

Combining Touch and Vision for the Estimation of an Object's Pose During Manipulation

Joao Bimbo, Lakmal D. Seneviratne, Kaspar Althoefer and Hongbin Liu *

Abstract—Robot grasping and manipulation relies mainly on two types of sensory data: vision and tactile sensing. Localisation and recognition of the object is typically done through vision alone, while tactile sensors are commonly used for grasp control. Vision performs reliably in uncluttered environments, but its performance may deteriorate when the object is occluded, which is often the case during a manipulation task, when the object is in-hand and the robot fingers stand between the camera and the object.

This paper presents a method to use the robot's sense of touch to refine the knowledge of a manipulated object's pose from an initial estimate provided by vision. The objective is to find a transformation on the object's location that is coherent with the current proprioceptive and tactile sensory data. The method was tested with different object geometries and proposes applications where this method can be used to improve the overall performance of a robotic system. Experimental results show an improvement of around 70% on the estimate of the object's location when compared to using only vision.

I. INTRODUCTION

Robot grasping and particularly the fine manipulation of an object by a robot requires very accurate sensing of the object's pose and of the acting forces. In fact, even for a human, tactile sensing is fundamental for performing tasks that require a great deal of accuracy. This was illustrated in a study by Rothwell [1] in which a man with impaired tactile sensing who, despite not having any motor problem and being able to perform most tasks using only vision to control his movements, failed to perform more accurate manipulation tasks such as the fastening of a button or using a pen to write. Similarly, a robot equipped with a camera can perform simple grasps but, to carry out fine manipulation tasks, must have very accurate tactile sensors, that not only sense the contact location but also the direction of the interaction forces. When using only vision to track an object's position and orientation, the obtained information might be inaccurate due to limitations of the hardware or bad calibration and might fail to track the object when it is occluded. An example of this situation is shown in Fig. 1, where the camera's point of view is shown along with a visualiser displaying the robot's posture and the object's pose as acquired from the vision system. Comparing the two images, it can be seen that the object is not represented in its correct pose with respect to the robot (it is intersecting the robot's middle and ring finger and the thumb does not

seem to be touching the object surface). These inaccuracies can hinder the success of a grasp or manipulation task.

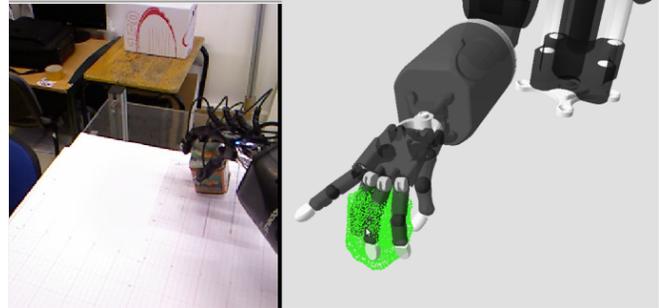


Fig. 1. Camera's point of view, and visualisation of the robot's current posture with a vision-acquired object pose

The fusion of sensor information from different sources has caught the attention of roboticists as the need for a robot to work in unstructured environments became an essential part of research in the field. The uncertainty associated with this context requires the usage of different sensor information which, ideally, can be complementary, providing redundancy and increasing a system's robustness [2]. In the field of robot grasping, the combination of vision and tactile sensing can increase the accuracy of the information on the object's pose, and also provide an estimate in cases where vision alone fails. Among the most relevant publications that use this approach are the works by Allen et al [3], who combined vision, force and tactile sensing to portray how these different sensing modalities could complement each other. Kragic [4] integrates vision with a grasp planner, using only vision to track the object. Petrovskaya [5] explored a method to localise an object using Scaling Series. Prats et al [6] combined tactile, force and vision information to locate and open a door handle and compared the performance under different sensing settings (only force, force and vision and tactile, force and vision), where the combination of the three modalities proved to outperform the other two, which is analogue to the previously mentioned results of Rothwell et al [1]. Estimation of an object's pose combining stereo vision and a force-torque sensor mounted on the wrist of a robot was reported by Hebert et al [7], who also used the joint position to estimate the location of the fingers with respect to the object's faces. Honda et al [8] used a combination of tactile and vision sensing to estimate an object's pose, assuming that the object is composed of plain and quadratic surfaces. This paper extends the authors' previous work in [9], where only the distance between fingers and object was considered

The authors are with the Centre for Robotics Research, King's College London, WC2R 2LS London, United Kingdom, L. D. Seneviratne is also with the College of Engineering, Khalifa University, Abu Dhabi, U.A.E. {joao.bimbo, k.althoefer, hongbin.liu}@kcl.ac.uk, lakmal.seneviratne@kustar.ac.ae

* Corresponding author

and no ground truth information was available and presents a method to track an object using the object geometry acquired from vision and tactile sensors on the hand which are used to complement the shortcomings of a vision system by either refining the pose given by the camera or estimate it when the camera is unable to do so. Given a grasped object's geometry and an initial estimate of its position and orientation, the proposed method finds a new estimate of a pose that satisfies the current contact information. The object's geometry and an initial estimate of the location was acquired using the 3-D vision-based reconstruction and tracking described in [10], [11] and the tactile information, contact location and normal and tangential force components was acquired using the sensors presented by Liu *et al* [12]. The objective is then to find a transformation (rotation and translation) that minimises the distance from the current contact location on the fingertips to a point on the surface of the object and also minimises the angle between the measured normal force and the vector perpendicular to the surface on that same point (contact normal). This improved information on the object's location can be directly fed into a manipulation planner or controller or, if similar corrections are obtained consistently, be used to improve the system calibration.

Both simulated and experimental data have been used to assess the algorithm, with different optimisation algorithms being tested and compared. The next section outlines of the proposed method, where the problem is presented, the algorithm is described and the simulation results compared. Section III details the experimental setup and the obtained results, while conclusions and possible future enhancements are described in section IV.

II. METHOD AND SIMULATION RESULTS

A. Problem Presentation

The problem of finding an object's pose according to the object's geometry and the contact location can be presented as finding a set of parameters that define a transformation (a rotation and a translation) from our current estimate of where the object is to one that is coherent to the current sensor measurements. This method is somewhat similar to Iterative Closest Point (ICP), a well known method used in computer vision to compute a transform between two sets of point cloud data [13] that minimises the distance between pairs of points. In this paper, instead, a matching is proposed that also takes into account the contact normal information provided by the tactile sensors and the object surface normal estimated through its geometry. To increase the performance of the algorithm, the desired transformation that is to be calculated is not on the object but on the contact locations, therefore avoiding having to transform all the points in the object in every iteration. Instead, the contact locations are transformed to coincide with the object, and this transform is then inverted and applied to the object. The quaternion representation was chosen to describe the rotations for its advantages in terms of computational efficiency and suitability for optimisation methods. Using quaternions to describe rotations not only requires less calculations when compared to rotating using,

for example, rotation matrices, but also the calculations involved are much easier for a computer to deal with, as there are no trigonometrical operations such as *sine* or *cosine*, improving the overall computation speed, besides allowing for smoother interpolation when compared to other rotation conventions [14], [15], [16]. The parameters which define a transformation (a rotation q and a translation \vec{t}) that are to be found are shown in (1):

$$\mathbf{x} = [q, \vec{t}]^T \quad (1)$$

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

B. Description of the Algorithm

In order to decrease the computational time of the algorithm, the first step is to find regions on the object, inside a defined neighbourhood, where the finger is expected to be in contact, avoiding iterating through all the points of the object's surface. This neighbourhood is set dynamically so that a minimum number of points in the object are selected. Equation (2) defines the sets $S^{(m)}$ for each finger that contains the points $s_1^{(m)}, s_2^{(m)} \dots s_n^{(m)}$ belonging to the object O , whose distances to each finger $f^{(m)}$ lie inside a neighbourhood ε . The result of this computation is shown in Fig. 2, with simulated and real data. The initial contact locations on the fingers are represented by the cross symbols and the coloured regions are each finger's respective region where the cost function will be evaluated.

$$S^{(m)} = \{s_i^{(m)} \in O : \|s_i^{(m)} - f^{(m)}\| \leq \varepsilon\} \quad (2)$$

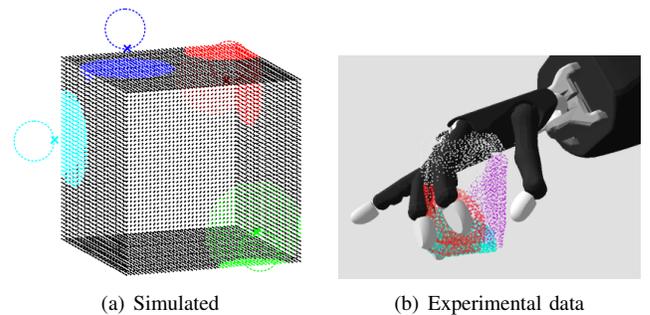


Fig. 2. Regions created in the object point cloud to improve the speed of the algorithm. Each colour represents points in the object in the neighbourhood of a finger in contact.

The algorithm then tries to find the parameters \mathbf{x} , described in (1) that yield a local minimum for a cost function $G(\mathbf{x})$, which takes into account not only the contact locations but also the direction of the normal component of the contact force. As the contacts are assumed to be rigid, this normal component of the force should coincide with the object surface normal direction. The desired cost function should then allow us to find a transform that, not only minimises the distance from the contact locations to the object surface, but also minimises the angle between the measured normal on the finger tip sensor and the surface normal calculated from its geometry. By taking into account the contact normals, one can also know which face of the object the robot is touching.

As an example, simply minimising the distance can give a wrong result if the contact is close to an edge or a vertex of the object. This error can put the success of a manipulation task at risk if the finger is assumed to be touching the wrong surface.

The objective function to minimise (3) was defined as the distance from the contact location to the closest point $s_i^{(m)}$ in its respective region S^m plus a weighting coefficient times a value that increases with the angle between measured normal direction of the contact force $\hat{u}^{(m)}$ and the surface normal \hat{n}_i at point $s_i^{(m)}$. The symbol $|a|$ denotes the absolute value, and $\langle \hat{u}, \hat{n} \rangle$ is the inner product. w_n is a weight attributed to the information of the normals, which is tuned according the requirements of the real system and how accurate we know the object model to be. By giving a large value to w_n , the algorithm will try to adjust the orientation of the object to fit the normals more than it will try to minimise the distance.

$$\mathbf{G}(\mathbf{x}) = \begin{cases} \min(\|(qf^{(1)}q^* + \vec{t}) - s_i^{(1)}\| + w_n|1 - \langle q\hat{u}^{(1)}q^*, \hat{n}_i \rangle|) \\ \min(\|(qf^{(2)}q^* + \vec{t}) - s_i^{(2)}\| + w_n|1 - \langle q\hat{u}^{(2)}q^*, \hat{n}_i \rangle|) \\ \dots \\ \min(\|(qf^{(m)}q^* + \vec{t}) - s_i^{(m)}\| + w_n|1 - \langle q\hat{u}^{(m)}q^*, \hat{n}_i \rangle|) \end{cases} \quad (3)$$

A summary of the algorithm in pseudo code is detailed below, in Algorithm 1.

Algorithm 1 Pose correction

Require: Object point cloud and number of fingers touching the object ≥ 2 .

for all fingers in hand **do**

if finger is in contact **then**

 Transform contact point ($f^{(m)}$) and contact normal $\hat{u}^{(m)}$ to palm coordinate frame

end if

end for

for all $f^{(m)}$ **do**

for all points p_i in object **do**

if $\|p_i - f^{(m)}\| \leq \varepsilon$ **then**

$s_j^{(m)} = p_i$

$j \leftarrow j + 1$

end if

end for

end for

Minimise: $\mathbf{G}(x) =$

$\min(\|(qf^{(m)}q^* + \vec{t}) - s_i^{(m)}\| + w_n|1 - \langle q\hat{u}^{(m)}q^*, \hat{n}_i \rangle|)$

if minimisation is successful **then**

 Invert transformation defined by x_i

 Apply transformation to object

end if

C. Optimisation Method

Fitting a small number of points to its surface often yields more than one solution, specially in cases where the object presents some kind of symmetry. As such, gradient-based iterative methods such as Gradient Descent or Levenberg-Marquardt present the advantage of finding a local minimum

close to the initial estimate. This local minimum is intuitively more likely to be a better solution than a global minimum that, despite obtaining a smaller residual, yield a pose that displaces the object further from the initial estimate, less likely to be the real pose of the object (*e.g.* upside down). Nonetheless, using these methods also presents some difficulties, as the objective function in (3) is not differentiable because we are testing a possible contact location against multiple points in the surface of the object. At each iteration, the point where we estimate the finger to be touching may be different, making the objective function discontinuous. To deal with this issue, the Finite Difference Jacobian is calculated at each iteration [17], using the forward difference approximation in (4).

$$\tilde{\mathbf{J}}_{\mathbf{G}} = (\nabla_h \mathbf{G})(\mathbf{x})_j = \frac{\mathbf{G}(\mathbf{x} + h\|\mathbf{x}\|\mathbf{e}_j) - \mathbf{G}(\mathbf{x})}{h\|\mathbf{x}\|} \quad (4)$$

The first iterative method that was used to find a satisfiable transformation was gradient descent. This method uses the update rule given in (5). The step size λ was tuned empirically, as it will make the convergence either faster or more accurate. The chosen value for λ was 2×10^{-3} . This method takes steps in the direction of the negative of the gradient to find a local minimum.

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \lambda(\tilde{\mathbf{J}}_{\mathbf{G}}(\mathbf{x}_i)^T \mathbf{G}(\mathbf{x}_i)) \quad (5)$$

The other tested iterative method was the Levenberg-Marquardt algorithm [18], which is a combination of Gauss-Newton and the Gradient Descent methods. The update rule is shown in (6). Given that the approximate of the Hessian matrix $\tilde{\mathbf{J}}^T \tilde{\mathbf{J}}$ may contain rows with only zeros, $(\tilde{\mathbf{J}}^T \tilde{\mathbf{J}} + \lambda \text{diag}[\tilde{\mathbf{J}}^T \tilde{\mathbf{J}}])$ can become singular and thus it is not always invertible. The workaround was to use the Moore-Penrose pseudo inverse [19].

$$\mathbf{x}_{i+1} = \mathbf{x}_i - (\tilde{\mathbf{J}}^T \tilde{\mathbf{J}} + \lambda \text{diag}[\tilde{\mathbf{J}}^T \tilde{\mathbf{J}}])^\dagger \tilde{\mathbf{J}}^T \mathbf{G}(\mathbf{x}_i) \quad (6)$$

D. Simulation Results

Tests were carried out first in a MATLAB simulation, to assess and compare the performance of each particular approach. Four points belonging to different faces of the object's surface were selected. These points were then arbitrarily transformed (with both rotation and translation) and the algorithm was run. Table I summarises the obtained results for each iterative method for similar accuracies. It was also investigated whether the addition of the normal information can improve the overall accuracy of the method.

MDTS stands for the mean distance between the fingers and the surface, while **RME** stands for real mean error, which is the mean distance from the optimisation result to the previously selected initial point. This difference is fundamental for the understanding of this problem as, in a real system, we do not know which point on the object the robot is touching, so the algorithm will find a transform that minimises the cost function (3) in one of the points in the region defined in equation (2). In the case of simulated

TABLE I
COMPARISON OF OPTIMISATION METHODS

Method	Shape	Its.	MDTS	RME	MAE	Speed (s)
Gradient Descent	Cube	200	0.197	0.799	9.53°	16.21
	Cylinder	154	0.111	0.265	4.45°	15.97
Levenberg-Marquardt	Cube	37	0.113	1.013	10.63°	4.99
	Cylinder	25	0.096	0.420	5.67°	5.50
GD with normals	Cube	200	0.140	0.524	6.15°	17.46
	Cylinder	136	0.065	0.088	0.97°	14.12
LM with normals	Cube	37	0.033	0.629	4.81°	6.33
	Cylinder	25	0.087	0.324	4.03°	5.63

data, the points were selected *a priori*, so it is known where the algorithm should converge. **MAE** stands for mean angle error, which is the average angle between the ground truth and the result. The comparison between these different approaches shows that the Levenberg-Marquardt method is usually faster to converge, requiring fewer iterations to obtain a similar accuracy. Also, it is clear that the addition of the normals not only improves the convergence of the algorithm itself but also yields results which are closer to the ground-truth, without any significant increase in computation time.

Given that we want to find a transform in the object that matches the contact locations given by the kinematics of the robot and the tactile sensors, the resulting transform needs to be inverted and applied to the object. The result is plotted in Fig. 3, where the grey point cloud shows the initial pose of the object, the green point cloud represents the ground truth and the yellow point cloud shows the transformed pose using the inverse of the resulting parameters from the optimisation. It can be seen that the output object coincides almost perfectly with the ground truth. These results use the Levenberg-Marquardt method with information on the normals.

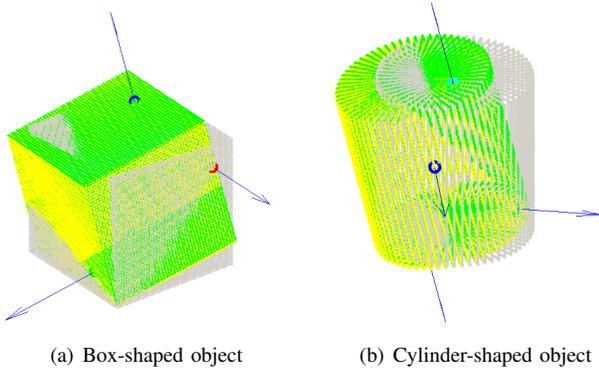


Fig. 3. Simulation results – green represents the ground truth, gray the initial misaligned pose and yellow the resulting object pose.

III. EXPERIMENTS AND RESULTS

A. System Overview

The proposed method was implemented in a real system, consisting of a Shadow Arm and hand robot with a Microsoft Kinect RGB-D camera mounted on the shoulder and equipped with custom designed fingertips with 6-axis ATI

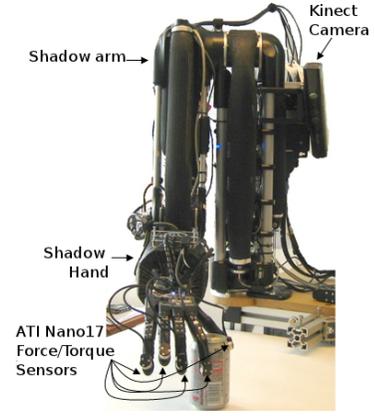
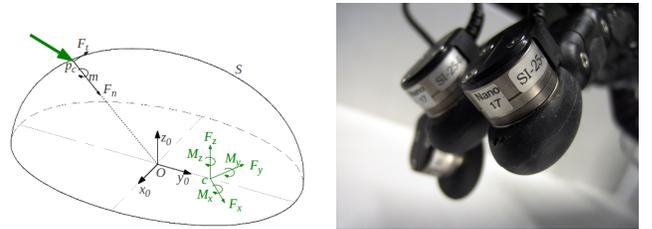


Fig. 4. Overview of the experimental setup

nano17 force and torque sensors. This tactile sensing technique was presented by Liu *et al* [12] where, by measuring the values of force and torque and using a parametrisable convex surface, one can calculate the contact location and the local torque (torque around the normal direction) which then allows to calculate the normal and tangential components of the interaction force.

The method consists of finding the point p_c in Fig. 5(a) that satisfies the equations of forces and moments for each dimension, assuming the only possible torque m is normal to the surface [20]. Equation (7) describes the force and moment balance for a single contact and the equation for the ellipsoidal surface S . The green coordinate frame is the sensor frame and the black is the ellipsoid frame. After the calculation of the contact location p_c , the acting force can be decomposed in its normal and tangential components. This method was evaluated in [21] and shown to have an accuracy on the calculation of the contact location of 266 μm and can run at frequencies of more than 800 Hz.

$$\begin{cases} p_c \times \vec{F} + \vec{m} = \vec{M} \\ S(x, y, z) = 0 \end{cases} \quad (7)$$



(a) Resulting forces and torques on an ellipsoid when a force is acting on point p_c (b) Detail of force and torque sensor mounted on the fingertip

Fig. 5. Intrinsic contact sensing fingertip design

The proposed algorithm was implemented in C++ using the ROS platform and can be triggered in two different ways: the initial estimate of the object pose can be manually given by the user, and the object model is downloaded from a database or it can be acquired directly from the vision system

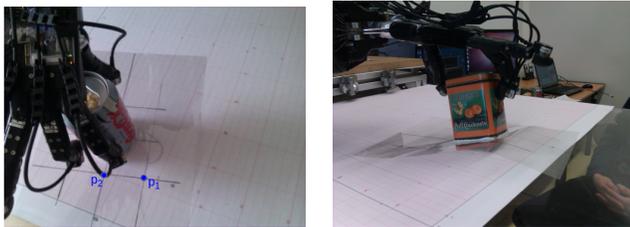
using the object recognition and tracking algorithm presented in [10]. The results on this paper use the latter approach in order to assess the advantages of this algorithm over a vision-only tracking system. Also, the object model is known to be imperfect, which should present further difficulties for convergence when compared to the simulated results, where the models were perfect cubes and cylinders.

B. Results

In order to validate the method, two objects were tested: one standard soda can and a cuboidal metal tea box. These two objects were chosen due to their different properties for the problem at hand. The cylindrical soda can presents rotational symmetry, while the tea box has very distinct faces and normals.

Due to the difficulty of having a continuous ground truth, the validation method consisted of glueing the object to a transparency with four squares with known dimensions, as shown in Fig. 6(a). The object's centre was coincident with the intersection of the squares drawn on the transparency and the table on which the robot manipulated the object was covered with millimeter paper, as shown in Figs. 6(a) and 6(b). By taking note of two corners of the squares, the location of the object could be determined using (8) for corner points p_1 and p_2 and object centre p_0 .

$$\begin{aligned} \vec{v} &= p_1 - p_2; \\ p_0 &= p_2 + \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \vec{v} \end{aligned} \quad (8)$$

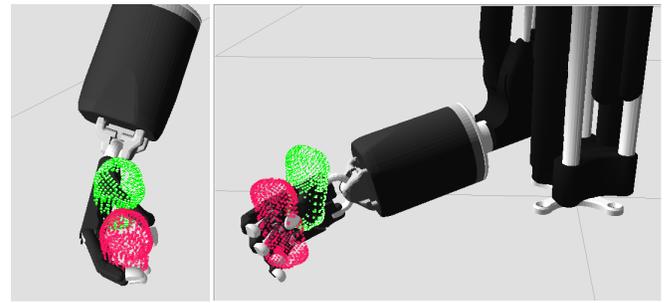


(a) Soda can glued to marked transparency (b) Box-shaped object (Tea box) being grasped

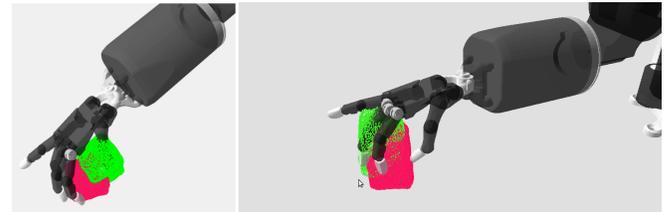
Fig. 6. Ground truth measurement method

The point clouds depicting the vision-acquired pose and the pose corrected with the proposed method are shown in Fig. 7(a) and 7(b), in green and pink respectively, along with a robot model showing the robot's current posture. It can be seen that the estimation of the object's location is improved, as it is located coherently within the robot hand. Fig. 8(b) and 8(a) show the x and y position of the centre of the object's base according to vision (as rings), corrected (as a line) and the ground truth at the recorded points (as dots). The resulting point cloud is expressed in the palm coordinates and stays fixed with the palm frame between corrections, as it is assumed that the grasp is stable and as such there is no relative movement between the hand and the object.

The algorithm's running time was 0.171 seconds on average, with an average number of iterations of 91.2. The mean



(a) Large diameter grasp on a soda can



(b) Tripod grasp on a cuboidal tea box

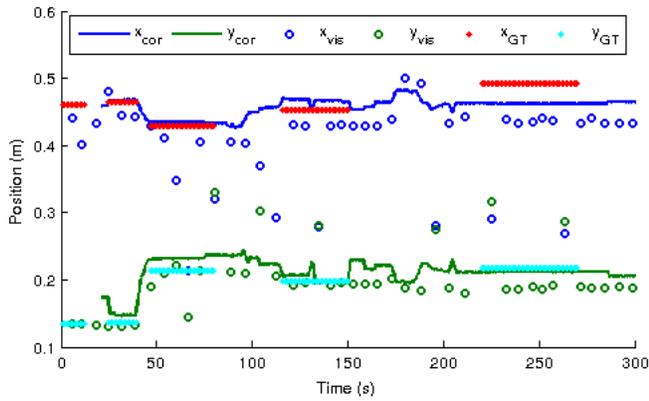
Fig. 7. Visualisation of a grasped object scene. The green point cloud represents the object in the pose detected by the vision system and the pink point cloud represents the object after its pose has been corrected using our approach

error of the vision estimates (taken only when the ground truth was known) was 8.58 cm for the cylindrical object and 8.02 cm for the cuboid, while after the correction it was reduced to 2.66 cm and 2.0 cm respectively.

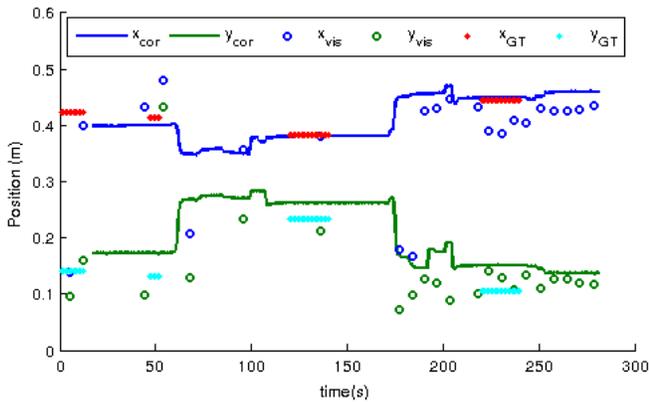
It is important to note at this point that the fact that the result of the optimisation algorithm might fall below distances of 1 mm between the fingers and the object, does not guarantee that the calculated pose is exactly the ground-truth correspondent. This is due to the reduced number of points to be fitted and the symmetry present in most hand-held objects, which typically allows myriad solutions. As an example, a cylindrical object will have countless solutions with orientations around its axis of revolution. While this may not be vital to the success of a manipulation task, further investigation should be done to narrow down the number of solutions. Some strategies are proposed in the end of the next chapter to address this problem.

IV. CONCLUSIONS AND FUTURE WORK

This paper presented a method to improve the estimation of an object's pose while it is being manipulated. While vision systems perform robustly in an uncluttered environment, during a manipulation task the fingers and the hand might occlude the object, which can greatly reduce the ability of such tracking system to be accurate and robust. By taking advantage of previously developed *intrinsic* contact sensing fingertips, an algorithm was developed to rectify an estimate of an object's position and orientation by finding a pose that satisfies the contact between the fingers and the object and also the contact normal. The algorithm succeeded in reducing the error by around 70%, while taking less than 0.2 seconds to compute. This allows the real-time tracking of an object



(a) Results for the cuboid box.



(b) Results for the cylindrical can

Fig. 8. Experimental results – blue and green represent x and y components, with the rings plotting the pose obtained by vision and the lines the pose estimated by the proposed method. Red dots are recorded ground truth

while it is being manipulated and can be a complement to a vision system both in terms of accuracy and update rate.

Further improvements on the robustness of the method can be made, such as ensuring that the resulting pose does not produce an intersection between the robot and the object. Also, taking advantage of the local torque information given by the *intrinsic* contact sensors to estimate the object's centre of mass location and take it into account when finding the best pose match. Future work will focus on adding other tactile information such as a tactile array in the robot hand's palm and fuse this information with the fingertip tactile sensing data, integrating a slip detector to trigger the correction algorithm. The application of this method in the field of blind grasping should also be looked into.

ACKNOWLEDGMENT

The authors of this paper would like to thank Prof. Véronique Perdereau and Dr. Guillaume Walck from Université Pierre et Marie Curie for their help in carrying out the tests on the Shadow arm and hand platform.

The research leading to these results has received funding from the European Commission's Seventh Framework Programme, projects COSMOS (Grant No: 246371) and HANDLE (Grant No. 231640).

REFERENCES

- [1] 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*, pp. 515–542, 1982.
- [2] R. C. Luo, Y. C. Chou, and O. Chen, "Multisensor Fusion and Integration: Algorithms, Applications, and Future Research Directions," in *Proc. of 2007 International Conference on Mechatronics and Automation*, pp. 1986–1991, 2007.
- [3] 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.
- [4] D. Kragic, A. T. Miller, P. K. Allen, and D. Kragid, "RealTime Tracking Meets Online Grasp Planning," *Proc. of 2001 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2460–2465, 2001.
- [5] A. Petrovskaya and O. Khatib, "Global localization of objects via touch," *IEEE Transactions on Robotics*, vol. 27, pp. 1–17, 2011.
- [6] M. Prats, P. J. Sanz, and A. P. del Pobil, "Vision-tactile-force integration and robot physical interaction," in *Proc. of 2009 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3975–3980, 2009.
- [7] P. Hebert, N. Hudson, and J. Ma, "Fusion of stereo vision, force-torque, and joint sensors for estimation of in-hand object location," *IEEE Transactions on Robotics and Automation*, pp. 5935–5941, 2011.
- [8] 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," *Proc. of 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1814–1819, 1998.
- [9] J. Bimbo, S. Rodriguez-Jimenez, H. Liu, X. Song, N. Burrus, L. D. Seneviratne, M. Abderrahim, K. Althoefer, and L. D. Seneviratne, "Object pose estimation and tracking by fusing visual and tactile information," in *Proc. of IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 65–70, 2012.
- [10] N. Burrus, M. Abderrahim, J. Garcia, and L. Moreno, "Object Reconstruction and Recognition leveraging an RGB-D camera," in *Proc. of IAPR Conference on Machine Vision Applications*, 2011.
- [11] S. Rodriguez-Jimenez, N. Burrus, and M. Abderrahim, "3D Object Reconstruction with a Single RGB-Depth Image," in *Proc. of International Conference on Computer Vision Theory and Applications (VISAPP 2013)*, 2013.
- [12] H. Liu, X. Song, J. Bimbo, L. Seneviratne, and K. Althoefer, "Object Surface Material Recognition through Haptic Exploration using a Intelligent Contact Sensing Finger," in *Proc. of 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.
- [13] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
- [14] E. Salamin, "Application of quaternions to computation with rotations," Stanford University Artificial Intelligence Laboratory, Tech. Rep., 1995.
- [15] A. Ude and R. Jamova, "Nonlinear least squares optimisation of unit quaternion functions for pose estimation from corresponding features," in *Proc. of Fourteenth International Conference on Pattern Recognition*, pp. 425–427, 1998.
- [16] 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, 1990.
- [17] C. T. Kelley, "Iterative Methods for Linear and Nonlinear Equations," Jan. 1995.
- [18] 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, 1963.
- [19] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*. Springer, 2003.
- [20] 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.
- [21] H. Liu, X. Song, J. Bimbo, and K. Althoefer, "Intelligent Fingertip Sensing for Contact Information Identification," in *Advances in Reconfigurable Mechanisms and Robots I*, pp. 599–608 Springer London, 2012.