Flower vase handle (25)



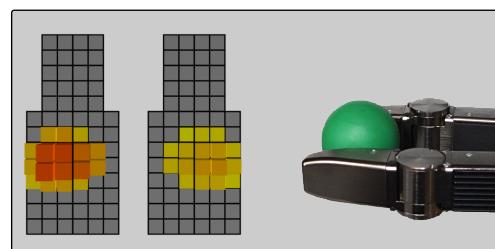
Golf ball (29)



Ping-pong ball (20)

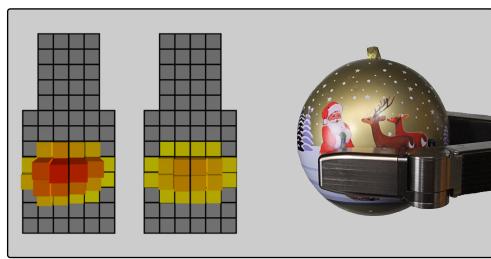


Green plastic ball large (19)

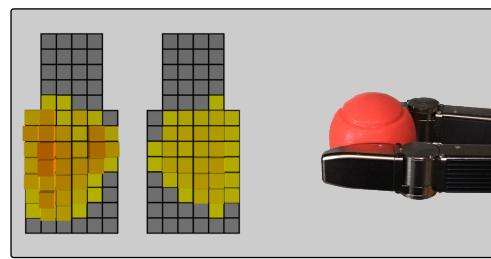


Green plastic ball small (26)

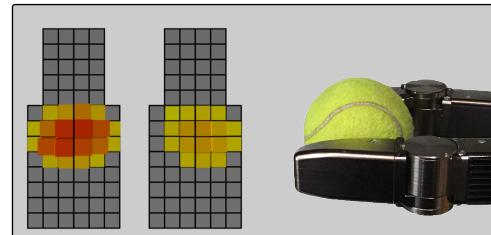
Figure 6.13: Learned grasp-object configurations



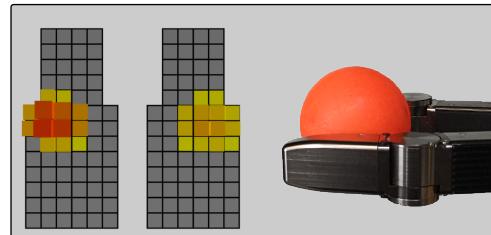
Christmas ball (20)



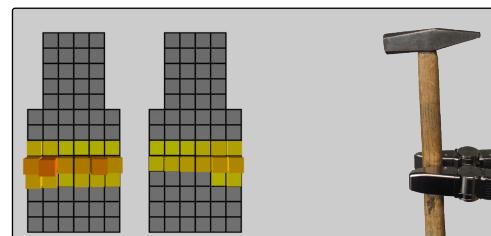
Foam ball (18)



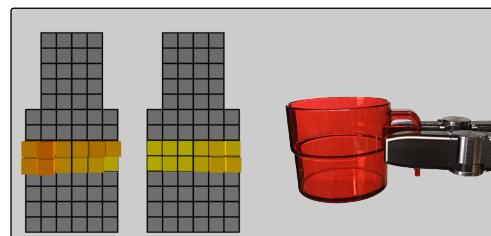
Tennis ball (28)



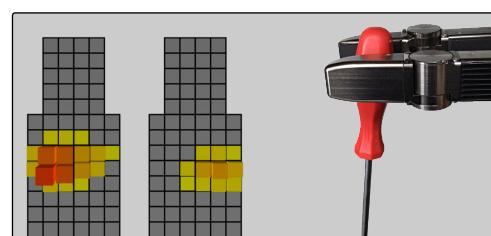
Hockey ball (14)



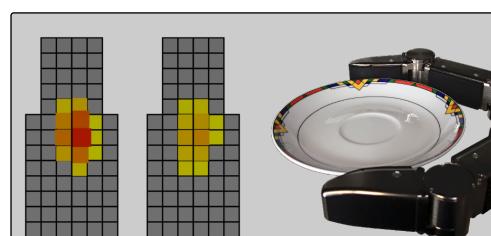
Hammer (21)



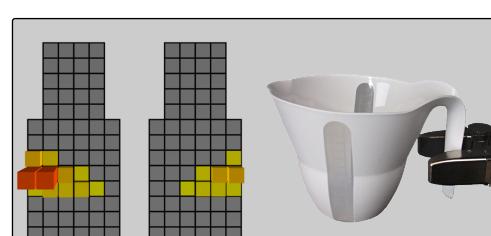
Plastic cup handle (24)



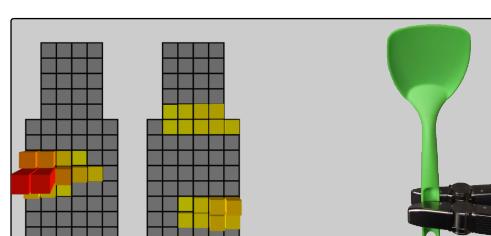
Screwdriver (21)



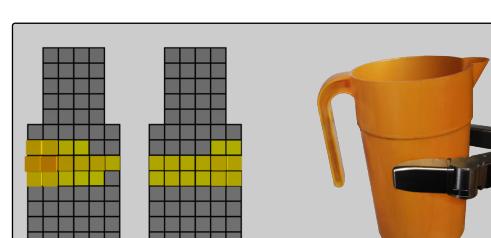
Saucer (28)



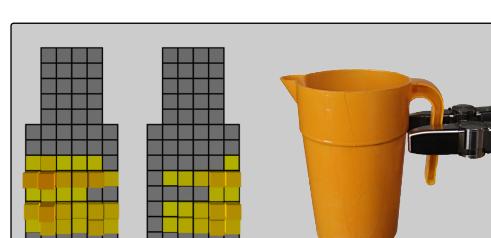
Jug white handle (22)



Spatula (L) (29)

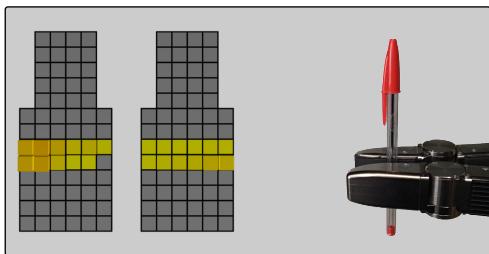


Jug orange body (22)

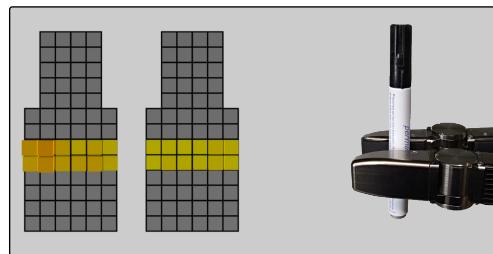


Jug orange handle (24)

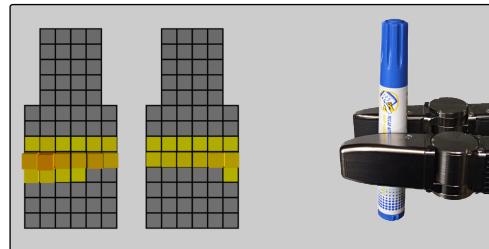
Figure 6.13: Learned grasp-object configurations (continued)



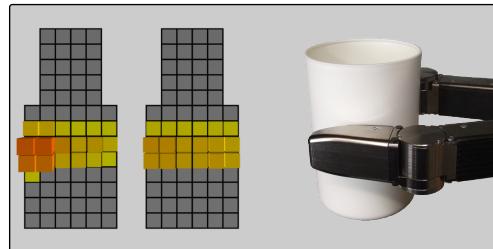
Pen thin (36)



Pen medium (20)



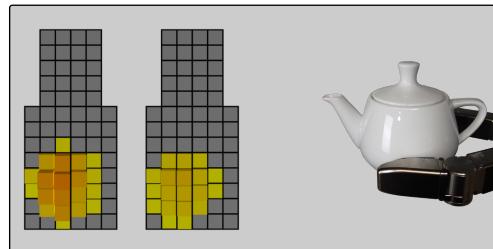
Pen thick (21)



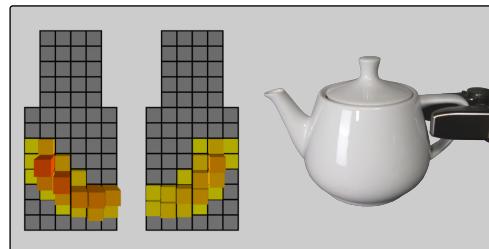
Plastic cup white (21)



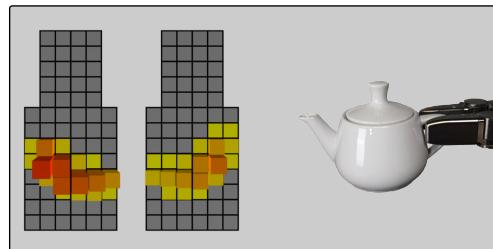
Teapot large body (25)



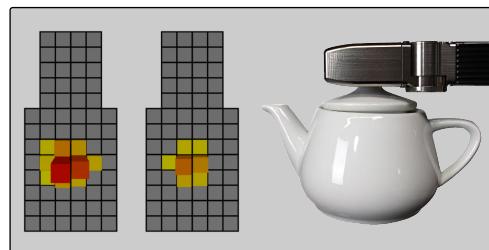
Teapot small body (28)



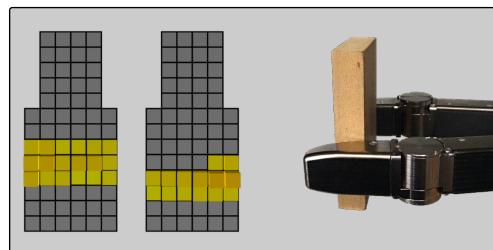
Teapot large handle (19)



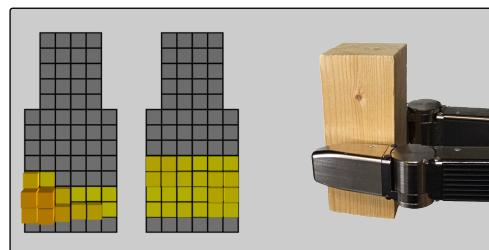
Teapot small handle (26)



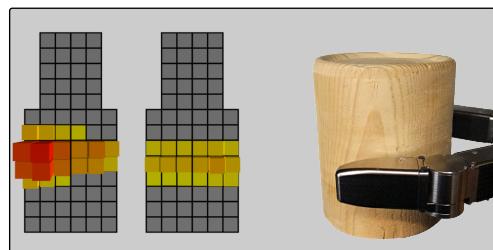
Teapot top (28)



Wooden block flat (29)



Wooden block small (23)



Wooden cylinder (19)

Figure 6.13: Learned grasp-object configurations (continued)

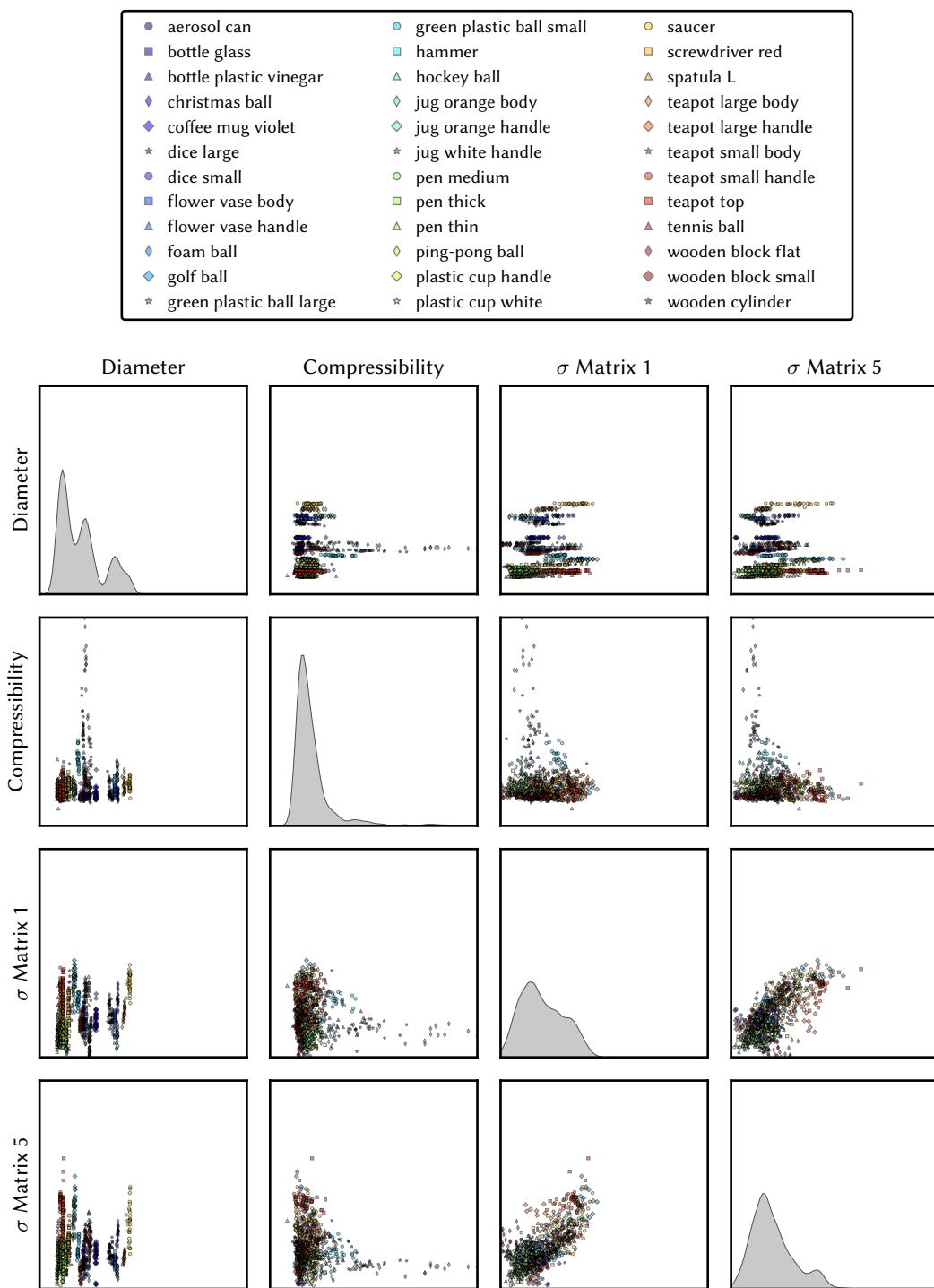


Figure 6.14: Scatterplot matrix of the first four standardized features. The object diameter is obviously a discriminatory feature with very little variation. The behavior of the examined compressible objects under deformation is complex. The plastic bottle for example is more flexible in the middle section than near the strengthened top or bottom. The applied force and thus the deformation also depends on the relative position of the object between the fingers due to the available torques and leverages. The resulting compressibility measures therefore show high variance. Note that the standard deviations of the intensity values are highly correlated.

6.8 Classification

As a result of the colloquial *no-free-lunch theorem* by Wolpert and Macready (1997), there is no classifier that satisfies all requirements, meaning that “for any algorithm, any elevated performance over one class of problems is offset by performance over another class”. It is therefore common practice to try several different methods on a given dataset. With the benefit of hindsight, one may then gain further insight from the performance of certain algorithms. Scikit-learn by Pedregosa et al. (2011) offers an easy to use environment to carry out various machine learning experiments. In particular, the provided SVM classifier is a Python wrapper for LIBSVM, Chang and Lin (2011). Additional tools like a generic grid search pipeline for hyper-parameter tuning or different cross-validation strategies are available. The thereby obtained results motivate the following sections.

6.8.1 Feature scaling

The range of values of individual features in the feature vector varies a lot. The diameter of objects for example varies between 7 and 140 mm, while compressibility is in the range of 0 – 11 mm. In order to avoid the domination of larger values in the classifier’s distance function, it is essential to scale the features to a common numeric range. A simple min-max scaling to the range [0, 1] resolves the problem. But in addition, different feature might be of varying importance for the classification task. This problem can be solved by normalizing the features according to the underlying distribution. Assuming normally distributed features, each dimension is scaled to zero mean and unit variance. Formally, the performed normalization or *standardization* is given as

$$\mathbf{x}' = \frac{\mathbf{x} - \boldsymbol{\mu}}{\sigma}$$

where $\boldsymbol{\mu}$ and σ refer to the element-wise mean and standard deviation of all available feature vectors. Both normalization constants are stored together with the fully trained classifier for the scaling of future data. There are other possible prescaling methods such as binning the feature vectors in case of sparse data or scaling them to unit length. But nonexhaustive experiments confirmed that the described procedure works best for the available data.

There also are numerical reasons for feature scaling in the context of SVMs which will become clearer in the next section. For one, it is well known that algorithms that depend on inner products tend to be instable in the case of large differences in the magnitudes of vector components. In addition, the number of iterations to approximately solve the Quadratic Programming (QP) optimization problem depends on the speed of convergence towards the Karush-Kuhn-Tucker conditions. And Sequential Minimal Optimization (Platt (1998)), the method implemented in LIBSVM to solve the QP problem, is sensitive to feature scaling.

6.8.2 Support Vector Machines (SVM)

Once all features of the recorded training set have been computed, it is time to classify the data in order to be able to make predictions for future, unlabeled input. While there are many classification techniques available, from the k -nearest neighbors algorithm to decision trees, logistic regression or neural networks, the decision was made to apply a support vector machine (SVM). SVMs are one of the most frequently used supervised classification techniques in machine learning. Their invention dates back to Vapnik and Lerner (1963) and Vapnik and Chervonenkis (1964) and has since lead to a theory of statistical learning, sometimes called Vapnik-Chervonenkis theory, see Vapnik (1999) for details. The popularity of the method grew with the extension to

nonlinear classification by Boser et al. (1992) using the then already known *kernel method*. The introduction of soft margins, suitable for non-separable data, by Cortes and Vapnik (1995) forms the basis of modern implementations.

For a better understanding, a brief description of the method based on Schölkopf et al. (1999) and Bishop (2006) is given in the following. Consider a supervised two-class learning problem where a set of feature vectors belonging to known categories should allow the prediction of class memberships for new observations. Given a set of N labeled training data (\mathbf{x}_n, y_n) where $y_n \in \{\pm 1\}$ is the label associated with $\mathbf{x}_n \in \mathbb{R}^D$, a *linear discriminant function* that predicts class labels is given as

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

where \mathbf{w} is a weight vector excluding w_0 for technical reasons. The set of points satisfying $f(\mathbf{x}) = 0$ form the decision boundary which divides the D -dimensional space into the two corresponding classes according to the sign of $f(\mathbf{x})$. In the linear case, the decision boundary is a $(D - 1)$ -dimensional hyperplane and bias b , formerly w_0 , determines its distance from the origin with $b/\|\mathbf{w}\|$. Projecting an arbitrary point orthogonally onto the decision boundary yields the distance relation $|f(\mathbf{x})|/\|\mathbf{w}\|$.

The question remains how to find the “best” hyperplane for the given training data in terms of generalization error on unseen data. Assuming separability, support vector machines aim to maximize the margins, that is the perpendicular distances from the decision boundary to the closest data points on both sides. For a correctly classified data point \mathbf{x}_n belonging to class y_n this distance to the hyperplane becomes

$$\frac{y_n \cdot f(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{y_n (\mathbf{w}^T \mathbf{x}_n + b)}{\|\mathbf{w}\|}.$$

The maximum margin solution with respect to \mathbf{w} and b is thus given as

$$\arg \max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n [y_n (\mathbf{w}^T \mathbf{x}_n + b)] \right\}.$$

By rescaling \mathbf{w} and b such that the closest point has unit distance, one obtains the so called canonical representation of the decision hyperplane, which leads to the simplified optimization problem

$$\begin{aligned} & \underset{\mathbf{w}, b}{\text{maximize}} \quad \frac{1}{\|\mathbf{w}\|} \\ & \text{subject to} \quad y_n (\mathbf{w}^T \mathbf{x}_n + b) \geq 1, \quad n = 1, \dots, N. \end{aligned} \tag{6.19}$$

which is equivalent to

$$\begin{aligned} & \underset{\mathbf{w}, b}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{subject to} \quad y_n (\mathbf{w}^T \mathbf{x}_n + b) \geq 1, \quad n = 1, \dots, N. \end{aligned}$$

Note that the initial maximization involves square roots in the norm of \mathbf{w} . This motivates the reformulation which is possible since $x \mapsto x^2$ is strictly increasing on $[0, \infty)$. The factor $1/2$ is arbitrary, but mirrors the fact that the combined margin equals $2/\|\mathbf{w}\|$ in the canonical representation.

The constrained optimization now turned into a quadratic programming problem for which various solution methods exist. But in practice, the data is often not separable. And even if a decision boundary can be found, it could be beneficial to allow a few misclassifications in

exchange for a more robust classifier. This observation leads to the introduction of soft-margins via a slack-variable $\xi_n \geq 0$ for each data point. The idea is that correctly classified points satisfy $\xi_n = 0$ while points for which $\xi_n > 1$ are misclassified. Points that lie on the correct side of the decision boundary but inside the margin satisfy $0 < \xi_n \leq 1$ and are considered to be margin errors. See figure 6.18b on page 103 for a better understanding of the working principles. The reformulated soft-margin optimization problem now maximizes the margin while penalizing margin errors and misclassifications.

$$\begin{aligned} \underset{\mathbf{w}, b}{\text{minimize}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n \\ \text{subject to} \quad & y_n (\mathbf{w}^T \mathbf{x}_n + b) \geq 1 - \xi_n, \quad \xi_n \geq 0, \quad n = 1, \dots, N. \end{aligned} \quad (6.20)$$

where $C > 0$ is a parameter that controls the trade-off between the size of the margin and the amount of slack. For small values of C , the optimization will favor larger margins even if points are misclassified while larger values lead to the minimization of training errors at the expense of generalization. For $C \rightarrow \infty$ the soft-margin SVM is equivalent to the original formulation. In this respect, the soft-margin parameter is related to regularization and controls the complexity or capacity of the model (hence the letter C). The slack variable can also be expressed in terms of a loss function. Reformulating the constraint yields $y_n f(\mathbf{x}_n) \geq 1 - \xi_n$. It follows that

$$\xi_n = \max(0, 1 - y_n f(\mathbf{x}_n))$$

The expression is known as hinge loss due to its shape which results from the linear penalty term depending on the distance of a data point from the one-sided margin and the sudden change to 0 for correctly classified points. The unconstrained optimization problem is thus given as

$$\underset{\mathbf{w}, b}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \max(0, 1 - y_n f(\mathbf{x}_n)).$$

Note that the combined cost function is convex since both $\|\mathbf{w}\|^2$ and the hinge loss are convex functions. This important property can further be exploited. As mentioned before, the quadratic programming problem can be solved using standard techniques. But in particular, it is beneficial to apply the well known Lagrange multipliers to obtain the dual representation of the optimization problem:

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n - \sum_{n=1}^N \alpha_n \left\{ y_n (\mathbf{w}^T \mathbf{x}_n + b) - 1 + \xi_n \right\} - \sum_{n=1}^N \beta_n \xi_n$$

with $\alpha_n, \beta_n \geq 0$ corresponding to the constraints in the primal formulation (6.20). The Lagrangian L must now be minimized with respect to primal variables \mathbf{w} and b and maximized with respect to the dual variables $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. This is equivalent to finding the saddle point of L . The partial derivatives therefore satisfy the constraints

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{0}, \quad \frac{\partial L}{\partial b} = 0 \quad \text{and} \quad \frac{\partial L}{\partial \boldsymbol{\xi}} = \mathbf{0}$$

which leads to

$$\mathbf{w} = \sum_{n=1}^N \alpha_n y_n \mathbf{x}_n \quad (6.21)$$

$$\sum_{n=1}^N \alpha_n y_n = 0 \quad (6.22)$$

$$\alpha_n = C - \beta_n \quad (6.23)$$

After eliminating the primal variables by substituting (6.21), (6.22) and (6.23) back in L one obtains the dual form

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \sum_{n=1}^N \alpha_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N \alpha_n \alpha_m y_n y_m (\mathbf{x}_n^T \mathbf{x}_m) \\ \text{subject to} \quad & \sum_{n=1}^N \alpha_n y_n = 0 \quad \text{and} \quad 0 \leq \alpha_n < C. \end{aligned} \quad (6.24)$$

The dual formulation is not only easier to solve due to the relaxed constraints but also has some interesting properties. For one, its solution is equivalent to the solution of the primal problem since strong duality holds for convex primal problems. In addition, it follows from (6.21) that only data points for which $\alpha_n > 0$ contribute to the weight vector \mathbf{w} . These points are called *support vectors* and are usually sparse. In contrast to dense methods like a k NN-classifier, only a subset of the training data is needed in order to make predictions. Note that in the case of hard-margin SVMs all support vectors lie on the decision boundary while the support vectors of soft-margin SVMs additionally include data points within the margin as well as those on the wrong side of boundary. And from (6.23) it follows that the influence of outliers on the weight vector \mathbf{w} is bounded by C . Finally, the decision function is derived from the set of margin support vectors S satisfying $0 < \alpha_n < C$ and $\xi_n = 0$ as follows:

$$f(\mathbf{x}) = \text{sgn} \left(\sum_{m \in S} \alpha_m y_m (\mathbf{x}_m^T \mathbf{x}) + b \right)$$

where

$$b = y_m - \sum_{m \in S} \alpha_m y_m (\mathbf{x}_m^T \mathbf{x}_m).$$

In practice, b is computed as the average of several margin support vectors.

But the main reason for the popularity of SVMs is their ability to efficiently classify nonlinear data. The naive approach to nonlinear classification using a linear classifier is the transformation of the original data via a nonlinear function $\phi : \mathbb{R}^D \rightarrow \mathbb{R}^{D'}$ such that the data becomes linearly separable. In other words, the linear discriminant function is modified as follows:

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b.$$

But the explicit mapping of the data to a (high-dimensional) feature space does not scale well for large N due to the increased complexity in D' . A popular solution to this problem is the use of kernels, dating back to Hilbert (1904) and Mercer (1909) and introduced to machine learning by Aizerman et al. (1964).

A kernel function $k : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$ is defined as

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$$

which computes the inner product of its arguments after a transformation ϕ .

The so called *kernel trick* is based on the observation that it is possible for valid kernels to compute the dot-product without explicitly evaluating the feature space mapping ϕ . For a proof, see for example Schölkopf et al. (1999). This surprising result allows the implicit use of high dimensional and even infinite feature spaces. Popular kernels include the polynomial kernel

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T, \mathbf{x}'^T + 1)^D$$

which maps to a space spanned by all monomials up to degree D and the Gaussian radial basis function (RBF) kernel defined by

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$

where normalization constant $1/2\sigma^2$ is often replaced by a parameter γ since probability densities are meaningless in this context. Note that the RBF-kernel corresponds to a dot-product in an infinite dimensional feature space. Kernel functions arise naturally from the dual representation, but they can also be applied to the primal as mentioned in Chapelle (2007). But as it turned out, the available dataset is pretty much linearly separable without nonlinear transformations.

So far, only binary classification tasks have been discussed. The extension to multiclass SVMs follows a one-vs-one approach first introduced by Knerr et al. (1990) in the context of neural networks. The basic idea is to construct a SVM for each pairwise comparison of the N classes. This amounts to $N(N-1)/2$ different binary classifiers which can be seen by expanding the comparisons: 0 vs 1, 0 vs 2, ..., 0 vs N , 1 vs 2, 1 vs 3, ..., 1 vs N , ..., $N-1$ vs N . The final label of a data point is then assigned to the class which received the most votes. In case of a tie, the confidence levels, that is the distance from the particular decision boundary is taken into account.

6.8.3 Feature selection

As mentioned before, the final feature vector contains 54 elements which can already be regarded as “high dimensional”. It is a well known fact in machine learning, that working with a large number of features comes with certain disadvantages.

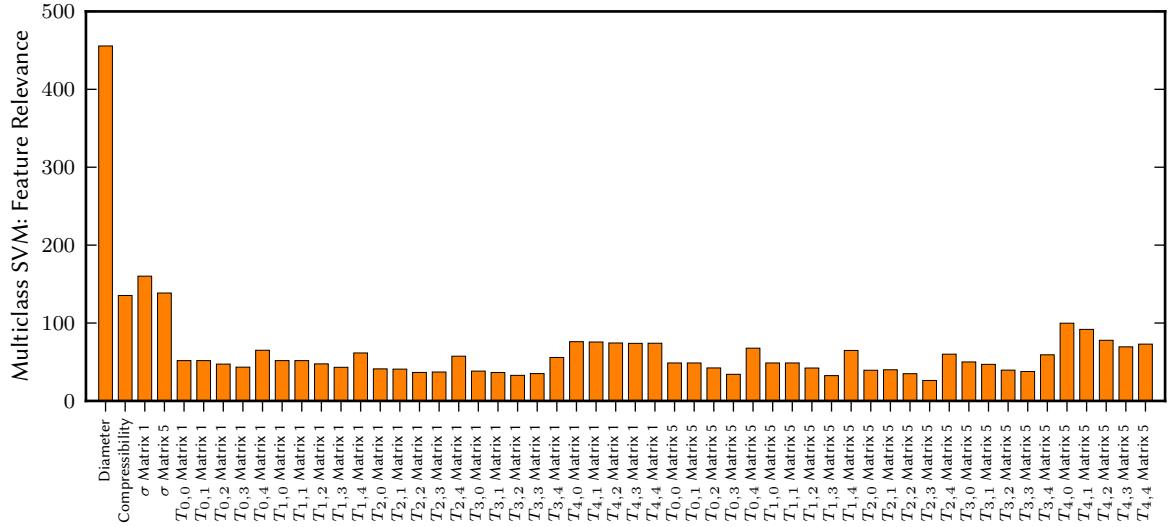


Figure 6.15: Relevance of individual features: The absolute values of the weight vectors of all binary SVM classifiers are taken into account. The assigned weights are therefore computed according to $\sum_{i=1}^{630} |\mathbf{w}_i|$ since there are $N(N-1)/2 = 630$ different hyperplanes for $N = 36$ classes. Note the disproportionate relevance of the diameter feature as well as the symmetry with respect to the sensor matrices. In addition, higher-order Chebyshev moments seem to be superior to lower-order ones.

Apart from leading to an increased processing time, irrelevant and redundant features increase the variance and thus lead to overfitting. This is especially true for a limited number of training samples. A smaller set of features also helps to avoid the *curse of dimensionality* since distance measures used for classification are more meaningful in densely populated spaces. A lower dimensional feature space is also beneficial for a better understanding of the structure of a given dataset. Figure 6.15 gives an impression of the relative importance of individual features based on the multiclass SVMs weight vectors obtained using the full training set. The criterion utilizes the absolute value of the hyperplane's coefficients in order to determine the features with best separation.

The simplest but also most inefficient method to find the best combination of features is an exhaustive search over all possible subsets. Since this procedure is computationally infeasible, various different methods have been proposed as outlined in Guyon (2003). They range from the removal of features with low variance to statistical correlation methods, randomized hill climbing or the sequential selection based on an information gain criterion. In general, a distinction is drawn between filter methods that select subsets of features based on independent knowledge and wrapper methods that rank features according to their predictive power with the help of the classifier itself. Embedded methods try to combine the best of both worlds and learn the selection of features using the regularization mechanism of the classifier.

Guyon et al. (2002) and Rakotomamonjy (2003) investigated several methods specially tailored to SVMs, among others Recursive Feature Elimination (RFE) based on $\|\mathbf{w}\|^2$. RFE makes use of the feature weights assigned by the classifier and recursively prunes the lowest ranking features from the current set until the desired number of features is reached. The results of such an approach can be seen in figure 6.15.

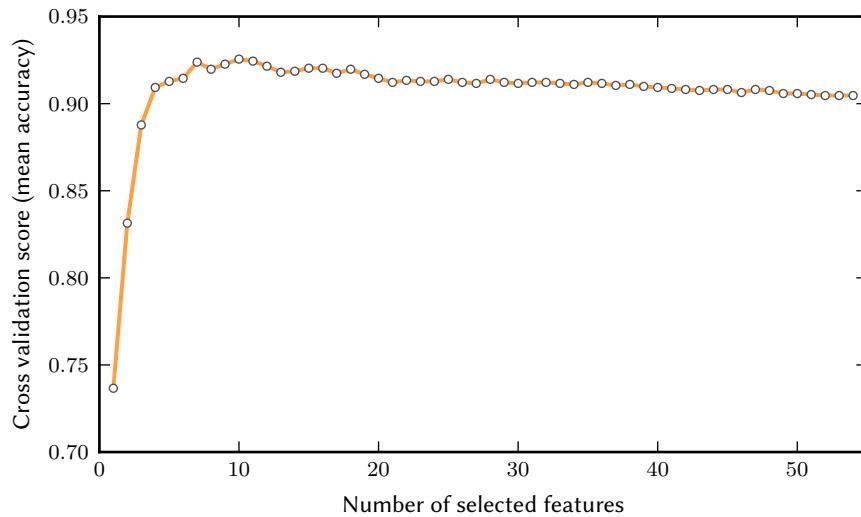


Figure 6.16: Recursive feature elimination with cross-validation accuracy. The most important feature alone, the diameter, already reaches good accuracy and the score keeps increasing for about 10 extra features. But the cross-validation accuracy (see subsection 6.8.5) decreases for every additional feature that is taken into account.

As an alternative to feature selection methods, dimensionality reduction based on data projection is discussed in the next subsection.

6.8.4 Linear Discriminant Analysis (LDA)

The investigated feature selection and elimination methods do not work well on the given dataset. The obvious conclusion to that observation is that some of the 54 features are partially redundant while none are completely irrelevant. But a dimensionality reduction is still desirable to avoid overfitting and to reduce training time. Although many linear and nonlinear dimensionality reduction techniques exist, the classic and relatively simple Linear Discriminant Analysis (LDA) was found to provide good results.

Put simply, LDA is a classification and dimensionality reduction technique with linear decision boundaries. There are two equivalent views on LDA: From a Bayesian perspective, class-conditional densities are modeled under the assumption that the data is normally distributed with identical class covariances. With the help of Bayes' theorem, the posterior probabilities can be obtained in order to predict class memberships. However, the original approach that maximizes the between-class separability while minimizing the within-class variability is computationally more feasible in practice. LDA is a generalization of the first algorithm for pattern recognition, called Fisher's linear discriminant, introduced by Fisher (1936).¹ The method was extended to deal with multiclass problems by Rao (1948) and several variants to solve the inherent eigenvalue problem exist. In addition, there are variants that use the kernel trick, e.g. Mika et al. (1999).

As illustrated in figure 6.17, LDA is related to PCA, but instead of projecting to the dimension of the greatest variance, LDA projects to a subspace that maximizes the separation of class means while minimizing the class overlap.

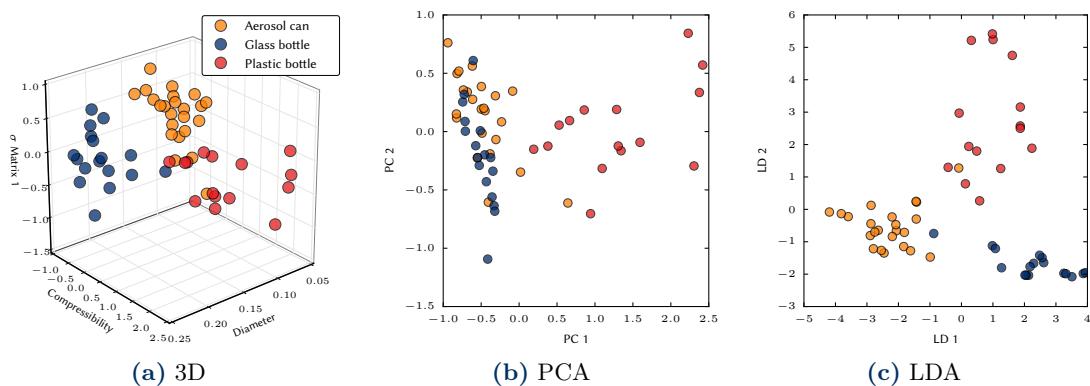


Figure 6.17: Illustration of a dimensionality reduction using the first three features of three very similar objects. PCA projects the data to the orthogonal coordinate system that maximizes the preserved variance, thus rendering the aerosol can and the glass bottle indistinguishable from each other. LDA, by contrast, takes the class labels into account and finds a projection that maximizes separability while minimizing the within-class variability.

The method is quite popular and Hastie et al. (2009) state that LDA “performs well on an amazingly large and diverse set of classification tasks” and attribute its performance to a beneficial bias-variance trade-off due to the simple linear model with high bias in combination with the Gaussian model with low variance.

A comprehensive review of the method can be found in Li et al. (2006), or again Bishop (2006). Given a set of D -dimensional training samples $\{\mathbf{x}_n\}$ where $n = 1, 2, \dots, N$ belonging to K different classes C_k , the *within-class* scatter matrix, a shared estimate of the covariance matrix

¹ The famous iris flower data set is part of this publication.

(up to scale) is defined as

$$\hat{\Sigma}_w = \sum_{k=1}^K \sum_{\mathbf{x} \in C_k} (\mathbf{x} - \bar{\mathbf{x}}_k)(\mathbf{x} - \bar{\mathbf{x}}_k)^T$$

where the sample mean of class C_k consisting of N_k data points is given by

$$\bar{\mathbf{x}}_k = \frac{1}{N_k} \sum_{\mathbf{x} \in C_k} \mathbf{x}.$$

That is, the total within-class scatter matrix is just the sum of the individual scatter matrices. The *between-class* scatter matrix is computed by

$$\hat{\Sigma}_b = \sum_{k=1}^K N_k (\bar{\mathbf{x}}_k - \bar{\mathbf{x}})(\bar{\mathbf{x}}_k - \bar{\mathbf{x}})^T$$

where the total mean vector $\bar{\mathbf{x}}$ is given by

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{k=1}^K N_k \bar{\mathbf{x}}_k.$$

Consider the one-dimensional projection of the D -dimensional feature vector \mathbf{x} similar to the linear discriminant function

$$y = \mathbf{w}^T \mathbf{x}.$$

For more than two classes ($K > 2$), this concept can be generalized with the help of a projection matrix \mathbf{W} of size $D \times D'$ which projects the D -dimensional feature vector \mathbf{x} onto a D' -dimensional subspace with $1 \leq D' \leq K - 1$. This results in

$$\mathbf{y} = \mathbf{W}^T \mathbf{x}$$

where the columns of \mathbf{W} consist of the one-dimensional weight vectors \mathbf{w}_k , $k = 1, \dots, D'$.

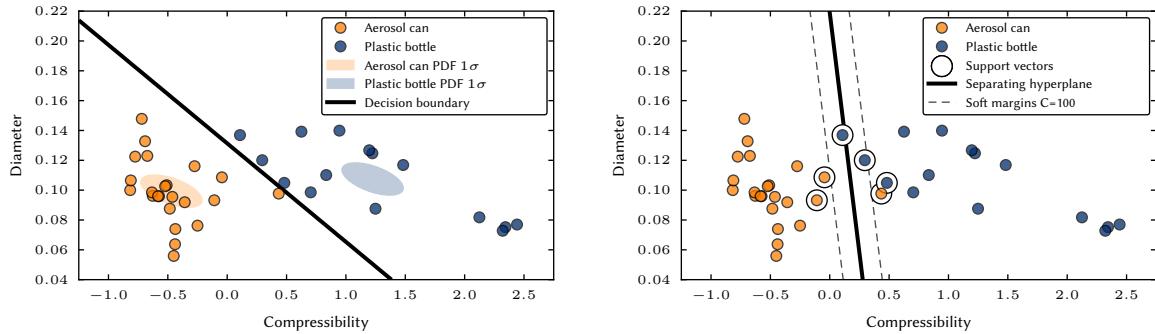
The LDA now tries to determine \mathbf{W} such that the within-class scatter $\hat{\Sigma}_w$ is minimized while the between-class scatter $\hat{\Sigma}_b$ is maximized. This is equivalent to maximizing the so called Fisher-criterion, i.e.

$$\arg \max_{\mathbf{W}} \frac{|\mathbf{W}^T \hat{\Sigma}_b \mathbf{W}|}{|\mathbf{W}^T \hat{\Sigma}_w \mathbf{W}|}$$

which is a generalized Rayleigh quotient. The solution can thus be found by solving the generalized eigenvalue problem

$$\hat{\Sigma}_w \mathbf{W} = \lambda \hat{\Sigma}_b \mathbf{W}.$$

Assuming $\hat{\Sigma}_w$ is invertible, \mathbf{W} can directly be determined by the eigenvectors of $\hat{\Sigma}_w^{-1} \hat{\Sigma}_b$. More precisely, the LDA solution \mathbf{W} consists of the D' eigenvectors corresponding to the largest eigenvalues in decreasing order. Although LDA can be used as a classifier as seen in figure 6.18a, its performs worse than SVMs on the dataset at hand. The final transformation matrix has to be stored alongside the support vectors for the dimensionality reduction of future observations before classification.



(a) Linear Discriminant Analysis: Assuming normally distributed classes which share the same covariance matrix, the separation of their means is maximized. Note that even though all points are correctly classified in this example, SVMs perform better overall on the given dataset. The method does not yield an explicit discriminant function and the decision boundary is thus found by evaluating the classifier on a dense grid.

(b) Soft margin SVM: The separating hyperplane is a linear combination of the support vectors only. Although two points are misclassified, the maximum margin classifier with slack performs generally better than a variety of other methods on the dataset at hand. In this example, a perfect separation could be achieved for large values of C , but at the same time the resulting generalization error on unseen data will increase.

Figure 6.18: Illustration of classification methods.

6.8.5 Model selection and performance measurement

Cross-validation is the method of choice to avoid overfitting and to estimate the performance of a classifier. By dividing the dataset into a training and test set, it is possible to control the generalization error of the multiclass SVM approach. Various possible variants to hold out parts of the available data exist. Forman and Scholz (2010) discusses the subtle differences between different cross-validation methods and scoring functions.

It is common practice for most applications to use a 5 or 10-fold cross-validation. Eventually, a stratified shuffle-split cross-validation is used throughout all experiments. The entire dataset is therefore randomly partitioned into training- and test sets with a ratio of 80/20 while maintaining the original percentage of samples for each class. The whole procedure is repeated 10 times with different shuffling and splitting operations. In other words, the method combines a 5-fold cross-validation with 10 randomized splits that try to preserve the original number of samples per class.

A common metric to compare the performance of classifiers is the average accuracy, i.e. the overall correctness of the model which is calculated as the sum of correct classifications divided by the total number of classifications. Table 6.1 gives an overview of the cross-validation performance of different subsets of features evaluated 10 times with random seeds.

Table 6.1: Overall cross-validation accuracy for different feature sets.

Features	Overall accuracy ($\pm \sigma$)
Diameter only	0.734 ± 0.003
First four features only	0.872 ± 0.008
Chebyshev moments only	0.711 ± 0.006
Combined features	0.903 ± 0.007
Combined features + LDA	0.976 ± 0.003

Regularization is another aspect where cross-validation plays an important role. Experiments revealed that nonlinear kernels do not improve the overall accuracy. The only remaining parameters to be tuned are therefore the soft margin parameter C and the number of LDA components. A simple but computationally expensive grid search was performed in order to determine the best combination. After initially computing the scores on a logarithmic grid, the most promising range of values was identified. The result of a second run on a finer linear grid can be seen in figure 6.19. Without applying the LDA, the regularization parameter C has a noticeable effect on SVM's the performance. But after transforming the data, only extremely small values of C affect the result. In addition, 14 LDA components seem to be a robust choice for the dimensionality reduction.

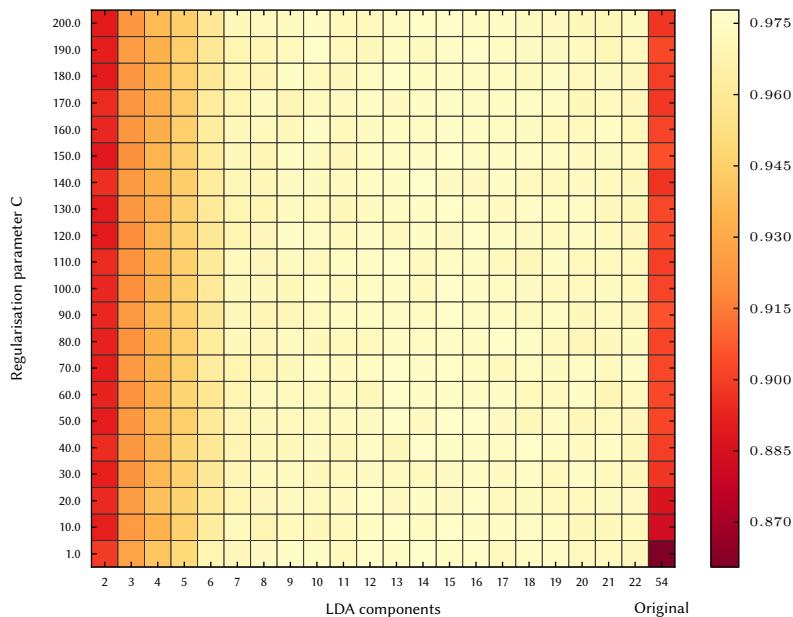
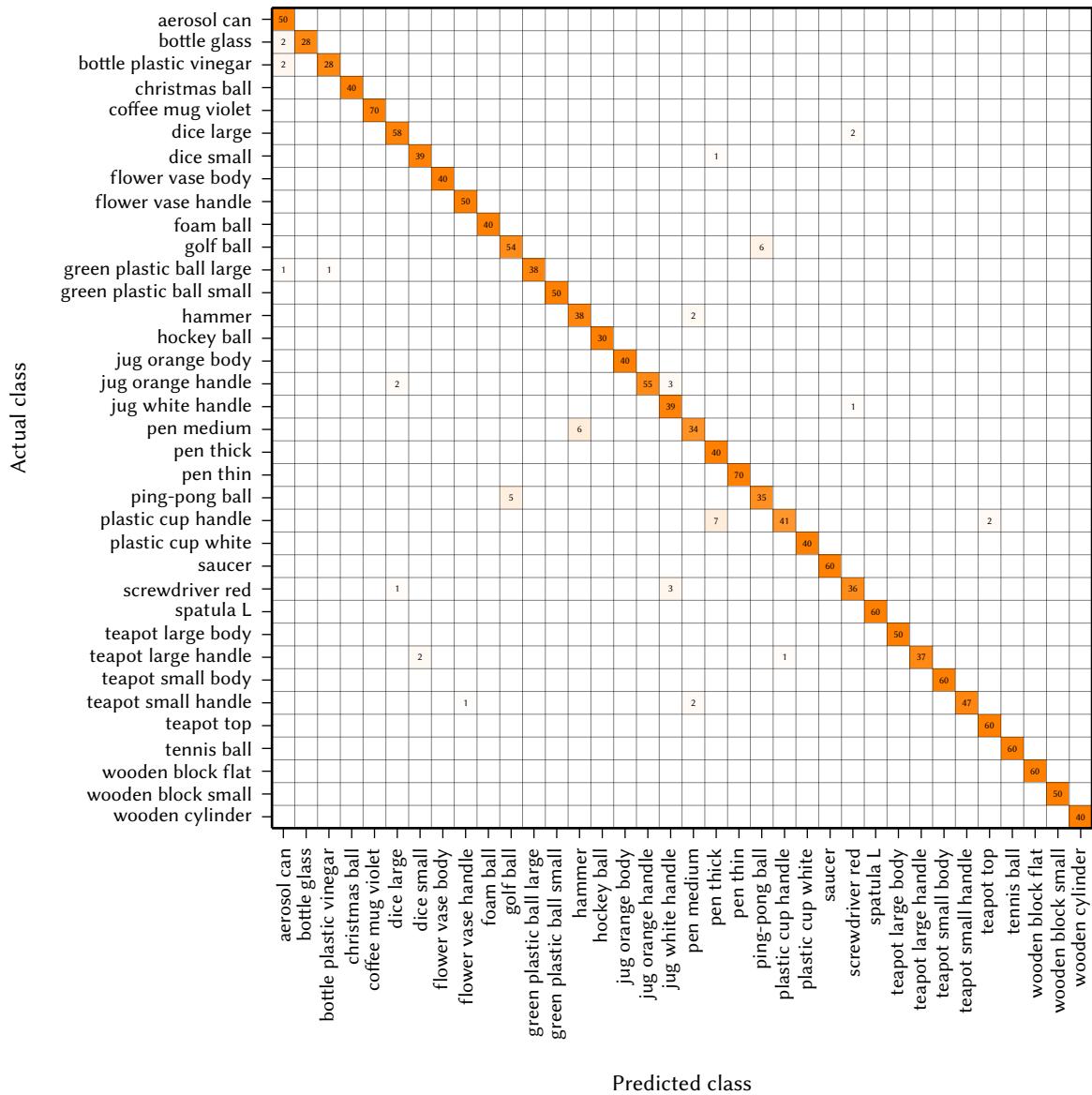


Figure 6.19: SVM hyperparameter optimization using a grid search strategy: Average cross-validation accuracy for different parameter combinations. More precisely, the randomized procedure is run 10 times with varying seeds for smoother results. The last column refers to the original feature vectors without preceding LDA. Note that after applying the LDA, the actual choice of the SVM's regularization parameter C is almost arbitrary within a large range of values.

6.9 Results and Discussion

Sokolova and Lapalme (2009) give a nice overview of various performance measures for classification. The confusion matrix of figure 6.20 depicts the cumulative result of the performed cross-validation. In general, it is considered bad practice to evaluate the final performance of a classifier with cross-validation. Instead, an independent training set should be composed while holding out a certain percentage of the data for testing. But in this case, the outcome is almost identical with the exception that the cumulative result is less sensitive to the actual split. Note that the final values might vary a little due to the randomized nature of the applied cross-validation scheme.

Symmetric misclassifications occur between the ping-pong and the golf ball as well as between the medium sized pen and the hammer shaft. The distances to the separating hyperplanes in the one-vs-one classifier scheme do not have to be equal between the same pair of classifiers.



where tp refers to the number of true positives, that is the values on the diagonal of the confusion matrix. fp refers to the number of false positives where the corresponding class was predicted even though it was not the actual condition. High precision therefore relates to low false positive rates. Similarly, recall is defined as

$$R = \frac{tp}{tp + fn}$$

where fn is the number of false negatives, that is the number of instances where the actual class was not correctly predicted. High recall indicates a low false negative rate.

Since there is a trade-off between precision and recall, the F-score as a weighted harmonic mean might be more suitable. It is defined as

$$F = 2 \cdot \frac{P \cdot R}{P + R}.$$

If either precision or recall is low then the resulting F-score will be poor as well. Table 6.2 summarizes the results. Note that the support does not refer to the actual number of training samples but to their occurrence during cross-validation.

Table 6.2: Summary of the classification performance metrics using a randomized and stratified cross-validation scheme.

Class	Precision (Reliability)	Recall (Accuracy)	F-score	Support
aerosol can	0.909	1.000	0.952	50
bottle glass	1.000	0.933	0.966	30
bottle plastic vinegar	0.966	0.933	0.949	30
christmas ball	1.000	1.000	1.000	40
coffee mug violet	1.000	1.000	1.000	70
dice large	0.951	0.967	0.959	60
dice small	0.951	0.975	0.963	40
flower vase body	1.000	1.000	1.000	40
flower vase handle	0.980	1.000	0.990	50
foam ball	1.000	1.000	1.000	40
golf ball	0.915	0.900	0.908	60
green plastic ball large	1.000	0.950	0.974	40
green plastic ball small	1.000	1.000	1.000	50
hammer	0.864	0.950	0.905	40
hockey ball	1.000	1.000	1.000	30
jug orange body	1.000	1.000	1.000	40
jug orange handle	1.000	0.917	0.957	60
jug white handle	0.867	0.975	0.918	40
pen medium	0.895	0.850	0.872	40
pen thick	0.833	1.000	0.909	40
pen thin	1.000	1.000	1.000	70
ping-pong ball	0.854	0.875	0.864	40
plastic cup handle	0.976	0.820	0.891	50
plastic cup white	1.000	1.000	1.000	40
saucer	1.000	1.000	1.000	60
screwdriver red	0.923	0.900	0.911	40
spatula	1.000	1.000	1.000	60
teapot large body	1.000	1.000	1.000	50
teapot large handle	1.000	0.925	0.961	40
teapot small body	1.000	1.000	1.000	60
teapot small handle	1.000	0.940	0.969	50
teapot top	0.968	1.000	0.984	60
tennis ball	1.000	1.000	1.000	60
wooden block flat	1.000	1.000	1.000	60
wooden block small	1.000	1.000	1.000	50
wooden cylinder	1.000	1.000	1.000	40
average total	0.971	0.969	0.969	1720

Chapter 7

Developed Software

The software package provided by SCHUNK, the SDHLibrary, comes in two variants for Python and C++. The latter forms the basis for the interaction with the Robot Operating System (ROS) API. Apart from the DSA-Explorer and the DSA-Toolbox by Weiss Robotics which offers a MATLAB/Simulink interface via UDP sockets, there is a framework based on the Real-Time Application Interface for Linux featuring a shared memory interface for MATLAB/Simulink RTW. As described in T. Haase and H. Wörn and Holger Nahrstaedt (2010), Simulink is used to design models and algorithms while QRtaiLab controls the execution in real-time.

But it was soon realized that additional features were needed to conduct repeatable experiments in a convenient manner. A framework that does not rely on proprietary software (excluding SDHLibrary) was therefore developed before the actual evaluation of the tactile sensors could begin. Written in C++, it basically consists of a frame manager and two separate frame grabbers. One grabber requests and processes the temperatures and axis angles received from the SDH-2 while the other captures the tactile sensor frames from the DSA controller. Both frame grabbers have to be manually paused before sending additional commands to the SDH-2 to avoid undefined states. If necessary, this limitation could be overcome with the help of command queues. It is worth mentioning, that some blocking function calls in the library were replaced with non-blocking versions at application level to support continuous grabbing. Position based grasping always relies on the feedback of axis states and collides with the desire to record axis angles. Unfortunately, it is not possible to incorporate both requirements without completely rewriting the existing library.

Figure 7.1 illustrates a GUI application developed on top of the capturing framework. It is used to perform grasps while observing the tactile sensor profile in real-time. If joint angles are captured, a 3D visualization using forward kinematics is available as well. When recording, the received sensor readings are stored in a data structure based on double ended queues and can be exported to a bzip2 compressed xml-file. These *.dsa files can later be opened and examined offline. Python bindings based on Boost.Python and Boost.NumPy were written to allow for a more convenient analysis of the recorded data. One advantage of the developed solution is that it benefits from a huge collection of free software based on the Python programming language. In particular, this project made extensive use of a variety of different libraries such as NumPy, Matplotlib, Scipy, Scikit-learn and OpenCV (C++ and Python). Both slip-detection and classification modules were prototyped in Python. The slip-detection and feature extraction was reimplemented in C++ and integrated into the frame manager for performance reasons. A real-time GUI based on cairo, a vector graphics library, visualizes translational and rotational slip, see figure 7.2 for details. In contrast, the classification is only available in Python. But with the help of Boost.Python an interpreter running the corresponding python scripts can be started from within the C++ application.

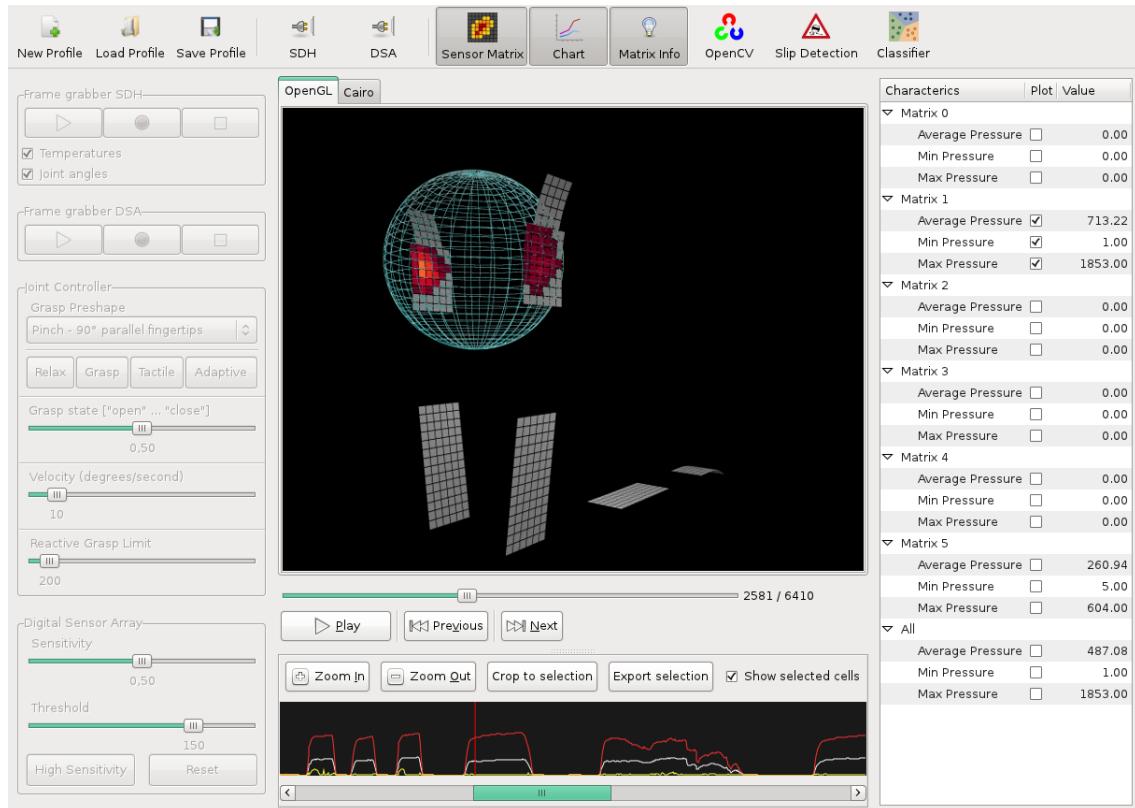
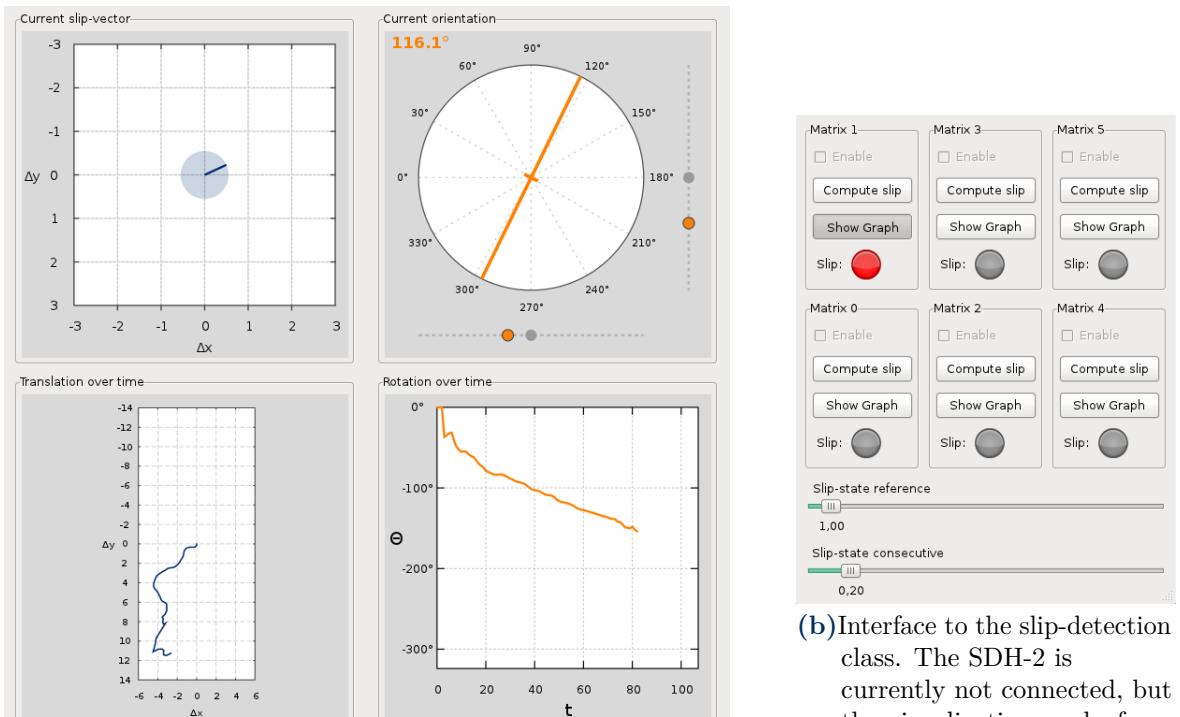


Figure 7.1: Screenshot of the framework's graphical user interface. If the SDH-2 is connected, the application allows the execution of predefined grasp primitives while recording pressure profiles, joint angles and temperatures. All data is visualized in real-time. The time series of the recorded pressure profiles can then be analyzed off-line. The implemented filters, slip-detection and classification can be accessed from the GUI for demonstration purposes.



(a) Translational (blue) and rotational (orange) slip-detection:
The graph is drawn using cairomm, a C++ API for the cairo vector graphics library. The orange knobs in the current orientation view refer to the shape's skewness and indicate a shifting of weight. Note that the magnitude of the current slip-vector is about 0.5 taxel units and thus higher than the threshold depicted in figure to the right.

(b) Interface to the slip-detection class. The SDH-2 is currently not connected, but the visualization works for recorded profiles as well. Since the magnitude of the current slip-vector exceeds the allowed limit between consecutive frames, the binary slip-state indicator reports a slip-condition.

Figure 7.2: Real-time visualization of the slip-detection.

Chapter 8

Conclusions

During the course of this thesis various aspects of tactile sensors have been addressed. In an attempt to smooth the noisy signal, several temporal and spatial filters have been examined. Keeping real-time constraints in mind, only a handful of algorithms come into consideration. Depending on the problem, a relatively simple spatial Gaussian filter or a spatio-temporal median filter will yield the best results. Since temporal smoothing and numerical differentiation did not achieve satisfactory results, an alternative curve-fitting approach was developed. A second-order model, motivated by the working principles of the tactile sensor, was therefore derived and fitted using nonlinear least-squares regression. With the help of this technique, a nonlinear calibration model that relates the measured pressure profiles and temperatures to the applied force was fitted on a per matrix basis. It should be noted, however, that the resulting calibration only works well under controlled conditions. The underlying working principles of the DSA9205 and DSA9210 simply do not match with the requirements of an accurate force sensor.

Instead, the developed methods were used to implement a high sensitivity mode for the SDH-2. Depending on the current temperatures, measured near the axis motors, the internal sensor thresholds are set to the lowest possible values. This procedure ensures maximum sensitivity while avoiding noise and thus low frame rates.

The ability to detect and avoid slip is another beneficial ability in robotic grasping. Several convolution based techniques to compute the displacement between consecutive tactile images were examined. The fastest approach which tracks the moving centroid of the convolution matrices was implemented. It is accompanied by rotational slip-detection based on the principal axis method. Special care was taken to detect undefined states in case of circular pressure profiles. The effectiveness of the resulting slip-detection module was demonstrated by means of a reactive-grasping application featuring a GUI for visualization. As mentioned before, the resulting framework can be used to prevent grasped objects from slipping while keeping the applied force as low as possible. Future grasping algorithms could make use of the detected slip state to estimate friction and to refine the grasp in an adaptive manner. Combining the spatial information of all involved sensor matrices, such an algorithm could also react to applied external forces with appropriate countermeasures.

Basis for such spatial awareness is the knowledge of the exact 3D position of the grasped object. The computation of forward kinematics was realized for individual taxels based on the Denavit-Hartenberg convention. This allows the estimation of the grasped object's size as well as its compressibility. It was demonstrated in this thesis that tactile images can be used to classify grasp-object configurations. Additional features such as the object's diameter and its compressibility further improve the recognition rate. This finding emphasizes the importance of combined tactile and kinesthetic perception in robotic applications.

The main difficulty of the object classification task was to find suitable features to represent typical pressure profiles. Translation and rotation invariant Chebyshev moments turned out to be a good choice to deal with the problem. However, the method is not very scalable and future developments such as high-resolution artificial skin will have to find other representations.

The implemented classification approach has several limitations and further investigation is necessary to build a generalized model for tactile sensing. For example, only two tactile sensor matrices were used for classification to reduce the complexity of the problem. But even with this simplification there is an unresolved problem concerning symmetry. It is easy to see that most objects can be grasped in many different ways and that a new class is needed for each grasp-object configuration. But in reality there is a lot of redundancy in the interaction between the grasped object, joint configurations and sensor matrices. To illustrate the problem, consider the spatula depicted on page 92. Turning the object 180° around its major axis (or grasping from the other side) results in a mirrored version of the otherwise identical feature vector. The problem gets even worse if all six sensor matrices are involved. A Bag-of-Words model would certainly be an easy way out, but at the same time all spatial relations of the tactile images would be lost. This observation is not new and other approaches such as part-based constellation models that relate features while maintaining geometric constraints are an active area of research in computer vision. Future work could also investigate the possibility of generative models. This would simplify the task of local grasp refinement by predicting and verifying the current grasp-object configuration. From there, modifications that improve the grasp could be inferred. A natural extension of the presented concept includes the active exploration of grasped objects. Combining the collected grasp-object configurations to a single object model could then be seen as an instance of haptic perception.

Even though tactile sensors were studied for over three decades, there is still no general-purpose solution available and it remains an open question how to make use of the rich information provided by tactile sensors. But more and more systems with special capabilities emerged in recent years. In that respect, the development of tactile sensors and their integration for robotic applications resembles the development of the field as a whole. It will be interesting to see what the future holds for tactile sensing and robotics in general.

Bibliography

- A. Aizerman, E. M. Braverman, and L. I. Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.
- Javier Adolfo Alcazar and Leandro G. Barajas. Estimating object grasp sliding via pressure array sensing. In *Robotics and Automation (ICRA)*, pages 1740–1746. IEEE, 2012.
- I.I. Argatov. Mathematical modeling of linear viscoelastic impact: Application to drop impact testing of articular cartilage. *Tribology International*, 63(0):213 – 225, 2013. ISSN 0301-679X. doi: <http://dx.doi.org/10.1016/j.triboint.2012.09.015>. URL <http://www.sciencedirect.com/science/article/pii/S0301679X12003118>. The International Conference on BioTribology 2011.
- K.J. Astrom and C. Canudas-de Wit. Revisiting the lugre friction model. *Control Systems, IEEE*, 28(6):101–114, Dec 2008. ISSN 1066-033X.
- Yasemin Bekiroglu, Renaud Detry, and Danica Kragic. Joint observation of object pose and tactile imprints for online grasp stability assessment. In *Manipulation Under Uncertainty (Workshop at IEEE ICRA 2011)*, 2011.
- Jiang-Tao Cui Bin Xiao, Jian-Feng Ma. Invariant Pattern Recognition using Radial Tchebichef moments. *Chinese Conference on Pattern Recognition (CCPR) 2010, Chongqing*, 2010.
- Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.
- P.L. Bonate. *Pharmacokinetic-Pharmacodynamic Modeling and Simulation*. SpringerLink : Bücher. Springer US, 2011. ISBN 9781441994851.
- Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM. ISBN 0-89791-497-X. doi: 10.1145/130385.130401. URL <http://doi.acm.org/10.1145/130385.130401>.
- H.F. Brinson and L.C. Brinson. *Polymer Engineering Science and Viscoelasticity: An Introduction*. Springer US, 2010. ISBN 9781441944788. URL <https://books.google.de/books?id=Ft6dcQAACAAJ>.
- A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65 vol. 2, June 2005. doi: 10.1109/CVPR.2005.38.
- R. Burden and J. Faires. *Numerical Analysis*. Cengage Learning, 2010. ISBN 9780538733519. URL <http://books.google.ca/books?id=zXnSxY9G2JgC>.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

Olivier Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19: 1155–1178, 2007.

Chee-Way Chong, P. Raveendran, and R. Mukundan. Translation invariants of zernike moments. *Pattern Recognition*, 36(8):1765–1773, 2003.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

J.J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley series in electrical and computer engineering: control engineering. Pearson/Prentice Hall, 2005. ISBN 9780201543612.

P. Crichton. Penfield’s homunculus: a note on cerebral cartography. In *Journal of Neurology, Neurosurgery, and Psychiatry*, volume 56, pages 329–333, 1994. URL <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1072893/>.

S. Cubero. *Industrial Robotics: Theory, Modelling and Control*. Pro-Literatur-Verlag, 2007. ISBN 9783866112858.

Ravinder S. Dahiya and Maurizio Valle. Tactile sensing for robotic applications. In J.G. Rocha and S. Lanceros-Mendez, editors, *Sensors: Focus on Tactile Force and Stress Sensors*. InTech, 2008.

Dabiri Dana. Cross-correlation digital particle image velocimetry - a review. *Associação Brasileira de Ciências Mecânicas*, 2006.

Hao Dang and Peter K. Allen. Learning grasp stability. In *ICRA*, pages 2392–2397. IEEE, 2012. ISBN 978-1-4673-1403-9.

Hao Dang and Peter K. Allen. Grasp adjustment on novel objects using tactile experience from similar local geometry. In *IROS*, pages 4007–4012. IEEE, 2013.

Hao Dang, Jonathan Weisz, and P.K. Allen. Blind grasping: Stable robotic grasping using tactile feedback and hand kinematics. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5917–5922, May 2011. doi: 10.1109/ICRA.2011.5979679.

Misganu Debella-Gilo and Andreas Käab. Sub-pixel precision image matching for measuring surface displacements on mass movements using normalized cross-correlation. *Remote Sensing of Environment*, 115:130–142, 2011.

A. Drimus, G. Kootstra, A. Bilberg, and D. Kragic. Classification of rigid and deformable objects using a novel tactile sensor. In *Advanced Robotics (ICAR), 2011 15th International Conference on*, pages 427–434, June 2011. doi: 10.1109/ICAR.2011.6088622.

C.U.S. Edussooriya, H.S.S. Hapuachchi, D. Rajiv, R.A.H. Ranasinghe, and S.R. Munasinghe. Analysis of grasping and slip detection of the human hand. In *Information and Automation for Sustainability, 2008. ICIAFS 2008. 4th International Conference on*, pages 261–266, Dec 2008.

Kaspar Fischer. Smallest enclosing balls of balls - combinatorial structure & algorithms, 2005.

- R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(7):179–188, 1936.
- Jan Flusser and Tomáš Suk. Rotation moment invariants for recognition of symmetric objects. *IEEE Transactions on Image Processing*, 15(12):3784 – 3790, 2006. doi: 10.1109/TIP.2006.884913.
- Jan Flusser, Barbara Zitova, and Tomas Suk. *Moments and Moment Invariants in Pattern Recognition*. Wiley Publishing, 2009. ISBN 0470699876, 9780470699874.
- George Forman and Martin Scholz. Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement. *SIGKDD Explor. Newsl.*, 12(1):49–57, November 2010. ISSN 1931-0145. doi: 10.1145/1882471.1882479. URL <http://doi.acm.org/10.1145/1882471.1882479>.
- Bernd Gärtner. Fast and robust smallest enclosing balls. In *Proceedings of the 7th Annual European Symposium on Algorithms*, ESA ’99, pages 325–338, London, UK, UK, 1999. Springer-Verlag. ISBN 3-540-66251-0. URL <http://dl.acm.org/citation.cfm?id=647909.740295>.
- Shaun S. Gleason, Martin A. Hunt, and W. Bruce Jatko. Subpixel measurement of image features based on paraboloid surface fit. In *Proc. Machine Vision Systems Integration in Industry, SPIE*, pages 135–144, 1990.
- D. Göger, K. Weiss, C. Burghart, and H. Wörn. Sensitive skin for a humanoid robot. In *Human-Centered Robotic Systems (HCRS ’06)*, Munich, October 6th and 7th,, 2006.
- D. Göger, N. Gorges, and Heinz Worn. Tactile sensing for an anthropomorphic robotic hand: Hardware and signal processing. In *Robotics and Automation, 2009. ICRA ’09. IEEE International Conference on*, pages 895–901, May 2009. doi: 10.1109/ROBOT.2009.5152650.
- M.S. Grewal and A.P. Andrews. *Kalman filtering: theory and practice using MATLAB*. [Wiley InterScience Online Books, Electronic and Electrical Engineering Collection]. Wiley, 2001. ISBN 9780471392545.
- Manuel Guizar-Sicairos, Samuel T. Thurman, and James R. Fienup. Efficient subpixel image registration algorithms. *Optics Letters* 33, 33:156–158, 2008.
- Isabelle Guyon. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
- Isabelle Guyon, Jason Weston, Stephen Barnhill, Vladimir Vapnik, and Nello Cristianini. Gene selection for cancer classification using support vector machines. In *Machine Learning*, 2002.
- H. Wörn und T. Haase. Motion control and reactive grasping skill concepts for the multi-fingered sdh2. *15th International Conference on Mechatronics Technology*, page 6, 2011.
- Thomas Haase. *Greifplanung und Greifskills für reaktives Greifen*. PhD thesis, Karlsruhe Institute of Technology, 2011. URL <http://dx.doi.org/10.5445/KSP/1000024291>.
- T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer, 2009. ISBN 9780387848587. URL <https://books.google.de/books?id=tVIjmNS30b8C>.
- David Hilbert. Grundzüge einer allgemeinen theorie der linearen integralgleichungen. *Göttinger Nachrichten, Erste und zweite Mitteilung, Reprinted 1912*, pages 49–91, 213–259, 1904.

Van Ho and Shinichi Hirai. Understanding slip perception of soft fingertips by modeling and simulating stick-slip phenomenon. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2011.

Robert D. Howe. Tactile sensing and control of robotic manipulation. *Journal of Advanced robotics*, 8(3):245–261, 1994.

Kaijen Hsiao, Sachin Chitta, Matei Ciocarlie, and E. Gil Jones. Contact-reactive grasping of objects with partial shape information. In *IROS*, 2010. URL <https://www.willowgarage.com/sites/default/files/contactreactive.pdf>.

Ming-Kuei Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, February 1962. ISSN 0096-1000. doi: 10.1109/TIT.1962.1057692.

Emil Hyttinen, Danica Kragic, and Renaud Detry. Learning the tactile signatures of prototypical object parts for robust part-based grasping of novel objects. In *IEEE International Conference on Robotics and Automation*, 2015.

Bernd Jähne. *Digital image processing, 6th ed.* Springer, 2005.

Nawid Jamali, Philip Byrnes-preston, Rudino Salleh, and Claude Sammut. Texture recognition by tactile sensing, 2009.

R. S. Johansson and Aa Vallbo. Properties of Cutaneous Mechanoreceptors in the Human Hand Related to Touch Sensation. *Human Neurobiology*, 1984.

M. Kaboli, P. Mittendorfer, V. Hugel, and G. Cheng. Humanoids learn object properties from robust tactile feature descriptors via multi-modal artificial skin. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 187–192, Nov 2014. doi: 10.1109/HUMANOIDS.2014.7041358.

Imin Kao, Kevin M Lynch, and J. Burdick. *Chapter 27, Contact Modeling and Manipulation.* Springer-Verlag New York Inc, 2008.

Doukeni Karakatsani. Object manipulation with two robotic fingers using tactile information. Master’s thesis, Universitat Politècnica de Catalunya, 2011.

Hae-Kwang Kim, Jong-Deuk Kim, Dong-Gyu Sim, and Dae-Il Oh. A modified zernike moment shape descriptor invariant to translation, rotation and scale for similarity-based image retrieval. In *IEEE International Conference on Multimedia and Expo (I)*, pages 307–310, 2000.

S. Knerr, L. Personnaz, and G. Dreyfus. Single-layer learning revisited: a stepwise procedure for building and training a neural network. In Françoise Fogelman Soulié and Jeanny Héault, editors, *Neurocomputing*, volume 68 of *NATO ASI Series*, pages 41–50. Springer Berlin Heidelberg, 1990. ISBN 978-3-642-76155-3. doi: 10.1007/978-3-642-76153-9_5. URL http://dx.doi.org/10.1007/978-3-642-76153-9_5.

D.E. Knuth. Two notes on notation. *Amer. Math. Monthly*, 99:403–422, 1992.

Roelof Koekoek and René F. Swarttouw. The askey-scheme of hypergeometric orthogonal polynomials and its q-analogue, 1998.

C. D. Kuglin and D. C. Hines. The phase correlation image alignment method. *IEEE Conference on Cybernetics and Society*, pages 163–165, 1975.

- J. P. Lewis. Fast normalized cross-correlation, 1995.
- Li Li, Bo Fu, Wen Xu, Bo Li, and Guojun Zhang. Image analysis by discrete radial tchebichef moments. In Maozhen Li, Qilian Liang, Lipo Wang, and Yibin Song, editors, *FSKD*, pages 569–572. IEEE, 2010. ISBN 978-1-4244-5934-6.
- Shan Li, Moon-Chuen Lee, and Chi-Man Pun. Complex zernike moments features for shape-based image retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 39(1):227–237, 2009.
- Tao Li, Shenghuo Zhu, and Mitsunori Ogihara. Using discriminant analysis for multi-class classification: an experimental investigation. *Knowl. Inf. Syst.*, 10(4):453–472, 2006. URL <http://dx.doi.org/10.1007/s10115-006-0013-y>.
- David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.
- Jan Lunze. *Regelungstechnik 1: Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen*. Springer, 2012.
- S. Luo, W. Mou, K. Althoefer, and H. Liu. Novel tactile-sift descriptor for object shape recognition. *Sensors Journal, IEEE*, PP(99):1–1, 2015. ISSN 1530-437X. doi: 10.1109/JSEN.2015.2432127.
- Nakamoto Masataka, Masashi Konyo, Takashi Maeno, and Satoshi Tadokoro. Reflective grasp force control of humans induced by distributed vibration stimuli on finger skin with icpf actuators. In *ICRA*, pages 3899–3904. IEEE, 2006.
- M. Meier, M. Schopfer, R. Haschke, and H. Ritter. A probabilistic approach to tactile shape reconstruction. *Robotics, IEEE Transactions on*, 27(3):630–635, June 2011. ISSN 1552-3098. doi: 10.1109/TRO.2011.2120830.
- Claudio Melchiorri. Slip detection and control using tactile and force sensors. *Mechatronics, IEEE/ASME Transactions on*, 5(3):235–243, Sep 2000. ISSN 1083-4435. doi: 10.1109/3516.868914.
- J. Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 209(441-458):415–446, 1909. URL <http://dx.doi.org/10.1098/rsta.1909.0016>.
- S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. R. Müllers. Fisher discriminant analysis with kernels. *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop*, pages 41–48, August 1999. doi: 10.1109/nns.1999.788121. URL <http://dx.doi.org/10.1109/nns.1999.788121>.
- S. Moisio, B. Leon, P. Korkealaakso, and A. Morales. Simulation of tactile sensors using soft contacts for robot grasping applications. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 5037–5043, May 2012. doi: 10.1109/ICRA.2012.6224629.
- Antonio Morales, Mario Prats, Pedro Sanz, and Angel P. Pobil. An Experiment in the Use of Manipulation Primitives and Tactile Perception for Reactive Grasping. *Robotics: Science and Systems - Robot Manipulation: Sensing and Adapting to the Real World*, 2007.

- V.B. Mountcastle. *The Sensory Hand: Neural Mechanisms of Somatic Sensation*. Harvard University Press, 2005. ISBN 9780674019744. URL <https://books.google.de/books?id=W0mqKSheygYC>.
- R. Mukundan. Improving image reconstruction accuracy using discrete orthonormal moments. In Hamid R. Arabnia and Youngsong Mun, editors, *CISST*, pages 287–291. CSREA Press, 2003. ISBN 1-892512-46-7.
- R. Mukundan. Some computational aspects of discrete orthonormal moments. *IEEE Transactions on Image Processing*, 13(8):1055–1059, 2004a.
- R. Mukundan. A New Class of Rotational Invariants Using Discrete Orthogonal Moments. *6th IASTED Conference on Signal and Image Processing - SIP2004*, 2004b.
- R. Mukundan, S. H. Ong, and P. A. Lee. Discrete Orthogonal Moment Features Using Chebyshev Polynomials. *International Conference on Image and Vision Computing - IVCNZ 2000*, pages 27–29, 2000.
- R. Mukundan, S. H. Ong, and Poh Aun Lee. Image analysis by tchebichef moments. *IEEE Transactions on Image Processing*, 10(9):1357–1364, 2001.
- R. Mukundan, S. H. Ong, and P. A. Lee. Radial Tchebichef Invariants for Pattern Recognition. *Proc. of IEEE Tencon Conference - TENCON05*, 2005.
- S.E. Navarro, N. Gorges, H. Worn, J. Schill, T. Asfour, and R. Dillmann. Haptic object recognition for multi-fingered robot hands. In *Haptics Symposium (HAPTICS), 2012 IEEE*, pages 497–502, March 2012. doi: 10.1109/HAPTIC.2012.6183837.
- Nihat Özkaya, Margareta Nordin, David Goldsheyder, and Dawn Leger. Mechanical properties of biological tissues. In *Fundamentals of Biomechanics*, pages 221–236. Springer New York, 2012. ISBN 978-1-4614-1149-9. doi: 10.1007/978-1-4614-1150-5_15. URL http://dx.doi.org/10.1007/978-1-4614-1150-5_15.
- M. Pawlak. *Image analysis by moments: reconstruction and computational aspects*. Oficyna Wydawn. Politechn., 2006. ISBN 9788370859664. URL <https://books.google.de/books?id=PspnAAAACAAJ>.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- Somrak Petcharatee. *Object Manipulation and Force Control by Using Tactile Sensors*. PhD thesis, Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, 2008.
- Z. Pezzementi, E. Plaku, C. Reyda, and G.D. Hager. Tactile-object recognition from appearance information. *Robotics, IEEE Transactions on*, 27(3):473–487, June 2011. doi: 10.1109/TRO.2011.2125350.
- John C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING, 1998.
- Pressure Profile Systems, Inc. Robotouch™tactile sensors with digital output for the barrett hand, 2010. URL https://www.pressureprofile.com/s/PPS-_RoboTouch_SpecSheet.pdf.

- John G Proakis and Dimitris G Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications*, 4/e. Pearson Education, 2007.
- P.A.Lee R. Mukundan, S.H.Ong. Discrete vs. continuous orthogonal moments in image analysis. *International Conference on Imaging Systems, Science and Technology CISST 2001*, pages 23–29, 2001.
- Alain Rakotomamonjy. Variable selection using svm based criteria. *J. Mach. Learn. Res.*, 3: 1357–1370, March 2003. ISSN 1532-4435.
- C. Radhakrishna Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society - Series B*, 10(2):159–203, 1948.
- B. Srinivasa Reddy and Biswanath N. Chatterji. An fft-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing*, 5(8):1266–1271, 1996.
- J. Reinecke, A. Dietrich, F. Schmidt, and M. Chalon. Experimental comparison of slip detection strategies by tactile sensing with the biotac on the dlr hand arm system. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2742–2748, May 2014. doi: 10.1109/ICRA.2014.6907252.
- Hannes P. Saal, Jo-Anne Ting, and Sethu Vijayakumar. Active estimation of object dynamics parameters with tactile sensors. In *IROS*, pages 916–921. IEEE, 2010. ISBN 978-1-4244-6674-0.
- José A. Sánchez-Durán, Ócar Oballe-Peinado, Julián Castellanos-Ramos, and Fernando Vidal-Verdú. Hysteresis correction of tactile sensor response with a generalized prandtl-ischlinskii model. *Microsystem Technologies*, 18(7-8):1127–1138, 2012. doi: 10.1007/s00542-012-1455-7.
- Simo Särkkä. *Bayesian filtering and smoothing*. Cambridge University Press, 2013.
- Eric L. Sauser, Brenna D. Argall, Giorgio Metta, and Aude G. Billard. Iterative learning of grasp adaptation through human corrections. *Robot. Auton. Syst.*, 60(1):55–71, January 2012. ISSN 0921-8890. doi: 10.1016/j.robot.2011.08.012. URL <http://dx.doi.org/10.1016/j.robot.2011.08.012>.
- A. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36:1627–1639, 1964.
- Andreas Schmid. *Intuitive Human-Robot Cooperation*. PhD thesis, Karlsruhe Institute of Technology, 2008.
- A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard. Object identification with tactile sensors using bag-of-features. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 243–248, Oct 2009. doi: 10.1109/IROS.2009.5354648.
- B. Schölkopf, C.J.C. Burges, and A.J. Smola. *Advances in Kernel Methods: Support Vector Learning*. Philomel Books, 1999. ISBN 9780262194167. URL https://books.google.de/books?id=_NYamXKkNM8C.
- M. Schopfer, M. Pardowitz, and H. Ritter. Using entropy for dimension reduction of tactile data. In *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pages 1–6, June 2009.
- Schunk GmbH & Co. KG. <http://www.pressebox.de>, June 19, 2008. BoxId=184485.

Carsten Schürmann, Matthias Schöpfer, Robert Haschke, and Helge J. Ritter. A high-speed tactile sensor for slip detection. In *Towards Service Robots for Everyday Environments*, volume 76 of *Springer Tracts in Advanced Robotics*, pages 403–415. Springer, 2012. ISBN 978-3-642-25115-3.

Huazhong Shu, Hui Zhang, Beijing Chen, Pascal Haigron, and Limin Luo. Fast computation of tchebichef moments for binary and grayscale images. *IEEE Transactions on Image Processing*, 19(12):3171–3180, 2010.

Mohan M Shyam, Naik Naren, R.M.O Gemson, and M.R. Ananthasayanam. Introduction to the kalman filter and tuning its statistics for near optimal estimates and cramer rao bound. Technical report, Dept. of Electrical Engineering, IIT Kanpur, India, 2015. URL <http://arxiv.org/abs/1503.04313>.

Steven W. Smith. *The Scientist and Engineer’s Guide to Digital Signal Processing, Second Edition*. California Technical Publishing, San Diego, CA, USA, 1999.

Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manage.*, 45(4):427–437, July 2009. ISSN 0306-4573. doi: 10.1016/j.ipm.2009.03.002. URL <http://dx.doi.org/10.1016/j.ipm.2009.03.002>.

Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Dynamics and Control, Second Edition*. Self-publish, 2004.

Michael Strohmayr. *Artificial Skin in Robotics*. PhD thesis, Karlsruhe Institute of Technology, 2012. URL <http://nbn-resolving.org/urn:nbn:de:swb:90-287827>.

C. Strub, F. Wörgötter, H. Ritter, and Y. Sandamirskaya. Correcting pose estimates during tactile exploration of object shape: a neuro-robotic study. In *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2014 Joint IEEE International Conferences on*, pages 26–33, Oct 2014. doi: 10.1109/DEVLRN.2014.6982950.

Herbert Suesse and Frank Ditrich. Robust determination of rotation-angles for closed regions using moments. In *ICIP (1)*, pages 337–340. IEEE, 2005.

T. Haase and H. Wörn and Holger Nahrstaedt. Shared Memory in RTAI Simulink for Kernel and User-space Communication at the Example of the SDH2: QRtaiLab For SDH-2 Matrix Visualization. In INCINCO 2010, editor, *7th International Conference on Informatics in Control, Automation and Robotics*, page 6, 2010.

M. R. Teague. Image analysis via the general theory of moments. *Journal of the Optical Society of America (1917-1983)*, 70:920–930, August 1980. URL http://adsabs.harvard.edu/cgi-bin/nph-bib_query?bibcode=1980JOSA...70..920T.

Cho-Huak Teh and Roland T. Chin. On image analysis by the methods of moments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(4):496–513, 1988.

S. Teshigawara, T. Tsutsumi, S. Shimizu, Y. Suzuki, Aiguo Ming, M. Ishikawa, and M. Shimojo. Highly sensitive sensor for detection of initial slip and its application in a multi-fingered robot hand. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1097–1102, May 2011. doi: 10.1109/ICRA.2011.5979750.

Tinne Tuytelaars and Krystian Mikolajczyk. *Local Invariant Feature Detectors: A Survey*. Now Publishers Inc., Hanover, MA, USA, 2008. ISBN 1601981384, 9781601981387.

- Heinz Unbehauen. *Regelungstechnik I*. Vieweg + Teubner, 2008.
- V Vapnik and A Chervonenkis. On a perceptron class. *Avtomat. i Telemekh*, 25:112–120, 1964.
URL <http://mi.mathnet.ru/eng/at11558>.
- V. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24., 1963.
- V. N. Vapnik. An overview of statistical learning theory. *Trans. Neur. Netw.*, 10(5):988–999, September 1999. ISSN 1045-9227. doi: 10.1109/72.788640. URL <http://dx.doi.org/10.1109/72.788640>.
- Ramiro Velazquez and Edwige Pissaloux. Tactile displays in human-machine interaction: Four case studies. *IJVR*, 7(2):51–58, 2008.
- F.E. Vina B, Y. Bekiroglu, C. Smith, Y. Karayiannidis, and D. Kragic. Predicting slippage and learning manipulation affordances through gaussian process regression. In *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*, pages 462–468, Oct 2013. doi: 10.1109/HUMANOIDS.2013.7030015.
- Joviša Žunić. Shape descriptors for image analysis. Mihaljević, Miodrag (ed.), Selected topics on image processing and cryptology. Beograd: Matematički Institut SANU. Zbornik Radova (Beograd) 15(23), 5-38 (2011)., 2011.
- J. Walker, A. Blank, P. Shewokis, and M. O’Malley. Tactile feedback of object slip facilitates virtual object manipulation. *Haptics, IEEE Transactions on*, PP(99):1–1, 2015. ISSN 1939-1412. doi: 10.1109/TOH.2015.2420096.
- K. Weiss and Heinz Worn. Resistive tactile sensor matrices using inter-electrode sampling. In *Industrial Electronics Society, 2005. IECON 2005. 31st Annual Conference of IEEE*, pages 6 pp.–, Nov 2005a. doi: 10.1109/IECON.2005.1569203.
- K. Weiss and Heinz Worn. The working principle of resistive tactile sensor cells. In *Mechatronics and Automation, 2005 IEEE International Conference*, volume 1, pages 471–476 Vol. 1, July 2005b. doi: 10.1109/ICMA.2005.1626593.
- K. Weiß, D. Göger, and H. Wörn. Reactive grasping in industrial automation. In *Tagungsband der 4. Deutschen Robotik-Konferenz, München*, 2006.
- Karsten Weiss and Heinz Woern. Tactile sensor system for an anthropomorphic robotic hand. In *IEEE International conference on manipulation and grasping IMG*, 2004.
- Weiss Robotics GmbH & Co. KG. Technical drawing: Tactile sensor dsa 9205, 2012a. URL <http://www.weiss-robotics.de/en/taktile-sensorik/taktile-sensoren/tactile-sensor-dsa-9205.html>.
- Weiss Robotics GmbH & Co. KG. Technical drawing: Tactile sensor dsa 9210, 2012b. URL <http://www.weiss-robotics.de/en/tactile-sensing/tactile-sensors/tactile-sensor-dsa-9210.html>.
- D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1):67–82, Apr 1997. doi: 10.1109/4235.585893.

Mingqiang Yang, Kidiyo Kpalma, and Joseph Ronsin. Shape-based invariant feature extraction for object recognition. In Roumen Kountchev and Kazumi Nakamatsu, editors, *Advances in Reasoning-Based Image Processing Intelligent Systems*, volume 29 of *Intelligent Systems Reference Library*, pages 255–314. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-24692-0. doi: 10.1007/978-3-642-24693-7_9. URL http://dx.doi.org/10.1007/978-3-642-24693-7_9.

Huang Ying, Fu Xiulan, Wang Min, Huang Panfeng, and Ge Yunjian. Research on nanosio₂/carbon black composite for flexible tactile sensing. In *Information Acquisition, 2007. ICIA '07. International Conference on*, pages 260–264, July 2007. doi: 10.1109/ICIA.2007.4295738.

Dengsheng Zhang and Guojun Lu. Content-based shape retrieval using different shape descriptors: A comparative study. In *Proceedings of the 2001 IEEE International Conference on Multimedia and Expo, ICME 2001, August 22-25, 2001, Tokyo, Japan*, 2001. doi: 10.1109/ICME.2001.1237928. URL <http://dx.doi.org/10.1109/ICME.2001.1237928>.

Dengsheng Zhang and Guojun Lu. Review of shape representation and description techniques, 2004.

Hui Zhang, Xiubing Dai, Pei Sun, Hongqing Zhu, and Huazhong Shu. Symmetric image recognition by tchebichef moment invariants. In *Proceedings of the International Conference on Image Processing, ICIP 2010, September 26-29, Hong Kong, China*, pages 2273–2276, 2010. doi: 10.1109/ICIP.2010.5654131. URL <http://dx.doi.org/10.1109/ICIP.2010.5654131>.

Hongqing Zhu, Huazhong Shu, Ting Xia, Limin Luo, and Jean-Louis Coatrieux. Translation and scale invariants of tchebichef moments. *Pattern Recognition*, 40(9):2530–2542, 2007a. doi: 10.1016/j.patcog.2006.12.003. URL <http://dx.doi.org/10.1016/j.patcog.2006.12.003>.

Qing Zhu, Bo Wu, and Neng Wan. A sub-pixel location method for interest points by means of the Harris interest strength. *The Photogrammetric Record*, 22(120):321–335, 2007b.