DOI:
10.1080/01691864.2014.1002531

Document Version
Peer reviewed version

Citation for published version (APA):
Bimbo, J. M., Kormushev, P., Althoefer, K. A., & Liu, H. (2015). Global estimation of an object's pose using tactile sensing. *Advanced Robotics*, *29*(5), 363-374. https://doi.org/10.1080/01691864.2014.1002531

To appear in *Advanced Robotics*
Vol. 00, No. 00, April 2014, 1–12

FULL PAPER

Global Object Pose Estimation Using Tactile Sensing

Joao Bimbo[a], Petar Kormushev[b], Kaspar Althoefer[a] and Hongbin Liu[a] *

[a]*Centre for Robotics Research, King's College London, London, United Kingdom*;

[b]*Advanced Robotics Department, IIT, Genoa, 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 method where tactile and force sensing together with the robot's proprioceptive data are used to find a suitable object 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. Also, 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: sensing; grasping; global search; tactile; 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 to be combined together with grasping 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]. Object tracking and pose estimation for grasping applications has also been a subject of recent work [6, 7]. These strategies 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. 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*proves that even humans fail to perform accurate manipulation tasks when their tactile sensory system is impaired [8].

Early work that combined vision and force sensing for robot grasping can be traced back to Son and colleagues [9], who investigated the advantages of combining these two sensing modalities and Allen *et al* [10] 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* [11] vision is

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

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 [12].

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 [13]. 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 [14]. 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 [15]. 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 [16]. Koval *et al* [17] 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 [18]. A collision checker combined with a particle filter was also used to estimate the in-hand object pose, starting from a initial pose acquired from vision and estimating the pose according to the hand's movements [19]. 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 [20–26].

The method proposed on 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 [27, 28], by making a global search instead of a gradient-based optimisation. The proposed method is extended to 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. 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 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 x which describe a 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 [29–31].

$$\begin{aligned} \mathbf{x} &= \left[q, \vec{t}\right]^T \\ \mathbf{x} &= [q_w, q_x, q_y, q_z, t_x, t_y, t_z]^T \end{aligned} \tag{1}$$

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 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 objects 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 a sum of two factors: the distance from the contact location on the finger f to a point on the surface s and the angle between the measured surface normal at that point $\hat{n}$ and the measured normal force direction $\hat{u}$. This sum is mediated by a weighting factor w_n that depends on the confidence on the object model, as inaccuracies in the geometric models can yield incorrect normals.

$$G(x) = \sum_{m=1}^{m} \min_i \left(\|(qf^{(m)}q^* + \vec{t}) - s_i\| + w_n(1 - \langle q\hat{u}^{(m)}q^*, \hat{n}_i \rangle \right) \tag{2}$$

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 their likelihood.

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 [32, 33].

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 [34]. 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 [35–37]. 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 the local search, starting 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 uniform pseudo-random distribution. For the rotation quaternion it is accomplished both through the method suggested by Marsaglia [38] and creating a small set of particles which contain binary randoms in their elements. After normalisation, this ensures there will be quaternions in all eight quadrants.

2.2.4 Resampling

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 a local search, 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.

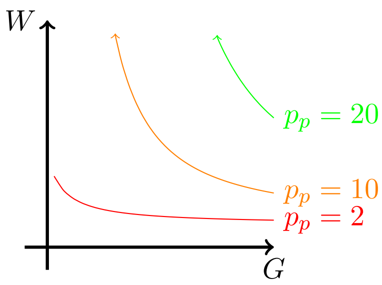


Figure 1.: Cost to Weight Function

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

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 resampled 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 [39], which conveniently creates a pair of normally distributed numbers each time. The way the standard deviation evolves over the particle number k, given a desired total number of iterations n_p is shown in (4). This noise tends to zero as it approaches the end of runtime and the speed at which it decreases is defined by changing the power p_n.

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

2.2.6 Evolution of the algorithm

The algorithm runs for a fixed 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.

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.

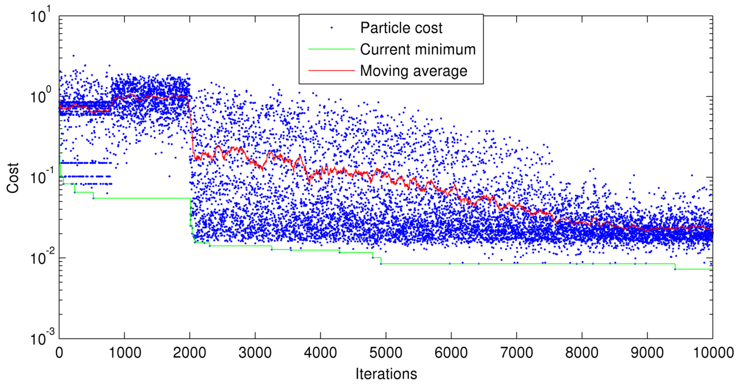


Figure 2.: Progress of algorithm – cost over iterations

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 vicinity of a number of points inside the robot (knuckles, palm, etc.). If a possible pose violates this condition it is discarded.

Finally, a Levenberg-Marquardt gradient search [40] is performed, further improving the estimate. The details of this step were previously shown by the authors [28]. 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 local search 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 resampled 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 the end of the array than adding from the beginning. This allows for a much faster resampling while maintaining the conditions for Importance Sampling.

Figure 3.: Weights of particles over time

Figure 4 shows the computation time to generate each particle. Typically, the generation of each particle would require increasing time with the number of previous particle it resamples 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.

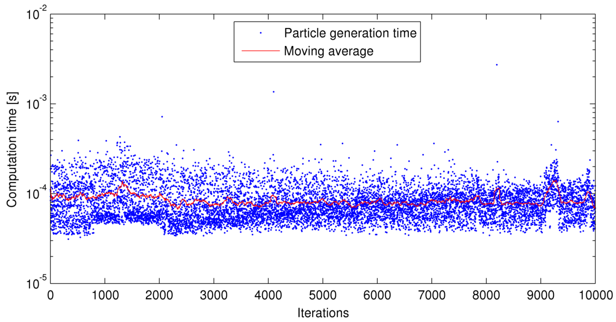


Figure 4.: Performance – Computation time to generate each particle

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. To make the figures clearer, the results are shown only in terms of the distance between the estimate and the ground truth and their angle in the vertical axis. 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. The algorithm ran at a frequency of over 2 Hz.

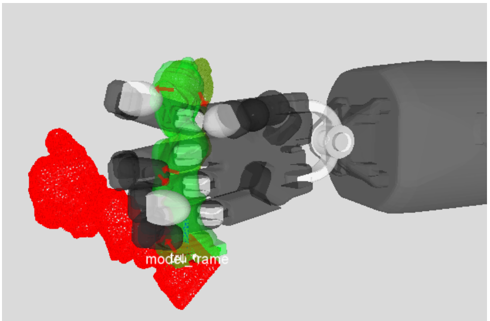


Figure 5.: Pose correction. Initial estimate in red, ground truth in green and result pose in olive green, force normals are displayed as red arrows

Figure 6 shows the results for twenty consecutive executions of the proposed algorithm. The blue dots represent the initial error in position and orientation before the execution of the algorithm and the red dots the error after pose correction. The mean absolute error in translation was reduced from 15.6 mm to 7.6 mm. As for orientation, the angular error in the z-axis was reduced from 37.7° to 5.14°. Besides, it can be seen that the error is reduced in every trial in both position and orientation.

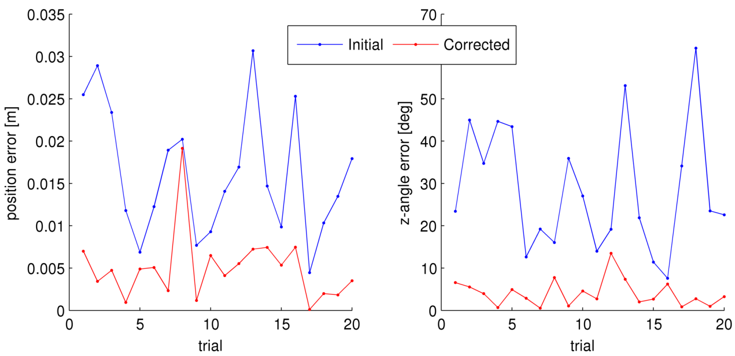


Figure 6.: Results for twenty executions of the pose correction algorithm with a reduced search space

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

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

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 7 shows a result of a trial where the object is put at an arbitrary location and the resulting estimated pose overlays the ground truth. The grasp performed on the object uses the little finger to touch the glass stem, allowing the estimation to detect the correct orientation, as such symmetric objects could yield "upside down" poses if the fingers were only touching the glass bowl. Figure 8 plots the results of the pose estimation. The mean absolute error was 9.9 mm for position and 10.6° for the vertical angle.

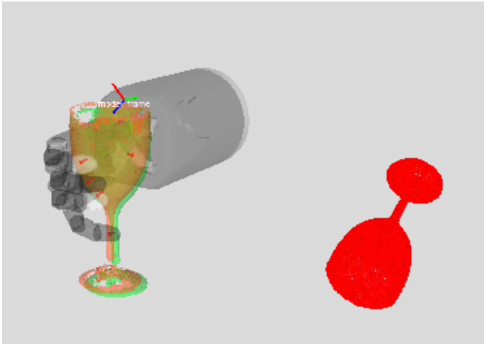


Figure 7.: Global pose estimation. Initial estimate in red, ground truth in green and result pose in orange, force normals are displayed as red arrows

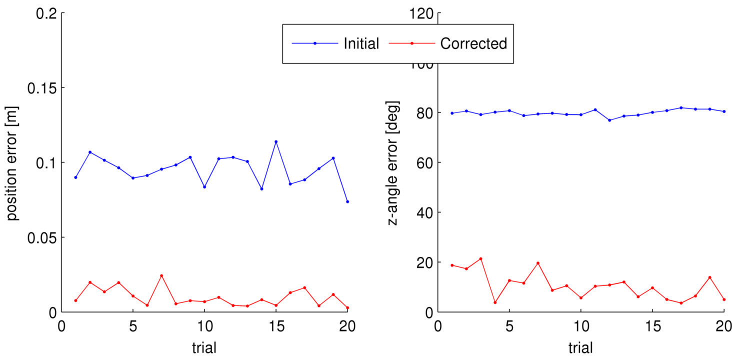


Figure 8.: Results for twenty executions of the global pose estimation

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™[1] 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* [41]. Equation 5 and Figure 9 illustrate this scheme where, using a 6 axis force-torque sensing under a parametrisable convex

[1] http://http://www.shadowrobot.com/products/dexterous-hand

shape – a semi-ellipsoid in this case – one can solve the system of equations consisting of the force and moment balance, yielding a unique solution for the contact location p_c. 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* [42] showing an accuracy of 0.224 mm.

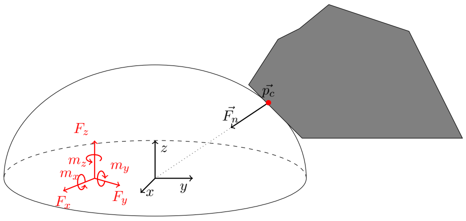


Figure 9.: Sensor Information

$$\begin{cases} p_c \times \vec{F} + \vec{m} = \vec{M} \\ S(x, y, z) = 0 \end{cases} \tag{5}$$

A Microsoft Kinect™[1] together with PCL[2] implementation of a point cloud tracker using a Particle Filter [32] 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. When the object is lying on the table, vision can successfully track its pose, but as soon as the robot hand grasps the object and creates occlusions, the performance of vision decays significantly. The pose correction method is then applied, accurately estimating the object's pose.

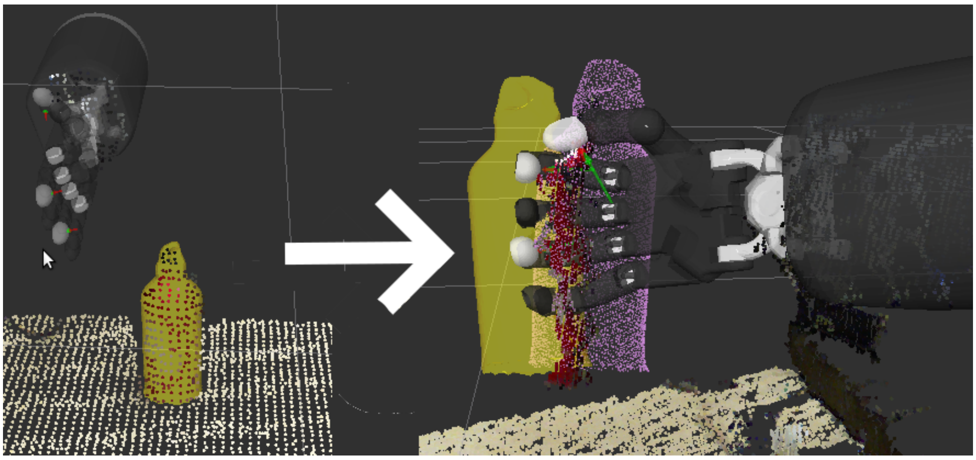


Figure 10.: Pose correction result – Image tracking performance before and after occlusions are created by the grasp in yellow. The pose corrected using the proposed method is displayed in pink

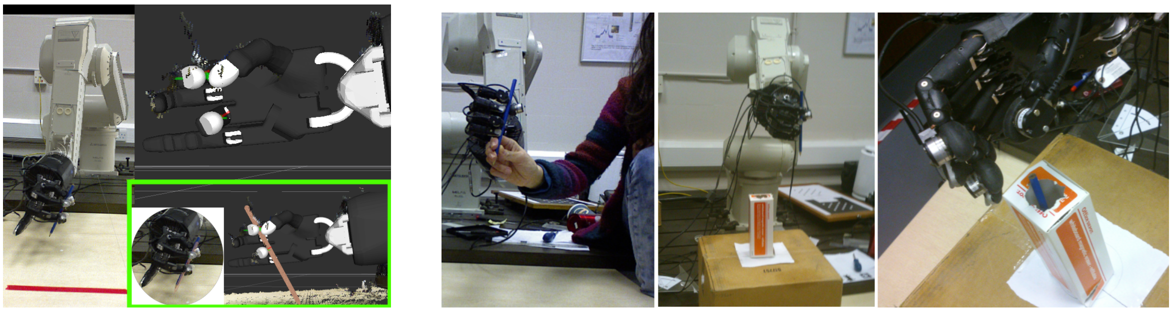
3.2.3 Interacting with humans – Hand over and place

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, requiring the estimate to be very accurate. Figure 11(a) shows a situation where the robot is grasping a pencil. The point cloud obtained with the RGB-D camera contains very few points

[1] http://www.xbox.com/en-GB/Kinect
[2] Point Cloud Library – http://www.pointclouds.org

belonging to the object, making it impossible to be tracked by vision. The 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 11(b).



(a) Clockwise from left: Robot grasping the pencil; Point cloud overlaid with the 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 11.: 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

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. Furthermore, in some situations it is not feasible to use vision. 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. Validation has shown an error below 1 cm on the global search 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] D. Kragic, A. Miller, and P. Allen, "Realtime tracking meets on line grasping planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2001, pp. 2460–2465.

[2] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm Computing Surveys (CSUR)*, vol. 38, no. 4, 2006.

[3] N. H. Lehment, D. Arsi, M. Kaiser, and G. Rigoll, "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] C.-M. Cheng, H.-W. Chen, T.-Y. Lee, S.-H. Lai, and Y.-H. Tsai, "Robust 3D object pose estimation from a single 2D image," *Visual Communications and Image Processing (VCIP)*, pp. 1–4, 2011.

[5] A. G. Buch, D. Kraft, J.-k. Kamarainen, H. G. Petersen, and N. Kr, "Pose Estimation using Local Structure-Specific Shape and Appearance Context."

[6] P. Azad, T. Asfour, and R. Dillmann, "Stereo-based 6d object localization for grasping with humanoid robot systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2007, pp. 919–924.

[7] a. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic Grasping of Novel Objects using Vision," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, Feb. 2008.

[8] J. C. Rothwell, M. M. Traub, B. L. Day, J. A. Obeso, P. K. Thomas, and C. D. Marsden, "Manual Motor Performance In A Deafferented Man," *Brain*, 1982.

[9] J. S. Son, R. D. Howe, J. Wang, and G. D. Hager, "Preliminary results on grasping with vision and touch," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1996, pp. 1068–1075.

[10] P. K. Allen, A. T. Miller, P. Y. Oh, and B. S. Leibowitz, "Integration of Vision , Force and Tactile Sensing for Grasping," *International Journal of Intelligent Machines*, vol. 4, pp. 129–149, 1999.

[11] K. Honda, T. Hasegawa, T. Kiriki, and T. Matsuoka, "Real-time Pose Estimation of an Object Manipulated by Multi-fingered Hand Using 3D Stereo Vision and Tactile Sensing," in *Proceedings of the 1998 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, no. October, 1998, pp. 1814–1819.

[12] S. Haidacher and G. Hirzinger, "Estimating Finger Contact Location and Object Pose from Contact Measurements in 3-D Grasping Institute for Robotics and Mechatronics," pp. 1805–1810, 2003.

[13] M. Prats, P. J. Sanz, and A. P. del Pobil, "Vision-tactile-force integration and robot physical interaction," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2009, pp. 3975–3980.

[14] C. Corcoran and R. Platt, "A measurement model for tracking hand-object state during dexterous manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[15] P. Hebert, N. Hudson, and J. Ma, "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, pp. 5935–5941.

[16] A. Petrovskaya and O. Khatib, "Global localization of objects via touch," in *IEEE Transactions on Robotics*, vol. 27, 2011, pp. 1–17.

[17] M. C. Koval, N. S. Pollard, and S. S. Srinivasa, "Pose Estimation for Contact Manipulation with Manifold Particle Filters," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, 2013, pp. 4541–4548.

[18] C. Zito, M. S. Kopicki, R. Stolkin, C. Borst, F. Schmidt, M. A. Roa, and J. L. Wyatt, "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)*, Tokyo, 2013, pp. 4013–4020.

[19] M. Chalon, J. Reinecke, and M. Pfanne, "Online in-hand object localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, 2013, pp. 2977–2984.

[20] D. Berenson, S. S. Srinivasa, and J. J. Kuffner, "Addressing Pose Uncertainty in Manipulation Planning Using Task Space Regions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, 2009, pp. 1419–1425.

[21] K. Hsiao, L. P. Kaelbling, and T. Lozano-Pérez, "Robust grasping under object pose uncertainty," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 31, no. 2-3, July 2011, pp. 253–268.

[22] F. Stulp, E. Theodorou, J. Buchli, and S. Schaal, "Learning to grasp under uncertainty," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Ieee, May 2011, pp. 5703–5708.

[23] Y. Su, Y. Wu, K. Lee, Z. Du, and Y. Demiris, "Robust grasping for an under-actuated anthropomorphic hand under object position uncertainty," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Ieee, Nov. 2012, pp. 719–725.

[24] J. Ilonen, J. Bohg, and V. Kyrki, "Fusing visual and tactile sensing for 3-D object reconstruction while grasping," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3547–3554, May 2013.

[25] J. Weisz and P. K. Allen, "Pose error robust grasping from contact wrench space metrics," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 557–562.

[26] Y. Bekiroglu, R. Detry, and D. Kragic, "Learning tactile characterizations of object- and pose-specific grasps," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Sept. 2011, pp. 1554–1560.

[27] J. Bimbo, S. Rodriguez-Jimenez, H. Liu, X. Song, N. Burrus, L. Senerivatne, M. Abderrahim, and K. Althoefer, "Object pose estimation and tracking by fusing visual and tactile information," in *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2012)*, Sept 2012, pp. 65–70.

[28] J. Bimbo, L. D. Seneviratne, K. Althoefer, and H. Liu, "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)*, Tokyo, 2013, pp. 4021–4026.

[29] E. Salamin, "Application of quaternions to computation with rotations," Stanford University Artificial Intelligence Laboratory, Tech. Rep., 1995.

[30] J. Funda, R. Taylor, and R. Paul, "On homogeneous transforms, quaternions, and computational efficiency," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 3, pp. 382–388, June 1990.

[31] A. Ude and R. Jamova, "Nonlinear least squares optimisation of unit quaternion functions for pose estimation from corresponding features," in *Proc. of Int. Conf. Pattern Recognition*, no. August, 1998, pp. 425–427.

[32] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*. Ieee, May 2011, pp. 1–4.

[33] M. Muja and D. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration." *International Conference on Computer Vision Theory and Applications (VISAPP'09)*, 2009.

[34] N. Metropolis and S. Ulam, "The Monte Carlo Method," *Journal of the American statistical association*, vol. 44, no. 247, pp. 335–341, 1949.

[35] S. Thrun and M. C. Localization, "Monte Carlo Localization With Mixture Proposal Distribution," 2000.

[36] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Robust Monte Carlo localization for mobile robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 128, no. 1-2, May 2001, pp. 99–141.

[37] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[38] G. Marsaglia, "Choosing a Point from the Surface of a Sphere," *The Annals of Mathematical Statistics*, vol. 43, no. 2, pp. 645–646, 1972.

[39] G. E. P. Box and M. E. Muller, "A note on the generation of random normal deviates," *The Annals of Mathematical Statistics*, vol. 29, no. 2, pp. 610–611, 06 1958.

[40] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, June 1963.

[41] A. Bicchi, J. K. Salisbury, and D. L. Brock, "Contact Sensing from Force Measurements," *The International Journal of Robotics Research*, vol. 12, no. 3, pp. 249–262, June 1993.

[42] H. Liu, X. Song, J. Bimbo, L. Senerivatne, and K. Althoefer, "Object Surface Material Recognition through Haptic Exploration using a Intelligent Contact Sensing Finger," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012)*, 2012.