

Grasping and Manipulation of Unknown Objects Based on Visual and Tactile Feedback

Robert Haschke

Abstract The sense of touch allows humans and higher animals to perform coordinated and efficient interactions within their environment. Recently, tactile sensor arrays providing high force, spatial, and temporal resolution became available for robotics, which allows us to consider new control strategies to exploit this important and valuable sensory channel for grasping and manipulation tasks. Successful dexterous manipulation strongly depends on tight feedback loops integrating proprioceptive, visual, and tactile feedback. We introduce a framework for tactile servoing that can realize specific tactile interaction patterns, for example to establish and maintain contact (grasping) or to explore and manipulate objects. We demonstrate and evaluate the capabilities of the proposed control framework in a series of preliminary experiments employing a 16×16 tactile sensor array attached to a Kuka LWR arm as a large fingertip.

Keywords Grasping · Tactile servoing · Online motion planning

1 Introduction

The sense of touch allows humans to perform coordinated and efficient interactions within their environment. Without the sense of touch, subjects have severe difficulties maintaining a stable grasp or performing a complex action such as lighting matches [1, 2]. Also in robot applications, lacking tactile feedback results in losing an initially grasped object or failing to robustly carry out manipulation tasks [3]. In recent years, the resolution and sensitivity of tactile sensors only sufficed for basic force feedback during blind grasping [4]. However, tactile sensor arrays providing high spatial and temporal resolution as well as high sensitivity [5, 6] emerged recently, allowing for more advanced control methods involving tactile feedback too.

R. Haschke (✉)

Cognitive Interaction Technology Excellence Cluster (CITEC),
Bielefeld University, Inspiration 1, 33619 Bielefeld, Germany
e-mail: rhaschke@techfak.uni-bielefeld.de

Such control approaches—which we denote as tactile servoing in accordance to corresponding control approaches involving direct visual feedback—require advanced tactile perception methods and their integration into control programs for direct robot control. Tactile servoing includes important tasks like sliding a finger tip along an object’s surface, tracking specific surface structures like ridges, searching for distinctive tactile patterns, or exploring the object shape by groping. Most of these tasks are essential for both in-hand object manipulation [7], and haptic object identification [8].

Drawing on ideas for visual servoing and applying image processing algorithms to the tactile force image provided by modern tactile sensor arrays, it is possible to extract basic tactile features in real time and employ them for robot control. The challenging mission is to find generic features, which not only work in specific hard-coded control scenarios on a specific type of tactile sensor, but that generalize to a rich set of control tasks and sensor types.

We argue for a unified and open control framework that can cover many grasping and manipulation tasks including tactile exploration. The proposed control approach facilitates the exploitation of task symmetries to unleash redundancies which can be efficiently utilized by subordinated tasks. Different, challenging tasks can be easily composed from a set of basic control primitives without the need for a detailed situation modeling (object and hand shape, friction properties, etc.), thus providing the foundation to yield robust manipulation skills also in unknown and unstructured environments.

The remaining chapter is organized as follows: In the next section we introduce the general concept of the control basis framework and discuss how efficient local motion generation methods can reduce the need for explicit planning in grasping of unknown objects. The subsequent Sect. 3 will introduce some recent tactile sensor developments and vision-based feature extraction methods to yield tactile features, which are at the basis of four tactile servoing control primitives. Finally, in Sect. 4 we describe and evaluate some tactile exploration tasks that impressively demonstrate the power of the proposed control framework.

2 Planning-Less Grasping in the Control Basis Framework

Gruen et al. first developed the idea of the control basis framework (CBF), which allows to realize complex tasks by composition of several basic controllers [9, 10]. Each of those controllers realizes resolved motion rate control, mapping updates of task control variables $\Delta \mathbf{x}$ to joint angle updates $\Delta \mathbf{q}$ of the robot. An important key idea was to stack controllers by priority allowing a subordinate controller to operate in the null-space of a higher-priority controller only, which can be easily achieved using appropriate null-space projections. Given any nonlinear relationship $\mathbf{x}(\mathbf{q})$ between joint and task-space variables, the relation of their velocities at any point t in time is linear and given as

$$\dot{\mathbf{x}}(t) = J(\mathbf{q}(t)) \cdot \dot{\mathbf{q}}(t), \quad (1)$$

where $J_{ij}(\mathbf{q}) = \partial x_i / \partial \mathbf{q}_j$ is the task Jacobian at time t . Then, the solution to realize three priority-ordered task space motions $\dot{\mathbf{x}}_1, \dot{\mathbf{x}}_2, \dot{\mathbf{x}}_3$ looks like this:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_1 + N_1 (\dot{\mathbf{q}}_2 + N_2 \dot{\mathbf{q}}_3) \quad (2)$$

$$= J_1^+(\mathbf{q})\dot{\mathbf{x}}_1 + N_1 (J_2^+(\mathbf{q})\dot{\mathbf{x}}_2 + N_2 J_3^+(\mathbf{q})\dot{\mathbf{x}}_3), \quad (3)$$

where J_i^+ denotes the Moore-Penrose pseudoinverse of J_i and $N_i = \mathbf{1} - J^+ J$ denotes the corresponding null-space projector of task $i = 1, 2, 3$.

To work in practice, it's important, that every controller's null-space is rich enough to accommodate lower-priority motions, i.e. that there is enough redundancy. However, classical motion planning approaches attempt to control the end-effector motion in all six degrees of freedom (dof) and thus do not leave the necessary redundancy. But, exploiting the inherent symmetry of many everyday tasks, we can restrict ourselves to a few task-relevant dofs and thus gain the required redundancy.

As a prominent example, consider the grasping of a spherical object. Nowadays grasp planning approaches attempt to generate and evaluate grasps that approach the sphere from all possible directions [11]. However, in this particular task, it's only important to drive the hand towards the sphere—no matter from which direction. This reduced task description only consumes a single dof, namely the hand-object-distance, and frees up all other dofs. The resulting task-space motion $\dot{\mathbf{x}}$ is a straight-line towards the goal, much like in classical Cartesian control. However, the redundant space at a given goal distance is the complete sphere around the target and any null space motion is automatically projected onto this sphere. In this manner, we can easily approach spherical objects for grasping from any direction, without the need to pre-compute a multitude of feasible grasps in advance. The corresponding task Jacobian $J_{\|\cdot\|}$ can be easily computed from the Jacobian J of the standard forward transform:

$$J_{\|\cdot\|} = (\mathbf{x} - \mathbf{x}_{\text{goal}})^t \cdot J \quad (4)$$

Similarly, grasping a cylindrical object, like a bottle, only requires to align the hand axis with the object axis—the orientation angle around this axis can freely be chosen [12]. To allow even more flexibility, one may specify a task-space interval instead of a unique target value [13]. Within the original control basis framework, Platt et al. also propose more abstract controllers, e.g. to maintain force closure, to optimize grasp quality, manipulability, or visibility [14].

2.1 Collision Avoidance

In the context of motion planning, an important subordinate control task is collision and joint-limit avoidance. Joint limits can be easily avoided minimizing a quadratic or higher-order polynomial function [12, 15]:

$$H_{\mathbf{q}} = \sum w_i (q_i - q_i^{\text{ref}})^p \quad w_i = (q_i^{\text{max}} - q_i^{\text{min}})^{-1}, \quad (5)$$

where \mathbf{q}^{ref} defines a reference pose, e.g. in the middle of the joint range, and the w_i 's weight the contribution of individual joints according to their overall motion range.

Local collision avoidance is achieved by a repelling force field originating from each object. To this end, Sugiura [16] proposed to minimize a quadratic cost function defined on the distance $d_{\mathbf{p}} = \|\mathbf{p}_1 - \mathbf{p}_2\|$ between the two closest points \mathbf{p}_1 and \mathbf{p}_2 on the robot and the obstacle:

$$H_{ca}(\mathbf{p}_1, \mathbf{p}_2) = \begin{cases} \eta (d_{\mathbf{p}} - d_B)^2 & d_{\mathbf{p}} < d_B \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Here, d_B acts as a distance threshold below which the force field becomes active and η is a gain parameter. The gradient of this cost function directly serves as a joint-level control target and can be easily computed in terms of the body point Jacobians $J_{\mathbf{p}_i}$ by applying the chain rule:

$$\dot{\mathbf{q}}_{ca} = -\nabla_{\mathbf{q}}^t H_{ca} = -2\eta (1 - d_B/d_{\mathbf{p}})(J_{\mathbf{p}_1} - J_{\mathbf{p}_2})^t (\mathbf{p}_1 - \mathbf{p}_2). \quad (7)$$

Thus we yield straight-line task-space motions (e.g. of the end-effector in Cartesian space), while the redundancy is exploited to circumvent obstacles as schematically shown in Fig. 1, left. To allow more flexible obstacle avoidance, Behnisch [17] proposed a relaxed motion control scheme, which allows deviations from straight-line motions, if the robot gets too close to obstacles:

$$\dot{\mathbf{q}} = J^+ (\dot{\mathbf{x}} - \beta \dot{\mathbf{x}}_{ca}) - N(\nabla H_{ca} + \nabla H_{\mathbf{q}}). \quad (8)$$

Here, additionally to the null-space motion, which minimizes a superposition of both cost functions $H_{\mathbf{q}}$ and H_{ca} , an obstacle avoidance motion $\dot{\mathbf{x}}_{ca}$ is directly allowed in task-space as well. This contribution is determined by projecting the cost gradient (7) to the task space:

$$\dot{\mathbf{x}}_{ca} = J \nabla_{\mathbf{q}}^t H_{ca}. \quad (9)$$

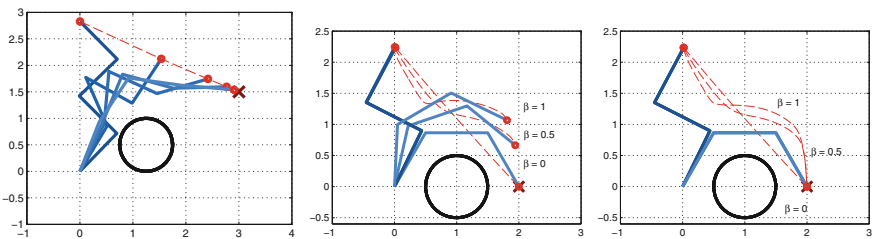


Fig. 1 Goal-directed task-space motion with collision avoidance. *Left* restricting avoidance motions to redundant space yields a *straight line* motion of the end-effector. *Middle* using relaxed motion control (8), the trajectory more strongly avoids the obstacle for larger weights β , but does not converge to the target anymore. *Right* dynamic adaption of β achieves both goals, target reaching and obstacle avoidance

Choosing different values of the weight β , we can smoothly adjust the importance of collision avoidance and target reaching as shown in Fig. 1, middle. However, because both contributions might be conflicting, the target is not always reached. To prevent this, we can ensure, that the goal-directed motion always dominates the collision avoidance motion with a margin ε by dynamically adapting β , such that the following condition is fulfilled:

$$\|\dot{\mathbf{x}}\| - \varepsilon \geq \beta \|\dot{\mathbf{x}}_{ca}\|. \quad (10)$$

The resulting motion is shown in Fig. 1, right. Please note, that this approach—is a local method—is prone to get stuck in local minima, if a straight target-reaching motion is not collision-free. To avoid this failure, a deliberative planning method at a global level is required. To this end, Behnisch [17] proposed to augment the local motion generation with a globally acting, sampling-based planning method, that, however, searches within the low-dimensional task-space instead of the full joint space. This sharing of workload between a local, reactive planner and a global, deliberative planner turned out to be very successful and computationally efficient.

2.2 Vision-Based Grasp Selection

Employing the outlined control basis framework to realize approaching motions for grasping and exploiting the passive compliance of modern, often underactuated hands [18, 19], grasp planning is extremely simplified: The fingers will automatically wrap around the object due to the inherent compliance of the hand. Thus, the only task for grasp planning is to choose a suitable grasp prototype and to align the hand to the object during the approach phase.

As already observed by Cutkosky, humans employ only a very small number of grasp postures that can be roughly separated into power and precision grasps. Cutkosky's taxonomy then further subdivides grasps by the shape of the object [20]. From our experience it suffices to use the three basic grasp prototypes shown in Fig. 2 (power, precision, and pincer grasp). To chose an appropriate grasp for a given

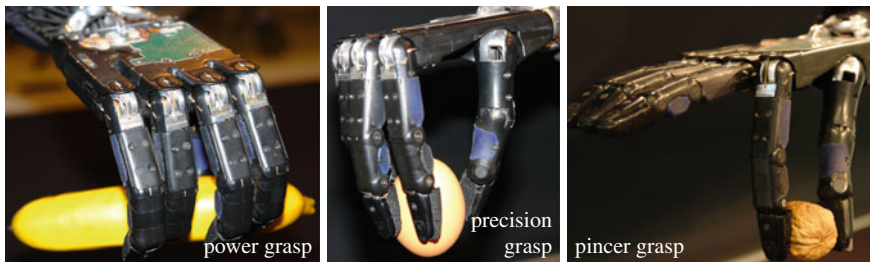


Fig. 2 Three basic grasp prototypes used for the Shadow Dexterous Hand. Depending on object size, estimated weight, and envisioned manipulation task we choose from a power grasp, a precision grasp, and a pincer grasp (*left to right*)

object, we employ a real-time, model-free scene segmentation method [21], which yields individual point clouds for all objects within the scene. Into each point cloud, a superquadrics model is fitted that captures the coarse shape of the object, smoothly varying between sphere, ellipsoid, cylinder, and box [22]. This model provides an estimation of the position and orientation as well as the coarse size and shape of the object. This information is utilized on the one hand to choose the grasp prototype and on the other hand to setup an appropriate approaching controller, utilizing the symmetries inherent to all recognized object shapes. A video illustrating the segmentation capabilities and the achieved grasping skills is available at youtube [22].

3 Tactile Servoing

In order to extend traditional grasp and manipulation planning approaches beyond a mere trajectory-centric view towards robust closed-loop controllers also integrating multi-modal feedback from proprioception, vision, and tactile sensing, in the following we discuss how the control basis framework (CBF) can be augmented by tactile servoing controllers. The main idea of these controllers is to define an inverted task Jacobian J_s^{-1} that directly maps errors in the tactile feature vector onto a suitable Cartesian velocity twist \mathbf{V}_s of the sensor frame. Subsequently we employ the power of CBF [23] to realize the computed sensor frame motion with appropriate joint motions. However, before looking into the details of these control primitives, we first review some recent developments in tactile sensing and discuss, which tactile features can be extracted from latest tactile sensing arrays.

3.1 High Resolution Tactile Sensing

In the past decades tactile sensors were developed exploiting a variety of physical principles—ranging from piezo-resistive or capacitive to optical or ultrasonic effects (cf. Dahiya et al. [24] for a compact review). The BioTac[®] sensor can be considered a breakthrough in tactile sensing, integrating high-frequency temperature and pressure sensing with a grid of electrodes to resolve the point of contact as well as normal and shear forces [25]. Analyzing high-frequency vibrations induced by slip-stick transitions, the sensor is able to detect incipient slippage and to distinguish various materials showing characteristic vibration patterns [26, 27].

Independently, Schürmann et al. developed a modular sensor design tailored towards high-frequency sensing for slip detection too, but also providing a high spatial resolution for normal force sensing (on an array of 16×16 tactsels spaced at 5 mm) at the high frame rate of 1.9 kHz. Employing a multilayer perceptron network, trained to predict slip velocities from Fourier coefficients of the tactile time series, they were able to adjust the required grasping force to stably hold an object

without knowledge about its weight or friction properties: Every time, when incipient slippage is detected, the grasping force is increased by a fixed amount. Otherwise, it is exponentially decaying to minimize the applied contact forces [28].

While these two sensors provide excellent sensitivity to high-frequency, small amplitude vibrations, they are both rather bulky and not suited to be integrated into human-sized robotic fingertips. Although there exists an adaptation of the BioTac[®] sensor to the anthropomorphic Shadow Dexterous Hand[™] [29], this integration design removes the distal finger joint, which is important in various manipulation tasks. Utilizing a new technology to realize 3D-shaped PCBs, Zenker et al. [30] miniaturized the tactile sensor array, integrating 12 tactels and the measurement electronics within a fingertip-shaped sensor-electrode that exactly matches the size of the robotic fingertip (cf. Fig. 3).

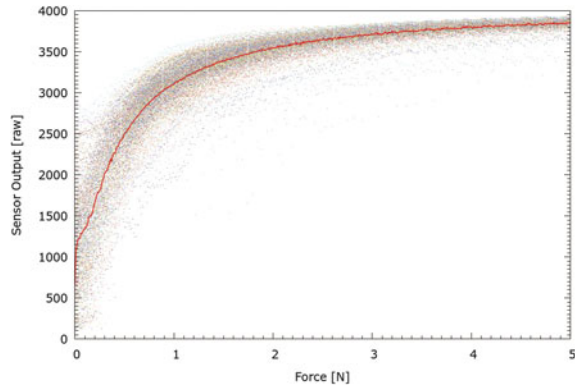
All these sensors are rather rigid and thus not suitable to be worn by a human. In order to measure interaction forces between the human hand and a manipulated object too, a more flexible sensor hardware is required. A first approach into this direction is the tactile glove developed by Büscher et al. [31] which is composed from conductive and piezo-resistive fabrics layers. In contrast to previous attempts to measure interaction forces, utilizing instrumented objects [32, 33], the sensorized glove allows to measure tactile interaction patterns with arbitrary objects. Its low construction height as well as the flexibility and stretchability of the fabrics, make this sensor concept well suited to cover larger parts of robots too, e.g. to yield a tactile-sensitive skin.

Given their high data frame rates, all sensor designs open up the opportunity to be employed for closed-loop robot control, thus for the first time offering large-scale reactivity to touch comparable to human sensitivity. Looking into the literature, only a very few approaches exist that directly utilize tactile sensor information for control, e.g. a very early [34] or a more recent one [35] on tactile contour tracking. However, a generic tactile servoing framework allowing to achieve a multitude of tasks from the composition of simple, basic controllers is missing so far. In the following sections we will get a glimpse on the enormous potential that can be unleashed when combining concepts from the control basis framework with tactile sensor information, thus lifting grasping and manipulation skills for robots to the next level of robustness and dexterity.



Fig. 3 Recent tactile sensors from Bielefeld University. From *left to right* a modular, flat 16×16 tactile sensor array, a 3D shaped tactile fingertip suitable for the Shadow Robot Hand, and a flexible tactile glove manufactured from conductive fabrics

Fig. 4 Sensor characteristics of all 256 tactels (and an individual one—*red solid line*) as acquired on a calibration bench



3.2 Feature Extraction from Tactile Images

Many tactile sensor designs propose an array of tactile sensing elements (tactels) providing *normal force* information [6, 36, 37] for each element. Sometimes it is also possible to compute *contact force directions* from this information [37]. Most array structures also have a reasonable spatial resolution to allow for an explicit control of the tactile force pattern sensed in a contact region. As a consequence, in our control framework, we assume the availability of a tactile sensor array providing a tactile image of normal force values measured by individual tactels.

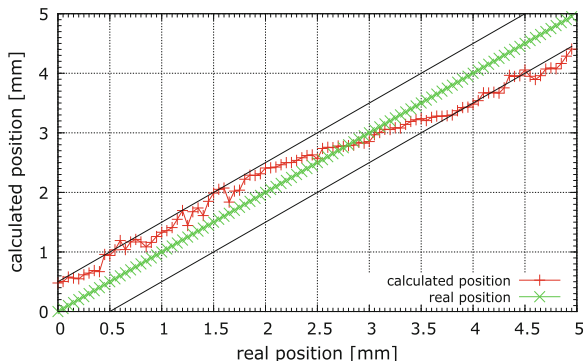
Particularly, the device employed in our experiments is the 16×16 sensor array depicted in Fig. 3, left. This sensor exploits the piezo-resistive sensing principle, measuring changes in resistance of a conductive foam due to an applied force. The analog measurement of each individual tactel is converted to a 12bit digital value covering a pressure range of 0.1–10kPa.¹ Due to varying local conductive properties of the foam, every tactel has a distinguished, squashed and noisy sensor characteristics as shown in Fig. 4. To obtain a coarse force calibration, the characteristic measurement function of each individual tactel is inverted in its linear range.

The intended tactile servoing tasks aim for controlling (a) the contact position on the fingertip, (b) the contact force, and (c) the orientation of an object edge relative to the sensor array. Hence, we propose feature extraction methods to provide the current values of these control variables.

As a first processing step, the contact region on the sensor is identified, which typically extends over several tactels due to the softness of the sensor foam. To this end, we employ connected component analysis [38], well known from image processing, to extract all connected regions in the binarized tactile image and choose the largest one as the considered contact region R —neglecting all smaller regions as originating from noise or spurious contacts. The binarization threshold is chosen rather small, just above the noise level, to consider as much tactile information as

¹The sensor’s sensitivity and force range can be adjusted to the task. Here, we have chosen the characteristics to provide a linear range from 0.1–1 kPa.

Fig. 5 Estimated (*red*) and expected (*green*) contact position (COP) of a 2 mm-diameter probe tip



possible. Subsequently, the overall contact (normal) force f is determined as the sum of forces f_{ij} within the contact region and the contact position \mathbf{c} as the force-weighted center of pressure (COP) of R :

$$f = \sum_{ij \in R} f_{ij} \quad \mathbf{c} = f^{-1} \sum_{ij \in R} f_{ij} \mathbf{c}_{ij}, \quad (11)$$

where c_{ij} are the discrete coordinates of the tactels on the sensor surface. Due to the averaging effect from multiple tactels composing a contact region, we obtain a sub-tactel resolution for the contact position as illustrated in Fig. 5. In this experiment a probe tip, 2 mm in diameter, was moved across the sensor from one tactel to another, i.e. about a distance of 5 mm. At every point, the estimated and real probe position (obtained from the robot's end-effector pose) are compared.

Usually, we want to control the contact pressure instead of the overall contact force. Considering manipulation of fragile objects, like an egg, it is the local pressure that should be limited to not damage the object. To obtain a pressure value, we normalize the overall measured force by the size of the contact region (measured as the number of pixels in R):

$$p = \frac{f}{|R|} \quad (12)$$

To extract the orientation of an object edge that maps onto a line-shaped contact region, we utilize the Hough transform, also well known from image processing [39].

3.3 Tactile Control Primitives

The proposed tactile servoing controller aims at realizing sliding and rolling motions about the contact point while maintaining a specified normal contact force during manipulation. Dependent on the actual task at hand, specific control primitives can be selectively turned on or off. Additionally to this purely tactile-driven motion,

an external task planner can provide a motion component $\mathbf{V}_s^{\text{ext}}$, which is a twist expressed in terms of the sensor frame O_s . This motion component allows to realize externally controlled tactile object exploration, e.g. to follow an object edge or to run the sensor over the whole object surface as detailed in the experimental Sect. 4.

The general control scheme of our proposed controller is depicted in Fig. 6. The control cycle starts by computing the deviation of the current tactile feature vector \mathbf{f} from the targeted one. This error is fed into PID-type controllers, acting independently on all feature-error components. The resulting control variable \mathbf{u} is a linearly transformed version of $\Delta\mathbf{f}$. Please notice, that for effective force control a non-zero integral component is required to compensate for static errors caused by a pure P-controller. Additionally, the derivative component is necessary to suppress undesired oscillations.

Subsequently, we compute a sensor motion $\mathbf{V}_s^{\text{tact}}$ aiming to reduce the feature error. This is realized with a fixed, task-independent, inverted Jacobian matrix J_s^{-1} . Both entities are expressed in terms of the sensor coordinate frame O_s , which is located in the center of the sensor surface and aligned with the sensor such that the z -axis equals the surface normal. This choice tremendously facilitates the determination of J_s^{-1} , which maps feature errors onto sensor motions.

The subsequent application of a task-dependent projector matrix P selecting certain twist-components for control and neglecting others, allows to selectively switch on or off specific motion components. To this end, P is a simple 6×6 diagonal matrix, where ones and zeros indicate, that the corresponding twist component is or is not used for control. Summarizing, the feedback-part of the tactile servoing controller is determined by the following equation:

$$\mathbf{V}_s^{\text{tact}} = P \cdot J_s^{-1} \cdot \left(K_P \cdot \Delta\mathbf{f}(t) + K_I \cdot \int \Delta\mathbf{f}(t)dt + K_D \cdot (\Delta\mathbf{f}(t) - \Delta\mathbf{f}(t-1)) \right). \quad (13)$$

Here $\mathbf{V}_s = [\mathbf{v}_s, \boldsymbol{\omega}_s]$ denotes the 6-dimensional twist vector composed of linear and angular velocity components $\mathbf{v}_s, \boldsymbol{\omega}_s$. $K_{P,I,D}$ denote diagonal matrices of PID-controller gains and $\Delta\mathbf{f}(t) = [\Delta x_s, \Delta y_s, \Delta f, \Delta\alpha]$ denotes the deviation of the feature vector composed of the positional error $\Delta x_s, \Delta y_s$, the normal force error Δf , and the

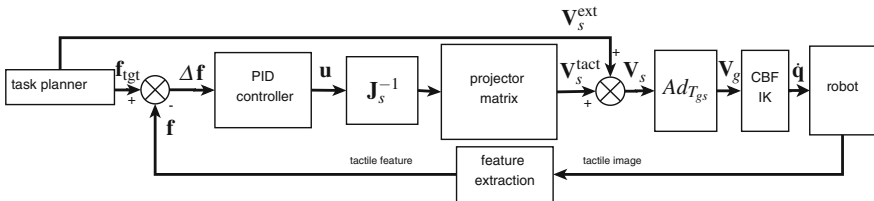


Fig. 6 Control scheme for tactile servoing: the core feedback part computes a sensor motion $\mathbf{V}_s^{\text{tact}}$ from tactile-feature deviations $\Delta\mathbf{f}$, which is superimposed with an external motion signal $\mathbf{V}_s^{\text{ext}}$ and subsequently fed into CBF's inverse kinematics

angular error $\Delta\alpha$ of the line orientation. Note, that the latter one is measured modulo π in order to obtain angular errors in the range $(-\frac{\pi}{2}, \frac{\pi}{2}]$ and thus circumventing singularities due to their circular nature. The rotational symmetry allows to restrict the errors to this range instead of $(-\pi, \pi]$.

Finally, the twists originating from the tactile feedback-loop and the external task planner are superimposed and fed to the inverse kinematics module of the control basis framework. To this end, the twist \mathbf{V}_s expressed in terms of the sensor frame O_s needs to be transformed to the global frame O_g , which is realized by the adjoint matrix derived from the forward kinematics $T_{gs} = (R_{gs}, \mathbf{p}_{gs})$:

$$Ad_{T_{gs}} = \begin{pmatrix} R_{gs} & \hat{\mathbf{p}}_{gs} R_{gs} \\ \mathbf{0} & R_{gs} \end{pmatrix} \quad (14)$$

At the core of the tactile-feedback controller is the inverse Jacobian that maps feature deviations onto a motion twist of the tactile sensor array:

$$\mathbf{V}_s^{\text{tact}} = J_s^{-1} \cdot \Delta \mathbf{f} = \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix} \cdot \begin{pmatrix} \Delta x_s \\ \Delta y_s \\ \Delta f \\ \Delta \alpha \end{pmatrix} \quad (15)$$

This matrix can be easily determined in the sensor coordinate frame O_s : Positional deviations are simply mapped onto corrective tangential motions in the x - y -plane of the sensor. Normal force errors are mapped onto a corrective translational motion along the z -axis of the sensor frame, which is normal to the sensor plane, pointing towards the object. These linear motion components are determined by the first three rows of J_s^{-1} . The rotational error $\Delta\alpha$ is mapped onto a rotational velocity around the z -axis (last row). The motion components corresponding to the fourth and fifth row of the inverted Jacobian realize a rolling motion of the sensor. These are triggered by positional deviations again. Thus, an error Δx_s is not only reduced by an appropriate tangential linear motion of the sensor, but also by a rolling motion around the y -axis of the sensor, that also moves the COP of the contact region closer towards its target location.

The task-dependent projector matrices P can be used to toggle these individual twist components on and off. For example, if contact position control is desired, one will choose $P = \text{diag}(1, 1, 0, 0, 0, 0)$. When additionally force control is required, the third diagonal entry should be set to 1 too. In order to enable or disable the orientation tracking of an object edge, you will set the last diagonal entry to 1 or 0 respectively. Finally, the fourth and fifth entries in the diagonal projector matrix determine, whether rolling is enabled or not. In the following section, we will discuss several application scenarios of the proposed tactile-servoing framework.

4 Experimental Evaluation

As shown in Fig. 7, we mounted the tactile sensor pad as a large fingertip to a 7-dof Kuka lightweight robot arm operated in joint-space compliance mode. The control basis framework maps Cartesian-space twists into joint-angle velocities, thus changing the equilibrium posture of the robot controller. The tactile sensor pad provides an array of 16×16 tactels measuring contact forces with 12bits resolution [6]. The sampling frequency of the tactile sensor as well as the control cycle frequency of the robot arm are set to 250Hz. We use manually tuned PID parameters for the tactile servoing controller.

All the experiments discussed in the following are also shown in a youtube video [40] and follow the same course: Initially the robot is moved to its working area, holding this posture until object contact is established. As soon as a pressure threshold is exceeded, the robot switches to a specific, previously determined tactile servoing task.

In order to reduce the noisiness of the feature signals, we apply a smoothing filter to both the force/pressure feature and the line orientation feature α . To this end, we average the ten most recent measurements, i.e. in a time window of 40ms. The position feature is smooth enough due to the averaging of Eq. (11).

4.1 Tracking Contact Points

Contact point tracking has an important application for multi-finger grasping and manipulation. In both cases, fingers need to maintain object contact with a given contact force and they should ensure, that the contact location remains on the fingertip area—optimally in its center—to avoid slipping off. Consequently, the task-dependent projector matrix has the form $P = \text{diag}(1, 1, 1, 0, 0, 0)$ enabling contact position and force control.

Fig. 7 Experimental setup: tactile sensor mounted on Kuka LWR

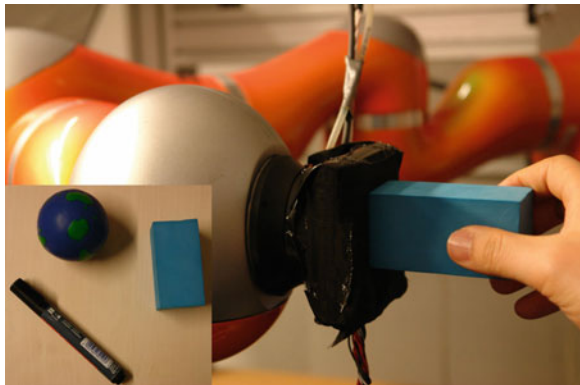


Table 1 Statistical tracking results for force and position control

Object	Steady state error	Standard deviation	Response time
Rigid pen	0.0032	0.039	2.5 s
Toy box	0.0026	0.039	2.0 s
Soft ball	0.0010	0.043	2.0 s
X	0.0041 pixel	0.1146 pixel	1.8 s
Y	0.0082 pixel	0.1158 pixel	1.8 s

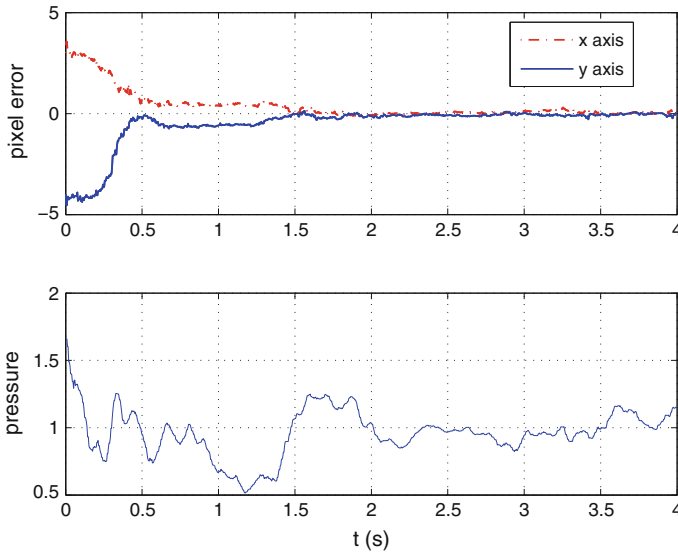


Fig. 8 Tracking results for combined position and force control

Please notice, that the quality of force control depends on the stiffness of objects (softer objects allow for a larger motion range given a fixed force range). We evaluated the control performance on various objects of different stiffnesses: a rigid pen, a toy box from stiff foam, and a soft ball. The results for maintaining a desired pressure level of $p = 1$ are shown in Table 1. As expected, stiffer objects take longer to converge to a stable tracking result (response time) and exhibit stronger force oscillations given similar deflections. However, in all cases the desired force level will eventually be well maintained with a small steady state error²

For contact position tracking, the goal is to maintain the COP of the contact region at the center of the tactile sensor frame. The evolution of the errors in contact position and force are shown in Fig. 8. As can be seen from the top sub-figure an initial position offset is corrected within half a second. The steady state error and

²The steady state errors and standard deviations are computed from a time series of 20 s duration starting after convergence (response time). All values are obtained by averaging over 20 trials.

response time are summarized in Table 1. As can be seen from bottom sub-figure the normal force applied in this experiment evolves randomly as it is not controlled. Note, that a large normal force—due to friction—will also cause large tangential forces, rendering the sliding motion more difficult. Hence, normal force control should be generally enabled.

4.2 Track Contact Point and Increase Contact Area by Rolling

The fourth and fifth row of the task Jacobian (Eq. 13) provide another mode of operation to compensate for positional errors of the COP: Instead of realizing a translational sliding motion, this control behavior realizes a rolling motion, thus changing the contact point both on the tactile sensor and the object’s surface. While previous approaches to realize rolling employed complex algorithms to determine the point of revolution and a corresponding joint-space robot motion [41], the tactile servoing approach proposed here, is conceptually much easier: a deviation in contact position is simply mapped to a rotational twist within the tangential plane of the sensor. Because we do not explicitly compute the point of revolution and do not know the shape of the object, the normal force will probably be disturbed due to this motion. However, the normal force controller, running in parallel, will counteract and maintain a predefined force level. The employed projector matrix equals $P = \text{diag}(1, 1, 1, 1, 1, 0)$, i.e. simultaneously realizing sliding and rolling as well as force control.

The resulting rolling motion is visualized in Fig. 9. An initial positional offset along the y -axis is compensated by a rolling motion about the sensor’s x -axis

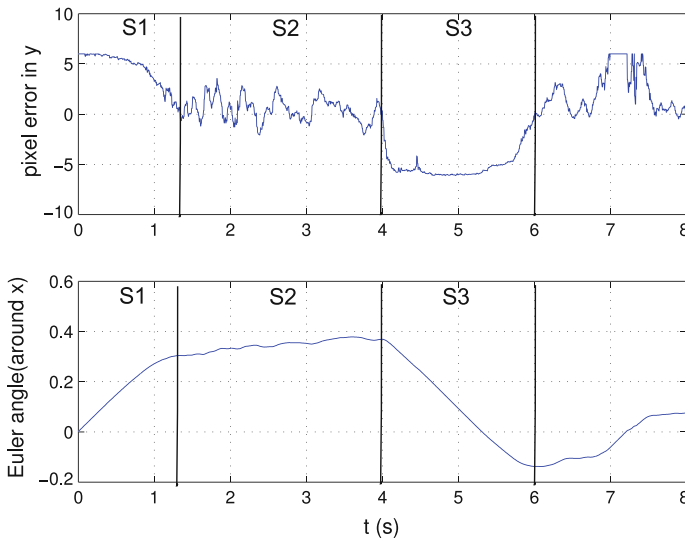


Fig. 9 Orientation control of surface normals by rolling

(stage S1). When the contact point error decays, the rolling motion ceases as well (stage S2). After 4 s the object was displaced yielding a negative position offset that was compensated by a rolling motion into the opposite direction (stage S3). This behavior can nicely be seen in the video [40] as well.

The rolling behavior has the beneficial side-effect of increasing the area of contact between the finger tip and the object. This is an important capability for grasp stabilization. Although classical grasp planning considers point contacts only, a large contact area naturally increases the grasp wrench space and thus increases the ability to resist to external disturbances. Furthermore, a prerequisite for successful tactile object exploration will be to maintain a large contact area during exploration in order to collect as much shape information about the object as possible.

How this side effect is achieved? Assuming large object and sensor surfaces, a small contact area typically results from a badly tilted sensor w.r.t. the object surface. In this situation the sensor only touches an object edge instead of the whole surface. This contact is often located off-center on the sensor array. The corrective rolling motion to move the COP into the sensor's center will also reduce the tilting and eventually result in the desired surface contact. This state also constitutes a fixed point of the controller dynamics, because the COP will be in the center of the tactile array in this case.

4.3 Tracking an Object Edge on the Sensor Surface

The orientation around the normal axis is controlled using the orientation angle α of a line in the tactile image emerging from an object edge on the sensor. For this control task the last row of the Jacobian matrix is important, resulting in a projector matrix $P = \text{diag}(0, 0, 1, 0, 0, 1)$. The tracking result for this experiment is qualitatively shown in the video [40] only. However, the next experiment also employs this control primitive and provides an evaluation in Fig. 10.

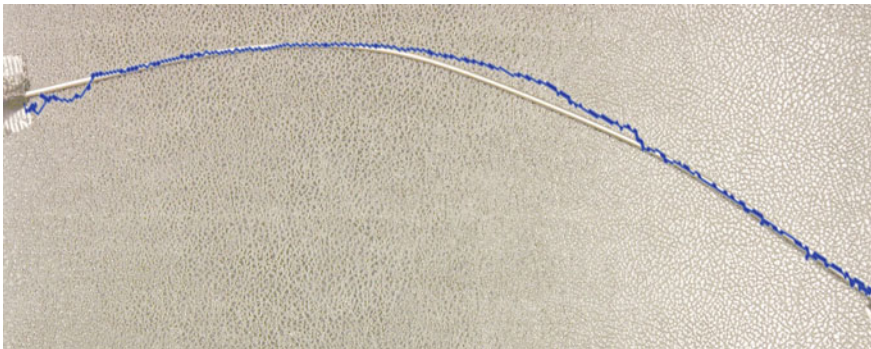


Fig. 10 Tracking of a cable of unknown shape: tracking result is superimposed onto a scene photo as a blue trajectory

4.4 Tracking of an Unknown Object Edge

The previous experiments illustrated the performance of the proposed tactile servoing controllers in various scenarios, neglecting external motion commands $\mathbf{V}_s^{\text{ext}}$. However, the aim of the following two tasks is to illustrate, that complex exploration behavior emerges if the tactile servoing motion and some externally provided guidance motion are combined.

In the first experiment, we consider the task of tracking the unknown shape of a cable lying flat on the table. To this end, the sensor should (i) be aligned to the local orientation of the cable, (ii) maintain the tactile imprint within its sensor boundaries (optimally in the center), and (iii) actively control the contact force. Accordingly we choose a projector matrix $P = \text{diag}(1, 1, 1, 0, 0, 1)$ selecting those subtasks. In order to follow the cable in space, we additionally impose an external tangential motion onto the sensor along its y -axis, which coincides with the desired orientation of the cable. Thus $\mathbf{V}_s^{\text{ext}} = [0, 1, 0, 0, 0, 0]^T$.

Figure 10 shows a photo of the tracked cable superimposed with the object shape (blue line) estimated from the forward kinematics of the robot arm when tracking the cable with tactile servoing. After some initial oscillations, the robot manages to align the cable imprint on the sensor with its y -axis.

4.5 Exploring the Shape of an Unknown Object

The second experiment illustrating the power of the proposed tactile servoing framework, aims at tactile object exploration: The sensor should slide over the unknown surface of the object in order to accumulate a dense shape model. Lacking an appropriate control framework, previous work acquired the corresponding tactile information by repeated establishment and breaking of object contact [42].

To realize this complex task, we decompose it into several phases: after establishing contact to the object, the robot maximizes the sensed contact area and aligns its y -axis with the major axis of the contact region applying the control schemes of Sects. 4.2 and 4.3 simultaneously.

Subsequently, by imposing a tangential motion along the sensor's x -axis (orthogonal to the major axis of contact region), we induce the exploratory motion. The tactile servoing controller maintains the optimal orientation and position of the tactile sensor on the object's surface by generating appropriate sliding and rolling motions. This task exploits all tactile servoing behaviors employing the projector matrix $P = \mathbf{1}$. As a result the object exploration behavior emerges automatically.

Similarly we can explore the object along the other direction, if we follow the contact's major axis instead (cf. previous task in Sect. 4.4). Please notice, that in the accompanying video [40] we change the direction of the external guidance motion $\mathbf{V}_s^{\text{ext}}$ in order to realize a scanning of the object into both directions. Figure 11 shows, how this exploration behavior can be utilized to construct an object shape estimation by touch.

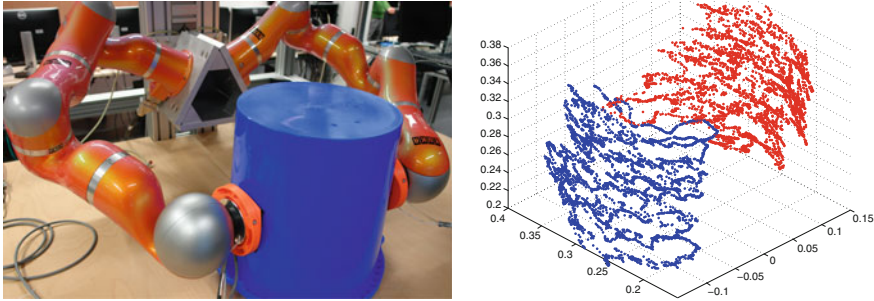


Fig. 11 Tactile object exploration using two tactile sensor arrays mounted onto Kuka LWR arms (*left*) and the resulting tactile point cloud as a local estimation of object shape (*right*)

5 Conclusion

The introduced tactile servoing control framework allows to realize a large range of tactile tracking and exploration tasks. To this end, it's only necessary to choose the task-specific projector matrix P choosing which tactile servoing primitives (sliding, rolling, turning, force control) should be applied.

The integration of an externally driven guidance motion $\mathbf{V}_s^{\text{ext}}$ allows to realize complex exploratory behavior. In the shown example tasks, we only used very simple, static guidance motions. However, if those guidance motions are computed from tactile feedback as well, one can easily realize even more complex exploration behavior, e.g. to drive the tactile sensor towards interesting spots on the object's surface, like ridges, edges or corners.

As you have seen, the formulation of tasks as a clever chosen set of primitive controllers relaxes the need for explicit planning and modeling to a large extend, such that both grasping and manipulation tasks become feasible also for unknown objects. Such situations frequently occur in unstructured human environments, like homes or hospitals, which are natural environments for service robots.

References

1. Jenmalm P, Johansson RS (1997) Visual and somatosensory information about object shape control manipulative fingertip forces. *J Neurosci* 17:4486–4499
2. Johansson RS, Westling G (1984) Roles of glabrous skin receptors and sensorimotor memory in automatic control of precision grip when lifting rougher or more slippery objects. *Exp Brain Res* 56:550–564
3. Steffen JF, Elbrechter C, Haschke R, Ritter H (2010) Bio-inspired motion strategies for a bimanual manipulation task. In: *Proceedings of international conference on humanoid robots*
4. Dang H, Weisz J, Allen PK (2011) Blind grasping: stable robotic grasping using tactile feedback and hand kinematics. In: *Proceedings of ICRA*
5. Ho V, Nagatani T, Noda A, Hirai Sh (2012) What can be inferred from a tactile arrayed sensor in autonomous in-hand manipulation? In: *Proceedings of CASE*, p 461

6. Schürmann C, Kőiva R, Haschke R (2011) A modular high-speed tactile sensor for human manipulation research. In: World haptics conference
7. Li Q, Haschke R, Bolder B, Ritter H (2012) Grasp point optimization by online exploration of unknown object surface. In: Proceedings of international conference on humanoid robots
8. Pezzementi Z, Plaku E, Reyda C, Hager GD (2011) Tactile-object recognition from appearance information. *Trans Robot* 27(3):473–487
9. Hart S, Sen S, Ou S, Grupen R (2009) The control basis API—a layered software architecture for autonomous robot learning. In: 2009 workshop on software development and integration in robotics at ICRA
10. Huber M (2000) A hybrid architecture for adaptive robot control. PhD thesis, University of Massachusetts
11. León B, Ulbrich S, Diankov R, Puche G, Przybylski M, Morales A, Asfour T, Moio S, Bohg J, Kuffner J (2010) OpenGRASP: a toolkit for robot grasping simulation. In: Proceedings of SIMPAR. Springer, Darmstadt, pp 109–120
12. Gienger M, Toussaint M, Goerick C (2010) Whole-body motion planning—building blocks for intelligent systems. In: Harada K, Yoshida E, Yokoi K (eds) Motion planning for humanoid robots. Springer, London, pp 67–98
13. Gienger M, Janßen H, Goerick C (2006) Exploiting task intervals for whole body robot control. In: Proceedings of IROS, pp 2484–2490
14. Platt R, Fagg AH, Grupen RA (2010) Null-space grasp control: theory and experiments. *IEEE Trans Robot* 26(2):282–295
15. Liegeois A (1977) Automatic supervisory control of configuration and behavior of multibody mechanisms. *IEEE Trans Syst, Man Cybern* 7(12):861–871
16. Sugiura H, Gienger M, Janssen H, Goerick C (2010) Reactive self collision avoidance with dynamic task prioritization for humanoid robots. *Int J Humanoid Robot* 7(01):31–54
17. Behnisch M, Haschke R, Ritter H, Gienger M (2011) Deformable trees—exploiting local obstacle avoidance. In: Proceedings of international conference on humanoid robots
18. Catalano MG, Grioli G, Farnioli E, Serio A, Piazza A, Bicchi C (2014) Adaptive synergies for the design and control of the Pisa/IIT SoftHand. *Int J Robot Res* 33(5):768–782
19. Odhner LU, Ma RR, Dollar AM (2013) Open-loop precision grasping with underactuated hands inspired by a human manipulation strategy. *IEEE Trans Autom Sci Eng* 10(3):625–633
20. Cutkosky M, Howe RD (1990) Human grasp choice and robotic grasp analysis. In: Venkataraman ST, Iberall T (eds) Dextrous robot hands. Springer, New York
21. Ückermann A, Haschke R, Ritter H (2013) Realtime 3D segmentation for human-robot interaction. In: Proceedings of IROS
22. Ückermann A, Haschke R, Ritter H (2012) Real-time 3D segmentation of cluttered scenes for robot grasping. In: Proceedings of international conference on humanoid robots. Video: www.youtube.com/watch?v=Z2SwggQTBC8
23. Schöpfer M, Schmidt F, Pardowitz M, Ritter H (2010) Open source real-time control software for the Kuka light weight robot. In: Proceedings of WCICA, pp 444–449
24. Dahiya RS, Metta G, Valle M, Sandini G (2010) Tactile sensing: from humans to humanoids. *IEEE Trans Robot* 26(1):1–20
25. Wettels N, Santos VJ, Johansson RS, Loeb GE (2008) Biomimetic tactile sensor array. *Adv Robot* 22(8):829–849
26. Fishel JA, Loeb GE (2012) Sensing tactile microvibrations with the BioTac—comparison with human sensitivity. In: International conference on biomedical robotics and biomechanics (BioRob), pp 1122–1127
27. Xu D, Loeb GE, Fishel JA (2013) Tactile identification of objects using Bayesian exploration. In: Proceedings of ICRA, pp 3056–3061
28. Schürmann C, Schöpfer M, Haschke R, Ritter H (2012) A high-speed tactile sensor for slip detection. In: Prassler E, Burgard W, Handmann U, Haschke R, Hägele M, Lawitzky G, Nebel B, Nowak W, Plöger P, Reiser U, Zöllner M (eds) Towards service robots for everyday environments, vol 76. Springer, New York, pp 403–415. Video: www.youtube.com/watch?v=mSq8e4PU90s

29. Shadow Robot Company. Shadow Dexterous Hand (2013). <http://www.shadowrobot.com/products/dexterous-hand>
30. Kōiva R, Zenker M, Schürmann C, Haschke R, Ritter H (2013). A highly sensitive 3D-shaped tactile sensor. In: International conference on advanced intelligent mechatronics (AIM)
31. Büscher G, Kōiva R, Schürmann C, Haschke R, Ritter H (2012) Tactile dataglove with fabric-based sensors. In: Proceedings of international conference on humanoid robots
32. Maycock J, Essig K, Haschke R, Schack T, Ritter H (2011) Towards an understanding of grasping using a multi-sensing approach. In: Proceedings of ICRA, pp 1–8
33. Roa M, Kōiva R, Castellini C (2012) Experimental evaluation of human grasps using a sensorized object. In: International conference on biomedical robotics and biomechatronics (BioRob)
34. Chen N, Zhang H, Rink R (1995) Edge tracking using tactile servo. In: Proceedings of IROS, vol 2. August 1995, pp 84–89
35. Martinez-Hernandez U, Lepora NF, Barron-Gonzalez H, Dodd TJ, Prescott TJ (2012) Towards contour following exploration based on tactile sensing with the iCub fingertip. In: Herrmann G, Studley M, Pearson M, Conn A, Melhuish C, Witkowski M, Kim J-H, Vadakkepat P (eds) Advances in autonomous robotics. Lecture notes in computer science, vol 7429. Springer, Berlin, pp 459–460
36. Schmitz A, Maiolino P, Maggiali M, Natale L, Cannata G, Metta G (2011) Methods and technologies for the implementation of large-scale robot tactile sensors. *Trans Robot* 27(3):389–400
37. Wettels N, Loeb GE (2011) Haptic feature extraction from a biomimetic tactile sensor: force, contact location and curvature. In: Proceedings of ROBIO, pp 2471–2478
38. Suzuki K, Horiba I, Sugie N (2003) Linear-time connected-component labeling based on sequential local operations. *Comput Vis Image Underst* 89(1):1–23
39. Duda RO, Hart PE (1972) Use of the Hough transformation to detect lines and curves in pictures. *Commun ACM* 15(1):11–15
40. Li Q (2013) A control framework for tactile sensing. Video: <https://www.youtube.com/watch?v=TcWipks3qJ0>
41. Schöpfer M, Ritter H, Heidemann G (2007) Acquisition and application of a tactile database. In: Proceedings of ICRA, pp 1517–1522
42. Meier M, Schöpfer M, Haschke R, Ritter H (2011) A probabilistic approach to tactile shape reconstruction. *Trans Robot* 27(3):630–635