

Improving robot manipulation through fingertip perception

Alexis Maldonado, Humberto Alvarez, Michael Beetz
{amaldo, humalvarez, beetz}@cs.uni-bremen.de

Abstract—Better sensing is crucial to improve robotic grasping and manipulation. Most robots currently have very limited perception in their manipulators, typically only fingertip position and velocity. Additional sensors make richer interactions with the objects possible. In this paper, we present a versatile, robust and low cost sensor for robot fingertips, that can improve robotic grasping and manipulation in several ways: 3D reconstruction of the shape of objects, material surface classification, and object slip detection. We extended TUM-Rosie, our robot for mobile manipulation, with fingertip sensors on its humanoid robotic hand, and show the advantages of the fingertip sensor integrated in our robot system.

I. INTRODUCTION

Our goal is to improve the reliability of robotic grasping and object manipulation by enhancing the sensing capabilities of robotic manipulators. Adding sensors to robot hands fills a widespread gap in humanoid and service robotics: As soon as a robot gets close and grasps an object, the view from the 'head' is heavily occluded by the hands and arms. Meaning that the best sensors available (cameras/range sensors) will not have a clear view of the object anymore, and the grasping and manipulation action will take place without good feedback information. The sensor that we propose in this paper attacks this problem in several ways.

Robots interact with the world using their manipulators, and there is a great range of them: from simple parallel-jaw grippers to complex humanoid hands with many degrees of freedom. Their basic function is to fixate objects within their 'fingers' by applying a certain amount of force, and letting friction do its job. An important limitation is that most of them only have proprioceptive sensing, usually the position and velocity of the fingers, so they can only indirectly gather information about the objects they are manipulating.

In this paper we propose a sensor that can be installed directly in the fingertips, and extend the perceptual capabilities of the hand to:

- 1) Reconstruct the 3D shape of objects, specially the occluded parts close to the chosen grasping area.
- 2) Recognize the surface texture of objects (quickly and accurately).
- 3) Detect slip of the object while manipulating (lifting/holding).

The authors are with the Intelligent Autonomous Systems / Artificial Intelligence chair, Computer Science Department, at the University of Bremen. We are also part of the Center for Computing and Communication Technologies (TZI), at the same University. We were previously at the Intelligent Autonomous Systems group, part of the Technische Universität München (TUM). The experiments described in this paper were done at TUM.

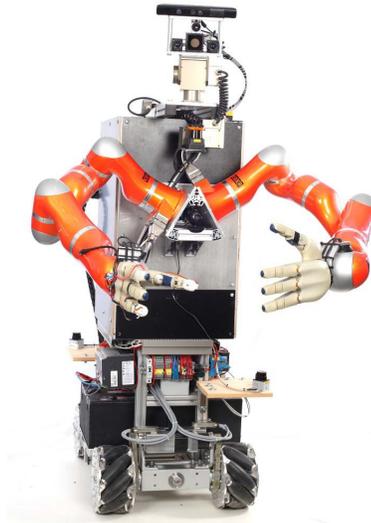


Fig. 1. TUM-Rosie, our mobile manipulation platform. Fingertip sensors installed in its right hand improve autonomous grasping, let it recognize object surfaces, and detect object slip.

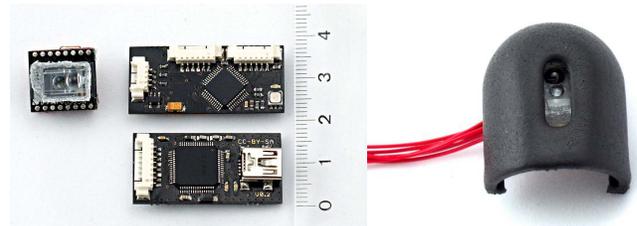


Fig. 2. System components (left image, clockwise): Fingertip sensor, SPI controller, FLOSS-JTAG. Right: A 3D-printed fingertip for the DLR/HIT Hand.

In order to realize those capabilities, we have designed a fingertip sensor (Fig. 2) that has the following basic modes:

- Proximity sensor: It can measure the distance to objects with a range of 1-10mm. This is the distance necessary for pre-touch sensing, that can be used for reconstructing the occluded shape of the object, or for pre-grasp finger positioning. This is particularly important for improving autonomous grasping.
- Surface image acquisition: The sensor transmits a 30x30pixel image from an area of the object of around 1 square mm. This data is used for surface recognition.
- Optical-flow measurement: The sensor can measure tangential displacements of the objects held in the fingers, and be used as a slip sensor.

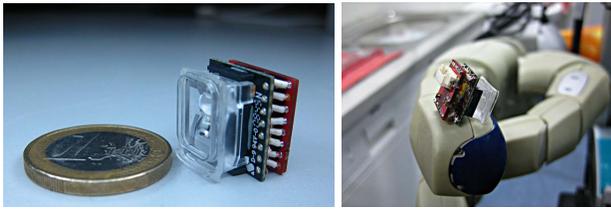


Fig. 3. Fingertip sensor size comparison, and the sensor on top of a fingertip of our DLR/HIT 4-Finger robot hand.

We have installed the fingertip sensors in our robot, TUM-Rosie (Fig. 1). In the following sections we show how they can be used to improve robotic grasping. Taking a typical pick and place scenario as an example: The robot will use the fingertip sensor’s functionality as described above. First, it can do a 3D shape reconstruction in order to more reliably grasp the object. As soon as the object is in the hand, surface images can be acquired to recognize the object’s surface texture and confirm its identity. Finally, during lifting and carrying of the object, the slip detection can alert of undesired movements of the object.

II. RELATED WORK

The idea of adding sensing to the manipulators is not new. In fact, in the early days of robotics, having sensing on the robot’s hand was a way to concentrate the limited computing resources to the place where the contact could happen. It was a type of data filter: Instead of observing a big area with a camera and processing a lot of data to find the position of the hand, it was better to have local sensors on the hand react to tactile or other input.

There was a lot of interesting work in tactile sensors, and how they could be used to gather information through contact and then to reconstruct object shapes[1]. Some years later there were already interesting overview papers[2], explaining principles of operation of many different sensors for manipulators: Tactile sensors, force/torque sensors, surface normal sensors, actuator effort sensors, and others. These ideas are still valid today. [3] is a recent reference of tactile sensing technologies for robots.

As computers became faster, it was possible to operate on bigger data sets, and there was a shift towards environment sensing. Initially this meant a lot of work in computer vision, and most recently in 3D sensors. As an example, today we see a lot of attention to algorithms that work on point clouds from real-time depth sensors which are even put together to make detailed maps of the environment. We are starting to come full circle, and the advantages of manipulator sensing become clear again. Thanks to miniaturization in electronics, and to more robust equipment, there are many new sensing modalities available for robot hands or grippers.

In this paper both approaches come together, as we use realtime 3D depth sensors to model the shape of an object, and fingertip sensors to reconstruct the missing information because of occlusions.

Recent work has shown the usefulness of pre-touch sensing. These systems use proximity sensors, typically based on



Fig. 4. Three fingertip sensors installed in the DLR/HIT hand (thumb, index and ring fingers).

capacitive[4] or infrared[5] principles, which are installed in a robot’s gripper. They are used to position the fingers correctly for grasping, detect interaction with humans, and to grasp objects that could not be detected precisely through other means. Also, a grasp controller based on human control schemes shows the advantages of tactile sensing in [6], including mechanisms to use grip force estimation and slip detection using pressure sensors on the PR2 robot.

We have also seen fast development in the field of robotic skin. Advanced designs[7] include multi-modal sensors, including infrared proximity sensors. There is still work being done on reducing the size of the skin units, and making them more reliable, but we expect to see a more widespread use. Designs using capacitive sensors [8] are already being used to cover fingertips and larger areas.

Also related are systems that include a stereo camera[9], or a depth sensor on the robot’s gripper. These systems deliver very rich point clouds from the environment in the close vicinity of the gripper, which has a very different point of view than the robot’s head. This is useful for dealing with occlusions. One disadvantage is that the arms are often not long enough to capture a view from behind an object.

As early as 2002, optical mouse sensors have been used for odometry in miniature robots [10], and their use as a slip sensor is a only natural progression. More recently, they have been used to measure slip for haptic applications during contact [11].

There are promising studies of surface texture recognition done by robots. [12] shows good results obtaining data from a metal pin stroking a surface. In [13], several machine learning algorithms are analyzed for surface texture recognition using data from an advanced biologically inspired finger.

III. SYSTEM OVERVIEW

TUM-Rosie is our mobile manipulation robot for human indoor environments. The robot has two KUKA LWR-4[14] lightweight robot arms, with DLR/HIT[15] hands as manipulators. Both arms and hands are state of the art devices, and have much better proprioceptive sensing capabilities than most other robots. The hands include 3-DOF torque sensing in each finger and report the complete state (position, velocity, torque) of all the joints with a high framerate. Additionally, in order to improve the capabilities of our system, we have installed three fingertip sensors (Fig. 4) in the robot’s right hand.

A. Fingertip sensor

The fingertip sensor is built around the Avago Technologies ADNS-9500[16], a high-end laser navigation sensor designed for computer mice. The ADNS-9500 is essentially a miniature 30x30pixel high-speed camera with an integrated laser emitter and a microcontroller. The collimated laser light makes minuscule cracks and surface characteristics visible to the camera, so that it can detect optical flow (motion) even on difficult surfaces like glass or porcelain. The internal registers and the image obtained by the on-board camera are accessible over a high-speed (2Mbps) SPI bus. This makes it possible to use the sensor as a simple optical proximity sensor (by looking at how much laser light is bouncing off the objects), as a miniature surface camera (for surface recognition), and as a slip sensor (measuring the displacement of the surface on the fingertips, like a regular computer mouse).

We developed the following components of the fingertip sensor system (Fig. 2):

- A miniature board (15x15mm) with support electronic components for the Avago ADNS-9500 sensor. Using this board, the sensor can be installed in small places, and only has one small connector for the SPI data bus and power.
- The SPI-controller: A microcontroller board that controls up to four ADNS-9500 sensors (one for each fingertip in our robot hand). This board is based on an STM32 72MHz microcontroller, and deals with the low level control of the Avago sensors.
- A slightly modified version of the FLOSS/JTAG, a high-speed USB to serial converter and JTAG programmer, for communication with the computer and easy programming of the firmware of the SPI-controller.

The fingertip sensor has been developed as a free software/free hardware project, so all the information necessary for building it is available online under free (as in freedom) licenses at this address: <http://toychest.in.tum.de/wiki/projects:fingertip>. This includes all the schematics, the circuit board designs, and the software.

The cost of the system is very low: It is possible to get all the components for a setup with four fingertip sensors, as needed for installation on humanoid robot hands, for approximately \$200 USD. We also used a 3D-Printing service to make the new fingertip housings, for a cost of around \$20 USD each. This has an excellent cost/benefit ratio, taking into account the additional capabilities given to the robot hand.

The size of each fingertip sensor is 15 x 15 x 16mm (See Fig. 3), small enough to be installed in the jaws of a parallel-jaw gripper, or the fingertips of many robot hands. The dimensions of the two other boards put together is 34.5 x 17 x 15mm.

The complete system communicates with the computer over a USB connection, which also provides the supply voltage.

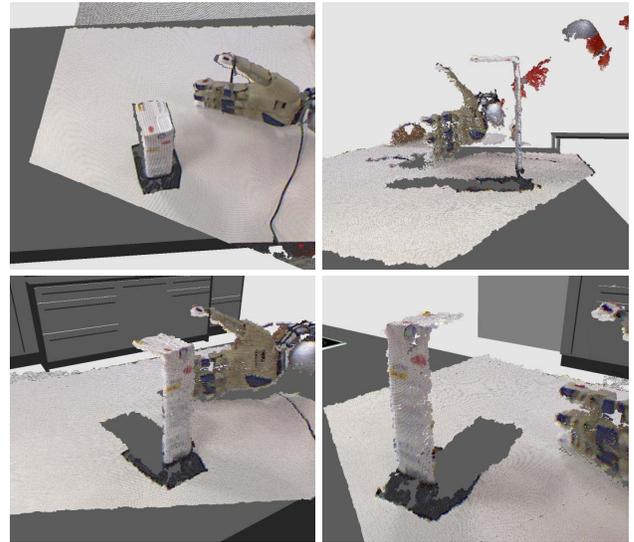


Fig. 5. TUM-Rosie's perception of a iced tea package on the kitchen counter. On the top left, the point of view of the robot is shown. The other images show the extension of the occlusion: the robot does not perceive any 3D points on the left side, right side, or the back of the object. This is a typical problem when using sensors mounted on the robot's head: the perspective is good to see a lot of the environment, but often only the frontal faces of the objects can be seen.

B. Integration into the robot hand of TUM-Rosie

We use ROS[17] as a communication middleware for integrating the different components of our robot, and the ROS nodes used for capabilities described in this paper are accessible as a part of the official TUM-ROS[18] repository.

The corresponding nodes stream data in a way that is easy to use for other ROS components. For example, the proximity sensing is combined with position information obtained from forward kinematics to generate a standard ROS 3D point cloud.

IV. 3D SHAPE RECONSTRUCTION: IMPROVING THE OBJECT SHAPE ESTIMATION FOR ROBOT GRASPING

The laser in the ADNS-9500 sensor emits infrared light at an angle, so that the maximum amount of light will be reflected when an object is at a certain distance from the sensor, and decreases as the object moves further away. It is possible to estimate the distance to the nearest object by evaluating the values of the internal registers of the ADNS-9500 sensor, specially the shutter speed and frame period of the internal camera. These values are adjusted continuously by the sensor, so that we can detect the presence of an object positioned 1-10mm front of the sensor reliably. This is the functionality required for its use as a proximity sensor. When such sensors are used in robotic grippers, they are sometimes known as pre-touch sensors.

We use the proximity information to reconstruct the shape of an object without touching it. Our robot has a RGB-D (Xbox KinectTM) sensor on its head that delivers high quality point clouds from objects, but they are often incomplete because of occlusions. The fingertip sensors complement our perception system for grasping by delivering shape

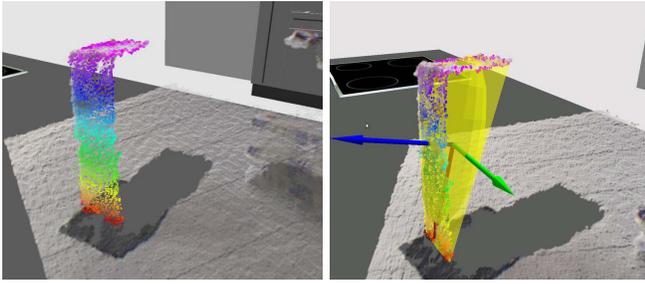


Fig. 6. Left: The point cloud corresponding to the iced tea package has been segmented and is shown in color. Right: The centroid and covariance of the points represent the object for use in grasp planning. The centroid is marked by the frame axes, and the ellipsoid represents the covariance. Note that the robot underestimates the size of the object considerably.

information from occluded areas, specially the ones around possible grasping locations.

A. The occlusion problem

Let us observe a rectangular iced tea package using the RGB-D sensor mounted on the robot’s head. Fig. 5 shows the point cloud data as it is perceived. The data looks complete from the point of view of the robot, but after looking at the points that represent the rectangular box from the side, the missing information becomes evident. The robot has basically no information from the sides or back of the box.

We use COP[19], [20], a software developed in our institute for robotic perception. One of the functions of COP is to search for point cloud clusters, which are segmented point clouds that belong to objects on the table. For this, it identifies the points belonging to a support plane (the table) in the complete point cloud obtained from the RGB-D sensor, and removes them. Now the point clouds belonging to objects on the table are left isolated from each other, and can be easily segmented. The result of this process can be seen in Figure 6. The point cloud representing the iced tea package has been correctly segmented and is shown in color on the left.

We use the simple-grasp-planning (SGP) library[21] to plan possible grasping positions for the robot. The SGP uses a simple description for the position and shape of the object, namely a 3D Gaussian distribution. This is just a point in space (x,y,z) describing the centroid of the object and the 3D covariance $(6 \times 6 \text{ matrix})$ describing the shape. Both are estimated from the segmented point cloud of the object. As can be seen in Figure 6 (right), the shape estimate of the object is wrong when large parts of the object’s surface are occluded.

A grasp pose suggested by SGP is shown in Fig. 7. After underestimating the size of the object (see Fig. 6), the robot will try to grasp the object from the side, thinking it is smaller than it really is, and will collide with the fingertips while doing so. Our robot can detect such collisions using torque sensors on the finger joints, and use a reactive grasping approach, so that it might still succeed in grasping, but it could push the object away and a re-detection would be

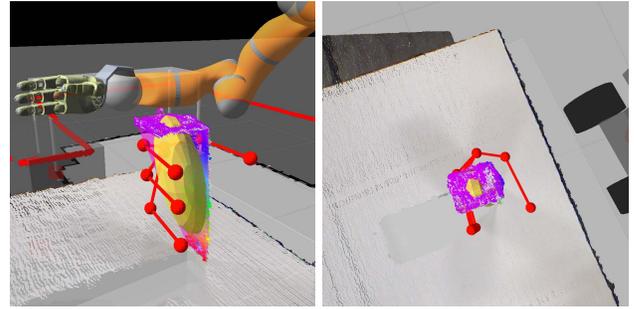


Fig. 7. A grasp planned by SGP based only on shape information from the RGB-D sensor, where the size of the object is underestimated. The representation of the hand is shown in dark red.

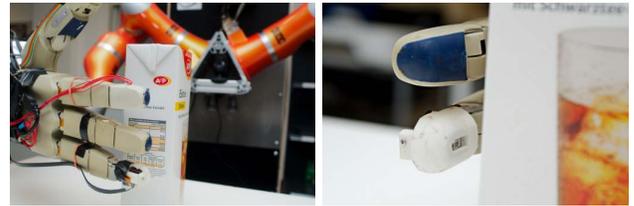


Fig. 8. The robot explores the shape of an object by moving the fingers close to it.

necessary. It is better to use the proximity sensors to avoid pushing the object in the first place.

B. Obtaining a point cloud using proximity data

Each time that an object is detected close to a fingertip sensor, we calculate the position of the detected point in space using forward kinematics. These points are continuously assembled into a point cloud that describes the shape of the surface seen by the fingertip.

Correct calibration of the robot’s arms and hands is necessary to reach a good spatial precision. Also important are a good time synchronization and high data rates for position streaming of all the involved joints.

In TUM-Rosie, we use a kinematic description of the robot that includes all the robot parts including the arms and hands. The position of the joints of the arms is reported at 1kHz, and the joints of the hands at 800Hz. The fingertip sensor streams proximity data at a rate of approximately 50Hz per sensor.

Our system calculates the forward kinematics of the chain from the base of the robot to the fingertip in real time, and publishes a new point cloud for the fingertip sensors at the same rate of the proximity data. Fig. 8 shows our robot gathering point cloud data from the back of the object.

C. Reconstructing the object shape (filling in the occluded faces)

To make a better shape estimate, we can combine the point cloud detected by the RGB-D sensor with the one from the fingertip sensors. The original method in SGP for finding the 3D Gaussian representing the object iterates over all the points in the point cloud and gives them the same weight. This does not work well with the combined point cloud

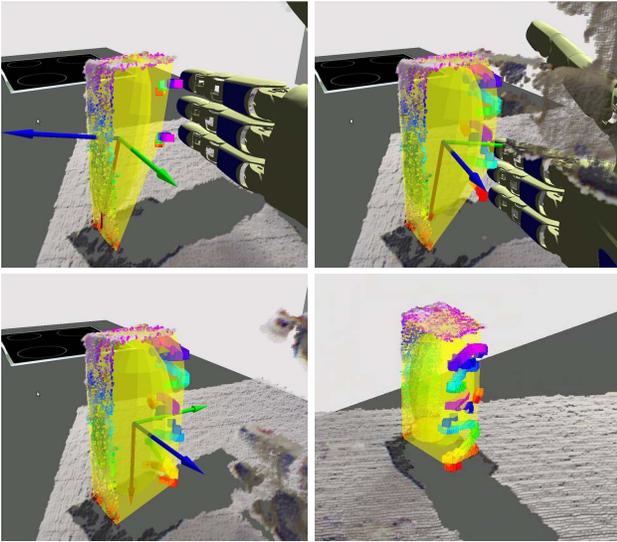


Fig. 9. Shape reconstruction of the object using the fingertips as proximity sensors. On the top, 3D points are acquired by moving the hand close to the object, but without touching it. In the bottom, the result of shape reconstruction using the combined point cloud from the RGB-D sensor and the fingertip sensors is shown.

because the RGB-D data has many times more points than the fingertip data, effectively giving a smaller importance to the last.

To solve this, we now use the following method for finding the 3D Gaussian representation of the object:

- The segmented object point cloud from the RGB-D sensor and the point cloud from the fingertip sensors are joined together.
- The convex hull of the complete point cloud is calculated.
- The centroid and covariance of the convex hull is calculated, and this is the 3D Gaussian point representation needed by SGP.

In order to find the centroid and covariance of a convex hull, we need to iterate over all the triangles that describe it, using their areas as a weighting factor. We implemented the algorithm as described by Ericson[22], which treats the convex hull's polyhedron as a hollow body.

The results of this method can be seen in Fig. 9. Here the robot explores the back of the iced tea package, moving the fingertips close to the object. As more points in the back of the object are detected, the estimated shape of the object keeps improving, until it resembles the real shape of the iced tea package closely.

The system runs the estimation of the shape continuously and new shape estimates are available at a rate of 2 Hz.

Another example of the surface reconstruction is shown in Fig. 10, where the robot perceives a bread toaster on the kitchen counter. Since the top of the toaster has many metallic parts that are not detected by the RGB-D sensor, almost all of the points in the segmented point cloud are in the frontal face of the toaster. The original shape estimation is particularly bad. After moving the robot's hand close to

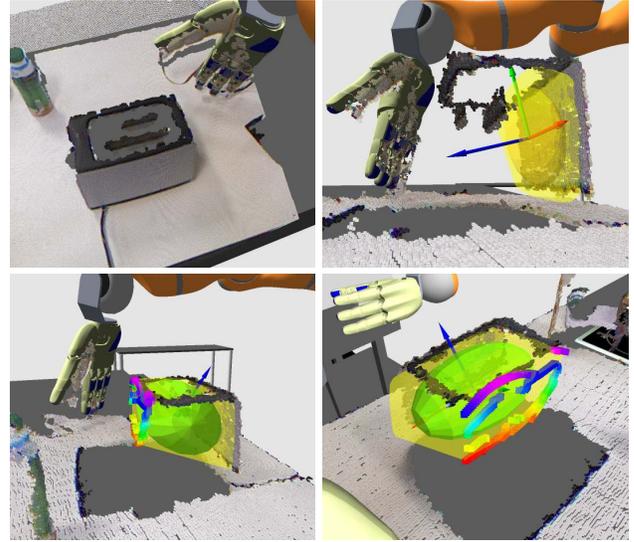


Fig. 10. Another example of the shape reconstruction. Top: A toaster is only partially perceived by the RGB-D sensor, and the 3D Gaussian (centroid + covariance) shape estimation work inadequately because most of the 3D points are located on the front face of the toaster. Bottom: The hand moved close to the back face and an additional point cloud was obtained using the fingertip sensors. The shape reconstruction represents the object much better.

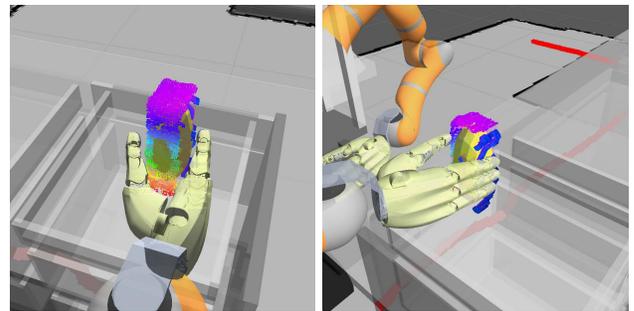


Fig. 11. Using the point cloud obtained from the fingertip sensors, the robot can now find better grasp poses.

the back face, the system created a shape estimate using the described method that was much closer to reality.

Fig. 11 shows a two grasp positions suggested by SGP, taking into account the improved object shape estimation. Using this system, the robot is now able to grasp objects more reliably.

V. SURFACE CLASSIFICATION

When the fingertip sensor is close enough to an object, the sensor can obtain detailed images of the surface's texture. This can be done while holding an object, or sliding the fingertip over it. These images contain enough information to differentiate one object from another. The goal here is to give the robot the ability to learn the texture of objects, and use this information to recognize them later. When grasping an object, the robot can quickly decide if the right object was grasped.

Texture is an important characteristic when recognizing object surfaces. Gray Level Co-Occurrence Matrices



Fig. 12. Objects used for the surface classification experiment. Samples were also taken holding the sensor in the air. The pancake mix bottle was divided into two classes because their surfaces are very different: one for the label, and the other for the plastic material of the bottle.

ID	Class	Total frames taken	Training frames	Testing frames
1	acrylic table	200	160	40
2	wood	200	160	40
3	air	200	160	40
4	shampoo	400	320	80
5	lego	200	160	40
6	cup	600	480	120
7	tomato soup	600	480	120
8	bread board	200	160	40
9	pancake label	400	320	80
10	pancake bottle	400	320	80
11	yogurt	600	480	120
12	ketchup	600	480	120
13	iced tea	600	480	120

TABLE I

LIST OF THE DIFFERENT CLASSES USED IN THE CLASSIFICATION EXPERIMENT AND THE NUMBER OF SURFACE SAMPLES RECORDED FOR EACH ONE.

(GLCM) also known as Gray-Tone Spatial-Dependence Matrices [23] have proven to be a good method to abstract texture information from gray tone images. Using them as input for a support vector machine algorithm (SVM)[24], we classify surface images from the sensors.

To evaluate the performance of the surface recognition system, we chose several test objects available in our test kitchen. Thirteen object surfaces (see Fig. 12) were recorded by sliding a fingertip sensor over the objects. These surfaces are the classes to be recognized by the SVM. Objects with more heterogeneous surfaces require recording for a longer period of time in order to cover the different regions. This translates into more image frames. The shutter speed information from the sensor is also saved for each frame. Table I shows the list of classes and the amount of frames taken for each one.

Fig. 13 shows three images of the pancake bottle surface. The sensor reveals details that are not visible to the human

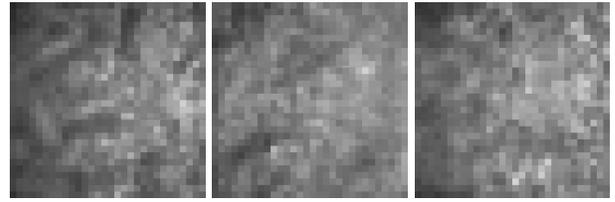


Fig. 13. Three different surface images from the pancake-mix bottle. The sensor registers details that are not visible to the human eye.

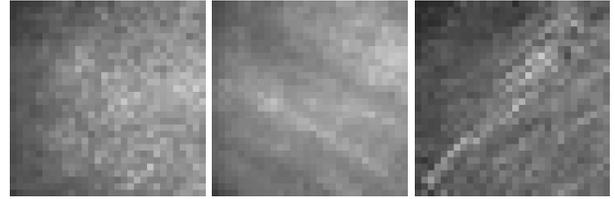


Fig. 14. Surface images corresponding to: acrylic table, wood and metal. The system is able to reliably classify the surfaces based on such images. It is possible to appreciate the texture differences.

eye. Figure 14 shows example surface textures of three different objects: The acrylic table, wood and a piece of metal.

Before running the classification experiments we randomly build a training and a testing set. For each object class, 80% of the pictures are used to train the SVM and the other 20% for testing. From each 900 pixel image, a 16x16 normalized GLCM is computed. GLCM is obtained by first mapping each pixel's original intensity value (0 - 255) into a new intensity value (0 - 15 in our case). This way, a new image with only 16 tones of gray is obtained. Next, the algorithm counts how often pixels with intensity value i are "close" to ones with intensity value j . We use the 8 closest neighbor pixels of the one that is currently checked. This means, the algorithm scans the whole image, and for each pixel px it checks the intensity values of its 8 neighbors $pn1, \dots, pn8$, incrementing the (i, j) element of the resulting GLCM, given that the intensity value of px is i and that of the neighbor being inspected is j . Reviewing all 8 neighbors, makes the GLCM and thus the classification more independent of image rotation.

The GLCM is then normalized and reorganized into a 256 element vector. The normalized shutter speed value of that frame is also added to the feature vector. This value adds important information, since with it, the sensor compensates for differences in surface reflectivity. With the feature vectors ready, the SVM can now be trained and tested. We used *libsvm*[24] with the following configuration: RBF kernel, C_SVC type, gamma = 1.0 and cost = 6.0.

The results of the classification experiment are registered in Table II. These values can vary slightly, depending on the randomly selected training and testing sets. The recognition system performs very well, specially considering that several surfaces were very similar to each other, like the product labels.

It is important to note that because of the high percentage

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	39												1
2	1	33			3		1			1		1	
3			40										
4				51		10	2		7	2	8		
5		1			36			1			1	1	
6			1	7		89	1	2	4		12	2	2
7	2	5		4	1	1	79		7	5	6	6	4
8		1			1			33			1	4	
9				3		1	10		66				
10		2		1			1			72	2	2	
11		1		5		12		4		2	66	23	7
12	2	3	1		1	4	2	1	1	3	15	79	8
13	3	3		1		2	3				6	9	93

TABLE II

CONFUSION MATRIX FOR THE FIRST CLASSIFICATION EXPERIMENT. THE LEFT COLUMN SHOWS THE REAL CLASS ID AND THE FIRST ROW SHOWS THE CLASSIFICATION RESULT.

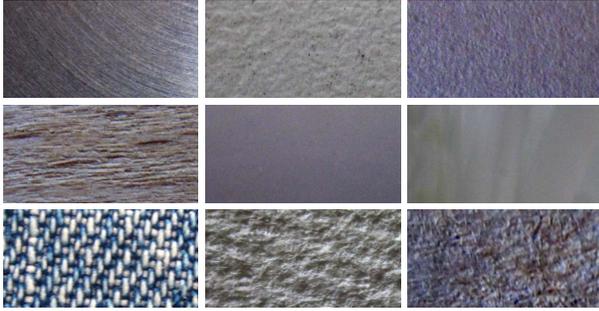


Fig. 15. Macro images of the different material surfaces used in the second classification experiment. From left to right and from top to bottom: (1)metal, (2)table melamine surface, (3)paper, (4)wood, (5)plastic, (6)glass, (7)fabric, (8)napkin paper and (9)cardboard.

of correctly classified vectors, the system can classify the surface correctly using a small number of samples. The sensor delivers images at approximately 50 Hz, so that a correct and certain classification can be expected by classifying the images obtained in under one second.

Several of the objects used for the classification experiment have very similar surfaces, especially the ones with printed labels. So we also evaluated the classification performance on surfaces of different raw materials. For each surface 240 frames were taken, and again, 80% of them were used to train the SVM and 20% were used for testing. Fig. 15 shows the textures to be classified. The results are registered in table III.

VI. SLIP DETECTION

The fingertip sensor is capable of detecting the relative motion of objects located at a short distance from the sensor's lens. The sensor is installed in the fingertip so that when the hand is grasping an object, the distance to the surface is optimal.

	1	2	3	4	5	6	7	8	9
1	48								
2		47		1					
3			47	1					
4				47					1
5					48				
6						48			
7							47		1
8								46	2
9				2			1	1	44

TABLE III

CONFUSION MATRIX FOR THE SECOND CLASSIFICATION EXPERIMENT. THE CLASS ID FOR EACH MATERIAL SURFACE IS GIVEN IN FIG. 15.

While doing pick and place tasks, the objects are rigidly grasped by the robot, and any movement detected by the fingertip sensors is directly translated to slip of the object.

Good slip estimation is difficult based on data from the typical sensors available on robotic grippers: Position sensors, or force/torque. A good example of this problem are glass bottles: Because of their regular shape and low-friction surface, they can slide out of the hand without making significant changes to the position and force signals of the gripper.

The proposed fingertip sensor excels at this task, and can deliver reliable slip data at approximately 50Hz per sensor or up to 200Hz if only one sensor is selected. The robot can use this information to adjust the grasping force while carrying an object, or decide to hold the object with a second hand in order to keep it from falling.

To demonstrate this capability we implemented the automatic adjustment of the grasping force to avoid object slip in the hand, keeping the force close to the minimum necessary. The robot grasps an object with an initial default force, and it reduces the applied force on the fingertips until slip is detected. Then it immediately holds the object a bit harder to keep it stable. If more slip is detected, the force is increased further. Using this simple method, the robot can hold objects between the fingers stably, but using a low force. A good side effect of this method is that the hands' motors stay much cooler while holding an object, since the torque they have to apply is now lower.

Fig. 16 shows the relationship between grasping force and detected slip during a grasping action. The slip is detected early and quickly enough to avoid any large movement of the object. Also during dynamic movements of the arm, the hand can adjust the grasping force before letting the object slip away more than a few millimeters.

VII. ACKNOWLEDGMENTS

We would like to thank Piotr Esden-Tempski, Julian Brunner, and Frank Ehmann, who have contributed to the development of the sensor and the project.

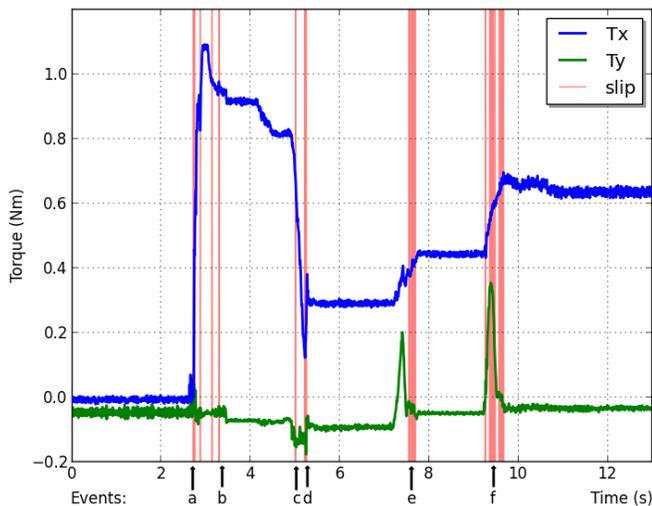


Fig. 16. Finger torque and slip signals during the automatic adjustment of grasping force. (a) The fingers moved to grasp the object, and they hold it. After (b), the torque is slowly reduced until the object moves a small distance, and slip is detected in (c). At (d), the fingers have now grasped the object stably again, but with a lower force. If the object is disturbed externally (e and f), the fingers apply more force.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we presented a new sensor for robot fingertips, and several ways how it can improve robotic grasping and manipulation. The sensor is very versatile, and there are still other possibilities to explore.

Each of the sensor's three modalities is useful in different sections of a typical pick and place action: In situations where only part of the object is visible to range sensors, it can complete the 3D shape information, and this in turn makes it possible for the grasp planner to give good grasp pose suggestions. As a surface texture sensor, the robot can use it to find objects whose texture it has learned previously, or to check the identity of an object it has grasped. Finally, the sensor can make carrying objects safer, as the robot gets a warning that the grasped object is slipping, and can take corrective actions.

In a follow-up paper, we plan to discuss methods for the automatic exploration of occluded sections of objects, and the functionality as a pre-grasp sensor: To position the fingers around objects to maximize the grasp success chances.

Finally, one of our long term goals is to have closed-loop grasping, where the pose of the object is tracked continuously in real-time, through a combination of proximity sensors in the fingertips and tracking algorithms for RGB-D sensors in the head. We believe that this will make possible the reliable grasping of moving objects.

REFERENCES

- [1] P. K. Allen and P. Michelman, "Acquisition and interpretation of 3-D sensor data from touch," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 4, pp. 397–404, 1990.
- [2] R. D. Howe, "Tactile sensing and control of robotic manipulation," *Journal of Advanced Robotics*, vol. 8, no. 3, pp. 245–261, 1994.
- [3] R. Dahiya, G. Metta, M. Valle, and G. Sandini, "Tactile sensing: From humans to humanoid," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 1–20, Feb. 2010.
- [4] B. Mayton, L. LeGrand, and J. R. Smith, "An electric field pretouch system for grasping and co-manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 831–838.
- [5] K. Hsiao, P. Nangeroni, M. Huber, A. Saxena, and A. Y. Ng, "Reactive grasping using optical proximity sensors," in *International Conference on Robotics and Automation (ICRA)*, 2009.
- [6] J. M. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K. J. Kuchenbecker, "Human-Inspired robotic grasp control with tactile sensing," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1067–1079, Dec. 2011.
- [7] P. Mittendorf and G. Cheng, "Humanoid multimodal tactile-sensing modules," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 401–410, 2011.
- [8] A. Schmitz, M. Maggiali, L. Natale, B. Bonino, and G. Metta, "A tactile sensor for the fingertips of the humanoid robot iCub," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 2212–2217.
- [9] A. Leeper, K. Hsiao, E. Chu, and K. Salisbury, "Using Near-Field stereo vision for robotic grasping in cluttered environments," *12th International Symposium on Experimental Robotics*, Dec. 2010.
- [10] F. Santos, V. Silva, and L. Almeida, "A robust selflocalization system for a small mobile autonomous robot," in *Proc. of the 2002 IEEE International Symposium on Robotics and Automation*, 2002.
- [11] R. Hover and M. Harders, "Measuring and incorporating slip in data-driven haptic rendering," in *Haptics Symposium, 2010 IEEE*, 2010, pp. 175–182.
- [12] O. Kroemer, C. H. Lampert, and J. Peters, "Learning dynamic tactile sensing with robust Vision-Based training," *Robotics, IEEE Transactions on*, vol. 27, no. 3, pp. 545–557, 2011.
- [13] N. Jamali and C. Sammut, "Majority voting: Material classification by tactile sensing using surface texture," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 508–521, June 2011.
- [14] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppel, A. Albu-Schäffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, et al., "The KUKA-DLR lightweight robot arm-a new reference platform for robotics research and manufacturing," in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, 2010, pp. 1–8.
- [15] H. Liu, P. Meusel, G. Hirzinger, M. Jin, Y. Liu, and Z. Xie, "The modular multisensory DLR-HIT-Hand: hardware and software architecture," *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 4, pp. 461–469, Aug. 2008.
- [16] Avago Technologies, "ADNS-9500 LaserStream Gaming Sensor," October 2011. [Online]. Available: <http://www.avagotech.com/docs/AV02-1726EN>
- [17] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," in *In IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 12–17 2009.
- [18] Intelligent Autonomous Systems Group, Technische Universität München, "TUM-ROS code repository," 2012. [Online]. Available: <http://www.ros.org/wiki/tum-ros-pkg>
- [19] U. Klank, D. Pangercic, R. B. Rusu, and M. Beetz, "Real-time CAD Model Matching for Mobile Manipulation and Grasping," in *9th IEEE-RAS International Conference on Humanoid Robots*, Paris, France, December 7–10 2009, pp. 290–296.
- [20] U. Klank, L. Mösenlechner, A. Maldonado, and M. Beetz, "Robots that validate learned perceptual models," in *IEEE International Conference on Robotics and Automation (ICRA)*, St. Paul, MN, USA, May 14–18 2012.
- [21] A. Maldonado, U. Klank, and M. Beetz, "Robotic grasping of unmodeled objects using time-of-flight range data and finger torque information," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 18–22 2010, pp. 2586–2591.
- [22] C. Ericson, *Real-Time Collision Detection*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2004, pages 108–109.
- [23] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-3, no. 6, pp. 610 – 621, 1973.
- [24] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.