Proc. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

Computing grip force and torque from finger nail images using Gaussian processes

Sebastian Urban[1], Justin Bayer[1], Christian Osendorfer[1], Göran Westling[2], Benoni B. Edin[2], and Patrick van der Smagt[1]

***Abstract*— We demonstrate a simple approach with which finger force can be measured from nail coloration. By automatically extracting features from nail images of a finger-mounted CCD camera, we can directly relate these images to the force measured by a force-torque sensor. The method automatically corrects orientation and illumination differences.**

Using Gaussian processes, we can relate preprocessed images of the finger nail to measured force and torque of the finger, allowing us to predict the finger force at a level of 95%–98% accuracy at force ranges up to 10 N, and torques around 90% accuracy, based on training data gathered in 90 s.

I. Introduction

Understanding human hand use can be realised at the level of kinematics: observing the position of the fingers may not be simple, but methods exist [1] with which the position and orientation of fingers can be accurately measured, allowing for a reconstruction of the grip or even of the finger kinematics [2].

To analyse human finger and hand use, the position of the fingers or finger tips is an important source of information. However, interaction of our fingers with the objects that we hold can only be measured by knowing the interaction forces, i.e., the force between the finger (tip) and an object.

Various methods have been proposed to measure this force. These methods are however mostly based on directly measuring the force at the interaction point, either by creating instrumented objects, which include a force or a force-torque sensor, or by putting force sensors on the finger tips.

Especially this latter approach is very clumsy. Typically, thin FSRs [3] are used which can only measure one component of force. Putting such sensors on the finger tips changes tactile sensation as well as the surface properties during grip, and natural experimentation is no longer possible. The alternative approach, instrumenting the object, is cumbersome. If unlimited gripping must be possible, the object must be equipped with a tactile skin, a technology which is not yet universally available; also, these usually only measure perpendicular force, and only limited data are available. Alternatively, a number of force–torque sensors can be integrated into the object, but then fingers must be accurately placed on the object, and free grip and manipulation is no longer possible. This approach also severely limits the number of objects as well as their size in experimentation.

[1]These authors are with the Faculty for Informatics, Technische Universität München, Germany

[2]These authors are with the Department of Integrative Medical Biology, Physiology Section, Umeå University, Sweden

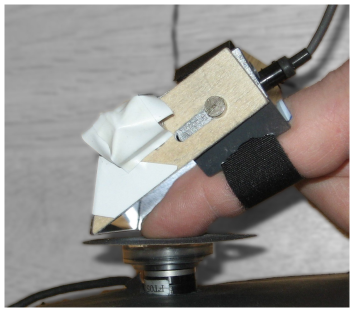
Fig. 1. Picture of our image recording setup. A velcro strap holds the camera on the back of the finger. A two-mirror setup is used to observe the nail.

Our method is based on measurements of finger tip deformations caused by pressure on the volar side. It has been shown [4] that a clear relationship exists between dorsal finger deformation due to volar finger pressure. Inspired by initial work by Mascaro *et al.*, our approach uses the force-dependent blood distribution underneath the finger nail. As shown by those authors, changes in nail colouring can be used to reconstruct forces F_x, F_y, and F_z up to a level of ± 5 N.

In a number of publications starting with [5], Mascaro *et al.* devised various setups to measure finger force from "fingernail touch sensors" based on LEDs and a grid of photodiodes. The sensor, which can be easily worn but must be custom made to ensure tight contact between the sensors and the nail, can be used to reconstruct forces up to ± 2 N in x, y, and z direction [6]. The same group also devised a camera-based system to obtain similar results [7] and a Bayesian classifier to compute force direction from the same source [8]. In their 2008 paper [7], the system estimated finger forces with an accuracy of 5%–10% for a force range of up to 10 N. The setup requires uniform lighting which is obtained by placing the camera, LEDs and finger in a closed dome. The accuracy of the system is obtained by estimating the force of a fixed number of points on the nail and finger bed, which are located after the visual finger recording is normalised based on a previously obtained 3D model of the finger tip.

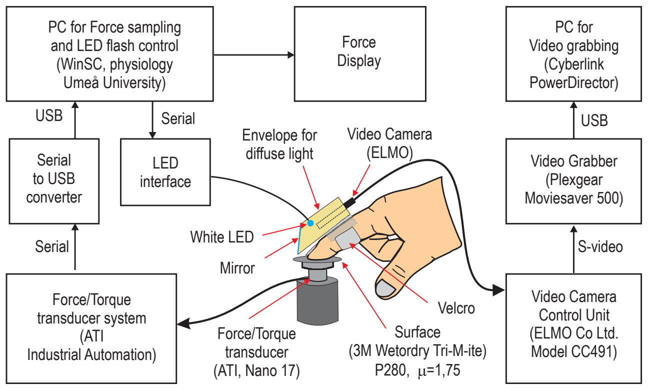
PC for Force sampling and LED flash control (WinSC, physiology Umeå University)
Force Display
PC for Video grabbing (Cyberlink PowerDirector)
USB
Serial
USB
Serial to USB converter
LED interface
Envelope for diffuse light
Video Camera (ELMO)
Video Grabber (Plexgear Moviesaver 500)
White LED
Serial
Mirror
S-video
Velcro
Force/Torque transducer system (ATI Industrial Automation)
Force/Torque transducer (ATI, Nano 17)
Surface (3M Wetordry Tri-M-ite) P280, μ=1,75
Video Camera Control Unit (ELMO Co Ltd. Model CC491)

Fig. 2. Recording setup. An ELMO CCD camera, attached to the finger with a strap band, records visual data of the finger nail while pushing an ATI Nano force-torque sensor.

Contrary to that approach, our method does not need to acquire models of the finger. To improve the visual information our approach uses a finger-mounted camera. A force range of $\pm 5\,\mathrm{N}$ with prediction errors below 5% are obtained by automatically segmenting the nail from the image, normalising the image to obtain lighting independence, and using Gaussian processes to predict the output. Our system does not only allow for force prediction in the x, y, and z direction, but also the torques τ_x, τ_y, and τ_z can be accurately predicted from the pixel image. Apart from obtaining a prediction of the force level, our approach also gives a confidence interval with which the accuracy of the prediction can be evaluated. A training set gathered in 90 s at a frame rate of 25 fps suffices to train our system.

II. Setup

A. Hardware setup

Our final goal is to record grip force from all five fingers of a hand simultaneously, without obstructing the grasp. We therefore aim at recording grip force by visually recording colour changes and deformation from the back of the finger. While we are working on a setup in which a remote camera can observe the complete hand, segment the fingers, and reconstruct grip force from there, in our current setup we simplified the segmentation part by fixing a camera to the finger and recording the nail from there.

The recording setup is depicted in Fig. 2. Visual data is recorded at 25 fps, while the force-torque sensor (FTS) is read out at 100 Hz and subsequently downsampled to 25 Hz. Synchronisation between the visual and force–torque data is realised by flashing a LED, visible by the camera, by the computer recording the FTS data.

B. Processing pipeline

An overview of the processing pipeline is shown in Fig. 4.

1) Nail tracking and stabilisation: Since the camera is mounted on the proximal phalange of the finger, the position of the nail in the recorded video stream is not fixed, i.e., the pixels of the nail move in the recorded picture when the finger is stretched or retracted. The algorithm stabilises the position of the nail in the video by tracking points on the nail and inverting the estimated affine transformation matrix.

Let us denote the (yet unknown) set of points that constitute the nail in the first recorded video frame of a session by $N \subset \mathbb{R}^2$. We will now describe the procedure to track the nail in the video. In the first frame we acquire the positions of the four black, circular markers on the nail by sliding an appropriate template over the image and observing the four locations where the sum of squared differences between template and image patch is minimised. The locations of the four markers define the corners of a rectangular region of interest $R \subset N$, which, by construction, covers only a part of the nail but not the surrounding skin or background. We then determine a set of salient points $S \subset R \subset N$ on the nail by detecting corners in the region of interest R using the Shi-Tomasi [9] minimum eigenvalue method for corner detection. Typically this method extracts 30 points on the nail. During the successive frames of the video stream we track the position of the set of salient points S using the Kanade-Lucas-Tomasi point tracking algorithm [10], [11] with three pyramid levels and a neighbourhood block size of 31×31 pixels. Let us denote the location during frame n of the physical point, that was at $\mathbf{x}_{(1)}$ during the first frame, by $\mathbf{x}_{(n)}$. If a point cannot be tracked during any frame it is removed from the set S of salient points.

We assume that the transformation of the nail N in the video due to its movement relative to the camera is a combination of translation, scaling, rotation and shearing. Therefore we can write

$$\mathbf{x}_{(n)} = A_{(n)} \left[\mathbf{x}_{(1)}, 1\right]^T$$

for every $\mathbf{x}_{(1)} \in N$, where $A_{(n)}$ is the affine 2×3 transformation matrix for frame n.

Under these assumptions we can use the set of about 30 tracked points $S \subset N$ to reliably calculate an estimate $\hat{A}_{(n)}$ of the affine transformation $A_{(n)}$. Since $S \subset N$ and the nail is an approximately planar and rigid object, the result of transforming all points of frame n with the inverted matrix $\hat{A}_{(n)}^{-1}$ is a frame where all points $\mathbf{x} \in N$ of the nail are in the same location as in the first frame.

Thus after applying this processing step all pixels of the nail remain at their initial coordinates during the entire video stream.

2) Nail extraction: Although at this stage the location of the nail is fixed, the video stream still contains images of objects that do not belong to the nail, for example the skin around the nail and the background. To avoid confusion of the force learning and prediction algorithm by these unrelated objects, the pixels belonging to the nail must be recognised and extracted. Although it might

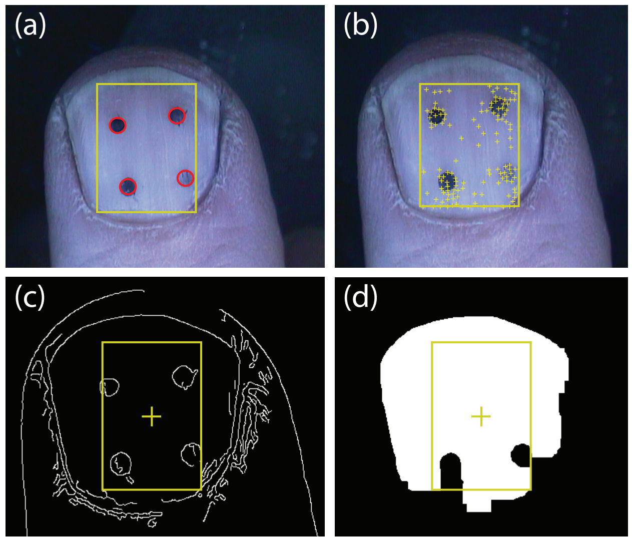

Fig. 3. Stages from the nail tracking and extraction algorithms. a) The four black markers on the nail are found using template matching and their location is used to define a rectangular region of interest. b) Points returned by the corner filter are tracked from frame to frame and used to estimate the affine transformation matrix. c) An edge filter is applied to the first frame of the video to extract the contour of the nail. d) The smoothed contour is filled from the centre point of the ROI rectangle giving the final nail mask.

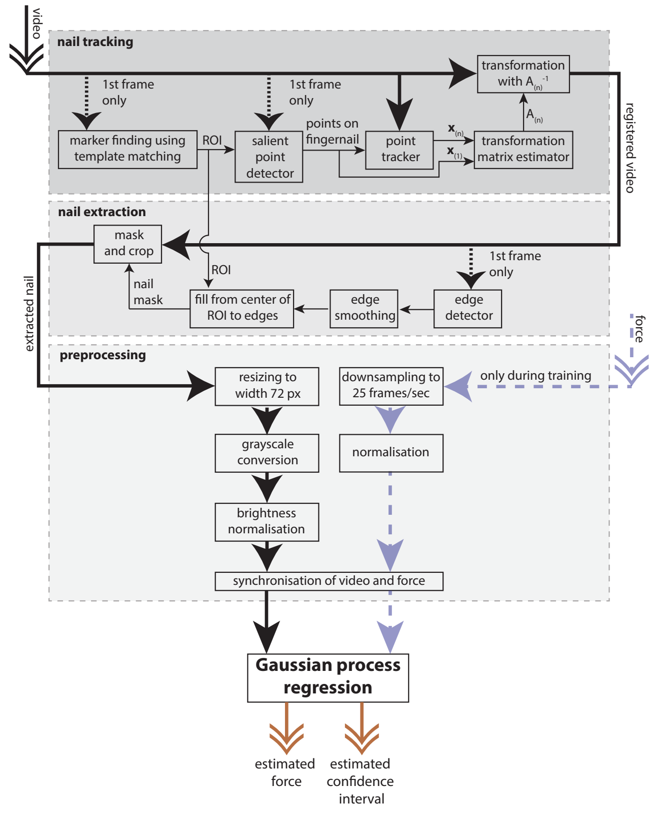

Fig. 4. Training and prediction pipeline.

be necesssary to consider the skin around the nail to extend the range of predictable forces beyond 10 N [7] we decided to exclude it because, unlike the nail, the skin cannot be treated as a rigid object and would require the use of an additional alignment algorithm for non-rigid bodies.

First we find edges in the first frame of the video stream using the Canny edge detection method [12]. The edges determined by this method give a good approximation of the contour of the nail, but it is usually not closed. To fill the gaps in the contour line we apply a morphological closure operation with a neighbourhood size of 25×25 pixels on the extracted edges. We then remove edges that do not belong to the contour line by discaring all edges that consist of less than 500 connected pixels. The final nail mask is given by all connected pixels that lie within the contour line and around the centre point of the four markers, whose location was determined during nail tracking within the computed contour line. Since the nail is stationary in the video stream we can apply the same mask for every frame.

After applying this processing step every frame contains only pixels belonging to the nail. Intermediate results from the image processing pipeline described above are shown in Fig. 3.

3) Preprocessing: After cropping the nail all frames are converted to greyscale and scaled to a width of 72 pixels while preserving the aspect ratio. On average the resulting frames have a height of 75 pixels. Each frame is normalised independently by dividing every pixel by the greyscale mean of all pixels.

The forces are normalised to have zero mean and unit variance over the training set.

The video and force data are recorded independently and synchronised using a bright LED flash in the video stream. This LED is controlled by the force-recording computer, thus allowing for time synchronisation. The force data is downsampled to match the video framerate of 25 frames per second.

C. Learning and prediction

We manually split each dataset into training and test set so that the test set consists of about 15% of the data. Training and prediction is done independently frame by frame.

1) Gaussian processes: A Gaussian process [13] $f(\mathbf{x})$ is a collection of random variables, any finite number of which have a joint Gaussian distribution. It is defined by its mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$,

$$\begin{aligned} m(\mathbf{x}) &= \mathrm{E}[f(\mathbf{x})] \\ k(\mathbf{x}, \mathbf{x}') &= \mathrm{E}\big[\big(f(\mathbf{x}) - m(\mathbf{x})\big)\big(f(\mathbf{x}') - m(\mathbf{x}')\big)\big] \end{aligned}$$

Although Gaussian processes can be defined with a variety of mean and covariance functions, here we will only consider the zero mean function, $m(\mathbf{x}) = 0$, and the squared exponential (SE) covariance function

$$k(\mathbf{x},\mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2l^2}||\mathbf{x}-\mathbf{x}'||_2^2\right) .$$

The hyperparameter l controls the characteristic length-scale of the Gaussian process. The model assumes that points which have distances to each other that are smaller than l should have similar values. The hyperparameter σ_f models the variance of the data.

Given a finite set of training points $\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x_N}$ with corresponding target values $\mathbf{y} := (y_1, y_2, \dots, y_N)^T$ the Gaussian process model predicts that the target value $f(\mathbf{x}^*)$ for a test sample $\mathbf{x}^*$ is normally distributed with mean

$$\mathrm{E}[f^*] = \mathbf{k}^{*T}(K+\sigma_n^2 I)^{-1}\mathbf{y}$$

and variance

$$\mathrm{Var}[f^*] = k(\mathbf{x}^*,\mathbf{x}^*) - \mathbf{k}^{*T}(K+\sigma_n^2 I)^{-1}\mathbf{k}^*$$

where $K_{ij} = k(\mathbf{x_i},\mathbf{x_j})$, $k_i^* = k(\mathbf{x_i},\mathbf{x}^*)$, I is the identity matrix and the hyperparameter σ_n^2 is the variance of the noise of the observations.

In our setting each pair of training point and target value $(\mathbf{x_i}, y_i)$ consists of a preprocessed video frame and the simultaneously measured force/torque. Hence each $\mathbf{x_i}$ is a vector of about 5,000 elements.

To determine optimal values for the hyperparameters σ_f, σ_n and l we calculate the log marginal likelihood

$$\log p(\mathbf{y}|X) = -\frac{1}{2}\mathbf{y}^T(K+\sigma_n^2 I)^{-1}\mathbf{y} - \frac{1}{2}\log\left|K+\sigma_n^2 I\right| - \frac{n}{2}\log 2\pi ,$$

which is a measure of how well the Gaussian process models the data, and maximise it using the conjugate gradient method on the training set with respect to these hyperparameters. This is done separately for each force/torque directions, thus after optimisation each force/torque direction has its own set of hyperparameters.

2) Neural network: We use a neural network with about 5,000 input units, one for each pixel of the preprocessed video frames, one sigmoidal hidden layer with 20 hidden units and a linear output layer with 6 units, one for each force/torque direction. Thus the predicted force/torque vector $\mathbf{f}(\mathbf{x})$ for a video frame $\mathbf{x}$ is given by

$$\mathbf{f}(\mathbf{x}) = W_2\sigma(W_1\mathbf{x}+\mathbf{b_1}) + \mathbf{b_2}$$

where W_2 is a 6×20 matrix, W_1 is a $20 \times M$ matrix, where M is the number of pixels in the video frame, $\mathbf{b_1}$ is a vector with 20 elements and $\mathbf{b_2}$ is a vector with six elements. The network is trained by minimising the objective function

$$E = (1-\lambda)\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{f}(\mathbf{x_i}) - \mathbf{y_i}\|_2^2 + \lambda\left(\|W_1\|_2^2 + \|W_2\|_2^2 + \|\mathbf{b_1}\|_2^2 + \mathbf{b_2}_2^2\right) ,$$

i.e., the mean squared error between the predicted forces/torques and the measured forces/torques is minimised and large values for the network weights are penalized. Our experiments showed that good results were achieved with $\lambda = 0.3$. This relatively large weight penalty ensures that the network considers a large ensemble of pixels for prediction with little significance of each individual pixel, thus improving the stability of the prediction and resilience to small movements of the nail, camera noise, and changes of lighting conditions.

Before training we form a validation set by randomly selecting a *sequence* of samples with a total length of 15% of all samples. Optimisation is done using the scaled conjugate gradient backpropagation algorithm with early stopping when there is no improvement of the objective on the validation set for seven consecutive iterations.

III. Results

We tested our method on data recorded during six sessions with a fixed force/torque sensor. The finger pressure in the sessions S1, S2 and S3 was only applied in the z, x and y directions respectively. Here the z direction is the direction perpendicular to the sensor surface. In the sessions C1, C2, and C3 the finger applied pressure in a circular motion parallel to the sensor surface and constant (C1, C2) or slowly increasing (C3) pressure perpendicular to the sensor. Subjects were not guided to vary the applied torques and there was no feedback during the recording sessions. The data of session C3 is shown in Fig. 5.

The force and torque predictions of the Gaussian process (GP) and the neural network (NN) are examplarily shown for the test set of session C3 in Fig. 6. The force predictions F_x, F_y, F_z of the GP are very accurate for all directions and the true force is contained in the predicted 99.8% confidence interval for nearly all frames. The force predictions of the NN closely follow the true forces, however they are less accurate and more noisy than the predictions made by the GP. The torque predictions τ_x, τ_y, τ_z of the GP and the NN are both qualitatively correct but for τ_y the predicted torque is offset by a fixed value and the predictions for τ_z have a smaller amplitude than the true torque. This can be explained by a systematical shortcoming of our experimental setup: The torque is not measured at the contact point between the finger and the force–torque sensor but at the central axis of the torque sensor. Between the data sets that were recorded, the finger was removed from the sensor and subsequently replaced at a slightly different position. As a result, the measured torque will change; this effect is clearly visible in Fig. 6. In subsequent experiments we will add markers to the sensor surface so that the position of the finger on the surface can be determined from the camera image and thus the measured forces and torques can be corrected accordingly.

In theory it should be possible to achieve better accuracy by using full color information for the force

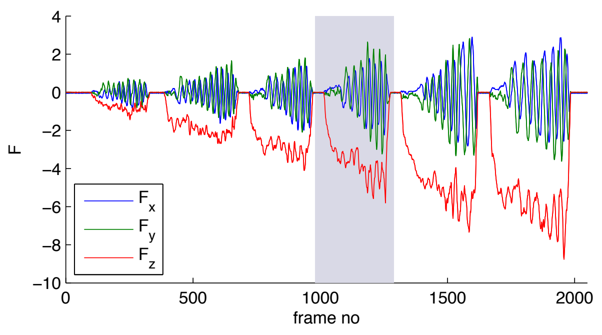

Fig. 5. Recorded forces from session C3. The total length of the recording is 99 seconds. The area shaded in blue is used as the test set. The finger applies forces parallel to the sensor surface in a circular motion and a slowly increasing force perpendicular to the sensor surface.

prediction. However we could not get any significant improvement by doing so, possibly because our video data contained quite high levels of color noise.

To quantitatively compare the performance of neural networks and Gaussian processes we calculate the coefficient of determination

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - f_i)^2}{\sum_{i=1}^{N}(y_i - \overline{y})^2}$$

where $\overline{y} = N^{-1}\sum_i y_i$ over the test set. Table I shows the comparison of the prediction accuracy of neural networks and Gaussian process regression. Although the differences in accuracies are small the GP performs almost always better than the neural network.

IV. Conclusion

We have demonstrated that a very simple, low-cost setup, consisting of an off-the-shelf camera and a force-torque sensor can be used to visually measure finger force and torque within a 10 N cq. 40 Nm range with an accuracy of over 95%.

The method can predict the finger force online at camera frame rate and training can be done in 90 s. The resulting system is therefore perfectly suited for experiments requiring grip force data, but also for detecting grip tremor or measuring other grip force deficiencies.

While the distinction between torque and force at the interaction between a soft finger and a stiff force-torque sensor may be negligible at low forces and torques the data that we record is not interpreted using models of the force-torque transduction, but directly used in learning the model-free representation and reconstructed in the same fashion from the images. Our method is stable independent of whether it is forces or torques that are measured; rather, we compare our images to the 6-dimensional output of the force-torque sensor independent of the physical interpretation.

Based on this simple yet powerful method we will extend our work by allowing for larger force ranges, but also extend our methodologies to the use of stationary cameras which can detect multiple fingers at the same time. Once this has been attained, fully interaction-free grip force measurement can be performed to obtain a full analysis of force and torque from imaging.

V. Acknowledgements

S. Urban was supported by the German Research Foundation (DFG) SPP 1527 Autonomes Lernen. The authors wish to thank A. Weiershäuser for her help with the design of the nail extraction method.

References

[1] D. Gierlach, A. Gustus, and P. van der Smagt. Generating marker stars for 6D optical tracking. In *IEEE International Conference on Biomedical Robotics and Biomechatronics*, 2012.

[2] A. Gustus, G. Stillfried, J. Visser, H. Jörntell, and P. van der Smagt. Human hand modelling: kinematics, dynamics, applications. *Biological Cybernetics*, 106:741–755, 2012.

[3] M. Cutkosky, R. Howe, and W. Provancher. Force and tactile sensors. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, volume C, chapter 19, pages 455–476. Springer Berlin Heidelberg, 2008.

[4] I. Birznieks, P. Jenmalm, A. Goodwin, and R. Johansson. Encoding of direction of fingertip forces by human tactile afferents the journal of neuroscience. *Journal of Neuroscience*, 21(20):8222–8237, 2001.

[5] S. Mascaro and H. Asada. Photoplethysmograph fingernail sensors for measuring finger forces without haptic obstruction. *IEEE Transactions on Robotics and Automation*, 17(5):698–708, 2001.

[6] S. Mascaro and H. Asada. Measurement of finger posture and three-axis fingertip touch force using fingernail sensors. *Robotics and Automation, IEEE Transactions on*, 20(1):26–35, January 2004.

[7] Y. Sun, J. Hollerbach, and S. Mascaro. Predicting fingertip forces by imaging coloration changes in the fingernail and surrounding skin. *IEEE Tr. Biomedical Engineering*, 55(10), 2008.

[8] Y. Sun, J. Hollerbach, and S. Mascaro. Estimation of fingertip force direction with computer vision. *IEEE Tr.Robotics*, 25(6), 2009.

[9] C. Tomasi and J. Shi. Good features to track. In *Proceedings IEEE Computer Society Conference on CVPR*, pages 593–600, 1994.

[10] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence*, pages 674–679, 1981.

[11] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, 1991.

[12] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[13] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

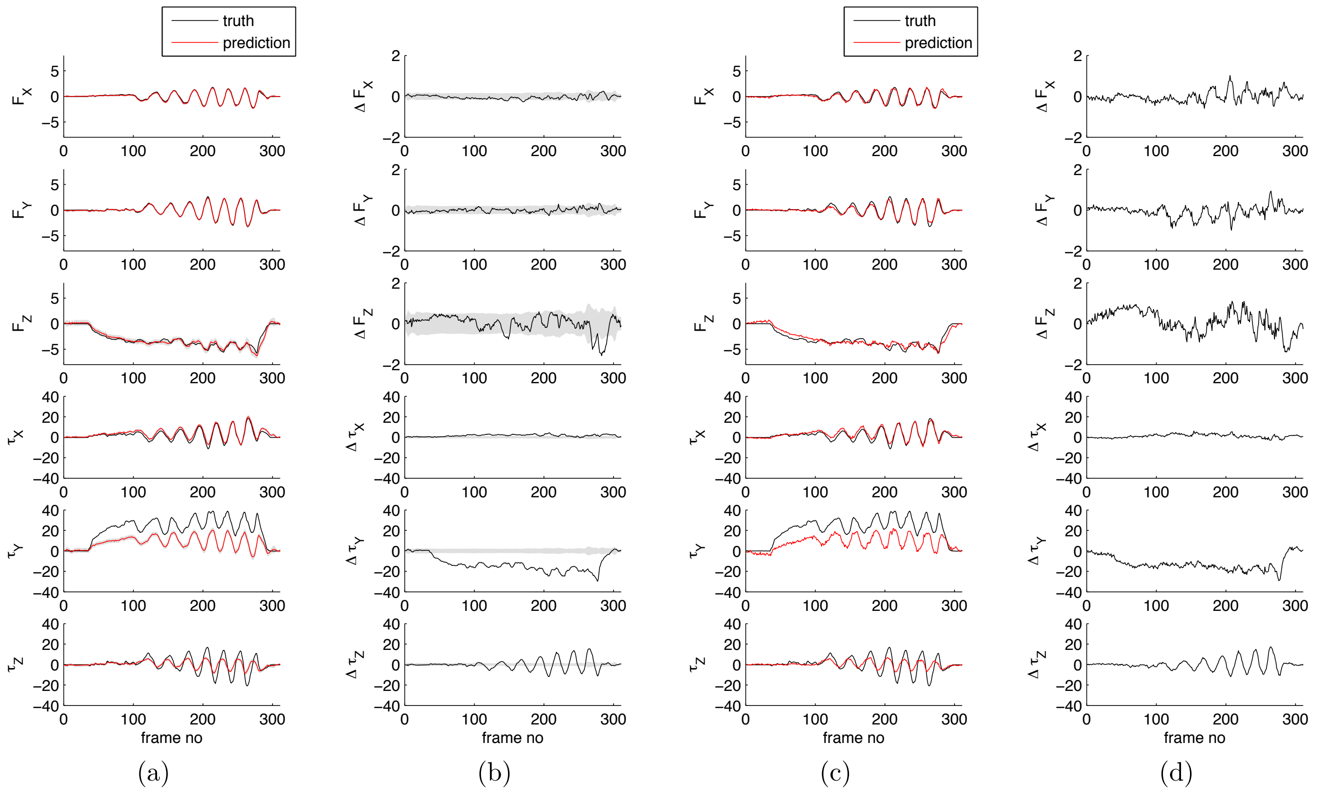

Fig. 6. Comparison of Gaussian process regression and neural network based prediction on the test set of session C3. a) The true value of the force or torque respectively is shown by the black line and the prediction by GP regression is shown in red. b) Difference between the prediction and the true value. In both plots the 99.8% confidence interval of the predictor is shaded in grey. c) Same as (a) using neural networks. d) Same as (b) using neural networks. The offset in the prediction of τ_y is explained in the text.

TABLE I

Accuracy of neural networks and Gaussian processes and the estimated standard deviation given by the GP on the test set.

The better prediction is printed in bold. Since during sessions S1, S2 and S3 the finger only applied forces in the z, y, and x directions, those directions for which not enough data was available to compute an accurate prediction are indicated by a dash.

Dataset		neural network $\sqrt{R^2}$			Gaussian process $\sqrt{R^2}$			GP rel. est. std. deviation		
		x	y	z	x	y	z	x	y	z
S1	force	–	–	0.963	–	–	**0.984**	–	–	0.099
	torque	**0.803**	**0.898**	–	0.787	0.888	–	0.104	0.126	–
S2	force	–	0.926	0.972	–	**0.978**	**0.990**	–	0.068	0.094
	torque	**0.911**	**0.809**	0.911	0.879	0.868	**0.966**	0.077	0.152	0.144
S3	force	0.951	–	0.957	**0.992**	–	**0.963**	0.088	–	0.088
	torque	**0.771**	0.637	0.000	0.695	**0.851**	**0.266**	0.163	0.175	0.333
C1	force	0.974	0.975	0.973	**0.988**	**0.990**	**0.974**	0.095	0.091	0.108
	torque	0.930	0.845	0.876	**0.937**	**0.921**	**0.960**	0.140	0.152	0.104
C2	force	0.933	0.932	0.949	**0.994**	**0.989**	**0.988**	0.086	0.085	0.086
	torque	0.937	0.810	0.818	**0.969**	**0.836**	**0.918**	0.125	0.169	0.125
C3	force	0.950	0.963	0.942	**0.992**	**0.996**	**0.967**	0.072	0.066	0.113
	torque	0.916	0.000	0.692	**0.925**	0.000	**0.733**	0.078	0.068	0.069
average	force	0.952	0.947	0.958	**0.992**	**0.988**	**0.978**	0.085	0.078	0.098
	torque	**0.878**	0.665	0.659	0.865	**0.727**	**0.769**	0.115	0.140	0.155