To appear in *Advanced Robotics*
Vol. 00, No. 00, April 2014, 1–15

FULL PAPER

Global Estimation of an Object's Pose Using Tactile Sensing

Joao Bimbo*, Petar Kormushev†, Kaspar Althoefer* and Hongbin Liu* *

*Centre for Robotics Research, Department of Informatics, King's College London, WC2R 2LS London, United Kingdom;

†Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), via Morego 30, 16163 Genova, Italy

(*v1.1 released April 2014*)

It is essential for a successful completion of a robot object grasping and manipulation task to accurately sense the manipulated object's pose. Typically, computer vision is used to obtain this information, but it may not be available or be reliable in certain situations. This paper presents a global optimisation method where tactile and force sensing together with the robot's proprioceptive data are used to find an object's pose. This method is used to either improve an estimate of the object's pose given by vision or globally estimating it when no vision is available. Results show that the proposed method consistently improves an initial estimate (*e.g.* from vision) and is also able to find the object's pose without any prior estimate. To demonstrate the performance of the algorithm, an experiment is carried out where the robot is handed a small object (a pencil) and inserts it into a narrow hole without any use of vision.

Keywords: tactile sensing; grasping; global search; pose estimation

1. Introduction

Robot grasping and manipulation in unstructured environments is often hindered by the inability of the robot to accurately estimate the pose (position and orientation) of the grasped object. This can lead to wrong assumptions on the stability of a grasp or failure in "pick and place" tasks.

Much research effort has been put into strategies which rely solely on vision (monocular, stereo and RGB-D). Vision-based object tracking combined with grasp planning was first proposed by Kragic *et al.* [1]. Yilmaz *et al.* [2] presented a review on different tracking strategies, and the state-of-the-art in vision-based object tracking has recently seen further improvement [3–5]. Strategies that rely solely on vision such as [6, 7], however, have limitations, particularly during manipulation tasks, where occlusions on the object are bound to occur as the robot fingers get in front of the object or it leaves the camera's field of view [8]. Furthermore, in hazardous environments such as disaster scenarios, robots need to operate in settings with reduced visibility. Examples include underwater operation, burning and smoke filled buildings or total darkness. Hence, object tracking systems need to be complemented with other sensing modalities, such as touch. In fact, an experiment by Rothwell *et al.* [9] proves that even humans fail to perform accurate manipulation tasks when their tactile sensory system is impaired.

This work was supported by the United Kingdom Technology Strategy Board under Grant #131290
*Corresponding author. Email: hongbin.liu@kcl.ac.uk

Early work that combined vision and force sensing for robot grasping can be traced back to Son and colleagues [10], who investigated the advantages of combining these two sensing modalities and Allen *et al.* [11] who, by adding different sensing capabilities to a robotic hand, showed the advantages of vision, force, tactile sensing and their combination. In Honda *et al.* [12] vision is used to track the object and tactile sensing to further refine its estimated pose by minimising the distance from the finger to the object's surface. Another approach uses a description of the object's facets that is done offline and, during runtime, finds possible combinations of facets that match the current sensor measurements [13].

More recently, significant research has been focusing on the combination of vision and tactile information to address the uncertainty on an object's pose. Different combinations of tactile, force and vision information for locating the handle and opening a door were tested and it was proved that the combination of all three modalities outperformed any other possible arrangement [14]. A particle filter approach was used to estimate a tube's pose using both positive and negative contact information – the knowledge of which fingers are touching the object and which are not [15]. Another approach was to model discrete states that contain the possible combinatorial arrangements between fingers and object surfaces using an hybrid systems estimator, estimating these discrete contact modes as well as continuous state variables – *i.e.* the object's pose [16]. Another method used Bayesian Filtering together with a technique called Scaling Series, which allows for successive refinement of the object pose estimate by increasing the granularity of the search region [17]. A framework where the grasp, sensing data, stability and object attributes such as pose are all modelled probabilistically is presented in [18]. Koval *et al.* [19] presented a method to continuously track a continuously pushed object in two dimensions using a modified particle filter. Object pose uncertainty can also be reduced by gaining tactile information from attempted grasps and replanning the grasp to increase the chances of success [20]. In [21], an approach is presented that uses tactile sensors to allow the recognition of the object and its 6-D pose by means of exploring the object's surface and edges using Iterative Closest Point combined with a particle filter. A collision checker combined with a particle filter was also used to estimate the in-hand object pose, starting from an initial pose acquired from vision and estimating the pose according to the hand's movements [22]. Extensively literature exists which deals with the uncertainty of the object's location and offer strategies to tackle the problem of unreliable information, proposing methods to increase the robustness of a grasp, but do not attempt to estimate the pose of objects [23–30].

The method proposed in this paper is based solely on force and proprioceptive data, and can estimate the pose of a grasped object given only its current state, using a global search method based on an evolutionary algorithm. The algorithm uses the rich contact information given by custom designed sensors and finds object poses which are coherent with the measured contact information. It extends the authors' previous work [31, 32], by making a global search instead of a gradient-based optimisation. The advantages of a global search are two-fold: it avoids local optima, which are prone to exist as the complexity of the handled object increases, and enables the estimation to be carried out without any initial estimate. As such, the proposed method can work under two different circumstances: correcting the pose information given by a 3D vision-based system and finding the pose of a known object given no prior knowledge of the object's pose. It uses the joint encoders and the contact position and force normal from tactile sensors and requires the object's geometry. The proposed algorithm can also be used to provide candidate poses which can be used as an initial proposal distribution in a sequential estimation such as a particle filter. Objects with any degree of complexity can be tracked as long as there is a sufficient number of contacts to discriminate between similar poses.

The problem is presented in the next chapter, as well as the description of the proposed method along with some implementation details. Section 3 describes and discusses the results both in simulation and with a real system. Section 3.2.3 presents a possible scenario where the method is used in interaction with humans. Section 4 presents the conclusions of this work.

2. Object Pose Estimation

2.1 *Problem Description*

The objective of this paper can be formulated as the estimation of a set of parameters $\mathbf{x}$ which describe the pose of an object – position and orientation – which matches the current tactile and kinematic information. In other words, given a certain hand posture and current contact information, what object pose(s) satisfy these measurements. The parameters to be estimated are a rotation and a translation (a vector and a quaternion), as shown in (1). The choice of quaternions for parametrising the rotation was made taking into account the computational and mathematical advantages over other notations, such as Euler angles or rotation matrices [33–35].

$$\mathbf{x} = [\ \mathbf{q}\ ,\ \mathbf{t}\ ] = [q_w, q_x, q_y, q_z, t_x, t_y, t_z]^T \tag{1}$$

Besides the geometric shape of the object, which needs to be known *a priori*, the available sensor information consists of the contact location on the fingertips and the interaction forces. This approach takes advantage of the fact that, for rigid contacts, the surface normal coincides with the measured normal force direction. Taking into account this normal force information not only improves the overall accuracy of the fitting but clearly discriminates on which of the object's face the finger is touching, which is fundamental for the success of a manipulation task. The objective is then to find $\mathbf{x}$ such that the distance between our measured contact location and the angle between the object surface normal vector and the measured contact normal are both minimised. Since object models usually consist of thousands of vertexes, applying transformations on all these points and their respective normals would be computationally very expensive. Instead, the goal becomes to find the transform applied on the contacts until a result is found. The inverse of this solution is then applied to the object.

The devised cost function shown in (2) consists of the sum of the cost for each of the k fingers touching the object. It takes into account the squared distance from the contact location $\mathbf{f}_m$ on the finger to a point $\mathbf{s}$ belonging to a surface S in the vicinity of that contact location and the angle between the surface normal $\hat{\mathbf{n}}$ at that point and the measured normal force direction $\hat{\mathbf{u}}$. This sum is mediated by a weighting factor w_n that is chosen according to the accuracy of the object geometric model and of the sensor. $\|\cdot\|$ denotes the Euclidean norm and $\langle\cdot,\cdot\rangle$ denotes the inner product.

$$G(\mathbf{x}) = \sum_{m=1}^{k} \min_{s_i \subset S} \left(\|(\mathbf{q}\mathbf{f}_m\mathbf{q}^* + \mathbf{t}) - \mathbf{s}_i\|^2 + w_n(1 - \langle \mathbf{q}\hat{\mathbf{u}}_m\mathbf{q}^*, \hat{\mathbf{n}}_i \rangle \right) \tag{2}$$

As mentioned in the previous section, this paper presents two scenarios for a manipulation task. First, the method can be used together with a vision tracker, starting from an initial guess detected by vision and setting a reduced search space, allowing for a very fast detection of the correct pose. The second method starts with no knowledge of the object pose and searches the whole space around the robot hand to find suitable pose(s), ranking them according to how well they fit the sensory data.

2.2 *Method*

2.2.1 Set up

The first step of the algorithm takes the object polygon mesh and computes the normal vector for each face, using the cross product between two vectors defined by the vertices. Active contacts

(contact force above a threshold) are then selected and transformed to be expressed on a common coordinate frame with object mesh being transformed likewise. A k-d tree is constructed with the object pointcloud to allow for easier distance queries. The creation of this k-d tree is done using PCL kdtree flann implementation [36, 37].

2.2.2 Search Algorithm

The search method used to obtain the transformation parameters belongs to a class of methods commonly called Monte Carlo, originally developed in the 1940's by Metropolis and Ulam [38]. These methods, while originally devised for mathematical physics problems, have been extensively used in the field of robotics, particularly in localisation problems for mobile robots [39–41] and in global search for optimal policies in reinforcement learning[42, 43]. The idea behind this class of methods is to randomly draw samples from an unknown distribution. The applications range from approximating parameters such as the expected value of a probabilistic event, simulate stochastic processes or, as is the case in this paper, to estimate parameters in an optimisation problem. More specifically, the used method can be classified as an Evolutionary Algorithm, where the purpose to find a set of parameters that minimise the cost function (2) by sequentially replicating the most suitable guesses (henceforth referred to as *particles*) with a probability related to each particle's fitness (coherence with the sensor data).

2.2.3 Generating the Population

An initial "population" of pseudo-random particles is created, ensuring a distribution inside the search space which is suitable for each of the two applications concerned in this paper. For pose correction, it starts from a rough pose estimate given from vision, it is sufficient to create random particles in a Gaussian distribution around the initial estimate. As for the global search, where there is no initial estimate, we need to ensure the search space is evenly covered. For the translation vector, this is done simply by creating a uniform pseudo-random distribution. For the rotation quaternion it is accomplished both through the method suggested by Marsaglia [44] and by creating a small set of particles in which q_w is either 1 or -1 and one other element is 1. These latter ensure that, after normalisation, there will be particles in the population which represent the object in six "straight" orientations ("upside down", "left side up" etc.), which everyday objects typically tend to be in.

2.2.4 Sampling

After the initial population has been generated, the algorithm should replicate the estimates which best minimise the cost function. As such, an equation was devised which inversely relates the probability of a particle to be replicated (its "weight" W) to its cost G. Equation (3), shows the chosen equation where p_p can be adjusted, again depending on the desired application. A higher p_p is used for pose correction, allowing a quicker convergence and a more "aggressive" search, sacrificing however the possibility of finding multiple solutions. As for the global search, where it is crucial not to be trapped in local minima, this value is lowered. Figure 1 represents how the parameter p_p affects this cost to weight conversion.

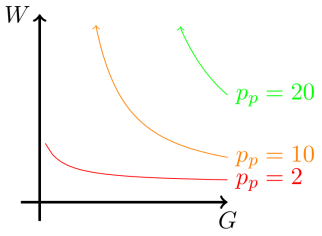


Figure 1.: Cost to Weight Function

$$W(\mathbf{x}) = \left(1 + \frac{1}{1 + G(\mathbf{x})}\right)^{p_p} \tag{3}$$

2.2.5 *Noise addition*

The addition of noise, or according to some authors, *perturbation* or *variation*, is another essential step for the algorithm, as it allows the search to be performed locally around the sampled particles. The selected scheme for adding noise consisted of creating normal pseudo-random values with decreasing variance on each iteration. Also, only two parameters were changed at one time – one in the rotation and one in the translation. These normal pseudo-random numbers were created using Box-Muller transform [45], which conveniently creates a pair of normally distributed numbers each time. The way the standard deviation evolves over the particle number j, given a desired total number of iterations n_p is shown in (4). This added noise is initially set to σ_0 and tends to zero as it approaches the end of runtime and the rate at which it decreases is defined by changing the power p_n.

$$\sigma = \sigma_0 \left(1 - \frac{j}{n_p}\right)^{p_n} \tag{4}$$

2.2.6 *Evolution of the algorithm*

The algorithm runs for a chosen maximum number of iterations and the best cost is saved, along with the last 1% of all particles. The fact that the population is always increasing and not replaced as it is commonly done in Genetic Algorithms has to do with the computational efficiency, as it would not present any advantage in terms of performance, as discussed in Section 2.2.8. In order to reduce the run time of the method, a stopping criterion was created, against which the current lowest cost was tested every thousand iterations. Since the cost $G(\mathbf{x})$ does not, on its own, provide enough information about the quality of this estimate, a confidence indicator was devised, which takes into account the number of contacts k and the selection of w_n. This confidence indicator $C(\mathbf{x})$ can be seen as the inverse of the average error over the k contacts, scaled to the range $0 < C(\mathbf{x}) \leq 100$.

$$C(\mathbf{x}) = 100/(1 + 100\left(\frac{G(\mathbf{x})}{k \cdot (1 + w_n)}\right)) \tag{5}$$

Figure 2 shows the evolution of the particle cost (note the log scale) where each particle is shown as a blue dot, the average cost is shown in red and the current best estimate in green. It can be seen that the algorithm keeps converging to particles with lower cost. In this example, the algorithm ran for the maximum number of iterations, not reaching the previously mentioned stopping criterion.

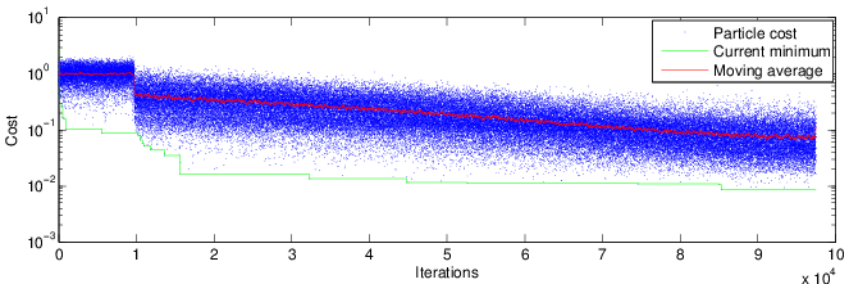


Figure 2.: Progress of algorithm – cost over iterations

2.2.7 *Post processing*

When finding the pose of an object without any initial estimate, some objects can yield multiple solutions. This can arise from having few fingers touching the object or from object symmetry. As such, the algorithm outputs a number of possible poses which can then be kept

for posterior evaluation. The algorithm requires that two solutions have sufficiently different positions or orientations to be deemed different.

After a group of solutions is obtained, these solutions are tested for collisions with the robot. In order to have a computationally fast evaluation, the collision checking was made as simple as possible, requiring only that the object does not have any of the points in its surface in a the vicinity of a number of points inside the robot (knuckles, palm, etc.). If a possible pose violates this condition, it is intersecting the robot's geometry and it is discarded.

Finally, a Levenberg-Marquardt gradient search [46] is performed, further improving the estimate. The details of this step were previously shown by the authors [32]. This step ensures that the solution found is a minimum in that region.

2.2.8 Computational Remarks

In order to improve the computational performance of the algorithm, different tactics were used on each step to allow the pose correction to be run at similar frequency as the vision tracker and the global search with no initial estimate to run within reasonable time (around two seconds).

The first strategy, as already described concerned the use of quaternions, allowing rotations to be applied without the use of trigonometric functions, known to be computationally expensive. The second consideration was to find the transformation on the finger, avoiding the operation to be done on the object, which could contain tens of thousands of vertexes and normal vectors at every iteration. Thirdly, the use of a k-d tree allowed evaluations of the cost function to be done much more efficiently.

Finally, the implementation of the importance sampling scheme was made carefully considering computational performance. Each time a particle is generated, its weight is saved into an array and added to an accumulated sum σ_W. To generate a new particle, a uniform pseudo-random number $r_n \in [0, 1]$ is multiplied by this accumulated sum to obtain a number r in the interval $[0, \sigma_W]$. The particle x_d to be replicated will be the one which, on the array of these accumulated weights, will be located where $\sum_{k=0}^{d} W(k) > r$. The procedure here is to begin the search from the end of the array, taking advantage of the fact that, as the algorithm progresses, particles with higher weight (lower cost) will be at the end of the array. Figure 3 shows an example of how these weights may be distributed. If the particle to be sampled sits at the position pointed by the red arrow, which is approximately in the middle point, much less operations will be needed if one starts subtracting from the end of the array than adding from the beginning. This allows for a much faster sampling while maintaining the conditions for Importance Sampling.

The choice of maintaining all previous particles, instead of having a smaller number of particles that are replaced, was made due to practical and computational considerations. First, it allows to keep the diversity of the population, avoiding "particle deprivation" [41]. Computationally, this strategy presents the advantage of not requiring the weights of each previous particle to be recalculated. The trade-off in this strategy is of course memory requirements, but given the availability of memory in modern computers, it does not pose a problem, as the required memory will be typically in the range of a few megabytes.

Figure 3.: Weights of particles over time

Figure 4 shows the computation time to generate each thousand particles. Typically, the generation of each particle would require increasing time with the number of previous particle it samples from. Using these strategies however, allows the algorithm to maintain nearly a constant duration, making the algorithm's computation time to depend linearly on the number of particles required. Depending on the selection of p_p in equation (3), a more "agressive" search will decrease the runtime of the algorithm. In the case of figure 4, which had a high p_p, sampling from the

existing distribution becomes even faster than the initial generation of population. This is due to the fact that to generate an initial particle requires six pseudo-random values, sampling and adding noise requires only three (one for importance sampling and two for noise addition).

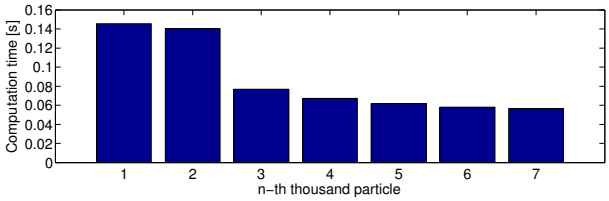


Figure 4.: Computation time to generate each thousand particles

3. Results

This section presents the results obtained in both simulation and a real robot, validating the proposed algorithm. The quantitative validation in section 3.1 takes place in a simulated environment as accurate ground truth values can be obtained directly. Two sets of experiments were carried out: On the first experiment, the object is displaced with a small rotation and a translation from its true location. This is done so as to simulate what is obtained from a vision-based object tracker when the object is enveloped by the robot hand. The second experiment uses a wine glass where no prior knowledge of its approximate location is available. The simulated sensor data was altered with increasing Gaussian noise on the contact location and normals to test the method's robustness. The results are shown with respect to added noise and number of fingers touching the object. It should be noted that each data point on the plot is a one-shot estimate and it does not rely on previously estimated poses. This choice was made in order to show the performance of the algorithm on its own, although in a practical situation the algorithm's initial condition could be the previously estimated pose. Section 3.2 shows results for a real system and qualitative evaluation, because of the difficulty to have a sufficiently accurate ground truth values.

3.1 *Simulation*

3.1.1 Pose correction

The first scenario uses the algorithm starting from a coarse estimate of the object's pose. The object was randomly displaced from its true location by a small amount in both rotation and translation. The pose correction algorithm is then set to use a reduced search space – angle under 45° and a maximum translation of 5 cm. Figure 5 shows a result of the correction algorithm. The tested object was a small 3D printed statue[1] and it can be seen that even for an object with such complex geometry, the solution is very close to the ground truth.

Figure 6 shows the results for pose correction, where the upper row displays the initial error in translation and rotation and the lower row the results after running the proposed algorithm. This initial error acts as, for example, a vision system, which gives an approximate estimate of the object pose, and was done in simulation by displacing the object randomly in every direction by a small amount. The average distance between the frame at the ground truth and this displaced pose was 33.2 mm and the average rotation angle was 16°. This is comparable to the typical accuracies of vision tracking when a small object such as the statue is being grasped.

[1]The bust of the poet Sappho was kindly provided by Artec3D – www.artec3d.com

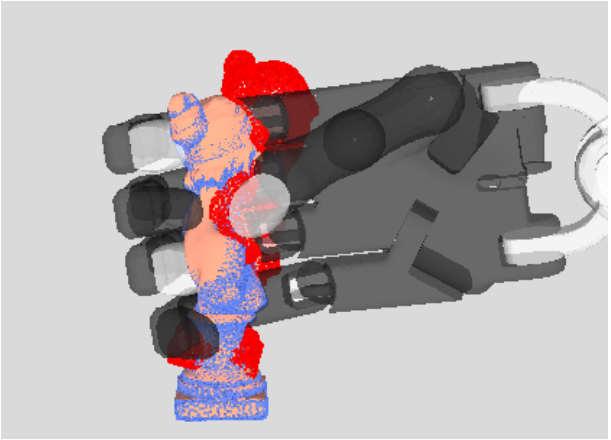


Figure 5.: Pose correction. Initial estimate in red, ground truth in pink and result estimated pose in blue

In this example, random Gaussian noise was added to the contact location and normal direction with means of 0.9 mm and 5°, which is above the errors existent on the real system described in Section 3.2 and five fingers were touching the statue. After applying the proposed method, the errors were reduced to an average of 4.05 mm and 5.0°, with standard deviations of 2.8 mm and 2.19° while the average run time was 0.64 seconds.

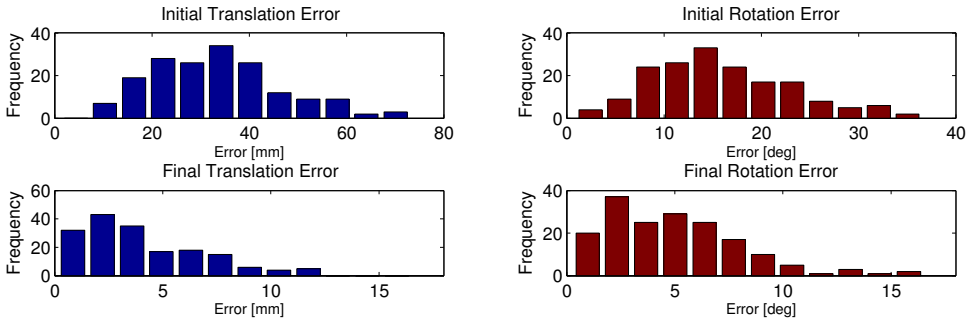
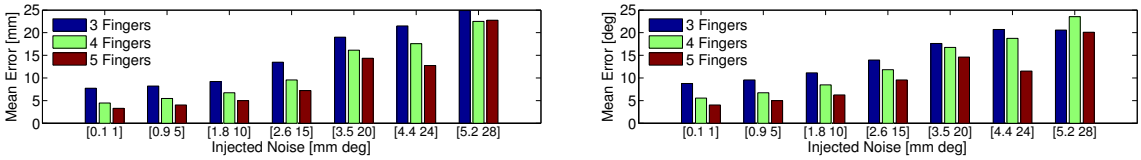


Figure 6.: Histograms for initial and final errors on rotation and translation for pose correction

To further evaluate the performance of the algorithm, the level of sensor noise was increased and simulations were carried out where the object was being grasped by three, four and five fingers. Also, an arbitrary threshold of 1 cm for translation and 15° for rotation was put in place, considering successful any result that stays within this limits. It should be pointed out that these errors are from the object's frame of reference, which in the case of the statue is at its base. The angle error is calculated according to the formula $\theta = 2cos^{-1}(q_w)$, which encompasses angle errors around every axis.

The histograms in figure 7 shows the mean error in rotation and translation according to the noise level and number of contacts. Figure 8 shows the percentage of success according to the previously defined criteria. It can be seen that the algorithm finds the correct pose over 90% of the times when the noise is below 1.8 mm and 10° and five fingers are touching the object.



(a) Translation error (b) Rotation error

Figure 7.: Mean errors after pose correction for different number of contacts and noise levels

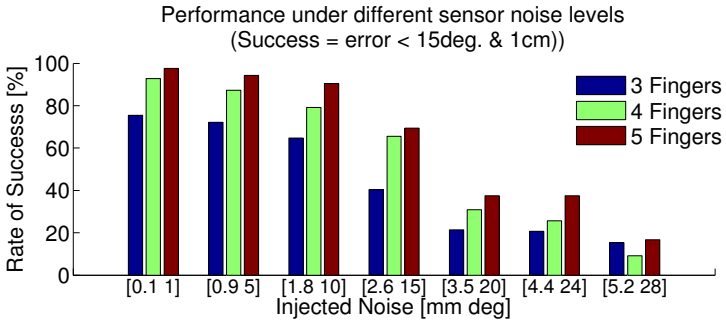


Figure 8.: Rate of success for pose correction for different number of contacts and noise level. A trial is considered successful if the error is under 1 cm and 15°.

3.1.2 Global pose estimation

This experiment shows how the pose of the object can be determined using no vision input, relying solely on the robot's proprioception and the force sensors on the fingertips. Applications of this method could range from situations or environments where it is unfeasible to have a functioning vision system. The proposed example uses a wine glass, which common image or RGB-D tracking systems would fail to track as it is transparent. Figure 9 shows a result of a trial where the object is put at an arbitrary location and the resulting estimated pose overlays the ground truth.

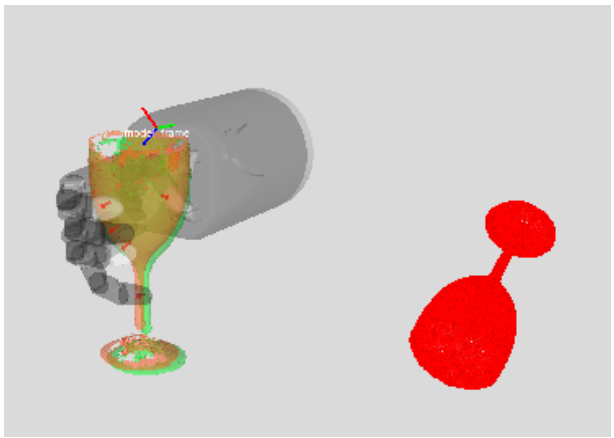
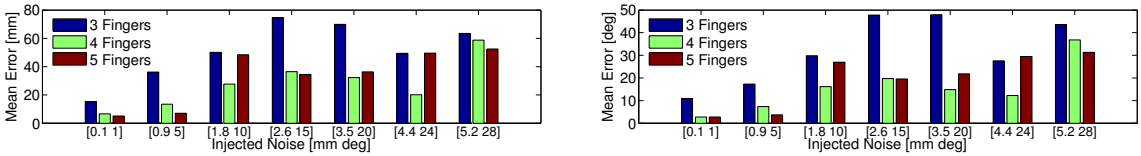


Figure 9.: Global pose estimation. Initial estimate in red, ground truth in green and result pose in orange, force normals are displayed as red arrows

The results in this case are much more dependent on the number of fingers touching the object than in the previous section where there was an initial estimate. This is due to the existing symmetries in the object which, for a small number of contacts, can have a variety of poses that fit those contacts. When using five fingers, and given that one finger is touching the glass stem,



(a) Translation error (b) Rotation error

Figure 10.: Mean error in global pose estimation for different number of contacts and noise levels

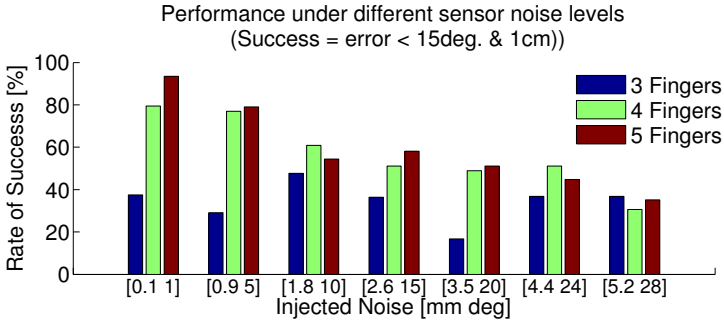


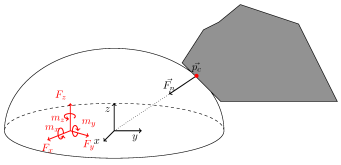
Figure 11.: Rate of success for global pose estimation different number of contacts and noise level. A trial is considered successful if the error is under 1 cm and 15°.

so as to not yield "upside down" poses, the mean absolute error was 7.1 mm for position and 3.72° for the vertical angle. Standard deviations were 9.1 mm and 5.88° respectively, with an average duration of 63 seconds.

3.2 *Real System*

3.2.1 Experimental Setup

The algorithm was implemented in a real system, using a Mitsubishi RV6SL robot and a Shadow Dexterous Hand™ with only three force-torque sensors mounted on the fingertips. The required contact information – contact location and normal force direction – are measured using a scheme called *intrinsic* contact sensing, described in Bicchi *et al.* [47]. Equation 6 and Figure 12 illustrate this scheme where, using 6 axis force-torque sensing under a parametrisable convex shape S – a semi-ellipsoid in this case – one can solve the system of equations consisting of the surface equation and the force $\mathbf{F} = [F_x, F_y, F_z]$ and moment $\mathbf{M} = [m_x, m_y, m_z]$ balance, yielding a unique solution for the contact location $\mathbf{p_c}$ and a torque τ around the contact normal $\hat{\mathbf{n}}$. From here, it is trivial to decompose the total measured force into its normal and tangential components. This approach has been previously validated by the authors in Liu *et al.* [48] showing an accuracy of 0.224 mm in contact location, corresponding to an error of 1.5° in contact normal. This sensing strategy has found several successful applications, ranging from slip detection [49] to surface following [50].



$$\begin{cases} \mathbf{p}_c \times \mathbf{F} + \tau = \mathbf{M} \\ S(x, y, z) = 0 \end{cases} \tag{6}$$

Figure 12.: Intrinsic contact sensing strategy

A Microsoft Kinect™ together with PCL implementation of a point cloud tracker using a Particle Filter [36] was used for tracking.

3.2.2 Pose correction from vision

The first example of the application is analogous to the experiment done in 3.1.1. Figure 13 shows a situation where vision successfully locate an object when it is lying on a table(left hand side), but as soon as the robot hand grasps the object and creates occlusions, the performance of vision decays significantly (right hand side). The pose correction method is then applied, accurately estimating the object's pose. Figure 14 shows results for other objects.

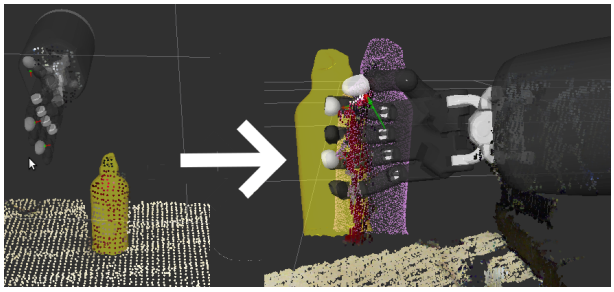


Figure 13.: Pose correction result – Vision based tracking results in yellow before and after occlusions are created by the grasp. The pose corrected using the proposed method is displayed in purple

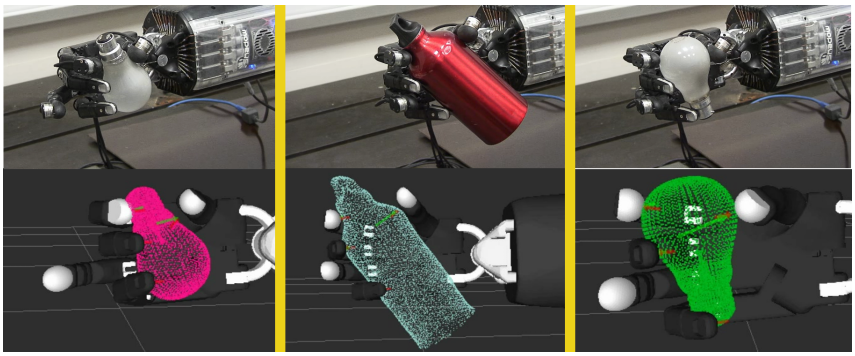
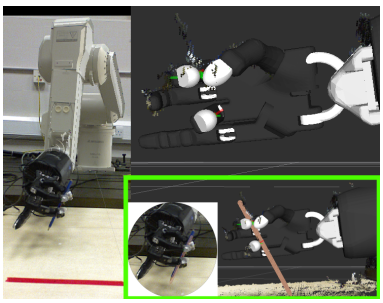


Figure 14.: Pose correction results with different objects

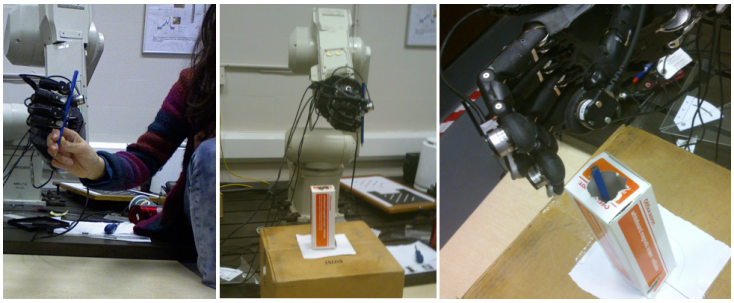
3.2.3 Interacting with humans – Hand over and place

To illustrate a possible application of the proposed method, an experiment was set up, where a robot collaborates with a human, in which the latter hands over an object to the robot, who grasps the object and places it in a designed location.

The example object was a pencil, as it poses difficulties to a vision tracker due to its size. The placing phase also entails some problems, as the pencil needs to be placed in a narrow hole in a box (around 1.5 cm radius), requiring the estimate to be very accurate (under 1.5 cm or 15° errors). Figure 15(a) shows the experimental setup. The point cloud obtained with the RGB-D camera contains very few points belonging to the object, making it impossible to be tracked by vision. The proposed method, however, can successfully estimate the pencil's orientation without any prior estimate of its pose. The result of the experiment is shown in Figure 15(b).



(a) Clockwise from left: Robot grasping a pencil; Point cloud overlaid with robot model; Result of the pose estimation



(b) Hand over and place experiment – a pencil is placed in the robot hand by a human operator, the goal is to place the object inside a box.

Figure 15.: Experiment - An operator hands over a pencil to the robot and the robot places it through a hole in a box. The experiment uses solely tactile and proprioceptive sensing

3.3 *Discussion*

This section presented results for the proposed algorithm in two different settings: correcting the pose with an initial prior (*e.g.* from vision) and estimating it without any information. Simulation results show that the approach is accurate and robust for the levels of noise encountered in the real system (contact location error smaller than 1 mm and 5°). Results also show that using a prior estimate increases the accuracy and allows the correct pose to be detected even in high levels of noise and low number of contacts.

Figure 16 shows the relation between translation and rotation errors during pose estimation (both scales in the same plot) and the green line indicates the boundary where the initial and final errors are equal. The right hand side of the green line indicates cases where the initial error was improved. It should be noted that even in the four cases where the final rotation error is higher than the initial, the translation error on these cases was reduced. It can be also seen from the same plot that there is no significant correlation between initial and final error, which is expected from a global search.

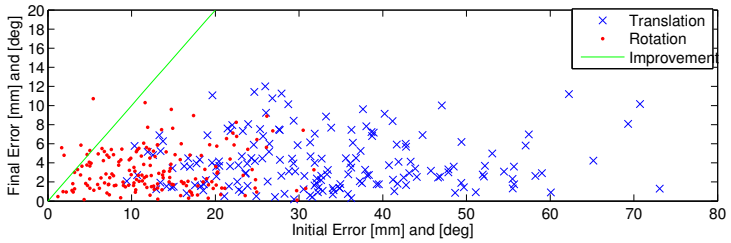


Figure 16.: Initial *vs.* final errors in pose correction. Line $y = x$ to outline the improvement from an initial estimate

The number of tests carried out were 2978 for the pose correction and 1329 for the global pose estimation. The runtime duration for pose correction and estimation was chosen to be around 0.5 seconds and one minute, as to compare it with other methods in the literature which have similar runtime, but could be reduced to around 10 seconds by lowering the confidence criterion in equation (5), allowing for less accurate estimations.

The search can also be tuned to fit particular situations, for example if the object is known *a priori* to be symmetric, the search space can be reduced (in a sphere, for example, the rotation search space can be set to zero). Furthermore, if the object is known to remain static or slowly moving, this algorithm can be run continuously, sending as initial estimate the previous estimation result and using new sensing data, increasing the accuracy of the estimation, particularly with high levels of noise. However, to deal with dynamic situations, where the object is moving

with respect to the hand, an update model should be developed to predict how the object will move, changing particles accordingly at every step.

Particularly in the global pose estimation without an initial prior, the number of fingers touching, as well as the choice of parameters can greatly influence the result of the method. Figure 17 displays an example of a situation where few fingers were touching the object, resulting in a mistaken pose. It can be seen that the bowl surface on the left side of the glass coincides almost perfectly in those two completely different poses. This problem can only be overcome if there are enough fingers contacting the object or if a exploration strategy is put in place.

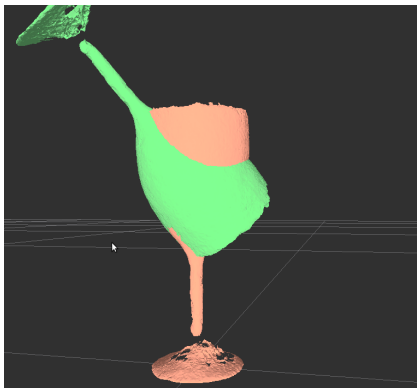


Figure 17.: Example of a situation that might cause the algorithm to fail. Parts of the object surface coincide almost perfectly in two different poses.

The results on this paper present a significant improvement on the authors' previous results in [31, 32] where the errors were over 2 cm for a pose correction from an initial vision estimate to 4 mm and 5° in the current form. These results are also comparable with other literature, which reports results with errors under 1 cm [21] or 2cm [19], 2 mm and 8° [15], 5.2 mm and 3° [17], 5.2 mm and 1.51° [16] but typically require exploration and/or objects with simple geometry.

4. Conclusions

Object grasping and/or manipulation typically relies on vision to estimate the pose of the target object. However, when the robot creates occlusions between the camera and the object the tracking performance decreases significantly. This paper presents a method to estimate this pose using current force, tactile and proprioceptive information, where an evolutionary algorithm is used to find an object's pose which is coherent with this sensor information. The proposed method can be used both to improve an estimate given by vision or globally estimating the pose when no prior estimate is available. The method can also be used to create an initial distribution of candidate poses which can be used in a sequential estimator such as a particle filter.

Validation has shown an error below 1 cm and 15° and a consistent improvement from an initial estimate. An example application was presented where the robot was handed over a pencil and accurately placed it through a narrow hole using no vision input. Both the results in simulation and the successful experiment show the validity of the proposed algorithm and the capabilities of an advanced tactile sensing system, particularly in situations where vision might not be available or accurate.

References

[1] Kragic D, Miller A, Allen P. Realtime tracking meets on line grasping planning. In: IEEE International Conference on Robotics and Automation (ICRA). 2001. p. 2460–2465.

[2] Yilmaz A, Javed O, Shah M. Object tracking: A survey. ACM computing surveys (CSUR). 2006; 38(4).

[3] Lehment NH, Arsi D, Kaiser M, Rigoll G. Automated Pose Estimation in 3D Point Clouds Applying Annealing Particle Filters and Inverse Kinematics on a GPU. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2010.
[4] Cheng CM, Chen HW, Lee TY, Lai SH, Tsai YH. Robust 3D object pose estimation from a single 2D image. IEEE Visual Communications and Image Processing (VCIP). 2011;:1–4.
[5] Buch AG, Kraft D, Kamarainen JK, Petersen HG, Kruger N. Pose estimation using local structure-specific shape and appearance context. In: IEEE International Conference on Robotics and Automation (ICRA). IEEE. 2013. p. 2080–2087.
[6] Azad P, Asfour T, Dillmann R. Stereo-based 6D object localization for grasping with humanoid robot systems. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2007. p. 919–924.
[7] Saxena A, Driemeyer J, Ng AY. Robotic Grasping of Novel Objects using Vision. The International Journal of Robotics Research. 2008;27(2):157–173.
[8] Kragic D, Björkman M, Christensen HI, Eklundh JO. Vision for robotic object manipulation in domestic settings. Robotics and Autonomous Systems. 2005;52(1):85–100.
[9] Rothwell JC, Traub MM, Day BL, Obeso JA, Thomas PK, Marsden CD. Manual Motor Performance In A Deafferented Man. Brain. 1982;.
[10] Son JS, Howe RD, Wang J, Hager GD. Preliminary results on grasping with vision and touch. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 1996. p. 1068–1075.
[11] Allen PK, Miller AT, Oh PY, Leibowitz BS. Integration of Vision , Force and Tactile Sensing for Grasping. International Journal of Intelligent Machines. 1999;4:129–149.
[12] Honda K, Hasegawa T, Kiriki T, Matsuoka T. Real-time Pose Estimation of an Object Manipulated by Multi-fingered Hand Using 3D Stereo Vision and Tactile Sensing. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). October. 1998. p. 1814–1819.
[13] Haidacher S, Hirzinger G. Estimating Finger Contact Location and Object Pose from Contact Measurements in 3-D Grasping. In: IEEE International Conference on Robotics and Automation (ICRA). 2003. p. 1805–1810.
[14] Prats M, Sanz PJ, del Pobil AP. Vision-tactile-force integration and robot physical interaction. In: IEEE International Conference on Robotics and Automation (ICRA). 2009. p. 3975–3980.
[15] Corcoran C, Platt R. A measurement model for tracking hand-object state during dexterous manipulation. In: IEEE International Conference on Robotics and Automation (ICRA). 2010. p. 4302–4308.
[16] Hebert P, Hudson N, Ma J. Fusion of stereo vision, force-torque, and joint sensors for estimation of in-hand object location. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2011. p. 5935–5941.
[17] Petrovskaya A, Khatib O. Global localization of objects via touch. In: IEEE Transactions on Robotics. Vol. 27. 2011. p. 1–17.
[18] Laaksonen J, Nikandrova E, Kyrki V. Probabilistic sensor-based grasping. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2012. p. 2019–2026.
[19] Koval MC, Pollard NS, Srinivasa SS. Pose Estimation for Contact Manipulation with Manifold Particle Filters. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2013. p. 4541–4548.
[20] Zito C, Kopicki MS, Stolkin R, Borst C, Schmidt F, Roa MA, Wyatt JL. Sequential Trajectory Re-planning with Tactile Information Gain for Dexterous Grasping under Object-pose Uncertainty. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2013. p. 4013–4020.
[21] Aggarwal A, Kirchner F. Object Recognition and Localization: The Role of Tactile Sensors. Sensors. 2014;14(2):3227–66.
[22] Chalon M, Reinecke J, Pfanne M. Online in-hand object localization. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2013. p. 2977–2984.
[23] Berenson D, Srinivasa SS, Kuffner JJ. Addressing Pose Uncertainty in Manipulation Planning Using Task Space Regions. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2009. p. 1419–1425.
[24] Hsiao K, Kaelbling L, Lozano-Prez T. Robust grasping under object pose uncertainty. Autonomous Robots. 2011;31(2-3):253–268.
[25] Stulp F, Theodorou E, Buchli J, Schaal S. Learning to grasp under uncertainty. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2011. p. 5703–5708.
[26] Su Y, Wu Y, Lee K, Du Z, Demiris Y. Robust grasping for an under-actuated anthropomorphic hand under object position uncertainty. In: IEEE-RAS International Conference on Humanoid Robots

(Humanoids). 2012. p. 719–725.
[27] Ilonen J, Bohg J, Kyrki V. Fusing visual and tactile sensing for 3-D object reconstruction while grasping. In: IEEE International Conference on Robotics and Automation (ICRA). IEEE. 2013. p. 3547–3554.
[28] Weisz J, Allen PK. Pose error robust grasping from contact wrench space metrics. In: IEEE International Conference on Robotics and Automation (ICRA). 2012. p. 557–562.
[29] Bekiroglu Y, Detry R, Kragic D. Learning tactile characterizations of object- and pose-specific grasps. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2011. p. 1554–1560.
[30] Dang H, Allen PK. Stable grasping under pose uncertainty using tactile feedback. Autonomous Robots. 2013;36(4):309–330.
[31] Bimbo J, Rodriguez-Jimenez S, Liu H, Song X, Burrus N, Senerivatne L, Abderrahim M, Althoefer K. Object pose estimation and tracking by fusing visual and tactile information. In: IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI). 2012. p. 65–70.
[32] Bimbo J, Senerivatne LD, Althoefer K, Liu H. Combining Touch and Vision for the Estimation of an Object's Pose During Manipulation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2013. p. 4021–4026.
[33] Salamin E. Application of quaternions to computation with rotations. Stanford University Artificial Intelligence Laboratory. 1995. Tech Rep.
[34] Funda J, Taylor R, Paul R. On homogeneous transforms, quaternions, and computational efficiency. IEEE Transactions on Robotics and Automation. 1990;6(3):382–388.
[35] Ude A, Jamova R. Nonlinear least squares optimisation of unit quaternion functions for pose estimation from corresponding features. In: International Conference on Pattern Recognition. 1998. p. 425–427.
[36] Rusu RB, Cousins S. 3D is here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation (ICRA). IEEE. 2011. p. 1–4.
[37] Muja M, Lowe D. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. International Conference on Computer Vision Theory and Applications (VISAPP'09). 2009;.
[38] Metropolis N, Ulam S. The Monte Carlo Method. Journal of the American statistical association. 1949;44(247):335–341.
[39] Thrun S, Fox D, Burgard W. Monte carlo localization with mixture proposal distribution. AAAI/IAAI. 2000;.
[40] Thrun S, Fox D, Burgard W, Dellaert F. Robust monte carlo localization for mobile robots. Artificial Intelligence. 2000;128(1-2):99–141.
[41] Thrun S, Burgard W, Fox D. Probabilistic Robotics. MIT Press. 2005.
[42] Kormushev P, Caldwell DG. Simultaneous discovery of multiple alternative optimal policies by reinforcement learning. In: IEEE International Conference on Intelligent Systems (IS). IEEE. 2012. p. 202–207.
[43] Kormushev P, Caldwell DG. Direct policy search reinforcement learning based on particle filtering. In: European Workshop on Reinforcement Learning. 2012.
[44] Marsaglia G. Choosing a Point from the Surface of a Sphere. The Annals of Mathematical Statistics. 1972;43(2):645–646.
[45] Box GEP, Muller ME. A note on the generation of random normal deviates. The Annals of Mathematical Statistics. 1958;29(2):610–611.
[46] Marquardt DW. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. Journal of the Society for Industrial and Applied Mathematics. 1963;11(2):431–441.
[47] Bicchi A, Salisbury JK, Brock DL. Contact Sensing from Force Measurements. The International Journal of Robotics Research. 1993;12(3):249–262.
[48] Liu H, Song X, Bimbo J, Seneviratne L, Althoefer K. Surface material recognition through haptic exploration using an intelligent contact sensing finger. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2012. p. 52–57.
[49] Song X, Liu H, Althoefer K, Nanayakkara T, Seneviratne L. Efficient break-away friction ratio and slip prediction based on haptic surface exploration. IEEE Transactions on Robotics. 2014;30(1):203–219.
[50] Back J, Bimbo J, Noh Y, Seneviratne L, Althoefer K, Liu H. Control a contact sensing finger for surface haptic exploration. In: IEEE International Conference on Robotics and Automation (ICRA). 2014. p. 2736–2741.