NAIST-IS-DD1561035

## Doctoral Dissertation

# Textile recognition and manipulation using tactile sensing based on active perception

Felix Wolf Hans Erich von Drigalski

March 15, 2018

Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Felix Wolf Hans Erich von Drigalski

Thesis Committee:

| | |
|---|---|
| Professor Tsukasa Ogasawara | (Supervisor) |
| Professor Kenji Sugimoto | (Co-supervisor) |
| Associate Professor Jun Takamatsu | (Co-supervisor) |
| Assistant Professor Ming Ding | (Co-supervisor) |
| Professor Akio Noda | Osaka Institute of Technology |

# Textile recognition and manipulation using tactile sensing based on active perception[*]

Felix Wolf Hans Erich von Drigalski

## Abstract

Tactile information is an attractive sensing modality, as it yields information about the contact that is otherwise inaccessible, depends neither on lighting conditions nor line of sight, and is computationally lighter than camera data. To make use of it during the robotic manipulation of textiles, in this work I investigate the sensing requirements, propose a suitable sensor, a gripper prototype and a method to extract tactile information, and finally showcase it in an application.

First, I investigate the deformation of the human fingertip while it is under load using stereo cameras and a 6D force plate. By measuring the force as well as the deformation of the fingertip's lateral surface, I relate the curvature of the fingertip surface to the force acting on the fingertip.

Second, I propose 1) an open-source gripper with a sensor that utilizes the deformation of a rubber half-dome to determine the normal and shear force acting on the fingertip and 2) a rubbing motion implemented on this gripper, which extracts information about the textile that is being grasped. I show that the data obtained by this exploratory motion allows not only the identification of textiles, but also to distinguish between a single and multiple layers of them, and propose a method to perform this classification. Furthermore, I show that by varying the speed and force of the exploratory motion, different information is obtained, and that this can be used to increase the accuracy of the identification.

To show the usefulness of the proposed method, I use the proposed open-source gripper and exploratory motion, as well as computer vision algorithms

[*] Doctoral Dissertation, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1561035, March 15, 2018.

and human tools, to fold shirts with a humanoid robot. I show that the use of haptic feedback as well as the use of human tools significantly reduce the time required to complete the folding task.

In summary, the proposed methods and gripper contribute a novel way to a) identify different textiles and distinguish between a single and multiple layers of them, b) manipulate textiles with a serial manipulator and c) complete the task of laundry folding with a humanoid robot. The code and mechanical designs used in this work are open source and available for download.

**Keywords:**

Textile Manipulation, Tactile Recognition, Haptic Sensing, Robotic Gripper, Service Robots

# 能動的な触覚センシングに基づく布の識別と折り畳み操作[*]

フォン　ドリガルスキ　フェリクス　ウォルフ　ハンス　エリヒ

## 内容梗概

触覚情報は他のセンシング方法では計測できない場所にまでコンタクトすることが可能であり，さらにカメラと比較し照明条件や視線に依存せず，計算速度も早いことからとても魅力的な計測手法である．本論文では，布製品をロボットが扱うために，必要となるセンシング技術について調査する．さらに，最適なセンサやグリッパーのプロトタイプ，接触情報を抽出するための手法について提案し，最後にそれらの応用例を示す．

はじめに，ステレオカメラと6軸力センサを用いて，荷重を加えたヒトの指先の形状を計測した．力と指先側面の変形を同時に測定することによって，指先の曲面と指先が発揮している力とを紐付けることができる．

次に以下の2つを提案する．1) ゴム半球の変形を用いることで，指先の垂直応力及びせん断応力を取得することができるセンサを用いたオープンソースなグリッパ．2) このグリッパによるこすり動作により，把持中の接触情報を取得する手法．

この動作により，事前に学習した布製品を識別するだけでなく，それらが複数枚重なっていることを識別するための手法とその結果を示す．さらに、こすり動作の速度と力を変化させることによって，異なる情報が得られ，これらを用いて推定の精度を高めることができることも示す．

提案手法の有効性を検証するために，提案するグリッパでこすり動作を行い，画像認識と折りたたみ補助道具を用いて，ヒューマノイドロボットを用いたシャツ折りたたみ動作を生成した．さらに，触覚フィードバックの使用だけでなく、

[*] 奈良先端科学技術大学院大学 情報科学研究科 博士論文, NAIST-IS-DD1561035, 2018 年 3 月 15 日.

補助道具を活用することで折り畳み動作を完了するために必要な時間を大幅に短縮することを示す。

まとめると、提案した手法とグリッパは以下の新規性を有する a) 布の識別及び布の層が 1 層か複数層かの識別．b) ロボットアームを用いた布の操作 c) ヒューマノイドロボットを用いた服の折り畳み．本研究のソースコードと３Ｄモデルはオープンソースでありインターネット上からダウンロードすることが可能である。

## キーワード

布の操作、 触覚による認識、 ハプティクセンシング、 ロボットグリッパ、 サービスロボット

# Acknowledgements

My heartfelt thanks go to:

My professors Tsukasa Ogasawara, Jun Takamatsu and Ming Ding for their patience, wise counsel and insight.

The DAAD and the Ministry of Education, Culture, Sports, Science and Technology for their support.

Prof. Tamim Asfour, whose recommendation opened the door to another world for me.

My precious colleagues, teammates and friends Gustavo Garcia, Lotfi El Hafi and Pedro Uriguen for their tireless efforts and mental support, and for our unforgettable adventures to Nagoya, Sweden and onto the stage of ICRA 2016.

Marcus Gall, Masayuki Watabe and all the students who joined my projects, for their contributions and giving me the chance to teach them what I could.

My brave mother, my family who I miss, my other half and all my friends who made my stay unforgettable.

The countless people who have worked on the open source software that I used.

It is my hope that this approach to sensing and manipulating textiles can be of use in research and industry applications, and help us usher in a future where robots fulfill their destiny and, at last, do our chores.

# Contents

# List of Figures

# List of Tables

”The tactual properties of our surroundings do not chatter at us like their colors, we have to make them speak... Eye movements do not create color the way finger movements create touch.” (Katz, 1925)

# 1. Introduction

If the living room, as Cynthia Breazeal put it, is the final frontier for robots, then surely soft objects and clothes are one of its great battlefields. Even though robots have demonstrated superhuman speed and strength with rigid objects, their performance when manipulating even simple unstructured textiles is orders of magnitude lower than humans'. Nonetheless, the manipulation of textiles is required for countless tasks in the household (*e.g.* laundry, bed making, table setting) as well as the workplace (*e.g.* car seat assembly, furniture manufacture, dry cleaning facilities, not to mention clothes manufacturing). Accordingly, the demand for robots that can manipulate textiles and clothing articles is significant.

The challenges in textile manipulation are numerous (complicated task representation, occlusions, large search space) and span fields as diverse as computer vision, task planning and material science. By exploiting the sensing modality of touch, we aim to relax some of these problems. Tactile sensing can extract information from objects that is hidden from sight, and it is with good reason that we as humans have difficulty imagining folding a shirt without feeling it with our fingers — tactile feedback is central to our internal task representation.

Bohg *et al.* [7] explore the notion that interacting with the environment creates rich signals that would otherwise not be present, and that only the combination of action and perception allows us to form meaningful representations of our environment. As an example, one might imagine picking up a crumpled clothing article in order to unfold and identify it. The importance of active perception for this task should be immediately evident.

In this thesis, I propose a method based on active perception to determine the material of a clothing article using a biomimetic finger rubbing motion, and to distinguish if a single or multiple layers have been grasped.

First, I present background on the human fingertip in section 2 and investigate the relationship between its deformation and the force applied on it. It grounds the design considerations for the open-source gripper introduced in section 3, where both the design and the exploratory rubbing motion are explained, as well as the results of the textile recognition experiments. In section 4, I implement a clothes-folding approach using a humanoid robot and a folding tool for humans. Finally, I conclude in section 5 with an outlook for future work and applications.

## 1.1 Contributions

To summarize, the contributions of this thesis are:

1. A method to distinguish different textiles and determine if a single or multiple layers of it are present
2. An open-source gripper design as well as code to implement the method
3. Experimental data showing the method's limits and requirements in terms of sensory data

# 2. Human Fingertip

## 2.1 Introduction

The contact at the human fingertip is an important source of information during human grasping. Particularly remarkable is the ability to sense a failing grasp before slippage occurs, and to sense even small textures. Understanding the contact at the human fingertip better should give us insight into

One can see that the lateral side of the fingertip is deformed when touching an object. We pose the question if this deformation can be used to to determine the force applied to the finger.

Successfully measuring the lateral deformation of the fingertip would have the added benefit of validating simulations of the deformation of the human fingertip. Simplified models of the fingertip have been proposed by Srinivasan [8] and Tada [9], but their results have only been compared with the fingertip's contour when it is placed under a line load.

Thus, the goal of this section is to:

- develop a measuring device and method that can measure the shape of the lateral surface of a human fingertip
- investigate whether the fingertip's deformation yields sufficient information to determine the contact force

## 2.2 Related work

### 2.2.1 Fingertip deformation

Srinivasan [8] has investigated the deformation of human and primate fingertips, and proposed a "waterbed" model of the fingertip as a membrane filled with a fluid under pressure. Tada and Pai [9] has extended this model by replacing the membrane with simplified shell elements with bending stiffness. Their results are compared to the deflection of a human fingertip's bottom side when subjected to a line load.

A number of recent works on fingertip deformation have concentrated on increasingly complex FE models of the fingertip in an effort to investigate the

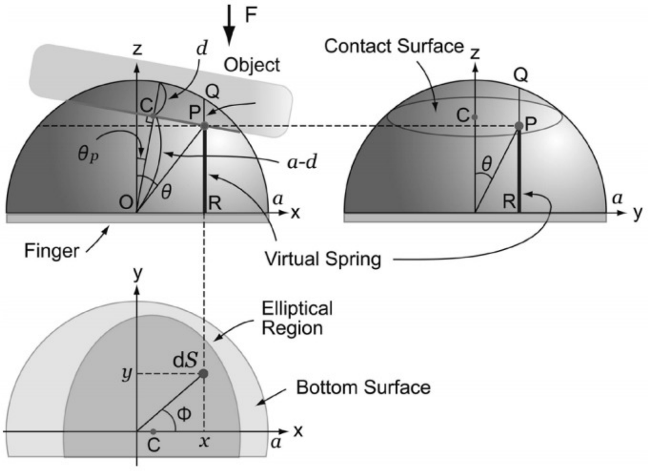


Fig. 1. Contact mechanism.

Figure 1. A deformable hemisphere is simplified to a linear spring in [1]

biomechanics behind fingertip sensation [10, 11]. These simulations investigate the strain and vibration frequencies present in different layers of skin. Real-time calculation will require a more lightweight approach, however.

Mainly in the interest of designing robot fingertips, a number of analytical solutions for simplified deformable bodies have been published, such as Inoue and Hirai [1], which presented equations to express a homogenous, deformable hemisphere as a spring, depending on the angle of contact. In the case of a robot hand with deformable fingertips, the results can also be used to incorporate its fingertip into the control system at reduced computational cost.

A review of the state of the art regarding tribological measurement of skin properties was published in Derler and Gerhardt [12].

### 2.2.2 Fingertip force sensors

**For measuring fingertip force in humans** [13, 14, 15, 16] have been pursuing the use of fingernail imaging to measure the fingertip force, direction, and angle of contact. Their idea is that the fingernail changes color in distinct patterns when the fingertip is charged, because of the constricting blood vessels in the flesh underneath the fingernail (see fig. 2) [17].

A first prototype sensor was based on 6 LEDs and 8 photoreceptors that were

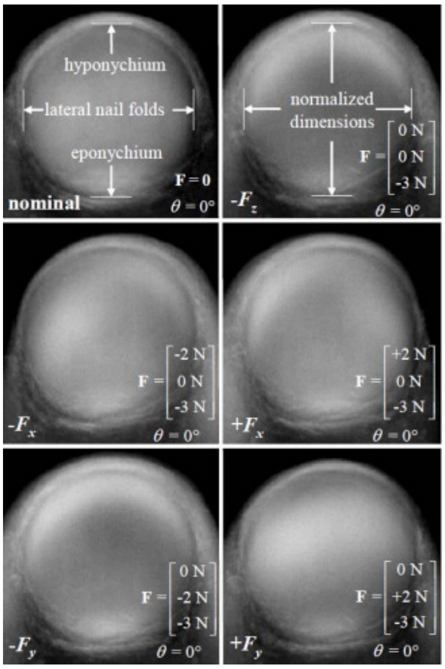


Figure 2. Average fingernail coloration patterns for 15 human subjects for 6 different force poses. 5x contrast is applied. [2]

attached to the fingernail to measure the coloration (see fig. 3). Sun *et al.* [18] mention that the sensor predicted normal force to within 1 N accuracy in the range of 2 N and shear force to within 0.5 N accuracy in the range of 3 N, and note the disadvantage of having to manufacture sensors fitted to each subject's fingernail. The most recent review of this approach compared the transmission qualities of different wavelengths in the fingernail [3].

More recently, the use of camera images taken from above the fingernail was proposed [19], and active appearance models used to transform the camera images to reference images [20]. Predictions of the normal force with an RMS error of 0.3 ± 0.1 N and shear force with an RMS error of 0.25 ± 0.1 N were reported.

A wearable sensor has been commercialized in 2011 by Keskato and Shiseido[1], which is supposed to measure the normal force at the fingertip. The sensor consists of two parallel beams, the ends of which are in contact with the lateral sides of the fingertip. When the fingertip is pressed on an object, the displaced

[1] Sold under the brand name "HapLog"

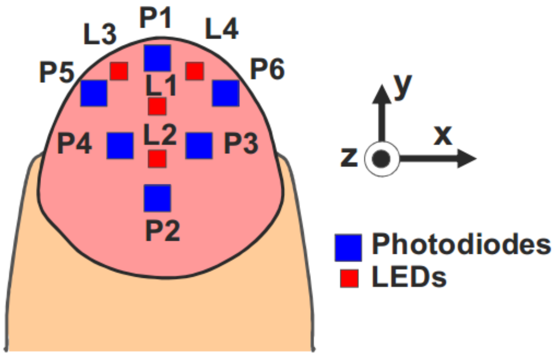


Figure 3. Schematic of the sensor array on the fingernail area from [3]

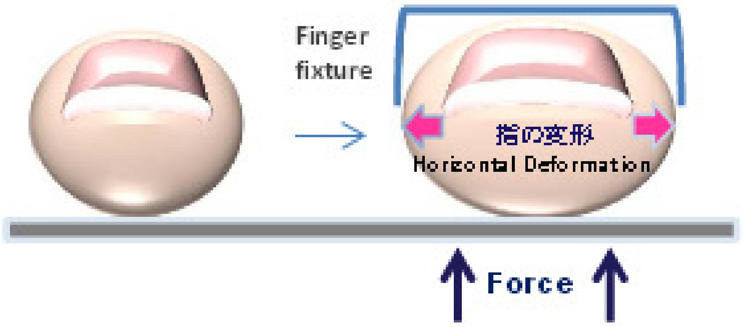


Figure 4. Schematic of a commercial fingertip force sensor. The deflected beams are arranged parallel to the force direction in this image. Reproduced from http://keskato.co.jp/products/HapLog.html

fingertip pulp causes the beams to bend outwards (see fig. 4). Strain gauges inside the beams measure their deflection, which is then connected to the fingertip force through reference training data previously recorded with the subject. The main disadvantages of this setup are the inability to measure lateral forces, and the constraining of the fingertip's deformation, which could affect the haptic perception of test subjects.

**For use in robotic hands** Many works on tactile sensors have concentrated on perceiving the same phenomena believed to be sensed by humans in order to predict the onset of slip. A review of sensor technology can be found in [21].

In this way, [22] have placed accelerometers in a foam matrix (a fingertip facsimile) and tried to detect vibrations caused by the slipping at the edges of the artificial fingertip ridges. This emulates the FA receptors in human skin. One problem with this approach is that the abrasion caused by wear of the artificial fingertip influences the behavior of the contact, and the vibrations induced during slip.

As noted in [21], the use of conventional arrays for slip detection has been suggested, but is only feasible if the scanning rate and array resolution are sufficiently high. In this vein, [23] propose the use of a piezo-resistive sensor pad evaluated at a very high frequency to detect the incipient slipping of an object. They used a robotic arm to slide a foam-covered 80 x 80 mm pad of pressure sensors over different surfaces, which the pad was sampling at about 1800 Hz. The high frequency allowed the extraction of spectral information, which was used to train an artificial neural net to detect if the object is slipping or at rest, as well as the texture of the object (e. g. a mouse pad or a cup).

Different types of artificial skin have been proposed, for example by [24], who claim to detect incipient slip with two inductive sheets separated by a rubber layer. [25] propose a transparent rubber matrix inside the fingertip which hold independent chips which measure the stress and transmit data optically. Their prototype is supposedly able to distinguish between rolling and slip.

## 2.3 Theoretical background

### 2.3.1 Stereo vision

In stereo vision, three-dimensional information is generated from two images of the same scene. If a point can be seen in both images, its depth or distance to the cameras can be calculated via triangulation. As we use stereo vision to measure the fingertip surface, we introduce the theoretical basics in this section.

**Disparity and triangulation** Disparity can be defined as the difference between pixels $d = x_l - x_r$. In an ideal camera configuration, it can easily be shown that the following formula holds true (see fig. 5):

$$Z = f_x \frac{T}{d}$$

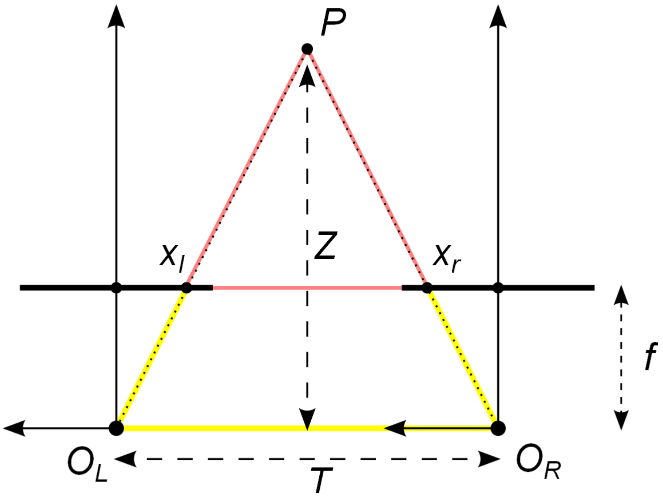


Figure 5. Depth calculation from triangulation in a stereo setup (similar triangles marked in colors)

with $T$ the baseline distance of the cameras and $f_x$ the focal length. One can see that depth and disparity are inversely related, which implies that fine-grain depth measurement is limited to nearby objects.

To get to this arrangement, however, general configurations of camera orientations have to be considered.

**Epipolar geometry** Epipolar geometry describes the relation of two pinholes cameras, using the following characteristics [26]:

- the point of interest $X$
- the centers of projection $O_L$ and $O_R$
- the points of intersection of the line $\overrightarrow{O_L O_R}$ with the two image planes, called the epipoles $e_L$ and $e_R$
- the projection of the point of interest $X$ on the image planes, $x_L$ and $x_R$
- the epipolar plane formed by $X$, $O_L$ and $O_R$, and the intersection of the epipolar plane with the image planes, called the **epipolar lines**

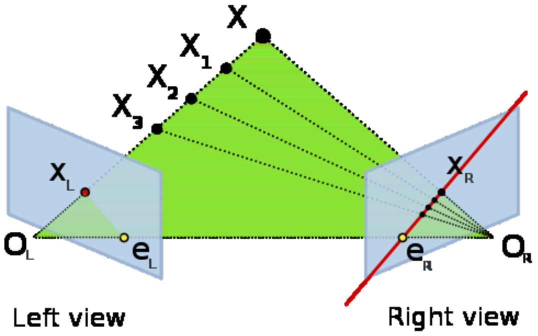


Figure 6. Epipolar geometry (cc by-sa 3.0 Arne Nordmann)

It can be shown that, if the projection $x_L$ of a point $X$ is known, then $x_R$ has to lie on the corresponding epipolar line (see fig. 6). [27] detail the derivation of the fundamental matrix $F$ and essential matrix $E$[2], which contain information about the rotation and translation of the two cameras and which can be used to find the epipolar line in world and camera coordinates respectively. The epipolar line is given by $(x'_R)^T E x_L = 0$, where $x'_R$ denotes the projection in the right hand side camera's coordinates. Notably, this property allows the restriction of the search space from two dimensions to only one.

If the projections $x_L$ and $x_R$ of a point are found, its depth can be calculated through the correspondence condition of the fundamental matrix:

$$(x'_R)^T F x_L = 0$$

**Rectification and calibration** A special case arises when cameras with identical focal lengths are placed such that the image planes are parallel. In that case, the epipolar lines are parallel, and the epipoles lie at infinity (see fig. 8). Most notably, the depth calculation is reduced to a term directly proportional to the disparity of the point:

$$X = \frac{b}{x_l - x_r} \begin{pmatrix} x_l \\ y_l \\ 1 \end{pmatrix}$$

An image pair can be transformed such that the epipolar lines become parallel.

[2] p. 240 ff.

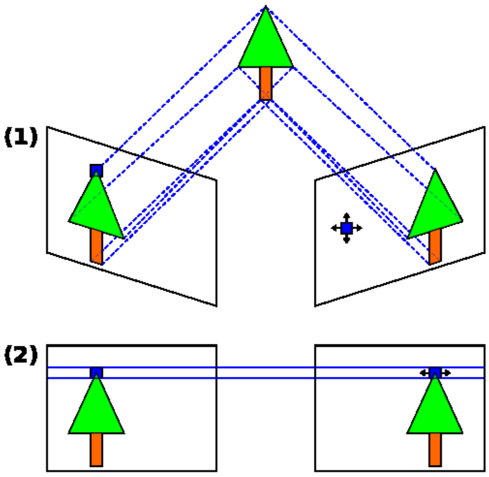


Figure 7. Image rectification illustration (cc-by-sa 3.0 Bart van Andel)

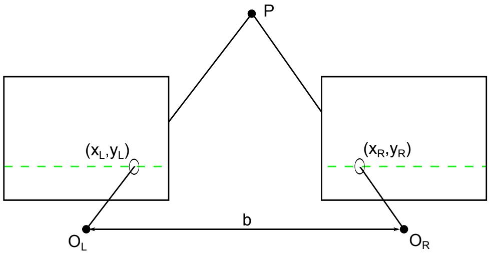


Figure 8. Special case after rectification: frontal parallel cameras

This transformation is called **image rectification**. Because of the increased simplicity and reproducibility when calculating with perfectly coplanar, row-aligned images, most image treatment algorithms work only on rectified images [28].

Rectification maps can be calculated on-line via Hartley's algorithm [29], which requires no computation of the camera's intrinsics, but is potentially less precise and offers no image scale. It is both more intuitive and appropriate to use calibration patterns to compute both the camera intrinsics and rectification maps in advance. This process is readily implemented in OpenCV as well as MATLAB.

Preparing a stereo setup will involve the following steps:

- calibration of each camera (finding the camera intrinsics matrix M, and the distortion coefficients)
- stereo calibration of the two cameras (finding the rotation and translation matrices R and T, and the essential and fundamental matrices E and F)
- computation of the rectification parameters via Bouguet's algorithm [30] (and optionally the reprojection matrix Q)

At the end of the OpenCV implementation stand two lookup maps which, when applied to the raw camera images, return undistorted and rectified pictures ready for use in stereo vision algorithms.

The last note should be that the reprojection matrix Q allows the calculation of real-world coordinates of a point:

$$Q \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix}$$

$W$ being a scale factor, such that the 3D coordinates are $\begin{pmatrix} X/W \\ Y/W \\ Z/W \end{pmatrix}$. [3]

### 2.3.2 The matching problem

Image registration methods can be categorized into intensity-based methods, which find correlating patterns in the grayscale values of the images, and feature-based methods, which try to match only especially distinct, characteristic points.

#### Intensity-based methods

**Block matching** Block matching is the simplest way of finding a corresponding point in the second image. It compares the neighborhood of a pixel in the reference image (a "block") with the neighborhood of a pixel in the target image. The similarity can be established by criteria such as the sum of absolute differences (SAD), sum of squared differences (SSD) or normalized cross-correlation (NCC) [31]. The size of the neighborhood can also differ between block matching algorithms.

If the two images are rectified, the search space in the right image is reduced to pixels further to the right than the pixel being searched ($y_R = y_L$ and $x_R > x_L$). Even then, the maximum search distance should be limited in order to minimize calculation cost. However, a small search distance can reduce precision and cause mismatches. This inherent conflict of interest in block matching algorithms can

[3] p. 435 in [28]

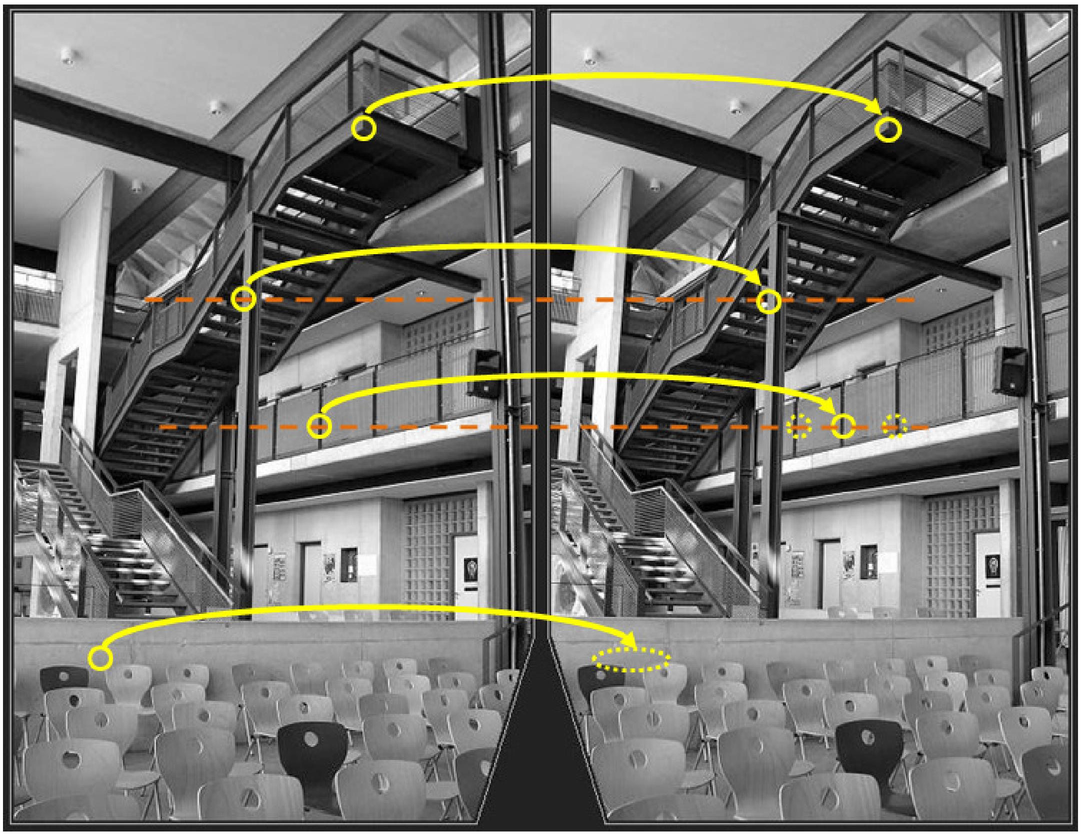


Figure 9. Problems of block matching algorithms. Photograph (c) Anika Henkel

be reduced by fine-tuning the search distance to a suitable value for the scene in question [4].

Block matching is a very versatile method of image registration, and requires no knowledge of camera parameters. For this reason it is also often used in optical flow estimation and video compression. However, block matching usually fails when patterns repeat in the picture, such as grids or line, or when the area in question is relatively homogenous, like a field of snow or clouds. The sensitivity to repeating patterns increases with the allowed search space, as well.

Block matching is also not inherently lighting invariant, but it can be if it is

[4] For example, video compression algorithms might reduce the search range for optical flow estimation when the frame rate of a movie file is high, and stereo correspondence searches might use knowledge about the scene to restrict the range.

applied to LoG[5]-filtered images. Because of its high speed at low search distances, it is common in motion tracking, too.

**Dynamic/adaptive block matching, variational methods** It can reasonably be assumed that contiguous objects make up a considerable portion of everyday scenes. For pixels that show the same object, the disparity would tend to fluctuate very little. One method to improve the results produced by block matching algorithms is thus to impose a smoothness constraint.

Some approaches change the window size adaptively, especially around object borders [32]. There also exist multilevel (pyramiding) versions of block matching algorithms [33].

Variational methods as proposed by [34] pose the smoothness constraint as as a flux problem between neighboring pixels in the x as well as the y direction. They are global rather than local (like block matching algorithms would be), and the solution of the resulting partial differential equation is computationally heavy and not real-time capable. It is also principally used for optical flow estimation rather than correspondence search in stereo vision.

**Feature-based methods** Feature point matching methods try to match only especially characteristic points in the two images. These correspondences can then be used to create a transformation map between the images, or to identify objects in a scene. Feature-based methods are a powerful tool for object recognition, but can also be used for general image registration.

A state-of-the-art feature matching algorithm might consist of the following steps. The implementation chosen by [35] for SIFT is given as an incomplete example.

1. Candidate generation / interest point detection (SIFT: Scale-space extrema detection)

2. Keypoint localization and selection (SIFT: Stability check, orientation assignment)

3. Descriptor extraction (SIFT: Oriented histogram of gradients)

[5] Laplacian of Gaussian

4. Calculation of feature similarity (SIFT: Euclidean distance of descriptor vector)
5. Descriptor matching (SIFT: Nearest neighbor (Best-Bin-First))
6. Exclusion of weak and/or ambiguous matches (SIFT: No match if second-nearest neighbor is >0.8 as close as nearest)

**Candidate detection and selection** Interest points or keypoints are well-defined points in the image that are rich in local information and stable against transformations.

Methods like SIFT and its variants (SURF, CenSure...) generate keypoint candidates from extrema in intensity as well as scale space, as scale invariance is desirable for stable view-based object recognition. If scale invariance is not required, as is the case in this work, other detectors such as the Harris corner detector or blob detectors might be used. An overview about different interest point detectors has been published in [36] and [37][6].

**Feature descriptors** Once a point of interest has been detected, a feature descriptor has to be extracted from it. A large number of descriptors has been proposed over the last few years which are still actively competing. [38] summarize the situation well:

> “A large number of vision applications rely on matching keypoints across images. The last decade featured an arms-race towards faster and more robust keypoints and association algorithms: Scale Invariant Feature Transform (SIFT) [35], Speed-up Robust Feature (SURF) [39], and more recently Binary Robust Invariant Scalable Keypoints (BRISK) [40] to name a few. These days, the deployment of vision algorithms on smart phones and embedded devices with low memory and computation complexity has even upped the ante [...]”[7]

Most of them attempt to offer improved properties in the following fields [41]:

[6] This paper was not available through the institute’s subscription and is mentioned here without verification.

[7] Citation numbers adjusted

- Invariance against transformation (rotation, lighting, scale)
- Low calculation cost (during extraction and matching)
- Low memory requirements
- High reproducibility

Binary feature descriptors have recently gained popularity because of their low memory requirements and computational cost. While feature descriptors based on oriented histogram of gradients, like SIFT, are represented as a vector of floating point numbers that is compared via euclidean distance, binary descriptors are both shorter and are compared via hamming distance.

**Exclusion of weak and ambiguous matches** After the similarity of all features has been evaluated, they have to be matched. Brute force matching with a fixed cut-off threshold can be applied, but the most common method appears to be some form of nearest neighbor matching, as practiced by SIFT, for example. After nearest-neighbor matching, the similarity to the second-nearest neighbor can be checked, and the match discarded if it is ambiguous. It stands to reason that the number of ambiguous matches can increase and disrupt matching quality if the buckets are very large.

## 2.4 Measurement setup and implementation

### 2.4.1 Force sensor setup

The force measurement setup consists of four force sensors placed at each corner below the acrylic plate on which the finger is pressed. Each force sensor measures the forces in three directions, such that a resultant 6 element vector containing both forces and moments can be calculated. The force data is captured at a frequency of 40 Hz.

The setup includes a camera pointed at the finger from below, which can be used to measure contact eccentricity and slip coefficient. The camera was not used during these experiments, however.

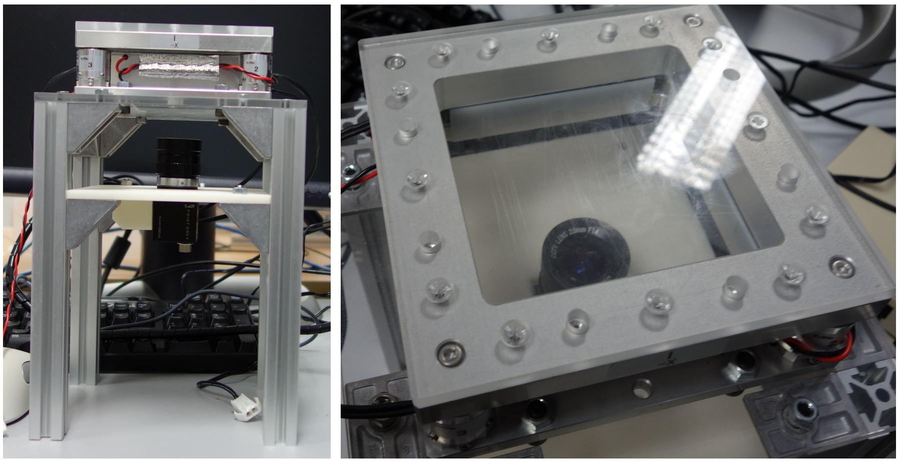


Figure 10. The force measurement setup before addition of stereo cameras.

The data received by the device is handled by a proprietary firmware by producer TecGihan. As the dll provided uses managed memory, a wrapper class based on gcroot was constructed.

### 2.4.2 Stereo cameras

The stereo setup consisted of two USB 3.0 cameras[8] with macro lenses, mounted on a 3D printed solid polymer frame (see fig. 11). The camera mount was then screwed to a simple scaffold of aluminum extrusion profiles. To ensure that the precision of the setup is sufficient, so the theoretical maximum depth resolution was evaluated with the formula provided in [21], p. 422:

$$\Delta Z = \frac{Z^2}{fT}\Delta d$$

Where $\Delta d$ is the physical pixel width of the camera's sensor, $\Delta Z$ the depth resolution, $f$ the focal length and $T$ the baseline distance between the two cameras. The pixel width is 4.8 µm while the macro lenses offer a focal length of 3.5 mm. The distance to the object filmed would normally be around 80 mm.

Fixing the cameras in hole pair 3 of the camera mount yields the lowest angle between cameras and a baseline distance of $T = 52$ mm. This means that the maximum available depth resolution has a theoretical value of

[8] Point Grey Research brand, model FL3-U2-13Y3M-C.

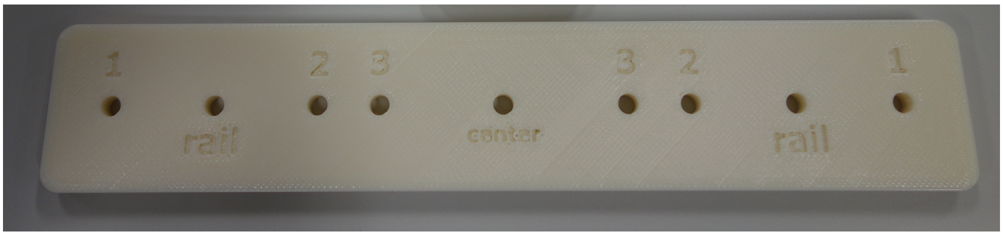


Figure 11. 3D printed camera mount. Hole pairs 1 – 3 allow for different baseline distances.

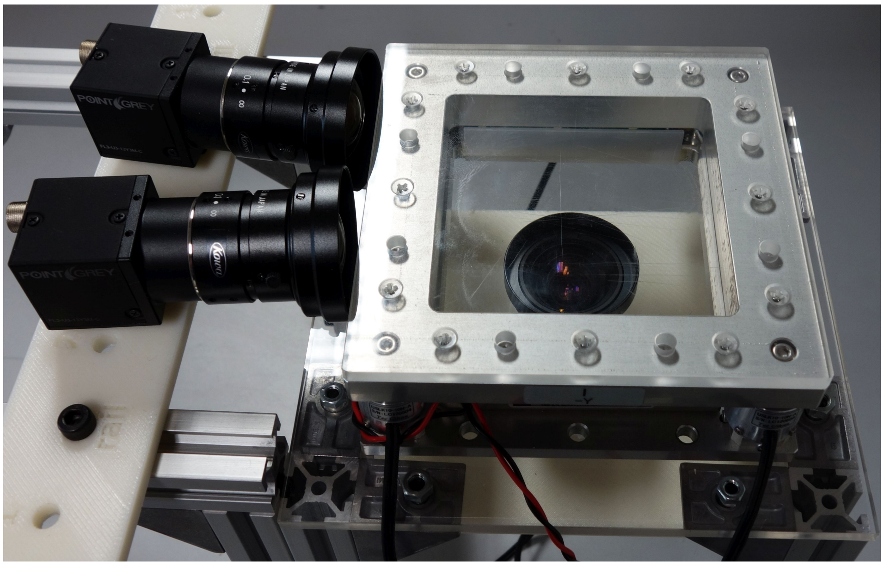


Figure 12. The measurement setup with cameras and force sensors

$$\Delta Z = \frac{(80mm)^2}{3.5mm \cdot 52mm} \cdot 4.8\mu m \approx 169\mu m$$

Images were recorded at frame rates of 150 Hz. As the region of interest on each camera's sensor is small (since the lower half is always obscured by the force sensor), finetuning the configuration of the cameras allows even higher frame rates, at the cost of signal noise. This frame rate was chosen as a compromise between image quality and frequency.

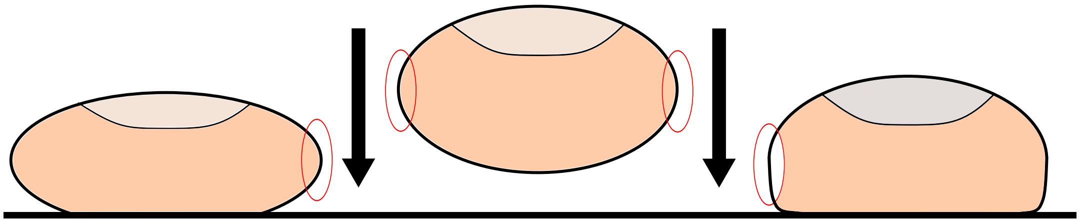


Figure 13. Possible deformation patterns of the fingertip

### 2.4.3 Evaluated parameters

It is evident that the most flexible part of the finger are the fingertips' bottoms, where most of the deformation will take place. When pressure is applied, fluid displacement inside the fingertip causes the lateral side to dilate. One might expect the overall curvature of the fingertip's side to decrease as it is pressed down on to an object, but the evolution of the fingertip shape and the curvature is a priori unknown (see figure 13). To quantify the curvature numerically, the fingertip surface is approximated by a paraboloid. The maximum curvature of a paraboloid $f(x) = a_0 + xa_1 + ya_2 + x^2a_3 + xya_4 + y^2a_5$ can be found easily by transforming its coordinate system to lie in its center at an angle such that the xy term disappears. It then equals $curvature_{max} = max(a_3, a_5)$.

A weakness of this approach is that the choice of points used for the approximation heavily affects the properties of the resulting paraboloid, although control over the number of points is limited, and the exact locations of the points on the fingertip cannot be easily predicted.

A further parameter that could be evaluated is the finger's thickness. However, other fingers in the background can disturb the extraction of the "main" finger's borders, and thus easily falsify the measurement. Furthermore, the thickness would be measured in pixels of the camera image and then transformed into millimeters, which would have to include a corrective calculation that accounts not only for the distance to the cameras, but also for the projection error caused by the close view (the upper and lower border seen in the camera image are not the two points on the cross section with the highest thickness). As low fidelity and reproducibility would have been likely, this parameter was not considered

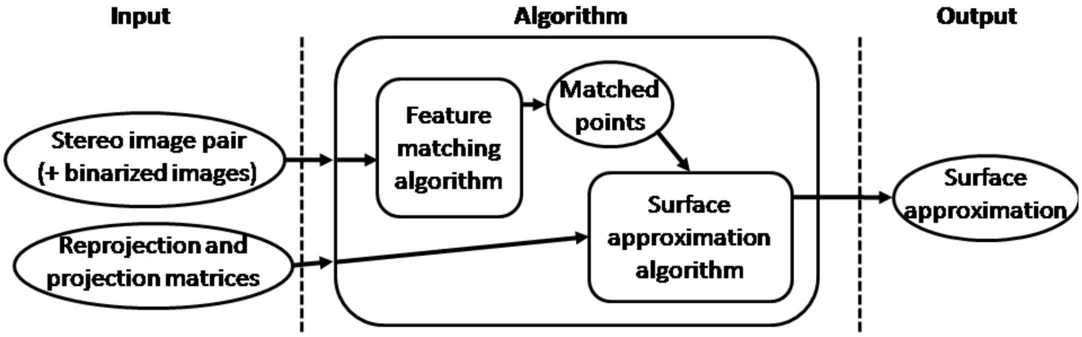


Figure 14. Structure of the proposed algorithm

further.

### 2.4.4 Proposed algorithm

The feature point matching algorithm accepts two rectified stereo images, their binarized versions and the reprojection matrix Q as input, and returns the approximation of the fingertip surface.

The algorithm can be split into the feature point matching and the iterative surface approximation. As seen in figure 14, the feature point matching algorithm passes matches on to the surface approximation algorithm. In the latter part, surface candidates are used to reject outliers in the matched set of points and to refine the approximation.

From the returned surface candidate, the curvature is extracted as described in section 2.4.3.

**Feature point matching** As mentioned in section 2.3.2, feature point matching is a highly empirical method primarily used for object recognition. It is an appropriate approach to the problem at hand, which consists of the recognition of just one object. It is also a more sophisticated method than block matching, and it stands to reason that it can be well-optimized to incorporate our prior knowledge about the scene. It is notable that fingerprint identification is also based on a form of feature point matching, which is more robust when presented with repeating patterns such as fingertip ridges.

This first part of the algorithm utilizes the rectified camera and binarized images, and returns the matched points found between the pictures.

The previous knowledge about the scene includes the very important information that only one visible object rotates in the view of the cameras, and all other parts of the scene are suppressed. This means that a) neighboring points are very unlikely to move very differently from one another, b) regions where points are hidden from view are scarce (e. g. at the edges of the object), c) due to the rectification, all visible points should be at the same height in the two pictures.

Going by the structure elaborated in section 2.3.2, the algorithm proceeds as follows:

1. Harris corner detection (excluding points where the binarized image is black)

2. Feature descriptor extraction

3. Exclude all matches with high y-disparity

4. Calculate feature similarity for the remaining matches

5. Match by nearest neighbor with a threshold cut-off

6. Calculate median x-disparity from current matches

7. Exclude matches with x-disparity too far from the median

8. Iterate the above three steps until no matches are excluded on the grounds of x-disparity

9. If too many matches are returned, increase the quality threshold and enter into step 5

The MATLAB implementation switches steps 3 and 4, as the calculation of feature similarity for all pairs is available in optimized code that outperforms the order described above, although it should normally be the faster option.

A FREAK descriptor was used for compliance with the standard in MATLAB version 2013a [38].

It was found empirically that if the number of matches is too large and too randomly distributed, the surface approximation algorithm fails to eliminate matches

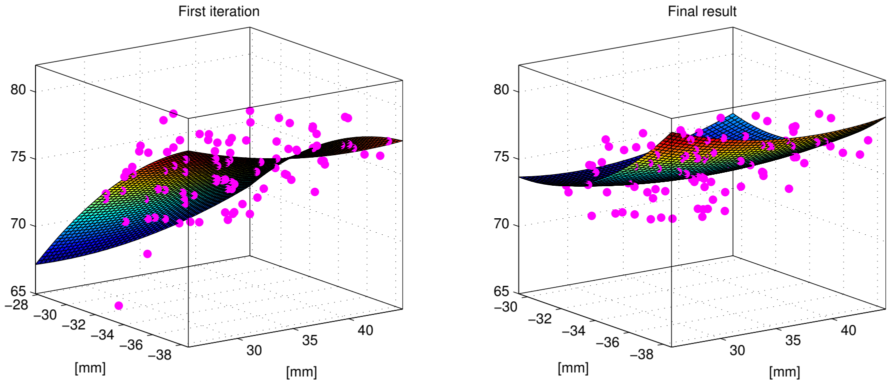


Figure 15. Successful iteration of a fingertip surface approximation

and to improve the approximation. To avoid pollution of the point set, step number 9 limits the number of returned matches to 120 pairs.

**Iterative surface approximation** For this method, it is assumed that the matches are unambiguous and that there are only few outliers that have to be recognized. In addition to the scene knowledge incorporated in the algorithm described in the section above, the surface approximation takes into account the relation of the two stereo cameras in space, as well as the presumed shape of the object (through the choice of the surface to be approximated).

This part of the algorithm takes as input the matched points in the two camera images, the projection matrix P1 and the reprojection matrix Q, and returns a paraboloid as a surface approximation. It proceeds as follows

1. Project 3D points from matches and their disparity (via reprojection matrix Q)
2. Fit a surface candidate $S'$ to 3D points
3. Using $S'$, project points from the surface as they would be seen in the left image (via projection matrix P1)
4. Calculate euclidean distance of projected points and real points
5. Discard outliers

6. Fit a new surface candidate $S'$ to the updated set of points

7. Repeat steps 2 – 6, accept candidate on convergence

It should be noted that the choice to project the 3D points into the left image is arbitrary, and the result would be the same if the right image (and the corresponding projection matrix P2) was used. This is due to the stereo information being contained within the disparity connected to each match.

The main parameters to tune are the cut-off threshold in step 5 and the convergence criterion used to exit the iteration. It is particularly problematic to reduce the cut-off threshold for the discarding of matches, as it is always a compromise between overfitting and tolerating incorrect matches. Moreover, if matches are too easily discarded, the algorithm does not converge until a lot of characteristic points are lost or there is very little heterogeneity in the data set.

In the current implementation, only points with a euclidean distance of over 4 times the standard deviation were discarded. To err on the side of caution, the iteration was not only stopped if no matches were discarded but also after a maximum of five iterations.

### 2.4.5 Validation via LIDAR

In order to assess the precision of the measured values, a mock run of the experiment was effected with a plastic replica of a human finger, and the results extracted were compared against a LIDAR scan of the replica's fingertip's side. A LIDAR sensor lends itself to this kind of test as it can have a very high depth resolution at the price of a low scanning speed.

All points from the LIDAR scan were exported as a point cloud and treated with MATLAB. Points used for the surface approximation were taken from a rectangle, drawn by hand on the fingertip side, to correspond with the region that the algorithm in question examined. Thus, the regions that both methods could use to extract points were the same. For comparison, the areas can be seen in figures 16 and 17.

The algorithm was applied to a number of images of the finger and the results plotted in figure 18. Each dot represents the result of one image. The red line represents the finger's curvature as measured by the LIDAR sensor. The standard

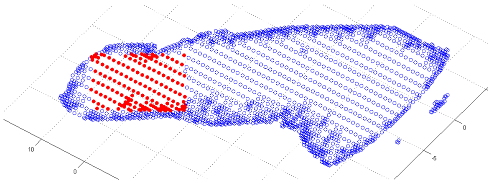


Figure 16. Points selected from the LIDAR point cloud $[mm]$

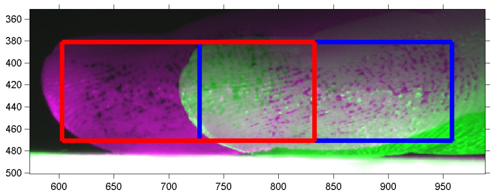


Figure 17. Regions in the stereo image pair from which feature points could be selected [pixel]

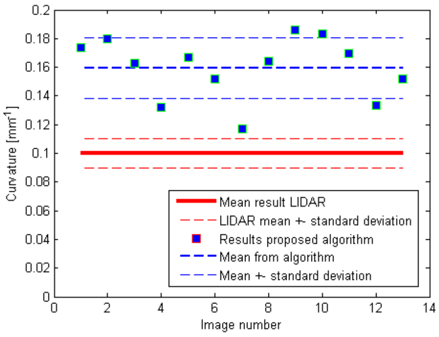


Figure 18. Curvature measured by LIDAR sensor and algorithm $[mm^{-1}]$

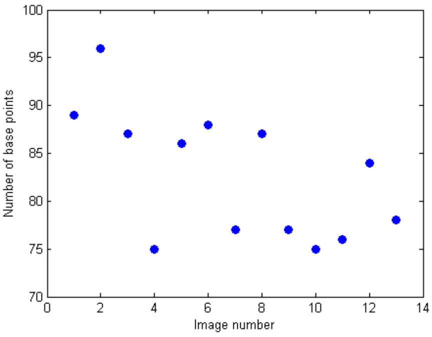


Figure 19. Number of feature points used by algorithm on LIDAR comparison frames

deviation on the curvature measured by the LIDAR sensor was 0.01 $mm^{-1}$ at a total of 3 measurements from different images. Additionally, an uncertainty of 0.008 $mm$ has to be taken into account for each 3D point.

The noise in the algorithm's results is about two times higher than in the LIDAR's. More notably, the algorithm overestimates the curvature systematically by about 60 %. This could be due to the selection of the region of interest, which affects results considerably. On the one hand, points near the top or bottom end of the fingertip are most affected by its curvature, but at the same time, those areas are badly visible and produce the most unreliable matches, as can be seen in figures 20 and 21.

The results of the validation are very sensitive to the choice of points that

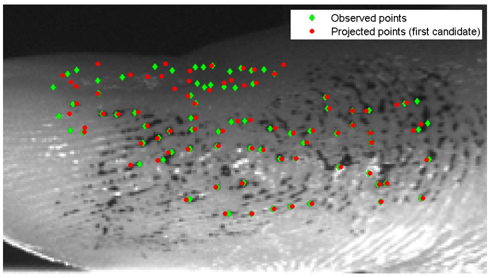


Figure 20. All feature points and their estimate's projection before iteration

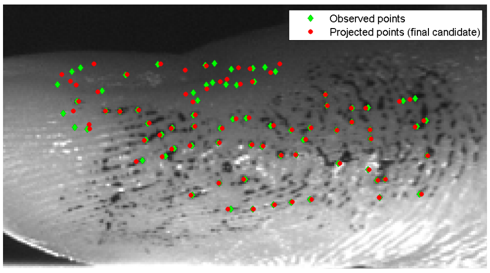


Figure 21. Feature points and estimated projections after iteration

are taken from the LIDAR image and included in the surface approximation. Because there were no precise markers on the fingers, the regions from which points were extracted could differ in size and position, and thus influence the resulting surface.

## 2.5 Experiment results

For the fitting of all models, the force data was linearly interpolated to the points in time when a frame was captured, i. e. the curvature was measured. About 10,000 frames were evaluated for the correlation calculations of each experiment.

For easier visualization, the contact forces were normalized to 1 in the graphs. All data is given to two significant digits.

### 2.5.1 Protocol

The procedure (excluding finger preparation) for each subject consisted of a first experiment, where only a normal charge was applied, and two further experiments that involved an additional lateral charge. In all experiments, the subject started at a hovering position above the acrylic plate. For the first experiment, the subject touched the acrylic plate and started to press down on it over the course of 3 seconds. The subject could lift or relax their finger after having reached the maximum force. In the two experiments that included a lateral charge, after pressing down on the acrylic plate, the subjects pulled their finger slowly and constantly to the left (in experiment 2) or right (in experiment 3). This resulted in a movement towards or away from the cameras, respectively.

Five runs of three different experiments were recorded for six subjects. Before the experiments were performed, four of the subjects lightly rolled their fingertip side on an ink pad, so that the contrast might be increased. The two remaining subjects used their fingers without any manipulation. In order to evaluate the effect of this approach, the goodness of fit statistics of the polynomial models were compared between the two subgroups (blank finger and inked finger). The subjects with inked fingers are number 2, 4, 5 and 6 in the tables.

The participants were healthy males between the ages of 18 and 25.

### 2.5.2 Normal charge (experiment 1)

This experiment, putting the finger under a normal charge exclusively, should paint the clearest picture of a correlation between curvature and normal force. To test this, the correlation coefficients[9] and the corresponding p-values were calculated, and a second-degree polynomial model was fitted to the measured curvature and normal force for each subject:

$$curvature(F_N) = a_1 {F_N}^2 + a_2 F_N + a_3$$

This allows a change in behavior over the range of normal force applied, which is to be expected, as the fingertip does not seem to deform visibly after a certain amount of force is applied, while avoiding overfitting.

**Correlation between normal force and curvature** Each subject's runs as well as the totality of all runs of the experiments were added separately to evaluate whether a correlation could be established between the normal force applied and the curvature that was measured. The correlation values reveal a highly significant, but only small linear correlation between the curvature and the normal force (see table 1). Moreover, the correlation values for different subjects can lie above and below 0. Despite the slightly positive correlation measured

[9] Assuming the force and the curvature to be variables of one observation, the correlation coefficients are calculated from the covariance matrix $C(i, j)$ with the formula $R(i, j) = \frac{C(i,j)}{\sqrt{C(i,i)C(j,j)}}$

Table 1. Correlation values for curvature and $F_N$ in experiment 1 (normal charge)

| **Subject** | **Correlation value (curvature, $F_N$)** |
|---|---|
| 1 | 0.19 |
| 2 | -7e-4 |
| 3 | 7e-4 |
| 4 | .043 |
| 5 | -.044 |
| 6 | .068 |
| All | .11 |

Table 2. Parameters and 95 % confidence bounds for $curvature(F_N) = a_1 {F_N}^2 + a_2 F_N + a_3$

| **Subject** | $a_1$ | $a_2$ | $a_3$ | 95 % ($a_1$) | 95 % ($a_2$) | 95 % ($a_3$) |
|---|---|---|---|---|---|---|
| **1** | -3.8e-05 | 0.006 | 0.17 | -5e-4, 5e-4 | 9.7e-4, .01 | .16, .18 |
| **2** | -0.0035 | 0.019 | 0.22 | -7.4e-3, 5e-4 | -.0034, .04 | .2, .25 |
| **3** | -8e-05 | 0.0042 | 0.32 | -1e-4, -5.7e-5 | .003, .006 | .31, .33 |
| **4** | 0.00036 | -0.0079 | 0.36 | 3.4e-05, 7e-4 | -.014, -.002 | .34, .39 |
| **5** | 0.00063 | -0.011 | 0.37 | -8e-4, .002 | -.026, .004 | .33, .4 |
| **6** | -0.00038 | 0.011 | 0.23 | -5e-4, -3e-4 | .0073, .01 | .21, .25 |

for all values, the variation between subjects implies that the correlation could not be conclusively distinguished from 0. This is discussed in more detail in the following section.

The confidence intervals and for the parameters of the curve fitting model are listed in table 2. It is helpful to keep in mind that the curvature could range from 0 to 0.5 $mm^{-1}$, and the normal force applied between 0 and around 20 $N$. It is also important to note that some confidence bounds cross zero, so one cannot be sure that those coefficients are different from zero.

It should be noted that for some subjects, a negative relationship between applied force and curvature is obtained (see fig. 23). This can be connected to excessive use of force, which can distort the data set. While many graphs give the impression of an acceptable fit (see fig. 22), one has to keep in mind that the expected curvature changes very little over the range of the independent variable.

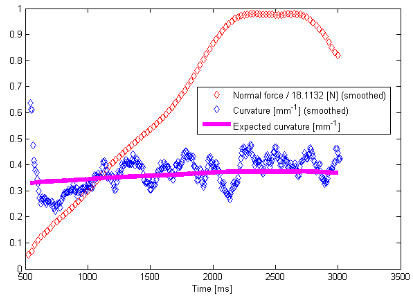


Figure 22. Modeled curvature with data from experiment 1 (normal charge)

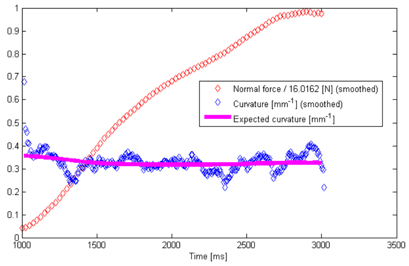


Figure 23. Modeled curvature with data from experiment 1 (normal charge)

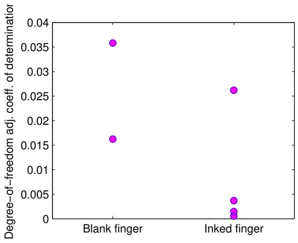


Figure 24. Goodness of fit (adjusted r-squared) for each subject in experiment 1

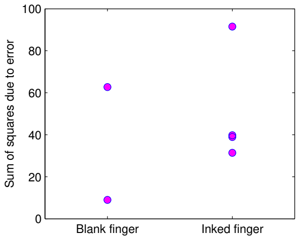


Figure 25. Goodness of fit (sum of squares) for each subject in experiment 1

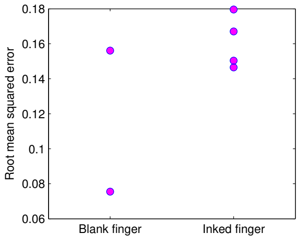


Figure 26. Goodness of fit (standard error) for each subject in experiment 1

**Comparison inked and blank fingers** The goodness of fit of the inked subjects' models was compared to those of the subjects with clean fingers in figures 24, 25 and 26. It appears that, instead of improving the contrast in the image and enriching the texture, rolling the finger on an ink stamp pad caused a risk of insufficient lighting and can make it harder for feature points to be detected and matched. This interpretation is supported by the sometimes spurious quality of matches seen in fig. 33.

Table 3. Correlation values between curvature and $F_N$, $F_T$ in experiment 2 (lateral charge to the left)

| **Subject** | **Correlation with $F_N$** | **Correlation with $F_T$** |
|---|---|---|
| **1** | 0.13 | -0.061 |
| **2** | 0.05 | -0.04 |
| **3** | 0.16 | 0.21 |
| **4** | -0.15 | -0.21 |
| **5** | -0.14 | 0.11 |
| **6** | 0.024 | 0.04 |
| **all** | 0.02 | 0.05 |

### 2.5.3 Lateral charge (experiments 2 and 3)

Charging the contact laterally changes the distribution of the fluid inside the fingertip pulp, which causes the deformation of the fingertip and changes the visible curvature in complex ways. Again, the correlation between the contact forces and the curvature was investigated. Due to the potential complexity of the mechanism, two polynomial models of degrees 2 and 3 were evaluated, with $F_N$, $F_T$ as independent variables and the curvature as the dependent one.

**Correlation between normal and lateral force and curvature** The runs for each subject were added and evaluated separately in the same way as described for experiment 1. It is noteworthy that, while the p-values are very low for the other correlations and experiments, the p-value for the correlation between curvature and $F_T$ in experiment 3 is at a relatively high 0.32. This implies that even the low correlation that was found might well be due to chance (see table 4).

The correlation values for $F_N$ in both experiments 2 and 3 are lower and less stable than in experiment 1 where only the normal force was evaluated. If this is not an effect of chance, it suggests that the measurement becomes less reliable when a general grasping situation is considered where the contact is charged laterally.

The second- and third-degree polynomial models taking into account both

Table 4. Correlation values between curvature and $F_N$, $F_T$ in experiment 3 (lateral charge to the right)

| **Subject** | **Correlation with $F_N$** | **Correlation with $F_T$** |
|---|---|---|
| **1** | 0.17 | -0.25 |
| **2** | 0.026 | 0.31 |
| **3** | -0.092 | 0.097 |
| **4** | -0.008 | -0.007 |
| **5** | -0.096 | 0.16 |
| **6** | -0.083 | 0.074 |
| **all** | 0.12 | 0.009 |

Table 5. Parameters for $curvature(F_N, F_T) = a_{00} + a_{01}F_T + a_{10}F_N + a_{02}(F_T)^2 + a_{11}F_TF_N + a_{20}(F_N)^2$ (Experiment 2) (lateral charge to the left)

| **Subject** | $a_{00}$ | $a_{01}$ | $a_{10}$ | $a_{02}$ | $a_{11}$ | $a_{20}$ |
|---|---|---|---|---|---|---|
| **1** | 0.072 | 0.0084 | 0.069 | -0.0038 | -0.0011 | -0.0099 |
| **2** | 0.16 | -0.036 | 0.13 | 0.0015 | 0.011 | -0.032 |
| **3** | 0.28 | 0.03 | -0.014 | 0.011 | -0.0098 | 0.0025 |
| **4** | 0.4 | 0.0024 | -0.015 | -0.014 | 0.0028 | 0.00078 |
| **5** | 0.31 | 0.21 | -0.15 | 0.031 | -0.18 | 0.098 |
| **6** | 0.27 | -0.055 | 0.015 | -0.0058 | 0.011 | -0.0033 |
| **all** | 0.3 | -0.02 | -0.0087 | 0.0073 | -0.0028 | 0.00095 |

normal and lateral force fare very similarly. Three goodness of fit statistics[10] are identical for both models within a margin of 5 %. It is thus reasonable to assume that the third-degree polynomial model performs not significantly better than the second-degree polynomial model, and preference should be given to the simpler solution. The parameters for the second-degree polynomial model are given in tables 5 and 8.

The sensitivity to movement to the right is visible in figures 28 and 29. There, the curvature decreases as the fingertip skin and liquid inside the finger is pulled towards the side viewed by the camera.

[10] Sum of squares due to error, standard error, adjusted $R^2$,

Table 6. 95 % confidence bounds for table 5 (part 1)

| **Subject** | $a_{00}$ | $a_{01}$ | $a_{10}$ |
|---|---|---|---|
| **1** | 0.046, 0.098 | -0.0032, 0.02 | 0.05, 0.087 |
| **2** | 0.03, 0.29 | -0.092, .021 | -0.045, 0.3 |
| **3** | 0.26, 0.31 | 0.0066, 0.053 | -0.026, -0.0012 |
| **4** | 0.35, .45 | -0.058, 0.063 | -0.033, 0.0039 |
| **5** | 0.17, .46 | 0.15, .28 | -0.38, 0.075 |
| **6** | 0.22, .32 | -0.086, -0.023 | -0.0058, 0.036 |
| **all** | 0.29, .31 | -0.027, -0.014 | -0.013, -0.0045 |

Table 7. 95 % confidence bounds for table 5 (part 2)

| **Subject** | $a_{02}$ | $a_{11}$ | $a_{20}$ |
|---|---|---|---|
| **1** | -0.009, 0.0013 | -0.0058, 0.0036 | -0.013, -0.0069 |
| **2** | -0.012, 0.015 | -0.016, 0.038 | -0.083, 0.019 |
| **3** | 0.0066, 0.014 | -0.014, -0.0056 | 0.0011, 0.0038 |
| **4** | -0.041, 0.013 | -0.0098, 0.015 | -0.0016, 0.0031 |
| **5** | 7.1e-4, 0.062 | -0.24, -0.11 | 0.0086, 0.19 |
| **6** | -0.014, 0.0026 | 0.0034, 0.019 | -0.0056, -9.6e-4 |
| **all** | 0.005, 0.0096 | -0.0048, -8.9e-4 | 3.7e-4, 0.0015 |

**Comparison inked and blank fingers** The goodness of fit statistics for the two types of finger preparation in both experiments are presented in figures 30, 31 and 32. There appears to be less of a difference between the blank and inked fingers in experiments 2 and 3, but this could be due to the quality of the fits, which is worse than during experiment 1. The results do not point towards the ink application improving accuracy, but they are overall inconclusive.

### 2.5.4 Irregularities and sources of error

The following things might have had an influence on the outcome of the experiments and should be noted.

- One subject used a lot of force and surpassed 50 N in normal force, while others stopped short of 5 N.

Table 8. Parameters for $curvature(F_N, F_T) = a_{00} + a_{01}F_T + a_{10}F_N + a_{02}(F_T)^2 + a_{11}F_TF_N + a_{20}(F_N)^2$ (Experiment 3) (lateral charge to the right)

| **Subject** | $a_{00}$ | $a_{01}$ | $a_{10}$ | $a_{02}$ | $a_{11}$ | $a_{20}$ |
|---|---|---|---|---|---|---|
| **1** | 0.072 | 8.4e-3 | 0.069 | -3.8e-3 | -1.1e-3 | -9.9e-3 |
| **2** | 0.16 | -0.036 | 0.13 | 1.5e-3 | 0.011 | -0.032 |
| **3** | 0.28 | 0.03 | -0.014 | 0.011 | -9.8e-3 | 2.5e-3 |
| **4** | 0.4 | 2.4e-3 | -0.015 | -0.014 | 2.8e-3 | 7.8e-4 |
| **5** | 0.31 | 0.21 | -0.15 | 0.031 | -0.18 | 0.099 |
| **6** | 0.27 | -0.055 | 0.015 | -5.8e-3 | 0.011 | -3.3e-3 |
| **all** | 0.3 | -0.02 | -8.7e-3 | 7.3e-3 | -2.8e-3 | 9.5e-4 |

Table 9. 95 % confidence bounds for table 8 (part 1)

| **Subject** | $a_{00}$ | $a_{01}$ | $a_{10}$ |
|---|---|---|---|
| **1** | 0.081,0.2 | -0.086, -0.0097 | -0.071, 0.07 |
| **2** | 0.21, 0.3 | 0.12, 0.17 | -0.028, 0.036 |
| **3** | 0.085, 0.23 | -0.13, -0.051 | 0.044, 0.17 |
| **4** | 0.37, 0.9 | -0.4, 0.14 | -0.72, -0.044 |
| **5** | 0.02, 0.36 | -0.063, 0.2 | -0.071, 0.31 |
| **6** | 0.25, 0.34 | -0.04, 0.022 | -0.0031, 0.02 |
| **all** | 0.26, 0.27 | 0.019, 0.032 | 0.0047, 0.0082 |

- Inked fingers were sometimes badly illuminated
- The lighting was changed slightly after the experiment was recorded with subject 1, so that the finger would also be illuminated from above

## 2.6 Discussion

### 2.6.1 Correlation between contact force and curvature

It was found in all three experiments that the bare correlation between the contact forces and the curvature measured was most often small, and could change sign even in different runs of the same subject. There can be multiple reasons for this:

1. the phenomenon can take on different forms in different subjects

Table 10. 95 % confidence bounds for table 8 (part 2)

| **Subject** | $a_{02}$ | $a_{11}$ | $a_{20}$ |
|---|---|---|---|
| **1** | -0.0035, 0.011 | -0.0071, 0.035 | -0.017, 0.023 |
| **2** | 0.028, 0.054 | -0.014, 0.018 | -0.00036, 0.0004 |
| **3** | -0.085, -0.054 | -0.062, -0.027 | -0.047, -0.019 |
| **4** | -0.023, 0.096 | -0.071, 0.23 | -0.0015, 0.21 |
| **5** | 0.0004, 0.049 | -0.053, 0.073 | -0.069, 0.032 |
| **6** | -0.0049, 0.0048 | -0.0029, 0.0039 | -0.0012, 0.00028 |
| **all** | 0.00048, 0.0056 | -0.00025, 0.0013 | 2.1e-05, 7.5e-05 |

2. the phenomenon is too complex to be described by a linear relationship or the parameters tested are out of the usable range

3. there are confounding variables that have not been considered or measured

4. there is little or no connection between the curvature and contact force

5. one or more of the measurements is plagued by noise

All of these reasons shall be considered.

**1. The phenomenon takes on different forms in different subjects** This explanation sounds feasible, since all human bodies are different. However, the correlation coefficient should then stay roughly the same for each subject, and only change between different subjects. The data shows that correlation coefficients can change sign in different runs of a single subject.

**2. The phenomenon is too complex to be described by a linear relationship or the parameters tested are out of the usable range** There is some evidence for this explanation: the deformation of the fingertip seems to reach a point of saturation or maximum deformation at a fairly low charge. At higher charges, trembling and repositioning of the finger would distort the measurement. It is a reasonable assumption that only a small range of fingertip force would correlate with the deformation, but this range is not easily determined.

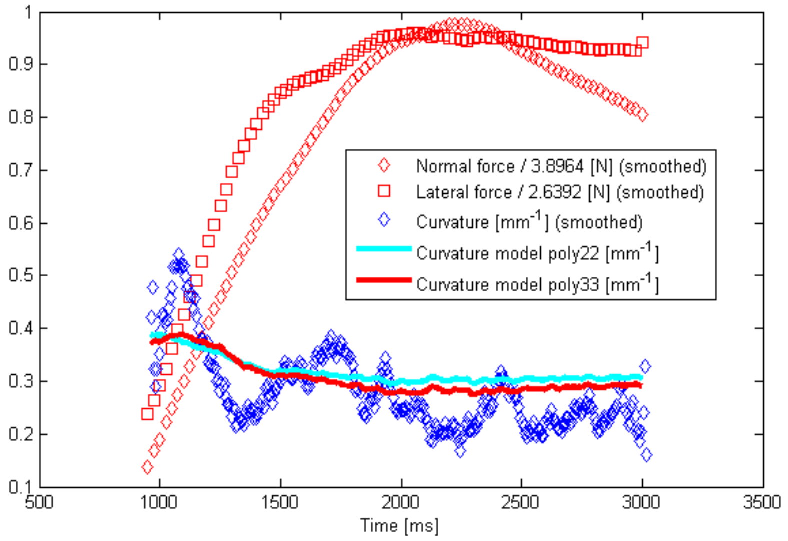


Figure 27. Two-variable curvature models with data from experiment 2 (lateral charge to the right)

**3. There are confounding variables that have not been considered or measured** The third idea, that some other confounding factor could influence the deformation of the fingertip, cannot be discarded off-hand. Indeed, one could envision temperature differences or physiological processes modifying the behavior of the dermis and thus influencing the measurement. However, the effect would have to be sizable enough to explain why the correlation coefficient would change its sign. Secondly, the unconsidered factor would have to change between runs recorded by the same subject. As all tests for each subject took no longer than 20 minutes, the experimental conditions are unlikely to have changed during that time. Finally, the coefficients for some of the subjects showed multiple sign changes in subsequent runs, which makes this explanation unlikely.

**4. There is little or no connection between the curvature and contact force** There is the possibility that curvature and contact force are not as intertwined as assumed. Previous work has shown that the fingertip deforms under a line load, and the commercial sensor mentioned in section 2.2.2 even exploits the deformation of the fingertip to measure the normal force. However, the

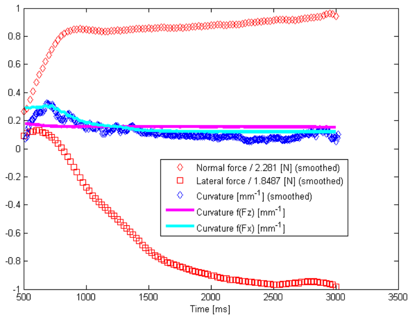


Figure 28. Simple curvature models with data from experiment 3 (lateral charge to the right)

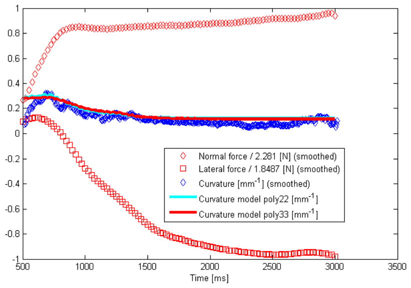


Figure 29. Two-variable curvature models with data from experiment 3 (lateral charge to the right)

normal force might not be reflected in the *curvature* of the deformed fingertip, rather than some other aspect of its deformation.

**5. One or more of the measurements is plagued by noise** There are good arguments to make for the conclusion that one of the measurements, specifically that of the curvature, is too unreliable and the results too noisy to make out a correlation, even if there is one. Apart from the calculated correlation coefficients, other signs that point towards this interpretation are the low amount of matches returned on many images, some of which fail later quality tests outright (see fig. 33). Another sign is that the fitted models do not change the curvature very much with the force, and barely exit the range of its standard error, as seen in the LIDAR validation in section 2.4.5. One could ask the question if the curvature genuinely does not change further than the tight boundaries that were observed - however, in that case the current measurement setup is not suited due to its high error.

The evidence that at least one part of the extraction of the curvature from the fingertip underperforms is plenty. What are the problems, and how could they be addressed?

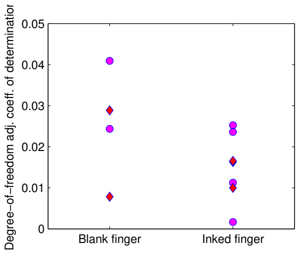


Figure 30. Goodness of fit (adjusted r-squared) for the models of each subject in experiment 2 (magenta) and 3 (red)

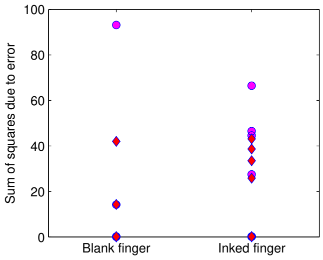


Figure 31. Goodness of fit (sum of squares) for the models of each subject in experiment 2 (magenta) and 3 (red)

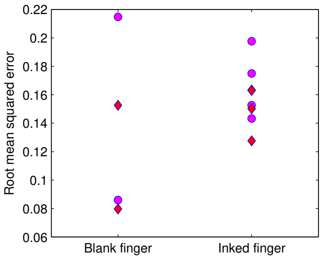


Figure 32. Goodness of fit (standard error) for the models of each subject in experiment 2 (magenta) and 3 (red)

### 2.6.2 Problems in the measurement setup and algorithm

The measurement of the fingertip curvature starts with the image acquisition, continues with the algorithm described in section 2.4.4, and ends with the mathematical extraction of the maximum curvature of the paraboloid surface approximation. While the theoretical depth resolution of around 170 $\mu m$ could be better for a phenomenon of this size, it remains at only $3-5$ % of the total measured range. The only way to improve the depth resolution outside of better technology would be a rearrangement of the stereo setup, which might exacerbate the difficulties posed by the large camera angle even further. As that would introduce even more difficulty to the matching process however, it is fair to say that the main source of error in the pipeline lies in the algorithm itself.

Its shortcomings could be described as follows:

- The region of interest (ROI) chosen for feature point detection is arbitrary and is prone to influence the result immensely

The fact that the position of the ROI can influence the result heavily while being very hard to localize precisely on the finger is a fundamental challenge to this approach. At the same time, the points near the top and bottom of the fingertip are the ones that carry the most information about the curvature[11], and the ones

[11] As the points near the top and bottom of the fingertip center are further from the fingertip center, they affect the quadratic term most.

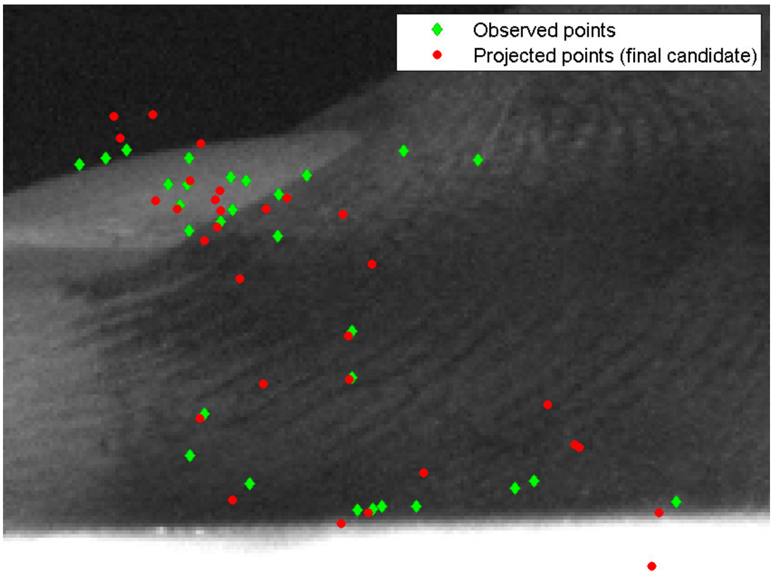


Figure 33. A failed matching landscape and the associated projected points

whose visibility is worst and whose position is thus badly defined.

- The unpredictable distribution of matches inside the ROI destabilizes the resulting surface approximation

On top of the fact that the placement of the ROI would affect the result even if the points inside it were distributed evenly, in reality the distribution of matches is extremely uneven, and depends on the texture richness of the fingertip and the feature detection and matching (see fig. 33). This further reduces the approximation quality and is a shortcoming of the basic nature of the feature matching algorithm employed. If no effort to create a dense disparity map is made from which to draw information, the regional density of matches is always a source of error.

- The fingernail is visible in some images and is not excluded from the allowed matches

Although the fingernail is one of the most rigid parts of the fingertip and will not deform appreciably during everyday tasks, feature points found on the fingertip are included in the surface approximation. These distort the result, as the fingernail does not have the shape of the surrounding skin. Care should be taken to include only points that lie on the skin of the finger. This leads conveniently into the next problem.

- The finger is not properly binarized

In the interest of simplicity and reliable localization of the finger in the image, the binarization routine focuses on finding the accurate contour of the bottom side and the front part of the fingertip. As the top side might be badly lit and other fingers can be in view of the camera, a rough overestimation is used for the top side of the fingertip. A more accurate binarization of the finger would come at a higher computational cost, but would allow better sorting of potential matches.

- The cleaning of matches during iterative surface approximation is insufficient. This is a multi-part problem:
  - The discarding of bad matches during surface approximation is too hesitant
  - There is no satisfactory criterion for convergence during surface approximation

As described in section 2.4.4, there is an inherent risk in reducing the threshold for the discarding of questionable matches. An excessively low threshold would discard too many matches, and not converge before the shape of the approximated surface is damaged and precision is lost. The current threshold is likely too high, and does not cull enough bad matches.

- The matching worked better for the finger replica used during LIDAR validation than for the subjects' fingers

It can be seen on a considerable number of images that the algorithm's performance in real situations is lower than it was during the validation (see figures 33 and 21). This points towards either the validation being inadvertently too easy, or the experimental images being of a drastically lower quality than during validation. Insufficient lighting may account for a part of the problem, as well as the reflective surface of the finger replica, which would make the image more rich in features than for human fingers, and thus better suited for analysis by the proposed algorithm.

- The contrast of the camera image was not monitored and optimized for inked fingers

During image acquisition, the cameras were not in manual mode and adjusted their exposure automatically, taking into account the brightness of the whole image, while only that of a small region of interest mattered to the algorithm. This seems to have lowered the contrast seen on the inked fingers, as their brightness values were pushed towards the edges of the available spectrum.

### 2.6.3 Avenues for improvement and associated problems

One way to improve the algorithm lies in more robust feature detection and matching. One avenue would be to apply techniques used in fingerprint classification to identify the ends of ridges and other characteristics of the skin. This could result in more robust features, at the cost of lowered density.

The performance on inked fingers might be improved by adjusting the brightness of the cameras with respect to the region of interest instead of the whole image. The region of interest, for which the contrast would have to be increased, should include only the fingertip side, and not the surrounding measurement apparatus and background noise.

An immediate option to improve the quality of the measurement would be to discard matches more rigorously during the surface approximation. The risk of the excessive discarding of matches should be avoided by forgoing the evaluation based on standard deviation. Specifically, a static cut-off threshold relaying real-world measurements to the model could be tested, since the euclidean distance being evaluated is already normalized by the reprojection matrix.

A more appropriate convergence criterion for the surface approximation phase of the algorithm could be constructed out of a combination of a) the current standard deviation of the set of points, b) the difference in standard deviation between the current and the last iteration and c) the number of discarded points by the last iteration. A well-chosen convergence criterion that assures the integrity of the surface estimate would, in return, allow a more radical match culling.

To check against failure of the surface approximation, the higher-order parameters of the last surface candidate could be evaluated to ascertain that the surface is a paraboloid and has no saddle point.

Another promising option would be to mend the surface approximation part of the algorithm with the feature matching stage. Instead of first sorting matches with a rough idea of which geometrical properties would be unacceptable (such as an extreme x-disparity) and then applying the previous knowledge about the shape of the object during the surface approximation, all the assumptions about the scene would work at the same time and produce the most appropriate matches. A previous version of the algorithm did not mix the geometrical sorting described in section 2.4.4 with the nearest neighbor matching that was formerly performed inside a MATLAB stock function. However, the improvements in match quality and number were considerable. If the synthesis of sorting criteria were to create a comparable jump in performance, then this would probably be the best approach.

A possible alternative route to retrieving the surface of the finger is to use an intensity-based image registration method. If a dense disparity map of the finger were available, one could evaluate it at evenly distributed points on the fingertip, and thus eliminate one source of instability. However, one of the bigger challenges for methods like block matching algorithms lies in repeating patterns, like the ridges of the fingertip. The current MATLAB implementation of image pyramiding and dynamic block matching was tested and did not deliver promising results.

It would be prudent to test the performance of any future version of the algorithm on multiple objects of different curvature and surface structure (reflectiveness, brightness, texture), in order to better predict the extent of the improvements.

Finally, one could optimize the lighting by projecting structured light onto the finger and either make it easier for the stereo matching algorithm to work, or calculate the shape directly via the deflection of the stripe. As the object is round and without holes, this could provide decent results. The use of a special light source might restrict the applicability too much to be generally useful, however.

## 2.7 Conclusion

We showed that stereo images of the human fingertip can be used to extract information about the shape of its lateral surface. With the presented camera setup, a theoretical depth resolution of 0.17 mm is achieved, which at results in a minimum uncertainty of 3 – 5 % for each measured point on the finger at a total range of interest of about 5 mm.

An algorithm based on feature point matching was proposed, which approximates the 3D shape of the fingertip via a paraboloid. The algorithm's performance was found to be low, as the stereo image of the fingertip presents several difficulties, such as the homogenous and repetitive texture and the high baseline distance required for a sufficient depth resolution.

The curvature of the surface approximation at the extremal point was extracted and tested for correlation with the normal and lateral fingertip force. While a slightly positive correlation between the normal force and curvature was found for the sum of observations, this was not strictly true for all subjects, and no significant correlation with the lateral force was found. While no direct correlation between contact force and fingertip curvature can be established on the basis of the measured values, the change in deformation is strongest at low force values.

We conclude that 1) if a force can be measured, then lower forces will be easier to distinguish, and 2) a different measurement mechanism than external cameras will be appropriate.

# 3. Textile sensing

## 3.1 Introduction

When manipulating objects, humans make extensive use of haptic sensation and feedback. Accordingly, reproducing the human sense of touch in robots has been the subject of numerous research and engineering efforts. While cameras are the most common sensor in modern autonomous robots interacting in unstructured environments, haptic and tactile feedback offers a way to gather data that would otherwise be inaccessible, such as the state of a grasp that is occluded by the gripper or a container *e.g.* when trying to grasp an object in a duffle bag). The ability to handle and distinguish textiles supports robots in the fulfillment of daily living tasks such as laundry and bed-making, as well as in commercial applications in unstructured environments, such as gift wrapping or covering objects.

Bohg *et al.* [7] explore the notion that interacting with the environment creates rich signals that would otherwise not be present, which affords learning algorithms not only the use of these signals, but combining the action and the resulting perception. This concept comes natural to us when it comes to touch, where making contact, grasping and forcefully interacting with an object gives rise to unique signals that humans learn to interpret skillfully. Indeed, Lederman and Klatzky [42] observed that one of the ways in which humans explore an object's properties is to evaluate the object's roughness by dragging their fingertips over its surface.

In this chapter, we present an active perception strategy: we investigate if moving the fingertip over the surface of a material can be used to distinguish between and classify textiles[12], by using a biomimetic back-and-forth fingertip motion to extract tactile signals from a grasped textile as shown in Fig. 45. We also investigate which motion and sensing parameters affect the recognition results and propose a method to discriminate textiles while they are grasped and during manipulation.

We show experimentally that our method can distinguish not only between different textiles, but also between multiple and single layers of the same material,

[12] We use the term ”textile” to mean thin, easily deformable objects with reversible deformation, including ones such as plastic and aluminium foil.

which is to the best of our knowledge the first time in robotic general purpose grippers. Our data set contains over 30,000 samples recorded on 18 different materials, explored at 3 different speeds and 2 different levels of force. Furthermore, we compare the performance of state-of-the-art classifiers as well as the effect of sampling rate, exploration speed and force on the recognition rate.

We also propose an open-source gripper design with two force sensors based on the Yale OpenHand M2 [43], which we have developed specifically for the manipulation of textiles, and which is equipped with only a 3-dimensional force sensor. We show that data from this 3D force sensor, even at low frame rates, is sufficient to distinguish everyday textiles. [44]

In summary, in this chapter we contribute:

- a novel method and algorithms to recognize a textile in-hand
- the comparison of multiple state-of-the-art classifiers on new tactile data
- the designs for the hardware required to perform both recognition and manipulation

We start by outlining how related works have explored material recognition using tactile sensors. Afterwards, we describe the hardware and method we use to discriminate textiles, detailing the design of the open-source gripper we propose to perform the motion. In the Experiments section, we explain the experimental setup and procedure used to evaluate our method and the performance of related methods on our data. Finally, we present and discuss the results we obtained.

### 3.1.1 Related Work

In this section, we first introduce related works that have attempted object manipulation and recognition with tactile sensors. Ordering by the type of sensor that was used, we describe *camera-based*, *acceleration-based*, *force-based* and *multimodal* sensor approaches. For the record, our setup uses a force sensor, and a review on tactile sensing technology can be found in Dahiya *et al.* [45].

**Force sensors** Kaboli *et al.* [46] use the same model of sensors as in our work, mounted on a 3-finger gripper to detect slip and control the grasp when manipulating deformable objects. They detect the slip by a sudden change in tangential force, determine the friction coefficient from it, and then use it to regulate the grasping force on each sensor. Their results support the assumption that this type of sensor yields sufficient information for slip detection and textile manipulation.

Strese *et al.* [47] proposed a database of 43 textures which they examine with a vibrotactile display and from which they attempt to extract features. Mukaibo *et al.* [48] proposed a tactile sensor with a structured fingerprint and embedded strain gages to distinguish textures based on friction and roughness. Boissieu *et al.* [49] identified 10 kinds of paper by exploring them with a MEMS force sensor element covered with a rubber skin. Schöpfer *et al.* [50] used a tactile sensor array recording at 1800 Hz to control slippage and attempted the classification of 5 surface textures.

In the CloPeMa project[13], Le *et al.* [51] developed a gripper for textile manipulation and recognition which contains a tactile sensor and can perform a rubbing motion. They use its tangential force measurement to estimate an appropriate grasping force on their items of clothing. While Le *et al.* [52] mention that the rubbing motion was intended for supporting the system's textile recognition, the authors were not able to locate a publication about this.

**Camera-based sensors** Visually tracking the inside of the contact surface has been used in a number of tactile sensors, such as the *TacTip* series of sensors summarized in Ward-Cherrier *et al.* [53], or the *FingerVision*, a transparent, flat gripper surface by Yamaguchi and Atkeson [54] which allows the observation of the grasped object, or the *GelSight* [55] which additionally tracks surface deformation via lateral projection of colored light. All of these approaches use cameras whose frame rate imposes a limit on the frequency of the signal they are able to extract from the contact interface. The sensor matrix may also dampen high frequency signals even if using a high-speed camera.

**Accelerometer-based sensors** Ho *et al.* [56] proposed a sensor knitted from conductive yarn, intended to detect the micromovements during the beginning of slip events, and draped over a steel ball. They test the sensor on three fabrics using different signal processing approaches, reporting success rates of up to 90 %

[13] Clothes Perception and Manipulation. http://www.clopema.eu/

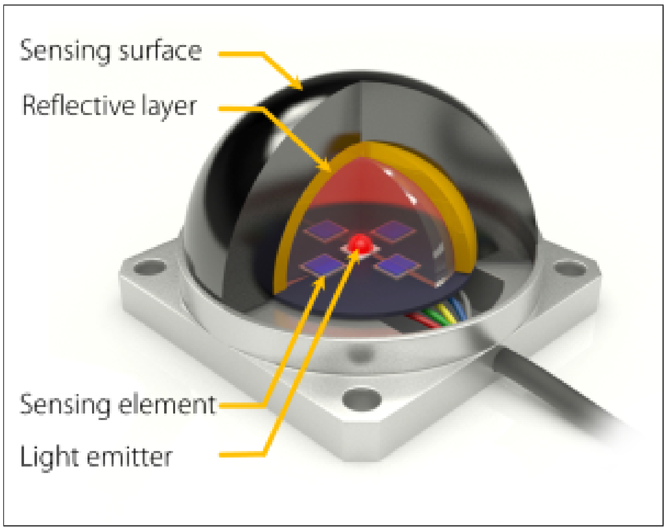


Figure 34. Structure of the tactile sensor. Figure adapted from [4]

using a feature vector constructed from Discrete Wavelet Transforms of their signal.

Sukhoy *et al.* [57] attached an accelerometer to an artificial finger nail and performed a scratching motions at different speeds and directions on 20 different surfaces. After five exploratory motions, they report a recognition rate of 80 %. Using a feature vector constructed from a spectro-temporal histogram of the magnitude of the acceleration, they applied different machine learning techniques to the data.

Giguere and Dudek [58] attached a tactile probe to a mobile robot, identifying different kinds of floor covering and using it as a weak form of localization.

## 3.2 Multimodal sensors

Fishel and Loeb [59] investigated a large amount of textures (117 materials) using a BioTac sensor mounted on a lever to apply a normal force, with the texture fixed to a linear stage to induce a movement. At a large number of different movement speeds and normal forces, they recorded three parameters that they related to "Traction", "Roughness" and "Fineness" of a texture, and which they postulate to be roughly orthogonal. These three dimensions are informed by the language with which humans describe textiles, which in turn should be based on the extensive experience of the human brain (an exceptional classifier) with textiles. They propose the use of Bayesian exploration (choosing the subsequent exploratory motion predicted to minimize the remaining uncertainty) for the classification of textures and report an overall recognition rate of 95.6 % after up to 10 exploratory motions. They use both the BioTac's sensor modalities and the current supplied to the linear stage as a measure of the tangential force, and chose a range of 36 exploratory motions mimicking the ranges typically used by humans (1 – 10 cm/s, 0.2 – 2 N).

Kaboli *et al.* [60] used a shadow hand with 5 BioTac sensors to identify an object held in hand by moving a fingertip over its surface. They proposed to use Hjorth parameters as tactile features, reporting an average 97 % recognition rate.

Tanaka *et al.* [61] explored cups made of different materials with a BioTac sensor, applying active learning on a 5-element feature vector obtained from the sensor's different modalities. They generated optimal actions from a set of dynamic motion primitives using object manifold learning.

Tada *et al.* [62] created an artificial fingertip by dispersing strain gauges and polyvinylidene fluoride films in two silicon rubber matrices of different stiffnesses, and used it to discriminate textures. Similarly, Jamali *et al.* [63] dispersed strain gauges and piezoelectric vibration sensors in an artificial finger, mimicking the human fingertip's slow- and fast-acting nerve cells, and moved it over different materials. They extracted local maxima of the recorded frequency spectrum as features, and report success rates of 80 % using one sample, and 95 % using several samples.

## 3.3 Texture benchmarks

While numerous groups have experimented on different numbers and classes of surfaces, as of yet no set of benchmark textures has been agreed upon, which makes the comparison between different works difficult. The difficulty in establishing such a benchmark lies not only in the selection of an appropriate set of textures, but also, more practically, in its accurate description. While some descriptive parameters are known to textile manufacturers or international standards[14], the parameters are not publicly known for commercially available objects.

[14] *e.g.* ISO 11036 for sensory analysis

Accordingly, most groups describe their specimen qualitatively (*e.g.* ”carpet”, ”coarse cotton”, ”rabbit fur”), allowing for at least a rough classification and internal consistency.

## 3.4 Hardware & Gripper Design

Both pushing and light grasping are important subtasks in the manipulation of deformable objects and textiles. Pushing (tucking) textiles into small openings is a challenging task in the manipulation of textiles and thin objects, which poses an obstacle to the automation of a number of manufacturing tasks. However, current robot grippers are almost exclusively designed for grasping objects, or imitate the human hand with very fine mechanisms that break easily, so that they cannot withstand the loads required by the pushing task. Additionally, many grippers cannot detect if a textile has been grasped, due to the thinness and deformability of textiles. Lastly, many grippers are not compliant, so that it is hard to realize a light grasp (one that applies little force). An overview of common grippers and their features is listed in Table 11. Finally, as textiles are very thin, grippers with pressure pads cannot easily confirm if a textile was grasped successfully or if the gripper is empty.

To advance in the automation of textile manipulation tasks, a gripper is desirable that is able to perform both sensitive precision grasping as well as the tucking task. In this chapter, we present the *NAIST OpenHand M2S*, a gripper design that can:

- detect the successful grasping of thin objects via the rubbing motion we propose
- sustain significant pushing loads in order to perform tucking tasks
- perform grasps with light forces to allow textiles to slide through its fingers

. The design is open source and can be 3D printed. All CAD files, assembly manuals and code reference can be found online at https://github.com/naist-robotics/naist-openhand-m2s.

It is based on the Yale OpenHand M2 gripper [43], an under-actuated gripper with two degrees of freedom on one finger, and a thumb without joints. We use the thumb to tuck textiles into cracks, and added two 3D force sensors to the hand's fingertips. This allows not only the compliant grasping of objects and textiles, but also to grasp with small amounts of force let a textile slide between the gripper's fingers. We also use the sensors to evaluate the grasp success of textiles by using a rubbing motion: the index finger contains two joints, which allows the fingertip to be rubbed against the rigid thumb.

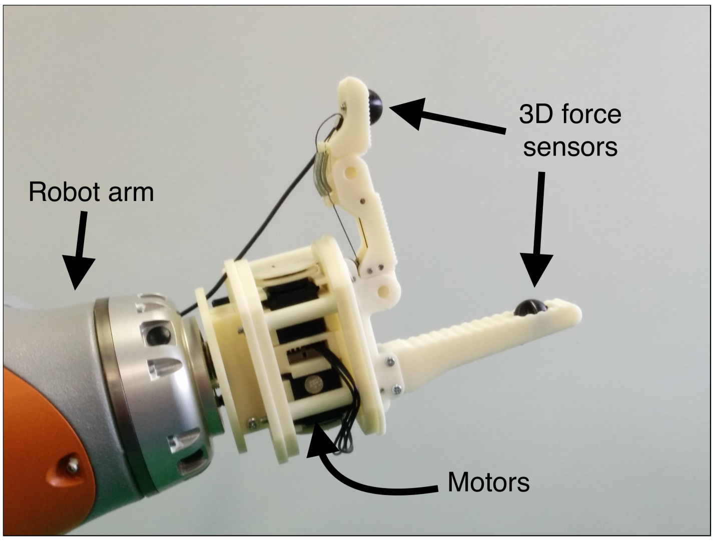


Figure 35. First prototype of the proposed gripper with two force sensors and a rigid thumb

## 3.5 Design & Hardware

### 3.5.1 Approach

We consider textile manipulation to consist of elementary manipulations such as grasping, sliding and tucking.

- *Grasping* creates a temporary, fixed connection between one or more points and the grasping surface of the grippers (*e. g.* the fingertips).

- *Sliding* is considered to occur when relative movement between the object and the grasping surface takes place, but the grasp is not necessarily lost after the operation, as shown in Fig. 37 and Fig. 41.

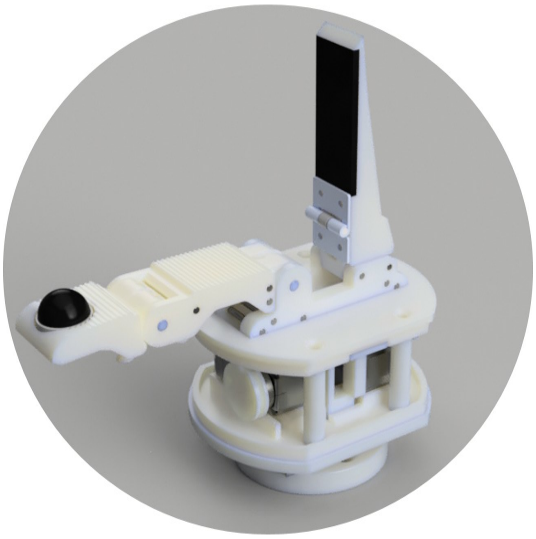


Figure 36. Second version of the gripper with one force sensor and a compliant thumb with a hinge joint, allowing it to be used on robots without torque control

- *Tucking* is the insertion of part of a textile into a small opening or crack by pushing onto the textile, as shown in Fig. 39, Fig. 40 and Fig. 42.

We enable all three of these elementary manipulations with our gripper design [44].

### 3.5.2 Gripper

The gripper shares most of its basic characteristics with the M2 gripper, such as one rigid thumb, a 2-DOF finger with an agonist and antagonist tendon arrangement, and a base with actuating motors. The agonist motor closes the second finger joint and results in an underactuated grasp, while the antagonist motor results in a fully actuated grasp where the second finger joint does not close.

In contrast to the Yale OpenHand design, we go to some lengths to reduce the friction in our wires and simplify the production process. For example, we added bearings with an outer diameter of 5 mm to each finger joint. With decreased friction, the force of the springs retracting the finger can be significantly reduced, which lowers the load on the motors that have to counteract the spring force during operation. While the bearings can be omitted, they make a significant

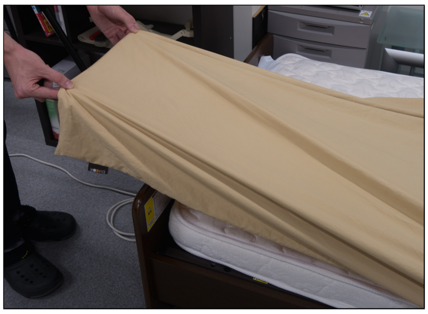


Figure 37. Human pulling a textile taut with two hands and light grasp force during bed making

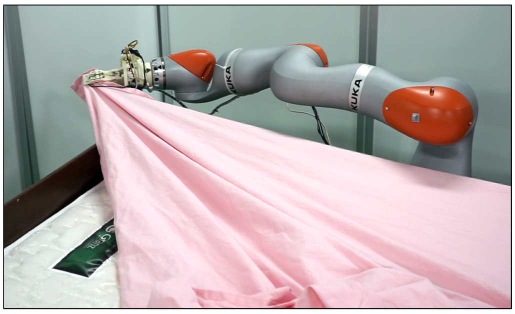


Figure 38. KUKA LBR iiwa with a prototype of the proposed gripper pulling a bed sheet over a mattress

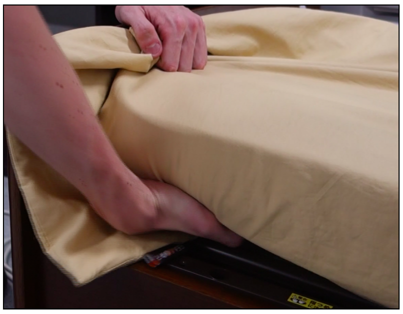


Figure 39. Human tucking a bed sheet in between mattress and frame during bed making

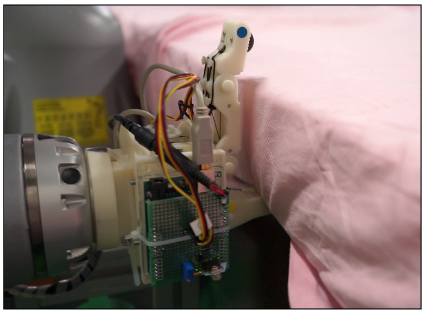


Figure 40. KUKA LBR iiwa with a prototype of the proposed gripper tucking a bed sheet under a mattress

difference when using less costly servo motors.

We also omit any rubber surfaces and hybrid deposition techniques, which makes it simpler to print and assemble than the original design. Our proposed design consists of only one 3D printed material and commercially available parts.

Another important addition to the original design are the two force sensors which allow us to perform tactile recognition with the method we describ in scetion 4.3. For the tactile sensor, we use hemispherical Optoforce 3D force sensors[15] with a high-friction rubber surface. While each sensor measures only three scalar force values at 1000 Hz, the whole surface of the sensors is sensitive

[15] OMD-20-SE-40N

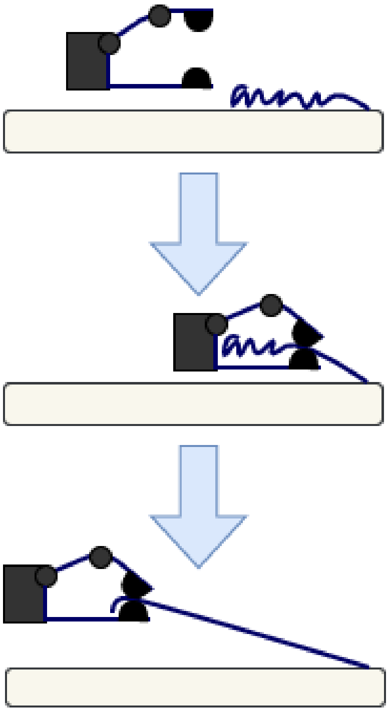


Figure 41. Representation of a textile being pulled taut while sliding through the gripper's fingers

and contributes to the force reading. The sensors function by measuring the deformation of a half dome of nitrile rubber via the light reflected off its inside surface [4]. The measuring range of 40 N covers the forces at which humans typically explore textures, with a non-linearity given at 2 %.

The motors are controlled at a rate of 50 Hz through an Arduino Uno microcontroller. While we worried that the control rate may cause vibrations that are visible in the force signal, this worry was unfounded. Most likely, they are absorbed by the under-actuated structure of the hand, the wires, tolerances and friction phenomena in the 3D-printed body. The underactuated structure of the hand implies that the input motor positions are not exactly mapped to output joint positions and forces, which reduces precision, but increases grasp stability and filters noise. This kind of robustness to noise is important for real-life applications on hardware with commercial tolerances.

Two different thumb structures are used as shown in Figs. 35 and 36: one with

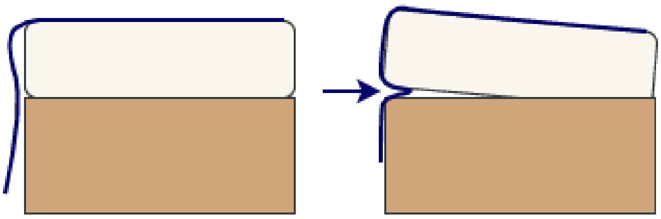


Figure 42. Representation of the tucking manipulation. The arrow represents a gripper pushing into the space between the two objects

a second sensor, and one with a single sensor and a flat nitrile rubber surface. The friction parameters of the flat surface are not exactly identical: The nitrile rubber was slightly harder, smoother and less adhesive than the surface of the sensor. The joint angles of the finger are not recorded, and are generally not known precisely due to the wire-driven structure of the hand.

In total, the gripper contains:

- 1 or 2 OptoForce 3D force sensors (OMD-20-SE-40N)
- 1 Arduino Uno microcontroller
- 2 HS-5585MH servo motors or 2 Dynamixel MX-28AT
- 3D printed base & finger links
- Bearings, pins, pulleys, cables

While the force sensors are capable of precision grasps, if an object is in a power grasp, it is in contact with more parts of the hand than the force sensors. In this case, the sensors do not report all the force acting on the object. Fully actuated grasps may also result in a grasp where the force sensors are not in contact with the object. We consider these limitations minor, as our main focus is the manipulation of textiles. As long as the force sensors are in contact with the object, slip can still be detected.

Aside from the addition of the bearings and the sensors, the assembly of the gripper can be completed by following the tutorial for the original design of the M2 gripper. If heavy objects are to be lifted, adding rubber to the grasping areas

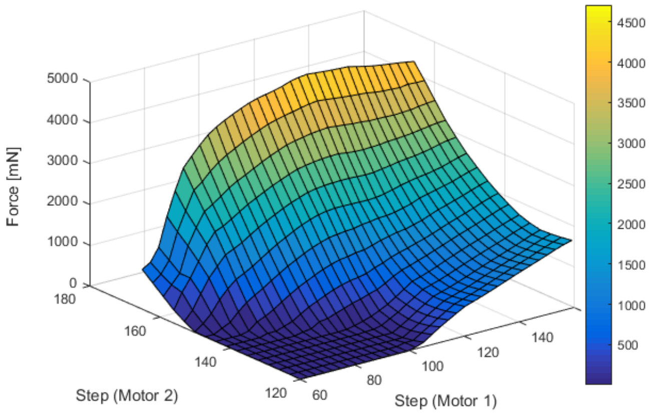


Figure 43. Grasp force at different motor positions. Motor 1: Antagonist tendon. Motor 2: Agonist tendon.

of the finger and thumb will help increase friction. Excluding the sensors and 3D printed material, the cost of the gripper is about $200 USD.

### 3.5.3 Maximum force in different positions

To evaluate the performance of the gripper, we sweep a grid of motor positions and record the force reported by the sensors, as shown in Fig. 43. This reveals the maximum grasping force in different configurations and grasping angles, and can also be used for the creation of a model-based controller. The maximum grasp force at the sensors is 4700 mN, as shown in the graph.

We note that the effective grasp force can be different for larger objects such as cylinders, and it may be only partially recorded by the sensors. Sensors were run at 30 Hz with a low-pass filter during this evaluation, as well as the following one.

We also used a simple controller to achieve a target grasp force for the hand and to evaluate the effective precision of the hand: the motor position of the antagonist is increased by one unit when the force is below the dead zone and vice versa, while the agonist tendon is still. As this utilizes the minimum discrete signal that a controller can use, it demonstrates the maximum precision of the setup. While a properly adjusted PID or model-based controller would be faster

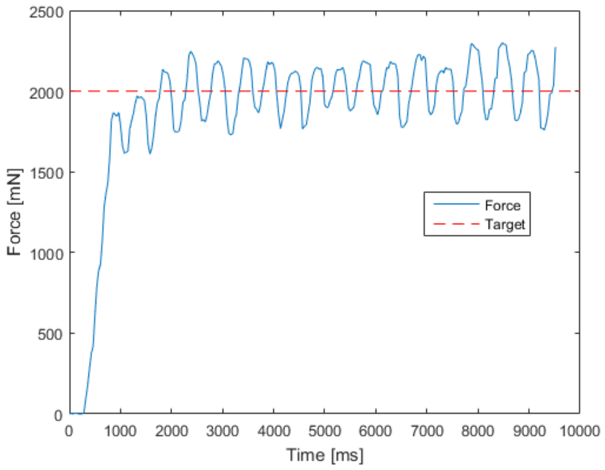


Figure 44. Total force seen by the force sensors when targeting 2 N

Table 11. Comparison of robot hands and their features.

| **Hand name** | **Tactile feedback** | **Axial load** | **Flat fingers** | **Relative sensor motion**[2] |
|---|---|---|---|---|
| Barrett Hand | ✓ | ~ | ✗ | ✗ |
| Robotiq 2-finger | ✗ | ✓ | ✓ | ✗ |
| Robotiq 3-finger | ✓ | ✓ | ✗ | ✓ |
| Schunk 5-finger | ✓ | ✗ | ✗ | ✓ |
| Schunk 3-finger | ✓ | ✗ | ✗ | ~ |
| Sake EZGripper | ✗ | ~ | ✓ | ✗ |
| **Our proposal** | ✓ | ✓ | ✓ | ✓ |

to reach the target force and may exhibit lower vibration, it will not be more precise than shown in this experiment unless the actuator is changed.

The controller was updated once every 150 ms. The data shows that at worst, the grasp force precision is about ±0.25 N. An example graph can be seen in Fig. 44.

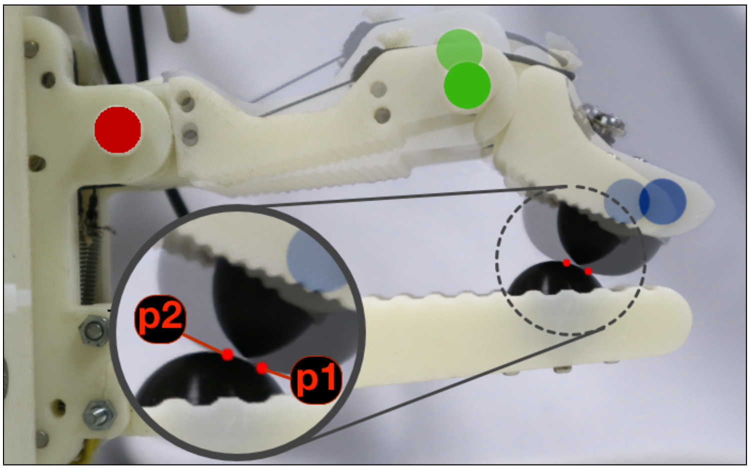


Figure 45. The finger performing the back-and-forth rubbing motion between the contact points $p_1$ and $p_2$, indicated in red.

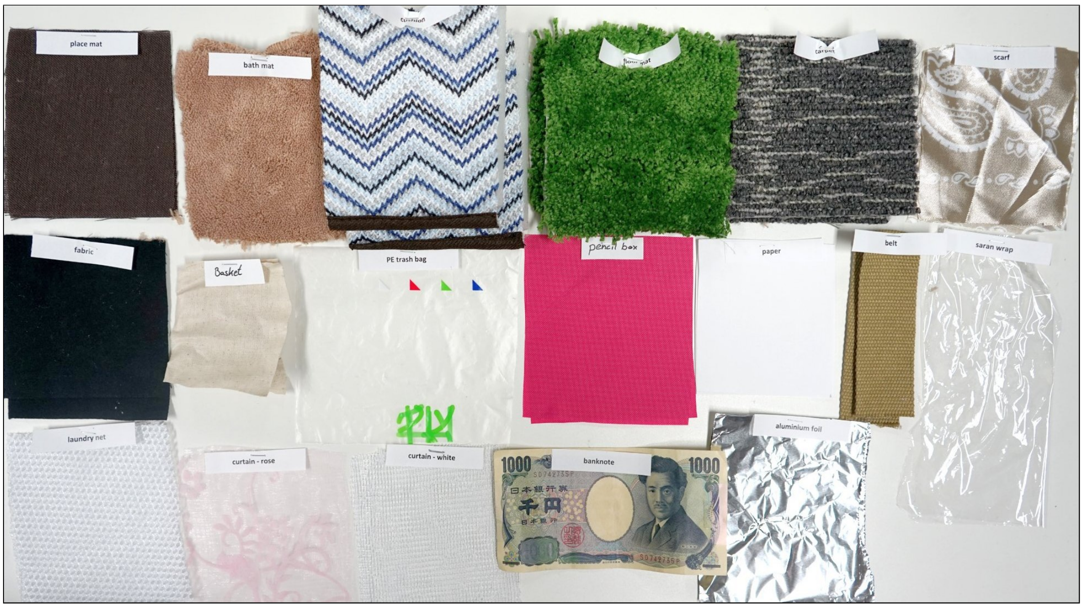


Figure 46. Single and multiple layers of the 18 materials were distinguished for a total of 36 cases. The case with two layers of cushion (top row, 3rd from the left) was excluded due to the high thickness.

## 3.6 Materials and Methods

### 3.6.1 Exploratory motion

Fig. 45 shows the basic, planar exploratory motion, which is a biomimetic rubbing motion as humans would perform with the tips of their index finger and thumb. The motion causes the sensors to rotate relatively to one another, which increases the relative motion in the contact interface. We surmise that this increases the amount of data issued by friction phenomena, allowing for easier detection.

The motion is defined by two sets of motor positions which place the fingertip on the thumb at a certain force. Fig. 45 shows the back-and-forth rubbing motion between the fingertips. To set up the exploratory motion, we record the motor positions near two points $p_1$ and $p_2$, at which the index finger's sensor touches the other sensor lightly. At $p_1$, the index finger is extended further than at the second point $p_2$, so that the point of contact is farther away from the robot and the distance between the two points of contact on the thumb's sensor are roughly 5 mm. To determine the final motor positions for a given exploratory motion, the motor positions at both $p_1$ and $p_2$ are increased slowly until the sensor reports a certain target force. These motor positions along with the values for $t_{move}$ and $t_{wait}$ fully define the motion.

One rubbing motion consists of setting the motor positions linearly from $p_1$ to $p_2$ over a time $t_{move}$, waiting for a time $t_{wait}$, and then doing the same for the movement back to $p_1$. The step trajectory that the motors follow is updated every 20 ms. Each rubbing motion starts and ends at the same point $p_1$, and thus consists of two *movement phases* and two *relaxation phases.* Fig. 47 shows the different phases during one motion.

**Algorithm 1** Rubbing motion

```
Input: in
Output: out
  Initialize parameters :
  $num\_mv\_cmds = round(t_{move}/dt)$
  $d_{step} = rubbing\_distance/num\_mv\_cmds$
  Back-and-forth rubbing loop
  for $i = 1$ to 5 do
    if $(i \geq 2)$ then
      Start recording
    end if
    Move finger back
    for $j = 1$ to $num\_mv\_cmds$ do
      Move motors back by $d_{step}$
      Wait for $dt$
    end for
    Wait for $t_{wait}$
    Move finger forward
    for $j = 1$ to $num\_mv\_cmds$ do
      Move motors back by $d_{step}$
      Wait for $dt$
    end for
    Wait for $t_{wait}$
    if $(i \geq 2)$ then
      Stop recording, save recorded data
    end if
  end for
```

During each *relaxation phase*, some of the stored mechanical energy in the sensors, materials and grippers is released. This can be seen in the decreasing force signal after the motors arrive at their destination (after about 100-150 ms). As the behavior during this phase differs between materials, we wait for $t_{wait} = 300ms$ before sending the next motor instruction. In our previous paper, we have used $t_{wait} = 1s$ under the assumption that the relaxation phase would yield more

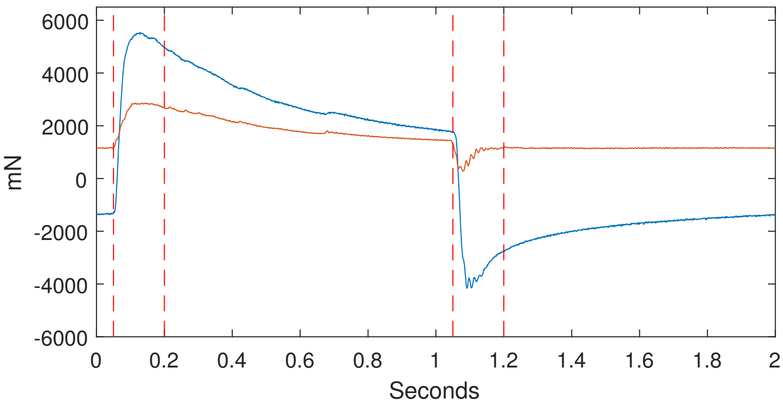


Figure 47. The two force signals of one of the sensors during a rubbing motion. The *movement phases* take place inside the red dotted lines.

information. However, in preliminary tests reducing $t_{wait}$ to 300-500 ms had only a negligible effect on the recognition rate. We thus chose a shorter wait time to speed up the data collection.

As the movement of the index finger is nearly planar, two of the three force values are used and the out-of-plane force value discarded.

### 3.6.2 Motion and sensor parameters

Humans vary the way they explore an object according to the information they wish to extract, and the results of [59] imply that moving the sensor at differing speeds and normal forces yields distinct insights into the material. In our experiment, the parameters that we vary to evaluate their effect on the recognition rate of the system are:

- Movement speed ($t_{move}$ of 50, 350 and 700 ms)
- Force (1 N and 2 N)
- Number of sensors (one or both)
- Sampling rate

We investigated the performance of the system at lower sampling rates to determine if camera-based force sensors can be used for material recognition.

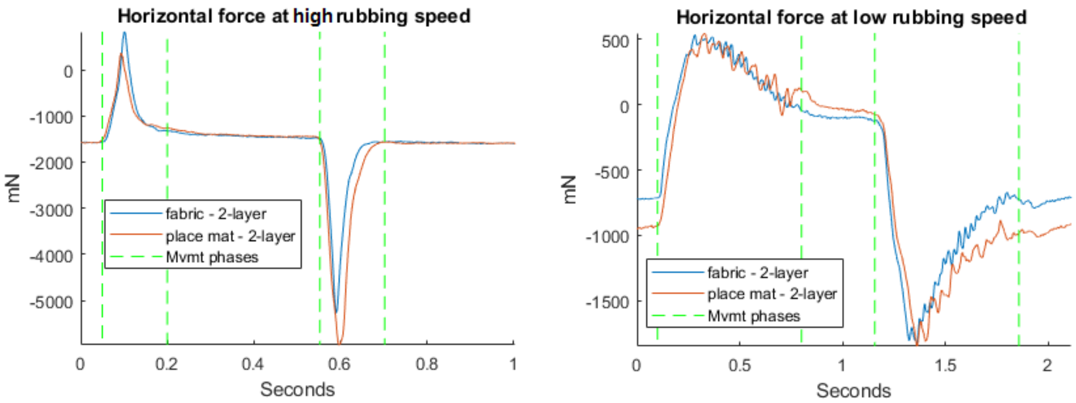


Figure 48. Left: Fast exploratory motion. Right: Slow exploratory motion. The horizontal forces of one sensor when rubbing 2 layers of two textiles of different roughness at two different speeds. Note the different vibration patterns during the slow motion.

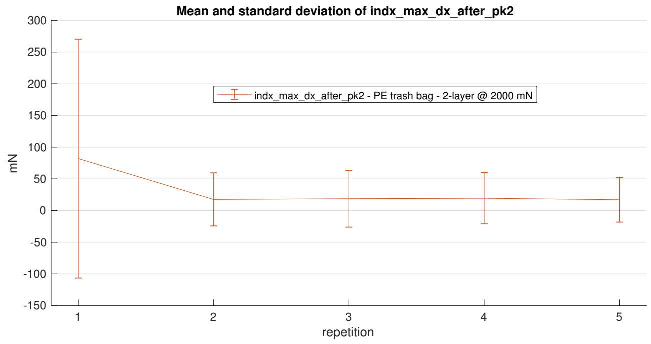


Figure 49. The mean and standard deviation of a feature over the course of successive rubbing motions. The noise in the first motion after closing the hand can be very high.

These low-cost sensors offer a high spatial resolution [64], which makes them attractive for manipulation tasks, and may be even be transparent, to allow more insight into the object being handled, such as in Yamaguchi and Atkeson [54]. However, they can only provide a low measuring frequency, which may be an obstacle for tactile material recognition.

We also investigated the number of sensors, because we expected that a single sensor would perform similarly to the pair of sensors, considering that both sensors should be subject to practically the same forces, at least for thin textiles.

For the different movement speeds and forces, we evaluate the confusion matrices for different exploratory motions to determine if different materials are more reliably distinguished by each exploratory motion.

Some of the materials had a structured surface that changes according to the orientation. To avoid adding complexity to these cases, all materials were arranged in the same direction during all measurements.

### 3.6.3 Classification algorithms

We evaluated the performance of different machine learning techniques and choices of feature representations.

In [65], we proposed a set of features and found that it yields the best results when using an OVO-MSVM[16] with a radial basis function. We used this SVM to compare the performance of each of the following features:

- The feature vector proposed in our previous work.
- A set of discretized spectral-amplitude-based features similar to Sukhoy *et al.* [57]
- Up to 15 frequency components as proposed by Jamali *et al.* [63]
- Roughness and Fineness as proposed by Fishel and Loeb [59]
- Activity, Mobility and Complexity as defined by Kaboli *et al.* [60] [17]

Finally, we compared their performance against a number of neural network models trained on the raw data. The performance of the SVM with each set of features was tested via a 10-fold cross-validation, and the neural network with a 5-fold cross-validation.

As the movement of the index finger is nearly planar, two of the three force values were used and the out-of-plane force value was discarded.

[16] One vs One Multicategory Support Vector Machines

[17] The feature vector in the original paper includes the correlation between 20 electrode signals and the pressure sensor in each BioTac. Because this cannot be calculated for our sensors, the feature vector contains 12 elements instead of 16. We consider each axis separately.

The calculation of the features proposed by Jamali and Sammut uses the peak-find algorithm graciously supplied by O’ Haver [66].

**Geometric features** The set of features is extracted from the filtered force sensor data according to human intuition, and called ”geometric” because it describes characteristic points defining the shape of the curve. It was originally developed for the fastest of the three exploratory motions, and does not attempt to extract information from vibrations.

From each of the four force signals that are recorded during one back-and-forth motion., the following values were used as features:

- The peak value during or just after the movement phase
- The amount of time between start of the movement phase and the occurrence of the peak value
- The absolute maximum value of the gradient in the relaxation phase after the peak value
- The static values at the end of the relaxation phases
- The values at the start and end of the motion

The bulk of the features, such as the peaks and the stationary force levels, are related directly to the friction coefficients. When a single layer of material is being measured, the friction coefficients are those of the interface between the sensor and the material. When multiple layers are present, it is predominantly the friction coefficient between the material itself that is being sensed, as the sensors’ high-friction rubber surface experiences little slippage.

**Binned frequencies** To extract the frequency-based features only from the *movement phases* as described in section 3.6.1, we define two intervals of $t_{move} +$ 50 ms length that begin when the motion command is sent to the motors, as most frequency variation is expected to happen during and shortly after the movement phases. We extract the frequencies present in each of these windows via fast Fourier transform from the raw sensor data. The 50 ms delay compensate for the lag of the the control chain.

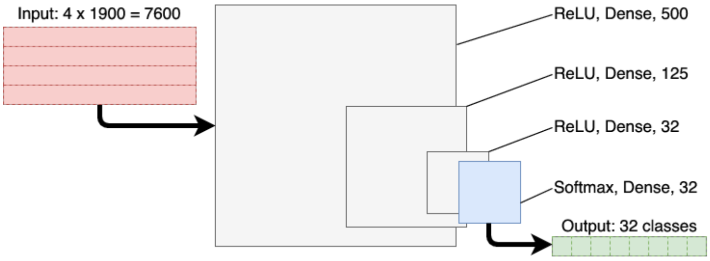


Figure 50. Structure of the neural network classifier Nr. 3 (medium size). All layers are fully connected. The last layer applies a softmax operation.

The single-sided amplitude spectrum of the signal in each interval is split into 10 equally sized bins and the values in each bin averaged. This results in a histogram of dominant frequencies. The feature vector consists of the histograms of all the intervals.

**Neural Networks** Neural networks have the strong appeal that they can learn feature representations, although they require large amounts of data to be effective. For our specific case, we examine the performance of five different configurations of neural networks. As training data in tactile applications is relatively sparse and expensive to obtain, finding an efficient configuration is of great interest.

We evaluated five different network structures. All of the layers were fully connected with an ReLU activation function, and each network ended with a softmax layer with 36 nodes. The numbers describe the number of ReLU neurons in each layer. Fig. 50 illustrates the third network in this list:

1. 125-75-36
2. 250-125-36
3. 500-125-36
4. 1900-500-125-36
5. 3800-1900-500-125-36

The input vector length depended on the speed of the exploratory motion and varied between 4000 and 8000 elements.

### 3.6.4 Material selection and data collection

To evaluate our method, we chose a set of 18 materials taken from different household objects, such as aluminium foil, saran wrap, cotton or carpet. Fig. 46 shows the full set listed in Table 12. More detailed information can be found in the appendix.

Table 12. List of materials. Ordered as in Fig. 46 left-to-right, top-to-bottom. Materials marked with a * had different material properties on each side.

| Article | Material |
|---|---|
| Place mat | 60% Cotton, 40% Polyester |
| Bath mat* | 90% Polyester, 10% Nylon |
| Cushion* | Polyester, Polyurethane |
| Floor mat* | Polyester, Latex |
| Carpet* | Bitumen (bottom), Polyamide (top) |
| Scarf | Polyester |
| Fabric | Cotton |
| Basket* | Polyester |
| PE trash bag | Polyester |
| Pencil case | Polyester |
| Paper | Paper |
| Belt | Polyester |
| Saran wrap | Polyvinylidene chloride |
| Laundry net | Polyester |
| Curtain - rose | Nylon |
| Curtain - white | Polyester |
| Banknote | Paper, treated |
| Aluminium foil | Aluminium |

A preliminary data set of 9000 fast motions was recorded with two sensors on a single exploratory motion with $t_{wait} = 1s$ in [65]. Of these, 2160 motions were preparatory as described in section 3.6.1, resulting in about 7200 usable samples.

Next, we collected 5400 samples each for the three different motion speeds and two different force levels, for a total of 32400 samples. These were used to

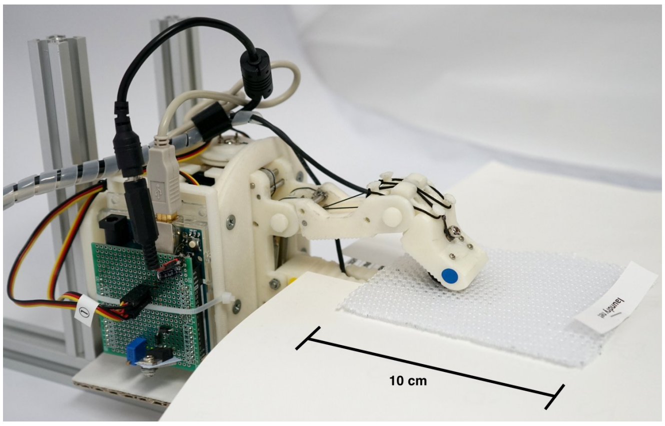


Figure 51. The experiment setup, with a NAIST OpenHand M2S rubbing its index finger on a piece of textile to obtain tactile information. The white guard supports the textile so it does not slip away while the fingers are opened.

evaluate the performance of the different features on different data.

Before every set of measurements and after the start-up of the system, the motor positions were recalibrated such that the force acting on the sensors would be constant during the rubbing motion. This was to avoid loosening wires or wear affecting the measurements.

The order of materials was randomized so that neither deterioration of the sensor nor possible debris from a fabric or other factors such as softeners would contaminate the measurements.

As shown in Fig. 51, the gripper is mounted horizontally with the rigid finger's sensor pointing upwards, with a small guard around the thumb so that the material would not fall off.

During the slower rubbing motions, the movement of the textile and collisions with the guard could cause vibrations and contaminate the signal. Furthermore, for samples of different sizes and high stiffness, such as the carpet, the way the object is placed on the hand may have an effect on the extracted signal. However, as this kind of noise would be unavoidable in a real application, we consider it a realistic difficulty.

For each recording of a material, we execute 5 consecutive exploratory motions after closing the hand, and another 5 motions after opening and closing the

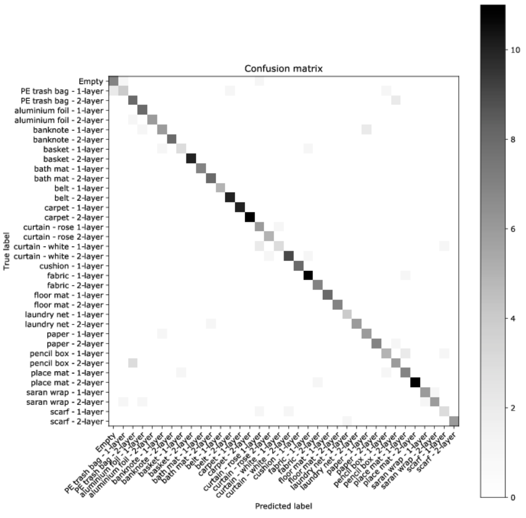


Figure 52. Confusion matrix for high speed motion at normal force 1 N, $t_{move} = 50\ \ \text{ms}$.

hand once more. We limited the number of successive rubbing motions to avoid damaging the material. During preliminary experiments, we observed that the first motion after closing the fingers yields less reliable values than the later movements, as shown in Fig. 49. This is caused by factors such as the initial configuration of the textile, the undefined angle of approach of the sensor when closing the hand and the state of internal forces in the sensors at the start. As these are difficult to predict, we only use the data from later motions for the data analysis.

### 3.6.5 Comparison with human performance

To compare the performance of the system and to evaluate the difficulty of our set of materials, we performed experiments with human subjects. We selected 9 textiles for which the confusion matrix showed difficulty, and which are listed in Fig. 56. Subjects had to distinguish between all the pair permutations between these textiles, as well as between 1 and 2 layers of the same object for all 18 materials. 8 human subjects consented to participate in the study in total.

The experiment procedure followed the one described in Fishel and Loeb [59]

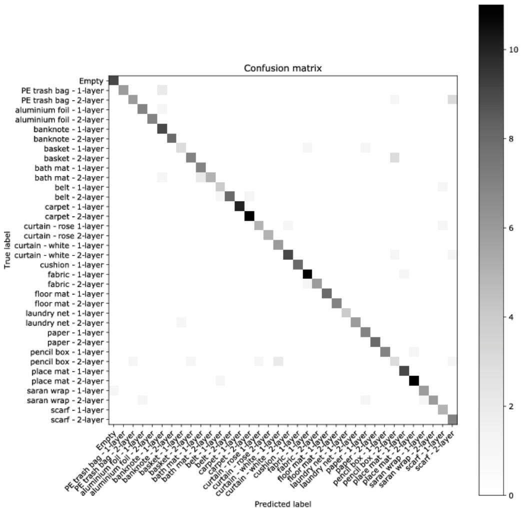


Figure 53. Confusion matrix for medium speed motion at normal force 1 N, $t_{move} = 350$ ms.

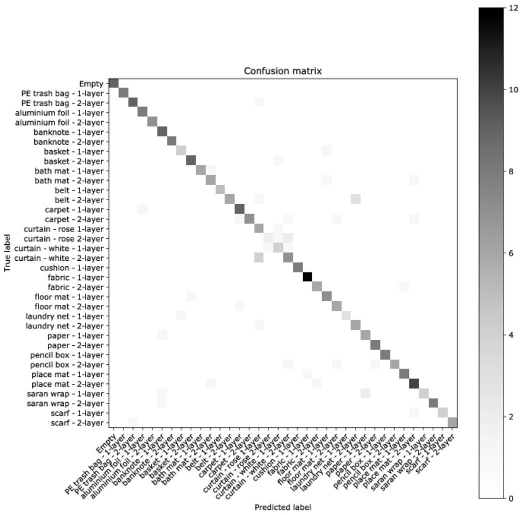


Figure 54. Confusion matrix for low speed motion at normal force 1 N, $t_{move} = 700$ ms.

closely. First, subjects were presented with two textiles and asked to rub them between their thumb and index finger until they felt comfortable distinguishing between them. Next, subjects were asked four times to feel one of the two textiles, randomly chosen, and state which one they believed it to be. There was no time limit. Subjects could not see the textile after the initial exploration and were given no feedback about the correctness of their answers until the end of their test. Subjects were aware that the selection was determined from a random number generator on the fly, and that they may encounter the same material multiple times in a row, or even by chance all four times.

One of the weaknesses of this setup is that humans have a number of advantages over the robot hand and force sensor:

- The robot can only move along 1 axis, but humans instinctively perform a more complex, multi-axis movement with their fingers
- Humans can hear the crumpling of certain materials (aluminium, paper)
- Proprioception. Humans sense the object's thickness from the position of their fingers, while the sensors measure it only indirectly via an increased normal force.

Some measures were taken to reduce the impact of these advantages. For one, the experimenter was imitating the motion that the subject was performing using the other textile, so that the sounds of both of the materials would be heard. Furthermore, subjects were reminded to use only their fingertips and not to touch the object with other parts of their hands or fingers.

## 3.7 Results

We first present the results of different classifiers on data obtained at different motion speeds and normal forces. Afterwards, we compare the effect of a lowered sampling rate, and the performance of humans in the recognition task.

### 3.7.1 Motion speed and force

Fig. 48 shows the horizontal force signal for two different materials at two different motion speeds. During the slow motion, the rougher material shows both lower

Table 13. Recognition rates using different classifiers, 1 N normal force

| $t_{move}$ | 0 ms | 350 ms | 700 ms |
|---|---|---|---|
| Geometric features (GF) | **86.4 %** | **90.9 %** | **87.5 %** |
| Binned frequencies (BF) | **76.5 %** | **79.8 %** | **88.6 %** |
| BF & GF | **86.9 %** | **88.6 %** | **86.8 %** |
| Jamali2011 | **13.8 %** | **16.9 %** | **23.9 %** |
| Fishel2012 | **48.2 %** | **54.4 %** | **22.4 %** |
| Kaboli2015 | **25 %** | **25.2 %** | **18 %** |
| Neural Net (Nr. 3) | **93.3 %** | **96.3 %** | **97.5 %** |

Table 14. Recognition rates using different classifiers, 2 N normal force

| $t_{move}$ | 0 ms | 350 ms | 700 ms |
|---|---|---|---|
| Geometric features (GF) | **90.9 %** | **93.8 %** | **91.3 %** |
| Binned frequencies (BF) | **79.5 %** | **83.2 %** | **86.3 %** |
| BF & GF | **91.3 %** | **92.9 %** | **92.1 %** |
| Jamali2011 | **17 %** | **24.4 %** | **39 %** |
| Fishel2012 | **51.4 %** | **49.4 %** | **31.8 %** |
| Kaboli2015 | **20.3 %** | **18.1 %** | **10.1 %** |
| Neural Net (Nr. 3) | **91.6 %** | **98.3 %** | **97.1 %** |

frequencies and higher amplitudes during the movement phase. When moving quickly, although almost no vibration is visible, the change from static to sliding friction is sudden and clear. When the sensor moves slowly, one can see the tangential force rise until slip starts to occur, after which vibrations appear at different frequencies. These vibration patterns are affected by stick-slip friction, surface roughness and other tribological phenomena.

Figs. 52, 53 and 54 show the confusion matrices for different motion speeds. It is notable that the matrices barely overlap, indicating that materials that are difficult to distinguish through one motion can more easily be determined by using another motion. The recognition rate of each individual motion is over 91 % for all of the neural network classifiers.

### 3.7.2 Performance of different features

Tables 13 and 14 show that the neural network outperforms all the other classifiers we have tested.

The geometric features we proposed in von Drigalski *et al.* [44] perform best at $t_{move} = 350ms$ (medium speed) and 2 N normal force, instead of the fastest motion and 1 N normal force, which is the movement they were based on.

The binned frequency features perform better when the rubbing motion is slower, likely because the sensor obtains more characteristic vibrations.

Jamali and Sammut's feature descriptor performs the worst on the fastest exploratory motion. We note that the single-sided amplitude spectrum in our data is relatively smooth, so that the peak-find algorithm had difficulty finding significant peaks: on 6000 samples of the fast motion in the 2016 data, 60 peaks should have been found, but the mean was 38.7 with an std of 5.

Neither the features proposed by Fishel and Loeb nor those by Kaboli et al. result in a high recognition accuracy. We discuss the reasons for this in section 3.8.

### 3.7.3 Effect of sampling rate & neural network depth

In previous work, we found that when using filtered data recorded at 1000 Hz, the third network configuration (500-125-36 ReLU activation layers and one softmax) reached 96 % recognition success on the first data set. In this investigation, we used the unfiltered data (7200 samples) in order to preserve vibration information, which lowers the recognition rate of the neural network by about 3 %. This is likely due to the increase in both signal and noise, which requires more training data for the network to adapt to.

Fig. 55 shows that with decreasing sampling rate, a decrease in both recognition and convergence speed starts to occur from 100-50 Hz, which indicates either a lack of useful information in the high frequencies of the data, or that the amount of training data was insufficient for the network to decipher the data at high frequencies.

All the neural network configurations performed similarly at all sampling rates, with no clear advantages in terms of generalization or convergence speed.

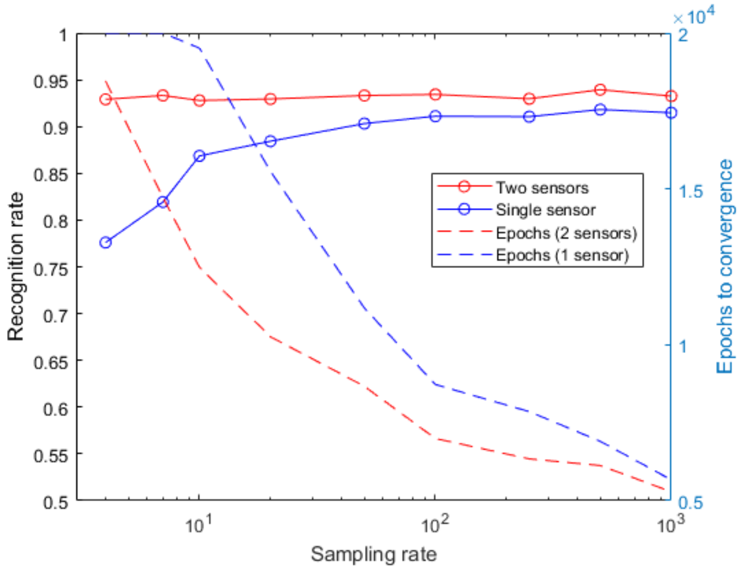


Figure 55. The performance of neural networks is affected by the signal sampling rate. Both recognition rate and convergence speed decrease with lower sampling rates. $t_{move} = 0$ ms, $t_{wait} = 300$ ms, $F = 1000$ mN.

### 3.7.4 Dual vs. single sensor

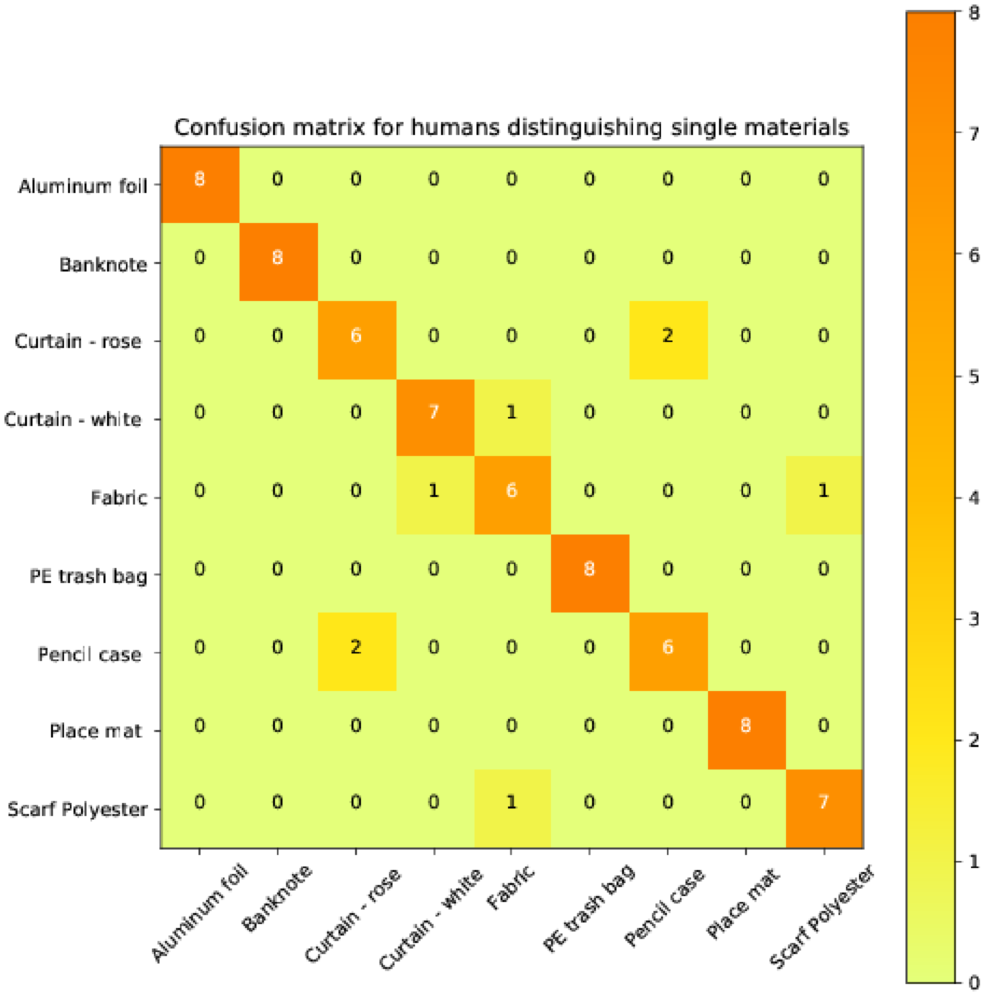
Using the neural network classifier, eliminating one of the two sensors costs about 3 % in recognition rate and slows down the convergence rate. This could be due either to the system requiring more data, or to the second sensor's signal containing information that is not captured by the first. With SVM classifiers, the recognition rate dropped by about 10-15 %. We expected the setup with dual sensors to be superior for materials who have a different texture on each side, but there was no significant difference.

### 3.7.5 Comparison with human performance

Fig. 56 shows the confusion matrix for single-layer materials. The overall success rate was 98.6 % on the restricted set of materials, whereas the best classifier[18] had an accuracy of 97.5 % on the same materials.

When distinguishing between one and two layers, the recognition rate was

[18] The neural network at medium speed and 2 N normal force.



Confusion matrix for humans distinguishing single materials

|  | Aluminum foil | Banknote | Curtain - rose | Curtain - white | Fabric | PE trash bag | Pencil case | Place mat | Scarf Polyester |
|---|---|---|---|---|---|---|---|---|---|
| Aluminum foil | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Banknote | 0 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Curtain - rose | 0 | 0 | 6 | 0 | 0 | 0 | 2 | 0 | 0 |
| Curtain - white | 0 | 0 | 0 | 7 | 1 | 0 | 0 | 0 | 0 |
| Fabric | 0 | 0 | 0 | 1 | 6 | 0 | 0 | 0 | 1 |
| PE trash bag | 0 | 0 | 0 | 0 | 0 | 8 | 0 | 0 | 0 |
| Pencil case | 0 | 0 | 2 | 0 | 0 | 0 | 6 | 0 | 0 |
| Place mat | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 0 |
| Scarf Polyester | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 7 |

Figure 56. The confusion matrix for single-layer materials in the human experiment. The lowest confidences were found for cotton(fabric) and the pencil case, which were confused for other polyester-based textiles 25 % of the time.

100 % for almost all materials, except for the floor mat (very springy) and saran wrap (very sticky), where the success rate was 75 %. Subjects reported the highest difficulty while distinguishing two layers of saran wrap, as the material tends to stick to itself. It was evident that when subjects were distinguishing between one and two layers of the same material, they were trying to induce sliding between the two layers. Once the sliding has been perceived, subjects were quick to answer.

After the experiments, some subjects expressed an interest in attempting to distinguish between 2 and 3 layers of banknotes. We recorded their attempts in 10 trials, where they achieved a recognition rate of 50 % – the same as random chance.

## 3.8 Discussion

First off, it is striking that the performance of the features proposed by Jamali *et al.* [63] and Fishel and Loeb [59] is lower than in the original papers. We believe that this has multiple reasons:

- features and classifiers are tuned to specific systems and methods
- the performance is closely linked to the sensors and the method that is used
- both the exploration methods and the classification approaches differ significantly between the related work and our proposed method

For instance, the sensor proposed by [63] is designed to capture data that is rich in vibration signals of a wide frequency range. Consequently, the features they proposed extract the peaks of the feature spectrum, and are particularly suited for this kind of data. Similarly, Fishel and Loeb's had an experimental setup that measured the resistance force accurately via a separate sensor, as well as a textured fingertip surface which augmented vibrations, and also used a different exploration method: they used a series of short, uniform exploratory motions selected by their method of Bayesian Exploration.

By contrast, our exploratory motion varies in force and relative pose of the sensors, is affected by the resistance of the materials because it is implemented in an underactuated gripper, and does not measure the resistance force separately.

The motion is also shorter than those used in related work, and the features we proposed in von Drigalski *et al.* [65] are intended for this type of input. Since both our movement and our sensors differ, it is not surprising that the methods proposed in other works performed worse on our data.

In our experiments, we found that the neural network had the highest recognition rates, but it is uncertain how this classifier would perform on a different system. There is also an argument to be made in favor of retaining some physical relevance in the feature description of a material.

In summary, it is logical that the recognition methods proposed for different sensor data play to different strengths and particularities of their respective systems. Our results highlight the difficulty of choosing an appropriate recognition approach for tactile recognition problems, and how strongly it is connected to the parameters of the problem and the equipment that is used.

## 3.9 Effect of different movement speeds

One question we aimed to answer is the effect of lower movement speeds on the data we obtain. As Figs. 52, 53 and 54 show, the confusion matrices for the different motions are indeed different, and indicate that the information obtained by each motion differs as well. This shows that some exploratory motions reduce uncertainty about the system's belief more than a repeated or random one would, which is in line with the findings of [59] and [57].

[59] reported that some of their feature dimensions (*Traction*, *Roughness* and *Fineness*) perform better on some materials and worse on others, with significant differences. By contrast, all of our exploratory motions yield reasonable recognition rates, with only small differences between them. We believe that this is because our motions are less controlled, and record a wider range of data at once.

In the investigation of the sampling rate, we found that the recognition rate of the fast exploratory motion stayed stable, even when reducing the measuring frequency down to 50 Hz. This is in line with the results of our previous work, where we found that even when using features that did not capture vibrations, recognition performance was acceptable, even though we had expected that high-frequency vibrations would be more important. The finding shows that our method may be used to perform textile recognition using sensors whose measuring frequency is low, but whose spatial resolution is high, such as camera-based sensors. We note that this means that robotic skin in particular does not require a high measuring frequency to be useful for this and other applications.

## 3.10 Number of sensors

The results of the single sensor data show that reliable identification can be performed even with a single sensor, especially if multiple exploratory movements are permitted. Performance did increase when using a second sensor, but only slightly, so the cost should be kept in mind. We attribute the increased performance to the second sensor reducing the effect of noise.

## 3.11 Physical interpretation and limitations

While we did not confirm this in a separate experiment, it is our strong impression that a high friction coefficient on the fingertip sensor[19] is an important requirement for our exploratory motion to yield useful data. When two layers of the same material are to be distinguished, both humans and our method detect this by inducing a relative movement of the two layers. If the friction coefficient of the sensor surface were too low, then the layers would not separate, and the sensory experience would be similar to a single layer. One can imagine that, if the sensor were extremely slippery, it would simply slide over the surface, capturing only a small tangential force and the vibrations caused by moving over the textile's surface. This view is supported by our results in the case of two layers of saran wrap, where test subjects noted particular difficulty because the layers did not separate easily. Even in the single-layer cases it appears that high friction is helpful, by increasing the signal-to-noise ratio when measuring the friction parameters of the material, and by increasing the range of tangential forces the sensor can perceive.

One limitation of our method is that it cannot distinguish between 2 and 3 layers of the same textile. However, as mentioned in section 3.7.5, we have found not only that humans have the same limitation, but also a reason for it. The contact interface material-material, as shown in Fig. 57, behaves differently than the interface material-sensor, or material-fingertip, and it is this change

[19] and the surface on the opposing finger, if only one sensor is used

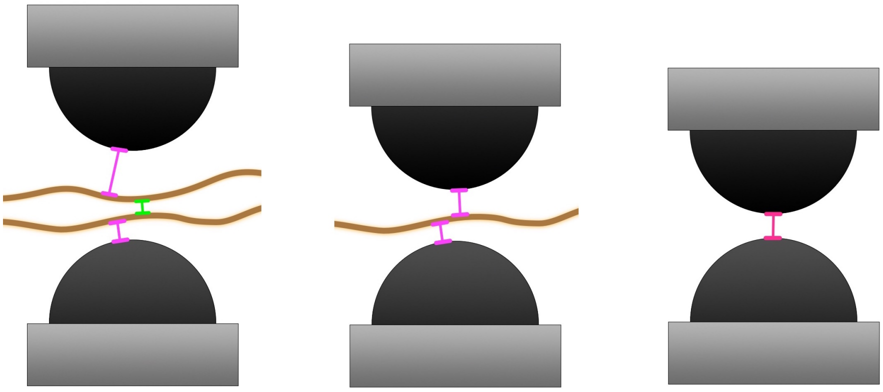


Figure 57. From left to right: Two layers of textile, a single layer of textile, and no object between two sensors. Contact interfaces between different materials have different friction coefficients and behaviour. If the interface sensor-material (magenta) has a higher friction coefficient than material-material (green), the three cases can be distinguished reliably.

in behavior that is revealed by the rubbing motion and detected by the human fingertip and our sensors. When attempting to distinguish 2 and 3 layers of a textile, the contact interface material-material is present in either case, so the sensory experience is identical except for thickness.

A more general limitation of our method – and indeed all the other recognition methods we have found – is the need to train on the different materials. It would be desirable if a database of surfaces and objects could be used for recognition, so that the sensor could identify a material it has not seen before. In future work, we plan to evaluate if it is possible to obtain a sensor-agnostic description of surfaces, and if it can be based on physical parameters, such as roughness, surface structure or stiffness.

## 3.12 Conclusion

In this chapter, we presented a method for the identification of textile-like materials that can be used by any robotic gripper that can induce a tangential relative movement of its fingers. By rubbing the gripper's fingertips together, it can not only detect the material, but also distinguish between one and multiple layers of the material between the tactile sensors. We tested the method's performance on 18 different materials at different speeds and normal forces, and compared it to humans' performance on the texture discrimination task. Furthermore, we showed that the method performs well even at low measuring frequencies, indicating that it can be used with other tactile sensors as well. Additionally, we investigated the descriptors of tactile data that have been proposed by earlier work by comparing their performance on our data set, compared to a neural network classifier. The recognition rate of our method is over 92 % after a single motion, and can be increased by performing additional exploratory motions. Lastly, we proposed the designs for the open-source gripper that can perform the exploratory motion and which we have used in our experiments. It can be used freely to reproduce our work.

# 4. Clothes folding

## 4.1 Introduction

To test the sensing method and gripper proposed in the previous section, we implement it in a system that folds clothing articles. We propose to improve both the speed and reliability of the process, as well as the quality of the result, by:

- integrating the tactile sensing into the picking and unfolding process, and
- using a tool made for humans during the folding process.

In spite of the proliferation of washing machines and dryers and the demand for further automation, laundry-related tasks still take up over 10 percent of total time spent on household tasks [67]. They also represent one of the areas in which the performance of modern robots is lacking.

We describe a series of manipulations and algorithms to bring a shirt from an arbitrary initial state on a table in front of the robot to a folded configuration. First, we describe the related work. Afterwards, we discuss the effect of our proposed rubbing motion and the folding tool for humans on the performance of the system. We close with directions for future work and limitations of our solution.

## 4.2 Related Work

In the body of research regarding clothes perception, a large number of works deal with a) person recognition using their gait and clothing and b) simulation, e.g. for computer graphics or online stores. These research directions can serve as sources for cloth modeling and representations, but in this section, we present works that propose either methods to manipulate textiles and garments, or perception methods that enable such manipulation.

Two noteworthy efforts in this field are from the group of P. Abbeel at UC Berkeley, and the EU-funded CloPeMa project that ran from 2012 to 2015, who each presented a complete pipeline to fold textile objects. In Maitin-Shepard *et al.* [5], the former folded a heap of towels using a PR2. Doumanoglou *et al.*

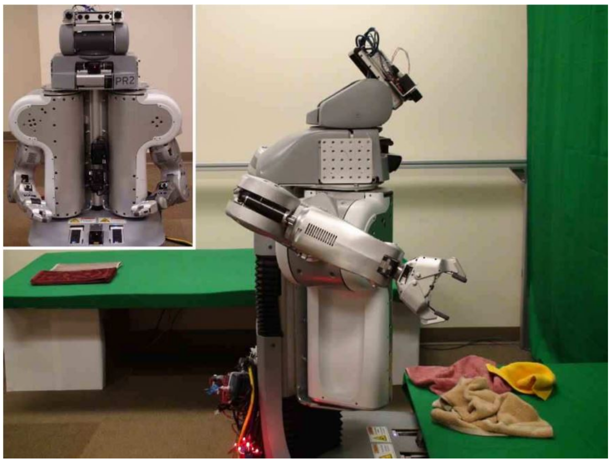


Figure 58. The PR2 that was used to fold towels in Maitin-Shepard *et al.* [5]. The table in the back is used as a folding surface.

[6] presents the results of the latter, which uses two robot arms with parallel grippers, and can be considered the current state of the art.

Additionally, some commercial products such as the Foldimate and the Laundroid have been announced, which claim to perform part of or the entire folding process. However, they have not been released yet.

We present the remaining works grouped by the subproblem that they focus on: grasping, unfolding, folding, flattening and classification/pose estimation.

### 4.2.1 Grasping

This subtask concerns finding a grasp point to pick up the clothing item, usually from a flat surface in a randomly arranged state, and often piled up with other items. Finding grasping points during manipulation is covered in separate sections.

Hamajima and Kakikura [68, 69] described the complete laundry folding process as bringing a pile of unfolded garments to a stack of folded items. They focus on separating items of similar color to obtain the grasping point of a single item in the pile.

Willimon *et al.* [70] segment the RGB image of the pile into similar regions, and attempt grasping in the highest segmented region above the table. Hata

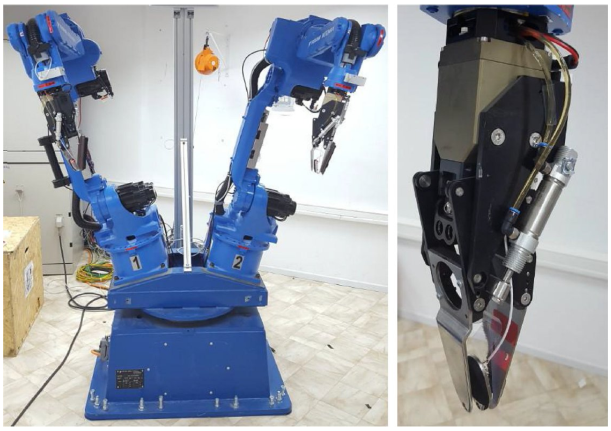


Figure 59. The two Motoman MA1400 with custom jaw grippers used in the CloPeMa project [6]. The vertical aluminium frame in the back holds two Nikon D5100 cameras for stereo vision.

*et al.* [71] and Bersch *et al.* [72] choose the highest point of the clothes pile as the grasping location.

In Cusumano-Towner *et al.* [73], the boundary of a crumpled garment is detected and then grasped from the side. Ramisa *et al.* [74, 75, 76] proposed grasping highly wrinkled parts of the clothing article to increase the success rate, and in [75] described a feature descriptor to detect such wrinkled parts. They also provide a database of wrinkled clothes and semantically labeled areas of those clothes in [76]. Yamazaki and Inaba [77] proposed a wrinkle detector to detect clothes in unstructured environments.

Doumanoglou *et al.* [6] proposed to first detect fold lines in a rectified depth image using a filter, and detected grasp point candidates on these fold lines. A suitable grasp point is then selected based on the fold geometry and an RGB image segmented by texture, preferring points that are easy to grasp and inside a region of one homogenous texture, so as to reduce accidental grasps of two clothes at once.

Alenyà *et al.* [78] present a benchmark for grasping textiles and explain the general difficulties when comparing textile manipulation systems.

In our method, we also use the RGB image segmentation to inform our grasp position, but otherwise prefer the highest point of the pile.

### 4.2.2 Classification and Pose Estimation

Kita and Kita [79], Kita *et al.* [80, 81, 82, 83] treated the recognition and pose estimation problem in a series of works. In [79], they fit the silhouette of a hanging piece of clothing to that of its mass spring model and thus infer some 3-D information about the garment. They improve the method in [80, 81, 82] by adding multiple view angles, 3D point clouds and matching the cloud to a Maya simulation of the clothing item. In [83], they demonstrate their method unfolding a pullover on an HRP2 humanoid robot.

Li *et al.* [84, 85, 86] simulated grasping and hanging clothing items to classify and finally unfold a clothing item held by a robot. [84, 85] focused on the classification of the item, which is observed by a depth sensor from multiple viewpoints, and then classified into a category and pose using an SVM. In [85], the similarity is calculated according to the merged volumetric model from multiple views, and the simulated virtual model of the garment. [86] extended this by matching the model to the observed item using non-rigid deformation and then unfolding the clothing item through a series of regrasps with a dual-armed robot.

Petit *et al.* [87] tracked textureless 3D objects using an RGB-D sensor by first registering their initial shape and then simulating the object's internal forces using ICP and a linear FEM model.[20] Similarly, Willimon *et al.* [88, 89] tracked 3D surfaces using an RGB-D sensor, a deformable mesh and an energy minimization approach without prior knowledge of the object's shape. They report difficulty with the borders of the object as well as sharp angles and folds. In Willimon *et al.* [70], they propose to classify a hanging garment via global features and edge information.

Stria *et al.* [90] proposed another method to fit polygonal clothing models to RGB input images, and report increased performance over Miller *et al.* [91].

We use our own algorithm based on an edge representation of the item's contour to estimate the pose and find the shoulder grasp points.

[20] While the goal of the authors' research is to create a pizza-making robot, their method applies to cloth and paper as well.

### 4.2.3 Unfolding

Hamajima and Kakikura [68, 69] proposed to regrasp the lifted clothing item until its hemlines are grasped, and thus unfold it. They detect the hemlines based on the shadows and shape of the item.

Osawa *et al.* [92] noted that grasping the lowest point of a freely hanging piece of clothing two times necessarily brings it into a finite number of states, and used template matching to categorize clothing articles.

Maitin-Shepard *et al.* [5] limited the possible cases by grasping two adjacent corners of a towel, and then rotating one of the grippers if the towel is not flat when pulled taut. The folding used the edge of the table and the mobile base of the robot. Cusumano-Towner *et al.* [73] used lowest point regrasps and the contour of the grasped item hanging between two robot arms to find appropriate grasp points.

Bersch *et al.* [72] use a t-shirt covered in fiducial markers which allows them to reconstruct its 3D shape. The grasp points are then found by searching for the point that is both reachable and closest to the shirt's shoulder.

Stria *et al.* [93] focused on the unfolding of a garment with few folds spread on a table. They proposed to detect the top and bottom layer of the garment using depth information, and then generate grasp points to unfold along the edge of the overlapping areas. This allows them to unfold garments without prior knowledge of their shape.

The approach we use resembles most closely that of [92], with some modifications to allow the use of a robot with a smaller workspace.

### 4.2.4 Flattening

Willimon *et al.* [94] propose to unfold a garment by first pulling the garment in 8 equidistant directions, and then using depth information to remove any remaining folds. They test their method on a washcloth.

Sun *et al.* [95] present a method to flatten a wrinkled but otherwise unfolded garment with two robot arms, by locating ridges and peaks of the wrinkles, and determining appropriate grasp points to pull the garment flat.

Doumanoglou *et al.* [6] compare the contour of the garment on the table to its target shape, and sweep over the concerned area towards the outside with a brush attached to the gripper, while the other gripper holds down the garment to prevent it from sliding.

### 4.2.5 Folding

Van Den Berg *et al.* [96] develop a method to plan folds of a garment along a folding line such that the remaining parts of the garment are not affected, using only the shape of the item and geometric considerations. The visual recognition and fitting of the shape of the items is described by Miller *et al.* [91, 97].

Stria *et al.* [98] proposed the method used in Doumanoglou *et al.* [6] to fold a flat clothing article on a table. It fits a contour to the observed garment and assumes a thin gripper that can fit one of its fingers underneath it from the side. Hou *et al.* [99] propose a similar recognition method, but discretize the garment into a mesh of particles.

Balaguer and Carpin [100] propose a motion planning algorithm for robot arms folding towels using an arc trajectory, focusing on collision avoidance and execution speed. Li *et al.* [101] calculate an optimized folding trajectory based on the contour of the object and target fold line, using the friction and material properties of the garment and table.

Although not focused on garments, an origami folding method and robot has been proposed by Balkcom and Mason [102]. Elbrechter *et al.* [103, 104] tracked the deformation of a piece of paper via fiducial markers and finite elements of varying stiffness, and folded it with two shadow hands.

Koustoumpardis *et al.* [105] and Kruse *et al.* [106] propose approaches for collaborative cloth handling with a human, with [105] focusing on collaborative folding with RGB-D and force sensor feedback.

## 4.3 Method

The folding process can be split into four distinct steps:

1. Picking
2. Unfolding
3. Grasping shoulder points

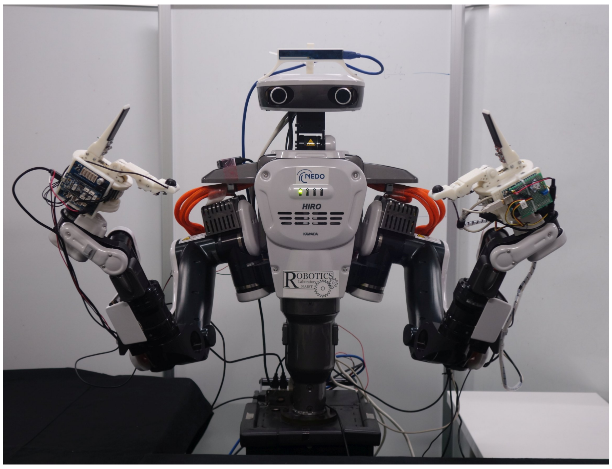


Figure 60. The HIRO NX used in our experiments, equipped with two NAIST OpenHand M2S grippers and a head-mounted Realsense R200.

4. Folding using the tool

The solution we describe uses a Kawada HIRO NX equipped with two NAIST OpenHand M2S grippers as shown in Fig 60. The grasp points are extracted from RGB and depth images recorded using a Realsense R200 mounted on the robot's head.

Detailed sequences of the subtasks as demonstrated in a previous version of this solution can be seen in Fig. 65, Fig. 66 and Fig. 67.

### 4.3.1 Picking

The goal of this subtask is to pick a single garment from a pile of randomly ordered, crumpled clothes, so that it can be manipulated later. Two common modes of failure consist of the robot either accidentally grasping two articles of clothing, or missing the garment and grasping nothing at all. The former, especially, is hard to avoid using only robotic vision, and causes problems when two articles of clothing are inadvertently transferred to the unfolding area. By using the rubbing motion we proposed in chapter 3, we aim to catch failures and detect successes faster, and thus make the process more reliable.

Fig. 62 shows a flow chart of our picking strategy. After picking the highest point of the clothes pile, we perform two regrasps at the point furthest to the

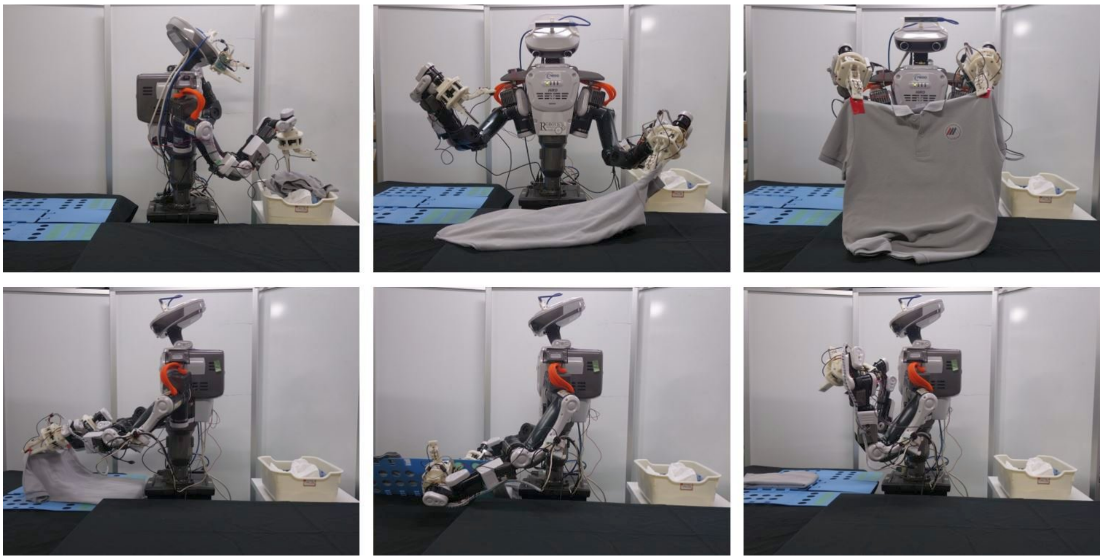


Figure 61. The different subtasks are performed in order. From left-to-right, then top-to-bottom: picking, unfolding, grasping shoulder points, placing the item on the folding tool, finshing.

side of the hanging garment. We perform the rubbing motion after each grasp attempt, to check if the grasp is successful and if only one layer is grasped. If no material is detected, we can immediately move the gripper closer and reattempt the grasp. Without this tactile feedback, a failed grasp would have to be detected visually by dropping the garment.

If only a single layer is detected, the process is considered successful and the robot proceeds to the unfolding section immediately. Tab. 15 shows that single-layer grasps occur over 25 % of the time for shirts, which implies that our strategy has the potential to improve performance.

After each grasp, we also shake the gripper lightly to separate items that may be stuck together. Fig. 63 shows one step of the process.

### 4.3.2 Unfolding

The goal of this process is to bring the shirt from an unknown, crumpled configuration to a planar, unfolded state. To this end, we adapted the method described

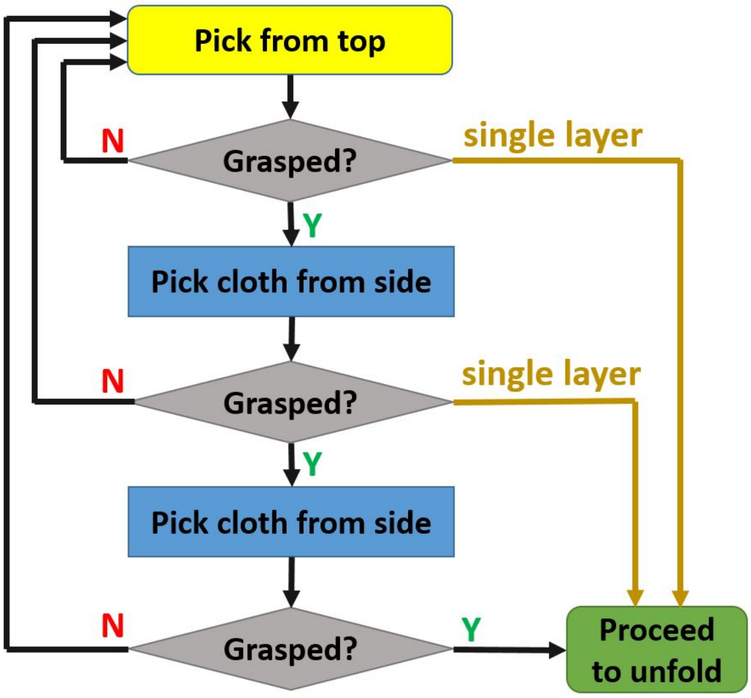


Figure 62. Picking strategy to avoid double grasps. When a single layer is grasped, the process can exit immediately.

by Osawa *et al.* [92]. As the working space of a humanoid robot is wider than it is high, we spread the shirt on a table and grasp a point furthest on the side, rather than the lowest hanging points.[21] The grasping points in this section are determined by canny filtering the camera image of the shirt on the table, binarizing the image and selecting the lateral extreme pixels of the shirt's contour. After two regrasps of the outermost points, the shirt is grasped with both grippers and laid out flat in front of the robot, moving away from the robot.

This leaves the shirt in one of 12 configurations, which are shown in Fig. 64.

### 4.3.3 Grasping shoulder points

The goal of this step is to lift the shirt by its shoulders, so that it can be placed cleanly on the folding tool. To achieve this, we need to determine appropriate

[21] *e. g.* the shirt is grasped and lifted with the right gripper, then pulled from left-to-right, so that the left gripper can grasp the leftmost point.

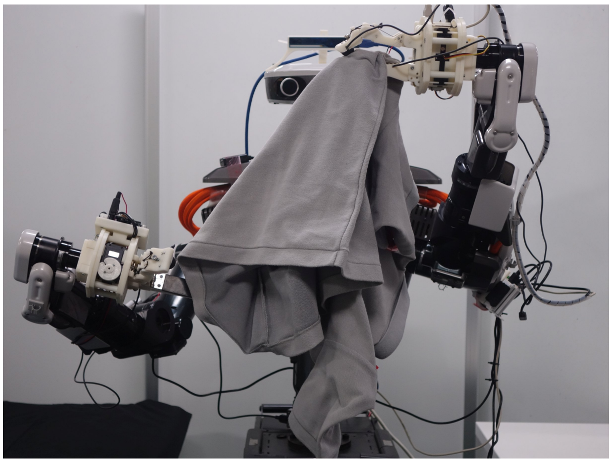


Figure 63. After the picking, the clothing article is lifted up, and the furthest point on its lateral side grasped with the other hand. In this picture, the gripper will recognize that only a single layer of textile is in the new grasp, which allows the system to proceed to the unfolding section immediately.

Table 15. The number of times a regrasp resulted in a multi- or single-layer grasp when handling a short-sleeved shirt during the picking subtask. Single-layer grasps are an opportunity to finish the picking subtask more quickly.

| | Count | Percentage |
|---|---|---|
| Total grasps | 100 | 100 % |
| Multi-layer grasps | 74 | 74 % |
| Single-layer grasps | 26 | 26 % |

grasping points near the shoulders or collar of the shirt. The method we propose to find the shoulder grasp points is based on the contour of the shirt after the unfolding procedure explained in the previous section.

First, we segment the outer edges as shown in Fig 68, by performing the following steps:

1. Binarize the input image. Shirt edges are white.

2. Apply the Guo-Hall algorithm[107] to thin the edges to a single pixel.

3. For each white pixel, count all white pixels at a certain Manhattan distance (we use a distance of 3).

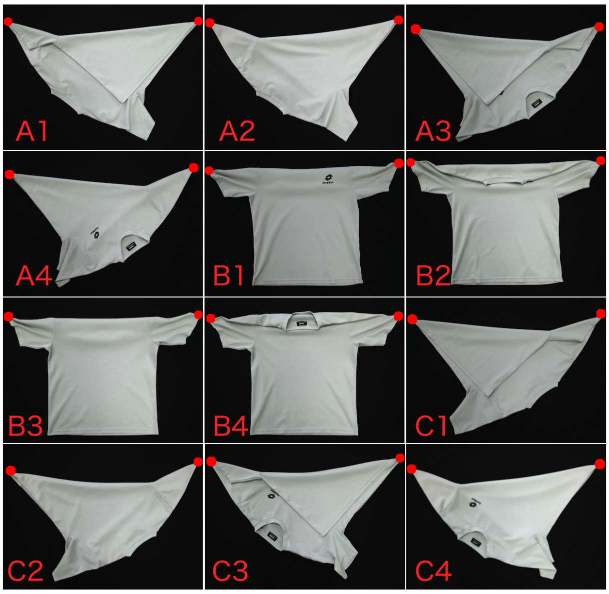


Figure 64. The possible configurations of the shirt after two lowest-point regrasps

4. If more than 2 pixels are white, paint the pixel black.

This deletes line junction points and results in a collection of edges. Each edge is then represented by four parameters: the edge's length, its position in x and y, and its angle[22]. Starting from the largest edge, we store this collection by relative transformation as a $4xn$ matrix $M_c$, and consider that it describes the contour of the garment.

The collection $M_c$ is then compared with the matrices derived from a set of template images in which the grasp points are labeled manually. The template image with the lowest $l_2$ distance is considered the closest match. Taking the largest edge from the matched template image as the base, the labeled grasp points are transformed to the camera image, and the shoulders of the t-shirt grasped. After grasping the shoulder points, the shirt can be placed on the folding tool.

[22] The angle between the x axis and the vector from the edge's first to last pixel. The direction of the vector from ”first” to ”last” pixel ($r_{p1}$ and $r_{p2}$) is chosen such that it points in a counter-clockwise direction around the contour's center point $r_c$.

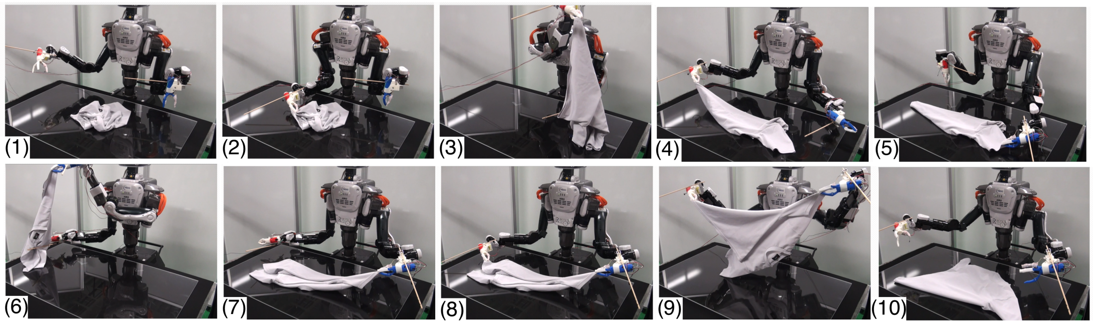


Figure 65. Unfolding sequence executed with 3-point grippers. After spreading the clothing item with one gripper, the point furthest from the hand is regrasped twice, leaving the clothing article in a number of finite states.

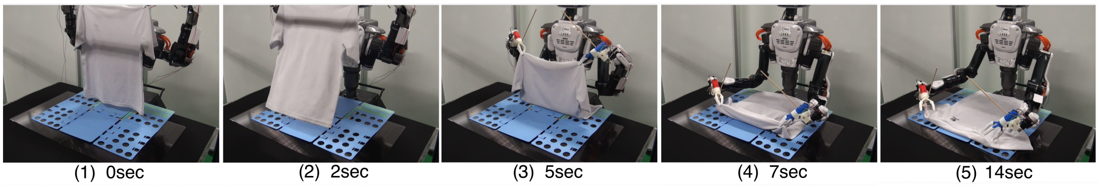


Figure 66. After the shoulder points are grasped, the shirt can be placed easily on the tool.

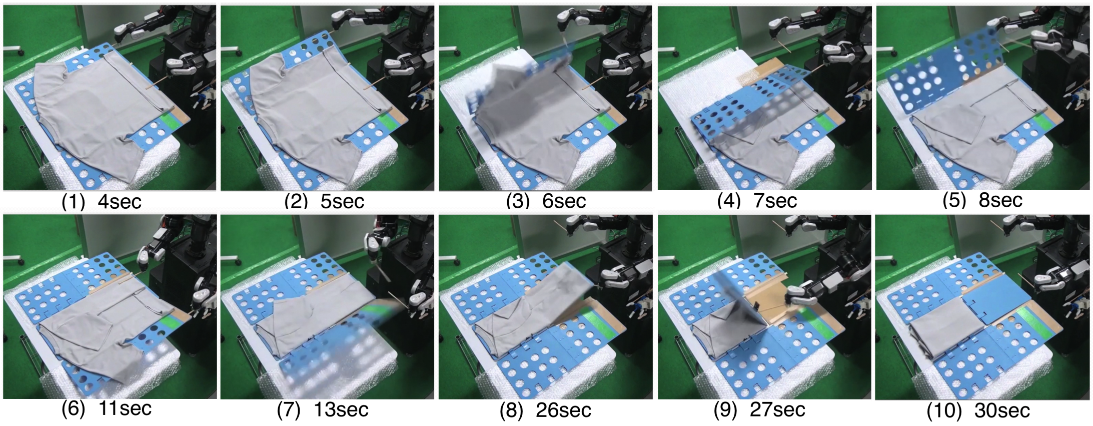


Figure 67. Details of the folding sequence using the commercial folding tool.

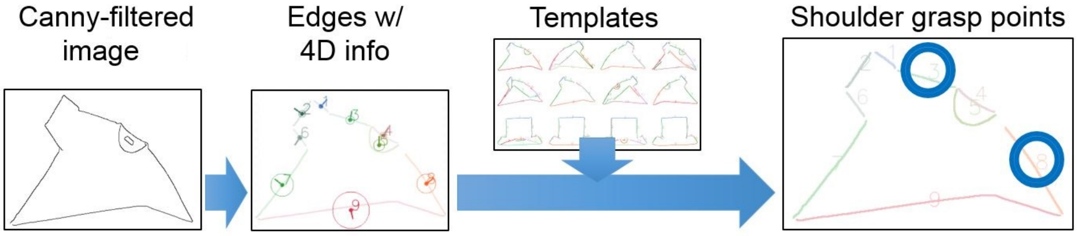


Figure 68. Image processing during the shoulder grasp point search.

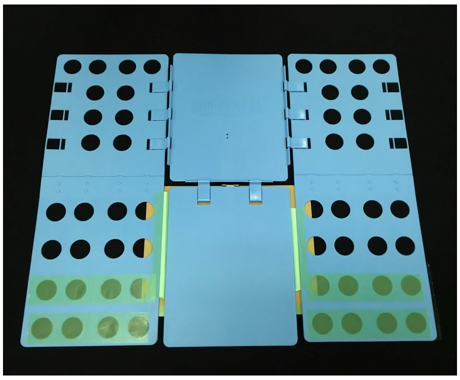


Figure 69. "QUICK PRESS" (Daisaku Shoji Ltd.) folding aid used to fold the shirt

### 4.3.4 Folding

The goal of this step is to fold the shirt using the folding tool seen in Fig. 69. The tool is situated to the right of the robot as shown in Fig. 70, and had a hinge attached to the bottom part of the tool[23] to make it easier for the robot to use with only sliding contacts.

The shirt is placed, centered and pulled taut on the tool, so that no unnecessary wrinkles are generated. The robot then folds and opens the flap on each lateral side, and finally the bottom side. At the end of this procedure, the shirt is completely folded.

[23] The hinge is visible in Fig. 65(9).

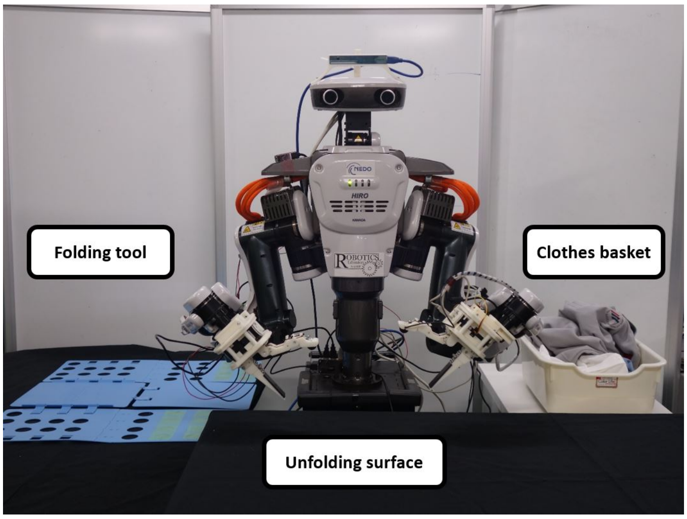


Figure 70. Overview of our folding pipeline. Right: Clothes basket. Middle: Unfolding surface. Left: Folding tool. The robot proceeds from right to left, first by picking, then by unfolding, and lastly folding each clothing item.

Table 16. Average time taken per section of the folding process

| Section | Time taken |
|---|---|
| Picking | 40 s |
| Unfolding | 180 s |
| Grasping shoulder points | 10 s |
| Placing shirt on tool | 10 s |
| Folding | 40 s |

## 4.4 Experiments

In the previous section, we explained the folding pipeline. For our experiments, we implement it as shown in Figs. 70 and 61, using a HIRO NX with two NAIST OpenHand M2S grippers that are equipped with one force sensor each and a thin version of the thumb that contains a compliant hinge joint. This allows the robot to make contact with the table easily without using impedance control, and to slide the passive finger underneath the textile to be grasped.

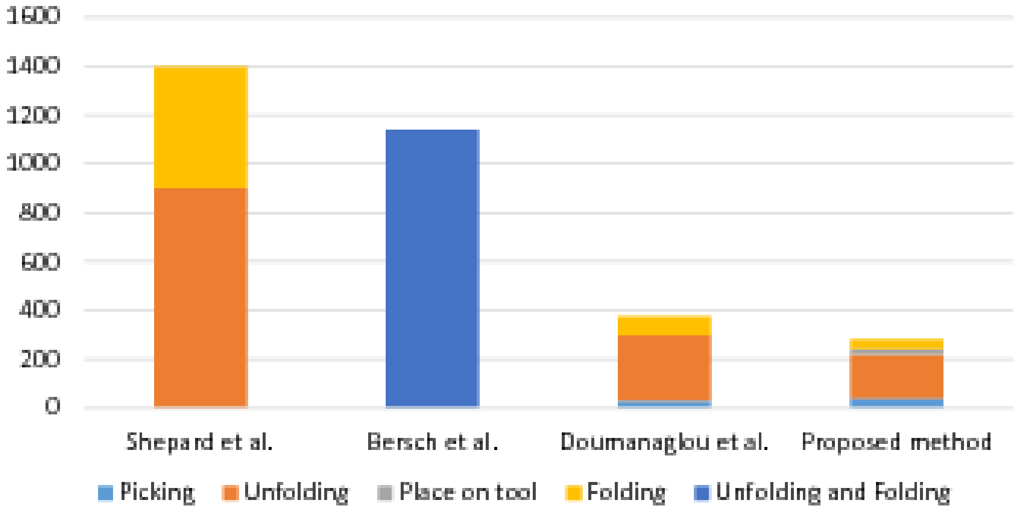


Figure 71. Folding times compared to past work that have reported their speed. Times for Doumanaglou et al. were extracted from their video.

## 4.5 Results

Tab. 16 shows the average time required for the task and subtasks. The Unfolding subtask had the biggest effect on the overall time, as the detection of the collar grasp points can fail, which requires the unfolding procedure to be repeated. This means that while individual unfolding procedures can finish in under a minute, perception failures that lead to repetitions affect the task time significantly.

## 4.6 Discussion

We showed that our proposed system works, and that the rubbing motion increases its reliability and performance. In a previous version of this system, which used a 3-finger gripper without tactile sensing, we found that single grasps during the unfolding section succeeded only 51 % of the time. This was due partially to the thickness of the gripper's finger, which did not always succeed to slide underneath the garment, as well as to noise in the vision system. Our hand improves on this with its thin tip, which can slide underneath the garment, and by the rubbing motion, which allows both a quick confirmation of the grasp, and an early termination during the picking subtask.

As shown in Fig. 71, the use of a tool for humans increases our folding speed significantly. It can also increase the usable workspace of the robot, even when the robot is stationary, by allowing it to interact with an object outside of its workspace. The tool is well suited to our gripper's design, as it can be operated easily by using a sliding contact along the edge of the gripper's thumb.

A frequent mode of failure is the grasping of multiple layers of the shirt when the initial state is very compact, or when many clothing items are tightly packed. We mostly solve this problem for the case of dry shirts by regrasping the hanging garment twice before moving it to the unfolding area, and improve the efficiency of this solution with the rubbing motion we proposed. However, while this process increases the success rate, it does not guarantee fully that a single garment is grasped. For situations where the clothes are highly entangled, a different approach may be required to separate garments.

One weak point in the system is the vision and recognition system. The detection of the collar grasp points can fail if the edges of the shirt are not detected cleanly, and the recognition of patterns with misordered edges is difficult. For future work, we would attempt different approaches such as end-to-end deep learning and shape registration.

The limited workspace of the robot leads to many detected grasp points being unreachable by the robot. This could be avoided if the robot could reposition the garment. If the table surface is smooth, the robot could shift the garment by touching a point in its center and moving it horizontally, so that the grasp points are more accessible. Such a repositioning process would likely be an effective measure that is worth developing further.

Other future work includes the application to more different kinds of clothing, integrating more advanced cloth models and perception methods, and more dextrous manipulation using tactile sensing, such as stretching the clothing item during unfolding.

## 4.7 Conclusion

We described a solution that brings a shirt from an unknown initial state to a folded configuration using our custom gripper, a two-armed robot, an RGB-

D camera and a commercially available folding aid. We showed experimentally that our solution can autonomously fold shirts, and that the rubbing motion that we propose with our gripper can improve the process by detecting earlier if the picking subtask was successful, and by improving the reliability in the other subtasks. Additionally, we showed that the use of a human folding tool improves the performance in the folding task significantly, finishing the task 20-40 % faster than past work, without requiring a large workspace. Future work includes the evaluation of different grasping point detection approaches, ensuring that grasping points are in reach of the robot, spreading the shirt out to reduce wrinkles and improving the manipulation of the folding press with the robot grippers.

# 5. Conclusion

In chapter 2, I investigated the human fingertip's deflection under load, concluding that the change in curvature is most pronounced at low loads.

In chapter 3, I proposed a design for an open-source gripper that utilize the deformation of a rubber half-dome to determine the normal and shear force.

I also showed that the biomimetic finger rubbing motion it can perform extracts tactile data from the material in its grasp, which can be used to recognize not only the material with an accuracy of over 92 %, but also distinguish between single and multiple layers of the same material. Furthermore, I show that by varying the speed and force of the rubbing motion, different tactile information is obtained, which increases the recognition accuracy of successive exploratory motions.

Lastly, in chapter 4 I present an application of a humanoid robot using a tool for humans and its grippers to fold clothes, using both haptic feedback as well as the use of human tools to significantly reduce the time required to complete the task. In summary, this thesis contributes a novel way to a) identify different textiles and distinguish between a single and multiple layers of them, b) manipulate textiles with a serial manipulator and c) complete the task of laundry folding with a humanoid robot.

All the designs and code required to manufacture the gripper and reproduce the experiments are provided open-source online under the name *"NAIST Open-Hand M2S"*.

# References

[1] T. Inoue and S. Hirai, "Elastic model of deformable fingertip for soft-fingered manipulation," *IEEE TRANSACTIONS ON ROBOTICS, VOL. 22, NO. 6*, 2006.

[2] S. Mascaro and H. Asada, "The common patterns of blood perfusion in the fingernail bed subject to fingertip touch force and finger posture," *Haptics-e: The Electronic Journal of Haptics Research*, vol. 22, no. 6, 2006.

[3] J. Abu-Khalaf and S. Mascaro, "Optimization of fingernail sensing technique based on optical experimentation and modeling," in *2011 IEEE Sensors Applications Symposium (SAS)*, 2011, pp. 283–288.

[4] OptoForce, "OMD-20-SE-40N 3-Axis Force Sensors," http://optoforce.com/3dsensor/.

[5] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel, "Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2010, pp. 2308–2315.

[6] A. Doumanoglou, J. Stria, G. Peleka, I. Mariolis, V. Petrik, A. Kargakos, L. Wagner, V. Hlavac, T. K. Kim, and S. Malassiotis, "Folding clothes autonomously: A complete pipeline," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1461–1478, 2016.

[7] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme, "Interactive Perception: Leveraging Action in Perception and Perception in Action," *IEEE Transactions on Robotics*, vol. PP, no. 99, pp. 1–19, 2017.

[8] M. Srinivasan, "Surface deflection of primate fingertip under line load," *Journal of Biomechanics*, vol. 22, no. 4, pp. 343–349, 1989.

[9] M. Tada and D. Pai, "Finger shell: Predicting finger pad deformation under line loading," *IEEE Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, 2008.

[10] Z. Wang, L. Wang, V. Ho, S. Morikawa, and S. Hirai, "A 3-d nonhomogeneous fe model of human fingertip based on mri measurements," *IEEE Transactions on Instrumentation and Measurement, Vol. 61, No. 12*, 2012.

[11] F. Shao, T. Chao, C. Barnes, and B. Henson, "Finite element simulations of static and sliding contact between a human fingertip and textured surfaces," *Tribology International 43*, 2010.

[12] S. Derler and L.-C. Gerhardt, "Tribology of skin: Review and analysis of experimental results for the friction coefficient of human skin," *Tribol Lett (2012) 45:1-27*, 2012.

[13] S. Mascaro and H. Asada, "Photoplethysmograph fingernail sensors for measuring finger forces without haptic obstruction," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp. 698–708, 2001.

[14] J. Abu-Khalaf, J. W. Park, D. Mascaro, and S. Mascaro, "Stretchable fingernail sensors for measurement of fingertip force," in *EuroHaptics conference, 2009 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2009. Third Joint*, 2009, pp. 625–626.

[15] Y. Sun, J. Hollerbach, and S. Mascaro, "Dynamic features and prediction model for imaging the fingernail to measure fingertip forces," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*, 2006, pp. 2813–2818.

[16] Y. Sun, J. Hollerbach, and S. Mascaro, "EigenNail for finger force direction recognition," in *2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 3251–3256.

[17] S. Mascaro and H. Asada, "Measurement of finger posture and three-axis fingertip touch force using fingernail sensors," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 26–35, 2004.

[18] Y. Sun, J. Hollerbach, and S. Mascaro, "Measuring fingertip forces by imaging the fingernail," in *2006 14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2006, pp. 125–131.

[19] Y. Sun, J. Hollerbach, and S. Mascaro, "Predicting fingertip forces by imaging coloration changes in the fingernail and surrounding skin," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 10, pp. 2363–2371, 2008.

[20] T. R. Grieve, J. M. Hollerbach, and S. A. Mascaro, "Force prediction by fingernail imaging using active appearance models," in *World Haptics Conference (WHC), 2013*, 2013, pp. 181–186.

[21] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics.* Berlin Heidelberg: Springer, 2008.

[22] W. J. C. M. R. Tremblay, M.; Packard, "Utilizing sensed incipient slip signals for grasp force control," *Japan-USA Symposium on Flexible Automation*, 1992.

[23] M. Schoepfer, C. Schuermann, M. Pardowitz, and H. Ritter, "Using a piezoresistive tactile sensor for detection of incipient slippage," in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, 2010, pp. 1–7.

[24] A. Mamun and M. Ibrahim, "New approach to detection of incipient slip using inductive sensory system," in *2010 IEEE International Symposium on Industrial Electronics (ISIE)*, 2010, pp. 1901–1906.

[25] K. Yamada, K. Goto, Y. Nakajima, N. Koshida, and H. Shinoda, "A sensor skin using wire-free tactile sensing elements based on optical connection," in *SICE 2002. Proceedings of the 41st SICE Annual Conference*, vol. 1, 2002, pp. 131–134 vol.1.

[26] E. R. Davies, *Machine Vision: Theory, Algorithms, Practicalities.* Elsevier, Dec. 2004.

[27] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge University Press, Mar. 2004.

[28] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library.* O'Reilly Media, Inc., Oct. 2008.

[29] R. I. Hartley, “Theory and practice of projective rectification,” *International Journal of Computer Vision*, vol. 35, no. 2, pp. 115–127, nov 1999.

[30] J.-Y. Bouguet, “Camera calibration toolbox for matlab.”

[31] B. C. Song, “A fast normalized cross correlation-based block matching algorithm using multilevel cauchy-schwartz inequality,” *ETRI Journal*, vol. 33, no. 3, 2011.

[32] C. S. Park and H. W. Park, “A robust stereo disparity estimation using adaptive window search and dynamic programming search,” *Pattern Recognition*, vol. 34, no. 12, pp. 2573–2576, Dec. 2001.

[33] P. Thevenaz, U. E. Ruttimann, and M. Unser, “A pyramid approach to subpixel registration based on intensity,” *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 27–41, 1998.

[34] S. Horn, “Determining optical flow,” *Artificial Intelligence*, vol. 17, 1981.

[35] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[36] C. Schmid, R. Mohr, and C. Bauckhage, “Evaluation of interest point detectors,” *International Journal of Computer Vision*, vol. 37, no. 2, pp. 151–172, Jun. 2000.

[37] T. Tuytelaars and K. Mikolajczyk, “Local invariant feature detectors: a survey,” *Found. Trends. Comput. Graph. Vis.*, vol. 3, no. 3, p. 177 – 280, Jul. 2008.

[38] A. Alahi, R. Ortiz, and P. Vandergheynst, “FREAK: fast retina keypoint,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 510–517.

[39] H. Bay, T. Tuytelaars, and L. V. Gool, “SURF: speeded up robust features,” in *Computer Vision - ECCV 2006*, ser. Lecture Notes in Computer Science, A. Leonardis, H. Bischof, and A. Pinz, Eds. Springer Berlin Heidelberg, Jan. 2006, no. 3951, pp. 404–417.

[40] S. Leutenegger, M. Chli, and R. Siegwart, “BRISK: binary robust invariant scalable keypoints,” in *2011 IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2548–2555.

[41] Schaffroth, “Vergleich von binary feature descriptors,” 2012.

[42] S. J. Lederman and R. L. Klatzky, “Hand movements: A window into haptic object recognition,” *Cognitive Psychology*, vol. 19, no. 3, pp. 342–368, 1987.

[43] R. R. Ma, A. Spiers, and A. M. Dollar, “M2 gripper: Extending the dexterity of a simple, underactuated gripper,” in *IEEE International Conference on Reconfigurable Mechanisms and Robotics (ReMAR)*, 2015.

[44] F. von Drigalski, D. Yoshioka, W. Yamazaki, S.-G. Cho, M. Gall, P. M. Uriguen Eljuri, V. Hoerig, J. Beltran, M. Ding, J. Takamatsu, and T. Ogasawara, “NAIST Openhand M2S: A Versatile Two-Finger Gripper Adapted for Pulling and Tucking Textile,” in *2017 First IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2017, pp. 117–122.

[45] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini, “Tactile sensing-from humans to humanoids,” *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 1–20, 2010.

[46] M. Kaboli, K. Yao, and G. Cheng, “Tactile-based Manipulation of Deformable Objects with Dynamic Center of Mass,” in *International Conference on Humanoid Robots (HUMANOIDS)*. IEEE, 2016, pp. 752–757.

[47] M. Strese, J.-y. Lee, C. Schuwerk, Q. Han, H.-g. Kim, and E. Steinbach, “A Haptic Texture Database for Tool-mediated Texture Recognition and Classification,” *2014 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*, pp. 118–123, 2014.

[48] Y. Mukaibo, H. Shirado, M. Konyo, and T. Maeno, “Development of a texture sensor emulating the tissue structure and perceptual mechanism of human fingers,” in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2005, 2005, pp. 2565–2570.

[49] F. D. Boissieu, C. Godin, B. Guilhamat, D. David, C. Serviere, and D. Baudois, “Tactile texture recognition with a 3-axial force mems integrated artificial finger,” *Houille Blanche Revue Internationale De LEau*, pp. 101–106, 2009.

[50] M. Schöpfer, C. Schürmann, M. Pardowitz, and H. Ritter, “Using a Piezo-Resistive Tactile Sensor for Detection of Incipient Slippage,” *Robotics ISR 2010 41st International Symposium on and 2010 6th German Conference on Robotics ROBOTIK*, pp. 14–20, 2010.

[51] L. Le, M. Zoppi, M. Jilich, R. Camoriano, D. Zlatanov, and R. Molfino, “Development and Analysis of a New Specialized Gripper Mechanism for Garment Handling,” in *The ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference IDETC/CIE 2013*, 2013, pp. 1–10.

[52] L. Le, M. Zoppi, M. Jilich, H. Bo, D. Zlatanov, and R. Molfino, “Application of a Biphasic Actuator in the Design of the CloPeMa Robot Gripper,” *Journal of Mechanisms and Robotics*, vol. 7, no. 1, p. 011011, 2015.

[53] B. Ward-Cherrier, N. Pestell, L. Cramphorn, B. Winstone, M. E. Giannaccini, J. Rossiter, and N. F. Lepora, “The TacTip Family: Soft Optical Tactile Sensors with 3D-Printed Biomimetic Morphologies,” *Soft Robotics*, vol. 00, no. 00, p. soro.2017.0052, 2018.

[54] A. Yamaguchi and C. G. Atkeson, “Combining finger vision and optical tactile sensing: Reducing and handling errors while cutting vegetables,” *IEEE-RAS International Conference on Humanoid Robots*, pp. 1045–1051, 2016.

[55] S. Dong, W. Yuan, and E. Adelson, “Improved GelSight Tactile Sensor for Measuring Geometry and Slip,” in *IEEE International Conference on Intelligent Robots and Systems*, 2017.

[56] V. A. Ho, T. Araki, M. Makikawa, and S. Hirai, “Experimental Investigation of Surface Identification Ability of a Low-Profile Fabric Tactile Sensor,”

in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 4497–4504.

[57] V. Sukhoy, R. Sahai, J. Sinapov, and A. Stoytchev, “Vibrotactile Recognition of Surface Textures by a Humanoid Robot,” *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 488—-497, 2011.

[58] P. Giguere and G. Dudek, “A simple tactile probe for surface identification by mobile robots,” *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 534–544, 2011.

[59] J. A. Fishel and G. E. Loeb, “Bayesian Exploration for Intelligent Identification of Textures,” *Frontiers in Neurorobotics*, vol. 6, p. 4, 2012.

[60] M. Kaboli, A. T. De La Rosa, R. Walker, and G. Cheng, “In-hand object recognition via texture properties with robotic hands, artificial skin, and novel tactile descriptors,” *IEEE-RAS International Conference on Humanoid Robots*, vol. 2015-Decem, pp. 1155–1160, 2015.

[61] D. Tanaka, T. Matsubara, K. Ichien, and K. Sugimoto, “Object manifold learning with action features for active tactile object recognition,” in *IEEE International Conference on Intelligent Robots and Systems 2014*. IEEE, 2014, pp. 608–614.

[62] Y. Tada, K. Hosoda, and M. Asada, “Sensing Ability of Anthropomorphic Fingertop with Multi-Modal Sensors,” *Intelligent Autonomous Systems 8*, pp. 1005–1012, 2004.

[63] N. Jamali, S. Member, and C. Sammut, “Majority Voting : Material Classification by Tactile Sensing Using Surface Texture,” *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 508–521, 2011.

[64] B. Winstone, G. Griffiths, T. Pipe, C. Melhuish, and J. Rossiter, “TACTIP - Tactile Fingertip Device, Texture Analysis through Optical Tracking of Skin Features,” in *Proceedings of the Second International Conference on Biomimetic and Biohybrid Systems*, vol. 8064, 2013.

[65] F. von Drigalski, M. Gall, S.-G. Cho, M. Ding, J. Takamatsu, T. Ogasawara, and T. Asfour, "Textile identification using fingertip motion and 3d force sensors in an open-source gripper," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2017, p. TBA.

[66] T. O' Haver, "An Introduction to Signal Processing in Chemical Analysis," *Dept. Chem. Biochem., Univ. Maryland, College Park, MD*, 2009.

[67] USA Bureau Of Labor, "American Time Use Survey 2015," http://www.bls.gov/TUS/CHARTS/HOUSEHOLD.HTM.

[68] K. Hamajima and M. Kakikura, "Planning Strategy for Task Untangling Laundry - Isolating Clothes from a Washed Mass -," *Journal of Robotics and Mechatronics*, vol. 10, no. 3, pp. 244–251, 1998.

[69] K. Hamajima and M. Kakikura, "Planning strategy for task of unfolding clothes," *Robotics and Autonomous Systems*, vol. 32, no. 2, pp. 145–152, 2000.

[70] B. Willimon, S. Birchfield, and I. Walker, "Classification of clothing using interactive perception," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2011, pp. 1862–1868.

[71] S. Hata, T. Hiroyasu, J. Hayashi, H. Hojoh, and T. Hamada, "Vision system for cloth handling robot," *2009 Iccas-Sice*, no. 1, pp. 3490–3495, 2009.

[72] C. Bersch, B. Pitzer, and S. Kammel, "Bimanual robotic cloth manipulation for laundry folding," in *IEEE International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 1413–1419.

[73] M. Cusumano-Towner, A. Singh, S. Miller, J. O'Brien, and P. Abbeel, "Bringing clothing into desired configurations with limited perception," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2011, pp. 3893–3900.

[74] A. Ramisa, G. Alenyà, F. Moreno-Noguer, and C. Torras, "Using depth and appearance features for informed robot grasping of highly wrinkled

clothes," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2012, pp. 1703–1708.

[75] A. Ramisa, G. Alenya, F. Moreno-Noguer, and C. Torras, "FINDDD: A fast 3D descriptor to characterize textiles for robot manipulation," in *IEEE International Conference on Intelligent Robots and Systems*, 2013, pp. 824–830.

[76] A. Ramisa, G. Alenyà, F. Moreno-noguer, and C. Torras, "Learning RGB-D descriptors of garment parts for informed robot grasping," *Engineering Applications of Artificial Intelligence*, vol. 35, pp. 246–258, 2014.

[77] K. Yamazaki and M. Inaba, "A Cloth Detection Method Based on Image Wrinkle Feature for Daily Assistive Robots." *IAPR Conference on Machine Vision Aplications*, pp. 366–369, 2009.

[78] G. Alenyà, A. Ramisa, F. Moreno-Noguer, and C. Torras, "Characterization of textile grasping experiments," *IEEE International Conference on Robotics and Automation*, pp. 1–6, 2012.

[79] Y. Kita and N. Kita, "A model-driven method of estimating the state of clothes for manipulating it," in *Proceedings of IEEE Workshop on Applications of Computer Vision*, vol. 2002-January, 2002, pp. 63–69.

[80] Y. Kita, F. Saito, and N. Kita, "A deformable model driven visual method for handling clothes," in *International Conference Pattern Recognition (ICPR)*, vol. 3. IEEE, 2004, pp. 243–247.

[81] Y. Kita, T. Ueshiba, E. S. Neo, and N. Kita, "Clothes state recognition using 3d observed data," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2009, pp. 1220–1225.

[82] Y. Kita, E. S. Neo, T. Ueshiba, and N. Kita, "Clothes handling using visual recognition in cooperation with actions," in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, vol. 1, 2010, pp. 2710–2715.

[83] Y. Kita, F. Kanehiro, T. Ueshiba, and N. Kita, “Clothes handling based on recognition by strategic observation,” in *IEEE-RAS International Conference on Humanoid Robots*, 2011, pp. 53–58.

[84] Y. Li, Y. Wang, M. Case, S.-f. Chang, and P. K. Allen, “Real-time Pose Estimation of Deformable Objects Using a Volumetric Approach,” in *IEEE/RSJ International Conference on Intelligent Robots and System*, no. Iros, 2014, pp. 1046–1052.

[85] Y. Li, C. F. Chen, and P. K. Allen, “Recognition of deformable object category and pose,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2014, pp. 5558–5564.

[86] Y. Li, D. Xu, Y. Yue, Y. Wang, S. F. Chang, E. Grinspun, and P. K. Allen, “Regrasping and unfolding of garments using predictive thin shell modeling,” in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, 2015.

[87] A. Petit, V. Lippiello, and B. Siciliano, “Real-time tracking of 3D elastic objects with an RGB-D sensor,” in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem. IEEE, 2015, pp. 3914–3921.

[88] B. Willimon, I. Walker, and S. Birchfield, “3D non-rigid deformable surface estimation without feature correspondence,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2013, pp. 646–651.

[89] B. Willimon, S. Hickson, I. Walker, and S. Birchfield, “An energy minimization approach to 3D non-rigid deformable surface estimation using RGBD data,” in *IEEE International Conference on Intelligent Robots and Systems*, 2012, pp. 2711–2717.

[90] J. Stria, D. Průša, and V. Hlaváč, “Polygonal models for clothing,” in *Proceedings - 15th Conference on Advances in Autonomous Robotics (TAROS 2014)*, 2014, pp. 173–184.

[91] S. Miller, M. Fritz, T. Darrell, and P. Abbeel, “Parametrized shape models for clothing,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2011, pp. 4861–4868.

[92] F. Osawa, H. Seki, and Y. Kamiya, "Unfolding of Massive Laundry and Classification Types by Dual Manipulator," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 11, no. 5, pp. 457–463, 2007.

[93] J. Stria, V. Petrík, and V. Hlaváč, "Model-free Approach to Garments Unfolding Based on Detection of Folded Layers," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[94] B. Willimon, S. Birchfield, and I. Walker, "Model for unfolding laundry using interactive perception," in *IEEE International Conference on Intelligent Robots and Systems*, 2011, pp. 4871–4876.

[95] L. Sun, G. Aragon-Camarasa, S. Rogers, and J. P. Siebert, "Accurate garment surface analysis using an active stereo robot head with application to dual-arm flattening," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, 2015, pp. 185–192.

[96] J. Van Den Berg, S. Miller, K. Goldberg, and P. Abbeel, "Gravity-based robotic cloth folding," in *Springer Tracts in Advanced Robotics*, vol. 68, no. STAR. Springer, 2010, pp. 409–424.

[97] S. Miller, J. van den Berg, M. Fritz, T. Darrell, K. Goldberg, and P. Abbeel, "A geometric approach to robotic laundry folding," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 249–267, 2012.

[98] J. Stria, D. Prusa, V. Hlavac, L. Wagner, V. Petrik, P. Krsek, and V. Smutny, "Garment perception and its folding using a dual-arm robot," in *IEEE International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 61–67.

[99] Y. C. Hou, K. S. Mohamed Sahari, L. Y. Weng, D. N. T. How, and H. Seki, "Particle-based perception of garment folding for robotic manipulation purposes," *International Journal of Advanced Robotic Systems*, vol. 14, no. 6, p. 172988141773872, 2017.

[100] B. Balaguer and S. Carpin, “Motion planning for cooperative manipulators folding flexible planar objects,” in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, 2010, pp. 3842–3847.

[101] Y. Li, Y. Yue, D. Xu, E. Grinspun, and P. K. Allen, “Folding deformable objects using predictive simulation and trajectory optimization,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2015, pp. 6000–6006.

[102] D. J. Balkcom and M. T. Mason, “Robotic origami folding,” *The International Journal of Robotics Research*, vol. 27, no. 5, pp. 613–627, 2008.

[103] C. Elbrechter, R. Haschke, and H. Ritter, “Folding paper with anthropomorphic robot hands using real-time physics-based modeling,” in *IEEE-RAS International Conference on Humanoid Robots*, 2012, pp. 210–215.

[104] C. Elbrechter, R. Haschke, and H. Ritter, “Bi-manual robotic paper manipulation based on real-time marker tracking and physical modelling,” in *IEEE International Conference on Intelligent Robots and Systems.* IEEE, 2011, pp. 1427–1432.

[105] P. N. Koustoumpardis, K. I. Chatzilygeroudis, A. I. Synodinos, and N. A. Aspragathos, “Human robot collaboration for folding fabrics based on force/RGB-D feedback,” in *Advances in Intelligent Systems and Computing*, T. Borangiu, Ed., vol. 371. Cham: Springer International Publishing, 2016, pp. 235–243.

[106] D. Kruse, R. J. Radke, and J. T. Wen, “Human-robot collaborative handling of highly deformable materials,” in *Proceedings of the American Control Conference*, 2017, pp. 1511–1516.

[107] Z. Guo and R. W. Hall, “Parallel thinning with two-subiteration algorithms,” *Communications of the ACM*, 1989.