Multi-sensor and prediction fusion for contact detection and localization

Javier Felip, Antonio Morales and Tamim Asfour

Abstract— **Robot perception of physical interaction with the world can be achieved based on different sensory modalities: tactile, force-torque, vision, laser, sonar, proprioception, accelerometers, etc. An important problem and research topic in robotics is the question of how to fuse multiple sensory modalities to provide the robot with advanced perception capabilities. However, in the context of contact localization in grasping and manipulation tasks, the fusion of multiple sensory information has not been addressed so far . We propose a sensory information fusion approach for contact detection and localization. The approach relies on the generation of contact hypotheses and the fusion of these hypotheses to determine the likelihood of a contact at a certain location leading to an improved robustness and precision of contact detection. In addition, the approach allows the integration of multiple sensors, environment, context and predictions. We have implemented the proposed approach on two dual-arm robots and validated it through several experiments.**

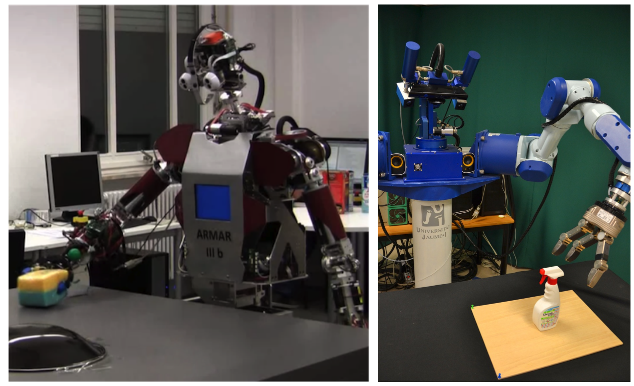


Fig. 1. Left: ARMAR-IIIb Humanoid robot. Right: Tombatossals

I. INTRODUCTION

Neuroscience studies carried out by Johansson et al., [1] pointed out that human grasping is driven by the creation and breaking of contacts with the environment. The prediction of these contact events is also important in the sensorimotor control of manipulation [2].

In the last decade, contact sensing has become a key element on all the robots with manipulation capabilities. Besides tactile sensors, there are other devices that can measure physical interaction, such as force-torque sensors or joint-torque sensors. Moreover there are other sensor modalities that can be used like vision or audio [3]. Unfortunately each data source has its own representation and the information from different sensors cannot be easily compared with that from other sources. In addition information about the environment can also be very useful to constrain where physical interaction can occur.

This paper proposes a robust, scalable and hardware independent framework for sensor fusion focused on contact detection and localization. Firstly, a common representation for all the cues and a method to fuse them is presented. Secondly the details about the required processing of different sensor inputs is demonstrated. Finally the approach is validated through several experiments on Tombatossals (see Fig. 1, right) and a simple use case of a contact driven controller on ARMAR-IIIb (see Fig.1, left) is implemented.

Using a common representation for all the sensory inputs enables contact based controllers to be more hardware independent. This makes systems more portable (same inputs), scalable (easy to add new sensors) and robust (failure tolerant). Moreover, as we show in this work, this level of abstraction enables the addition of non sensor data like context, control or predictions.

J. Felip and A. Morales are with Robotic Intelligence Laboratory at the Department of Computer Science and Engineering, Universitat Jaume I, 12006 Castellon, Spain. T. Asfour is with the Institute for Anthropomatics and Robotics at the Karlsruhe Institute of Technology, Karlsruhe, Germany. {jfelip,morales}@uji.es, asfour@kit.edu

II. RELATED WORK

Data fusion from different sources has been a largely studied problem in robotics. In the early 90's the theoretical basis of the current techniques were already settled [4]. More recently the evolution of parallel computation enabled the use of high computational cost probabilistic approaches (e.g. particle filters) [5]. On real scenarios fusion is often performed with a defined goal: fusion of audio and visual input to track a talking person [6], to track an object [7] or to recognize it [3]. Prats et. al. developed a framework that presents sensor fusion for robotic manipulation, where each sensor handles a controller that contributes to the resultant control applied to the robot [8]. In this paper, instead of focusing on control, we provide a common representation for contact detection and localization.

Using vision and force, Ishikawa et al. [9] proposed a method to detect contacts between a known grasped object and the environment. In that work the fusion method is task specific and the contact detection method is embodiment specific. Other works that perform contact localization, either use only one sensor modality [10] or process and fuse the data with an ad-hoc non scalable method. Hebert et al. [11] presented a probabilistic sensor fusion method to estimate the pose of a grasped object, although they obtained a very good precision (5mm) contacts were considered only on the fingertips.

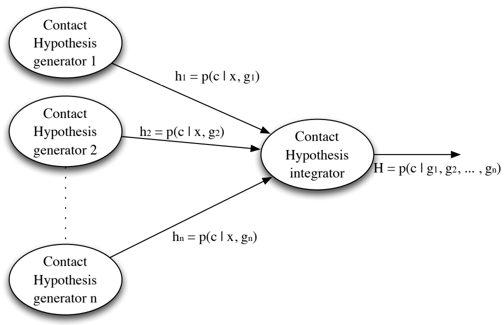


Fig. 2. System overview.

III. SYSTEM DESCRIPTION

The sensor fusion framework is composed of two independent parts; the contact hypotheses generators and the integrator (Fig. 2). Each generator creates contact hypotheses based on a defined criteria (e.g. force sensor, simulator) and sends them to the integrator. The integrator receives those hypotheses, combines them and uses the result to determine the likelihood of a contact at a location.

A. Contact hypothesis and hypothesis space

A contact hypothesis represents the likelihood that a contact happened at a specified location. The hypothesis space *HS* is a 3D Cartesian space discretized in voxels of a fixed size. The state of the *HS* is determined by the occupied voxels and the likelihood of each one. The state of the *HS* is updated by the integrator.

The integrator receives sets of contact hypotheses h_n from the generators $(g_1, \ldots, g_n)$ that represent the probability p that a contact c happened at a specified voxel $x \in \mathbb{R}^3$. Thus, the set of hypotheses generated by g_n is $h_n(x) = p(c|x, g_n)$ (see Fig. 2).

After performing the hypotheses fusion, the output of the integrator is a set of contact hypotheses H representing the probability that a contact happened at a specified voxel combining all the received inputs $H(x) = p(c|x, g_1, \ldots, g_n)$.

A contact hypothesis must have information about its location and likelihood. Beyond the required information, in the proposed framework, a contact hypothesis is composed by the following elements (required fields are **bold**):

- **Location**: Specifies the 3D position of the hypothesis.
- **Likelihood**: List of likelihoods of each source that contributed to this hypothesis.
- Timestamp: Time when was it generated.
- Force magnitude.
- Force direction.
- **Type**: Regular, Support or Null.
- **Source id**: List of sources that generated this hypothesis.

It is possible to have generators that create hypotheses without real evidence of physical contact (e.g. predictions), in order to separate those hypotheses the field "type" is used.

Support hypotheses are used to add contextual data or predictions to the estimation of the contact locations. Therefore if the sensors detect a real contact and generate hypotheses, those hypotheses that fuse with support hypotheses will increase their likelihood. On the other hand, *Support* hypotheses that are not fused with any other hypothesis from perceptual evidence will be discarded. *Null* hypotheses are used to draw a null space for the contact detection. Any hypothesis that fuses with a null hypothesis will be discarded. *Regular* hypotheses are those produced by real sensors from perceptual evidence.

B. Hypotheses fusion

The hypotheses fusion is performed by the integrator. The integrator receives contact hypotheses from multiple generators, then fuses the incoming hypotheses and produces estimations of contact locations (Fig. 3).

At the beginning, the hypothesis space is empty. When a cloud of contact hypotheses is received by the integrator, the hypotheses are processed one by one and added to the hypothesis space.

When adding a new hypothesis, the integrator uses its location to check whether that voxel is already occupied by a hypothesis. If so, both hypotheses are fused, otherwise the hypothesis is inserted into the voxel. When two hypotheses are fused, the resulting hypothesis keeps the location of the voxel, the time-stamp is updated (to the current time), and the force and direction are averaged using both hypotheses values weighted by their likelihood. The lists of likelihoods are combined keeping the newest value if a source is in both lists.

After receiving and combining the contact hypotheses from all the active sources, the fused likelihood of each occupied voxel is computed using the DeMorgan's law, see Eq.(1). We assume that the measurements of the sensors are independent of each other.

By combining the likelihoods using Eq.(1) we expose the integrator to be saturated by inputs like $p(c|x, g_n) = 1$. On the other hand, the saturation will occur only on determined voxels and will not affect the entire *HS*. Moreover the design of contact hypothesis generators should take into account that issue, and produce very high likelihoods only when necessary.

$$p(c|x, g_1, \ldots, g_n) = 1 - \prod_{i=1}^{n}(1 - p(c|g_i)) \qquad (1)$$

A hypothesis is discarded when its time-stamp is older than a configurable timeout parameter. This parameter should be adjusted depending on the update rate of the hypotheses. The fusion process is detailed in Algorithm 1.

It is possible that the incoming hypotheses are generated from different sources at different rate, to perform the hypothesis fusion, the integrator waits until the hypotheses of all the active sources are received. From faster sources the newest readings are used. The sources can be plugged and unplugged to the integrator dynamically.

Algorithm 1 Contact hypotheses fusion algorithm

```
function ProcessIncomingHypotheses(hypotheses)
    for all h in hypotheses do
        if IsVoxelOccuppied(h.location) then
            FuseHypotheses(h, GetVoxel(h.location))
        else
            SetVoxel(h)
        end if
    end for
end function
function FuseHypotheses(h1, h2)
    h3.location := h1.location
    h3.likelihood := h1.likelihood ∪ h2.likelihood
    h3.sources := h1.sources ∪ h2.sources
    h3.fMagnitude := weightedMean(h1.fMag, h2.fMag)
    h3.fDirection := weightedMean(h1.fDir, h2.fDir)
    h3.updateTimeStamp()
    SetVoxel(h3)
end function
```

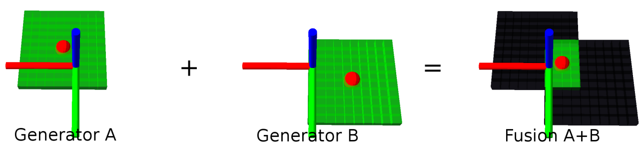


Fig. 3. Contact hypotheses fusion and contact detection. Black boxes are discarded hypotheses. Green boxes are hypotheses considered for contact detection. Red spheres show the result of the contact detection after clustering the thresholded contact hypotheses and calculating its centroid.

C. Contact condensation

When all the incoming hypotheses have been fused, the hypothesis space contains a cloud of contact hypotheses with different likelihoods. In order to summarize the information and provide the estimated location of the contacts, two strategies are proposed.

1) Threshold: A straightforward method is to set up a high threshold for the likelihood (e.g $H(x) \geq 0.6$), and filter the data to obtain the hypotheses that will be considered contacts. This strategy is eligible if all the generators produce very precise data, otherwise the likelihood will be distributed among many hypotheses and none of them will be over the threshold.

2) Threshold, cluster and centroid: On the other hand we can use a low threshold (e.g $H(x) \geq 0.1$) and calculate the global centroid weighted by likelihood. This will reduce the detected contacts to a single one. However if there are separated contact regions the result will be the average of those regions. Thus, before performing the centroid calculation, the contact regions are detected using an euclidean clustering algorithm[1]. Then the centroid (likelihood weighted) is calculated for each cluster (See red spheres in Fig. 3).

[1]Clustering algorithm taken from: `http://www.pointclouds.org/documentation/tutorials/cluster_extraction.php`

IV. Hypotheses generators

A contact hypothesis generator can use the information from a sensor, a controller, a simulator or any other source that provides data about a perceived or a predicted contact located in the space. Contact hypothesis generators provide at least a cloud of contact hypotheses (that determine the possible contact locations) and the likelihood of each generated contact hypothesis.

Contact hypothesis generators are classified into two main types: single-contact (can only detect one contact at a time) and multi-contact. Each contact hypothesis generator can use the information available to determine where to generate hypotheses, some examples are: the sensor's shape, robot geometry, sensitive geometry, constant volume and range data to name a few.

The likelihood is calculated depending on the type of the generator. For single-contact hypothesis generators, the probability of the detected contact has to be distributed among the generated contact hypotheses, see Eq.(2). The likelihood can be distributed uniformly or with any other distribution, depending on the contact hypothesis generator data.

$$\sum p(c|x_i, g_{single}) = 1 \tag{2}$$

For multi-contact hypothesis generators, we will allow $\sum p(c|x_i, g_{multi}) > 1$. The likelihood of each hypothesis can be calculated regarding the information type from the data source:

- Binary: (contact / no contact) it gives no clue about the contact distribution, in that case a fixed value for all the hypotheses is used.

$$p(c|x, g_{binary}) = constant \tag{3}$$

- Value: If the value of the input data is related to the contact likelihood. It can be calculated using Eq.(4) where $argmax(data)$ is used for normalization and represents the maximum value of the current reading.

$$p(c|x, g_{value}) = \frac{data_x}{argmax(data)} \tag{4}$$

- Distance: If the value for each hypothesis is a distance. Eq.(5) can be used to determine the likelihood of each hypothesis. Where λ determines the distance at which the likelihood will be 0.5, this has to be tuned depending on the precision of the range sensor and the calibration.

$$p(c|x, g_{distance}) = \frac{\lambda^2}{\lambda^2 + distance^2} \tag{5}$$

For example a bumper based contact hypothesis generator, would be single-contact and will use the sensor geometry to place the contact hypotheses on. On the other hand, a laser combined with the robot geometry would be multi-contact, use the robot geometry close to the range data to place the hypotheses and use the distance between the range data and the robot model as the likelihood of each contact hypothesis.

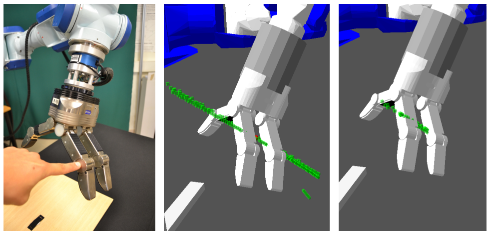


Fig. 4. Example of the hypotheses generated by the force-torque generator. Left: Real contact. Center: Generated hypotheses before hand geometry filtering. Right: Generated hypotheses.

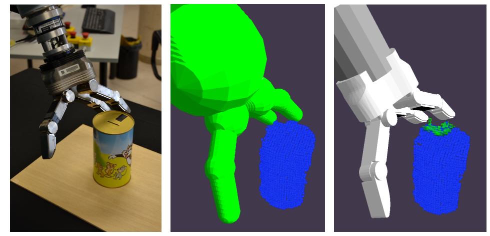


Fig. 5. Example of the hypotheses generated by the range sensor generator. Left: Real scenario. Center: Segmentation using spherical model. Right: Generated hypotheses.

V. Implemented generators

In this section we show the implementation of several contact hypotheses generators based on sensors that are nowadays quite common in robotic manipulators. To see a demo of the described contact hypotheses generators please refer to the attached video.[2]

A. Experimental platforms

For the validation experiments we have used the UJI Humanoid torso, also called Tombatossals. Tombatossals is a humanoid torso with 29 DOF (see Fig. 1 Right). It is composed of two 7 DOF Mitsubishi PA10 arms. The right arm has a 4 DOF Barrett Hand and the left arm has a 7DOF Schunk SDH2 Hand. Both hands are endowed with Weiss Tactile Sensor system on the fingertips. Each arm has a JR3 Force-Torque sensor attached on the wrist between the arm and the hand. The visual system is composed of a TO40 4 DOF pan-tilt-verge head with two Imaging Source DFK 31BF03-Z2 cameras. Attached to the forehead there is a Kinect™ sensor. The robot used for the use case experiment is ARMAR-IIIb (see Fig. 1, left), it is a humanoid robot with 43 actuated DOFs. For the experiment we have only used its right arm (7 DOF) and hand (7 DOF). It has a force-torque sensor on the wrist and tactile sensor pads on the palm and fingertips. See [12] for more details about ARMAR-IIIb.

B. Force-torque hypotheses generator

Uses 6D force-torque sensor readings to determine the contact hypotheses location. These types of sensors produce a 3D force vector f and a 3D torque vector τ. The system of equations shown in Eq.(6) can be used to determine the contact point, where τ are the torque readings and f the force readings for the three axis.

$$\begin{cases} \tau_x = f_z \cdot y - f_y \cdot z \\ \tau_y = f_x \cdot z - f_z \cdot x \\ \tau_z = f_y \cdot x - f_x \cdot y \end{cases} \quad \text{w.r.t. sensor frame} \tag{6}$$

Unfortunately that system has no single solution. To generate a set of possible solutions, we will use only two of those equations at a time (see Eq.(7)) and solve the leftmost group for a set of possible values for x (if $f_x \neq 0$), the center group for y (if $f_y \neq 0$) and the rightmost group for z (if $f_z \neq 0$). After this we have three lines of contact hypotheses (see Fig.4, center). The lines are theoretically equal but due to sensor noise and singularities they can be different.

The hypotheses that are not close to the robot hand surface will be filtered out (see Fig.4, right). The filtering uses the distance to the spherical model of the robot (see Fig.5, center).

$$\begin{cases} y = \frac{f_y \cdot x - \tau_z}{f_x} \\ z = \frac{\tau_y + f_z \cdot x}{f_x} \end{cases} \begin{cases} x = \frac{\tau_z + f_x \cdot y}{f_y} \\ z = \frac{f_z \cdot y - \tau_x}{f_y} \end{cases} \begin{cases} x = \frac{f_x \cdot z - \tau_y}{f_z} \\ y = \frac{\tau_x + f_y \cdot z}{f_z} \end{cases} \tag{7}$$

This is a single-contact hypothesis generator, it is not possible to determine the location of multiple contacts with a F/T sensor on the wrist. Moreover this generator assumes no external force to be applied on the robot. Another approach to determine the contact point regardless of end effector geometry is shown in [13].

As it gives no information about the contact distribution, the likelihood is uniformly distributed among all the generated hypotheses. The Force-torque hypotheses generator was implemented both for ARMAR-IIIb and Tombatossals.

C. Tactile sensor hypotheses generator

Tactile sensors typically produce an array of pressure values with measurements from a grid of sensing cells. In combination with the joint positions and the robot model it is possible to determine the spatial location of contacts. In this kind of sensors, a single contact may generate marginal readings in the nearby taxels. Thus we will consider that the taxel pressure value is related with the contact likelihood.

This hypothesis generator is multi-contact and uses the sensor values to calculate the likelihood of each generated hypotheses, see Eq.(4). The hypotheses location is determined by the activated taxels of the sensor. The tactile hypotheses generator was implemented both for ARMAR-IIIb and Tombatossals.

D. Range sensor hypotheses generator

Range sensors can also be used to detect contacts. The Kinect™ installed on Tombatossals' forehead combined with a robot spherical model and object tracking is used to generate contact hypotheses. The foreground, consisting of the robot arm, and probably several obstacles is separated from the background of the scene using a box pass through filter

[2]Watch the HD video at: `https://dl.dropboxusercontent.com/u/6172197/robot_videos/HUMANOIDS14_felip_morales_asfour.mov`

that isolates the workspace. A spherical representation of the arm and hand of the robot, together with proprioception are used to segment the robot within the point cloud, (Fig.5, center). The remaining clusters in the point cloud correspond to objects. Using the isolated object points, ICP is applied to detect object motion. When object motion is detected, the hand moving direction and the object points are used to determine the hypotheses location. More details about this contact detection and localization method can be found in [14].

The likelihood of each hypothesis is calculated using the distance to the spherical robot model, see Eq.(5). For our experiments we have set $\lambda = 0.01m$. Hypotheses with very low likelihood (i.e. $\leq$ 0.01) are discarded to save memory and computation time. The range sensor hypotheses generator was implemented only for Tombatossals, although it can be implemented also for ARMAR-IIIb using its stereo head as a range data sensor.

E. Finger pose feedback hypotheses generator

The variation of compliant hand finger positions is used to detect a contact on the fingers. Before starting the detection, the current pose of each finger joint is stored. When a variation on a joint is detected, the finger geometry is used to place the contact hypotheses. Like for the force-torque generator, the likelihood is uniformly distributed among all the generated hypotheses.

This generator assumes that the hand joints are not actuated to move. After moving the hand joints to a different position, the generator is reset to get the reference values updated. The finger pose feedback hypotheses generator was implemented only for ARMAR-IIIb. Tombatossals does not have compliant hands.

F. Motion estimation support hypotheses generator

Using the current motion of the robot, contact hypotheses are generated on the next predicted hand position. As this generator is not based on physical evidence, it produces only *Support* hypotheses. The generator assumes that the robot will continue moving as it did in the last time step. Thus the predicted position of the robot joints $q(t+1)$ is calculated using Eq.(8)

$$q(t+1) = q(t) + \Delta q \tag{8}$$

This generator can detect multiple contacts at a time and the data input is binary, thus the likelihood of the generated hypotheses depends on the weight we want to give to this support generator. During the experiments we found that a good value is 0.3. Motion estimation support hypotheses generator was implemented both for ARMAR-IIIb and Tombatossals.

G. Simulator predictions support hypotheses generator

This generator is only implemented for Tombatossals. It uses the integrated OpenGRASP simulator as a prediction engine to detect where contacts are supposed to happen.

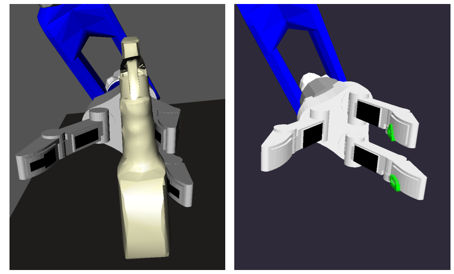


Fig. 6. Simulator prediction support contact hypotheses generator. Left: Simulator, Right: Support contact hypotheses generated (green voxels).

The simulator requires the model of the robot, the environment and the objects in the workspace. The generator looks for geometry contacts using the available methods in OpenRAVE/OpenGRASP and generates *Support* hypotheses where contacts in simulation are detected (Fig. 6). As the motion estimation generator, this is a multi-contact binary generator and the likelihood of the generated *Support* hypotheses depends on the relevance that the simulator will have. We found a good value on 0.5.

VI. Experiments

We have conducted two experiments: a validation and a use case. To do so we have implemented the method on the two robotic platforms, sec. V-A. On Tombatossals we have performed a validation test to show the method performance and some hypotheses generators. On ARMAR-IIIb we have applied the contact detection method on a real grasping situation; using the contact output information to drive a reactive grasp algorithm as in [15] or [16]. We have used the same voxel size in the generators and in the integrator, 5mm side. Thus the precision of the contact detection is limited to the 5mm resolution of the hypothesis space. The selected contact condensation method is the threshold, cluster and centroid.

A. Experimental validation

The experimental validation is performed using the Tombatossals robot. This experiment consists on touching three different objects (a box, a cylinder and a sprayer bottle) each one from 15 different approach directions. Then we compare the contact locations obtained by the sensor fusion method with ground truth data. The role of the simulator in this experiment is twofold, as a ground truth tool and as a prediction engine.

1) Ground truth data: The scenario consists of an object on a table in front of the robot. The 3D model of the object is known. To obtain the ground truth data, the position of the object is calibrated using the robot left arm. The calibration is manually performed: Touching several points of the real object and moving the simulated object to fit those positions. With the object position calibrated in the simulator, we have used the joint positions recorded from the experiment execution to get the exact hand-object contact points and use them as ground truth data. Two of the experiments did not really touch the object, they were removed leaving 43

TABLE I

RESULTS FOR EACH SENSOR MODALITY

Sensor Modality	Detected contacts	ε(cm)	σ(cm)
Tactile	9/43 (20.9%)	1.25	$\pm$0.20
Force	34/43 (79.1%)	5.37	$\pm$1.18
Range	25/43 (58.1%)	3.74	$\pm$0.73
All	39/43 (90.7%)	4.31	$\pm$1.10
All + Simulator	39/43 (90.7%)	3.33	$\pm$1.11

Number and % of detected contacts. ε shows the distance between the ground truth and the detected contact. σ is the mean dispersion of the centroid calculation.

touches. To keep the ground truth right the object is fixed and cannot be moved by the robot during the experiments.

2) Hypothesis generators: The simulator is used as a prediction engine to generate support contact hypotheses, for this purpose we have added error to the calibrated position of the objects. In order to model the uncertainty introduced by state of the art 3D object recognition and pose estimation methods we have added a Gaussian error, $\mathcal{N}(\mu = 0, \sigma = 2)$ in cm, to the objects calibrated position. The other contact hypotheses generators used are: tactile, force-torque and range.

3) Result discussion: The results, after the execution of 43 touches, are shown in Table I. The distance between the ground truth contact and the result of the contact condensation (See Sec.III-C) is used as the error measure ε. The standard deviation of the centroid calculation performed by the contact condensation is used as the precision measure σ. In Table I the mean accuracy and precision considering different sensor modalities is shown. Fig.7 depicts the individual results for each touch experiment considering different sensor modalities.

Although the tactile sensor modality has a small localization error (around 1cm $\pm$0.2) the contact detection is quite low, 20.9%. This low detection rate is related to the reduced area that the tactile sensors can cover, thus many contacts happen outside the tactile sensor patches. Regarding the force-torque generator, although the detection rate is good (79.1%) the localization error is around (5cm $\pm$1.18), the force-torque contact hypothesis generation method is very sensitive to noise and the effect of multiple contacts decreases the accuracy. The range modality shows average contact detection (58.1%) and good accuracy (3.7cm $\pm$0.73), the main problem of this modality are contacts on hidden surfaces or occluded by the hand.

Fusing the modalities, the detection raises to 90.7%. The accuracy depends on which sensors are detecting the contact. The fusion method takes the most precise sensor (i.e. the point with higher probability density). Note that the error (4.31cm $\pm$1.1) is increased by those cases where only the force sensor generates hypotheses. This problem is solved adding the predicted contacts from the simulator, then the accuracy is improved (3.33cm). Moreover, beyond the sensors precision, the object position uncertainty (modelled by $\mathcal{N}(0,2)$ cm), the robot model error and the joint encoders error also influence the total error.

B. Grasping application

For the real application problem, the experimental setup consists of a bottle of water in front of the right hand of ARMAR-IIIb. The objective is to grasp the bottle using the provided contact feedback, to do so we have implemented a robust grasp method similar to [15] that uses the contact location output from the proposed method. We assume that the bottle is in the trajectory of the hand but not exactly in front of it. In this case the object is not fixed and can be moved by the robot, only generators that depend on the object position (simulator predictions) would be influenced by this fact but for ARMAR-IIIb the simulator predictions are not implemented.

The result of the experiment is shown in Fig. 8 and in the attached video. The robot is able to detect the contact location fusing the information coming from the force sensor and from the arm motion. In this case, neither the tactile sensors nor the finger position did detect the contact.

VII. CONCLUSION

In this paper we have shown theory, validation and application of a sensor fusion method for contact detection. One of the contributions is that the method allows the integration of input from other sources but sensors, such as context, predictions or environment. We have shown that the projection of predictions or beliefs into the sensor space improves the results. The theoretical approach has been implemented in two different robotic platforms. The experiments carried out using Tombatossals have shown that the method is suitable to be used in real environments providing a framework to fuse sensor, simulation and prediction data to improve contact detection and localization. The experiments also show that the fusion of different sources performs better than the sources separated.

Moreover, we have implemented the method on another real platform, and conducted a grasping experiment where contact information provided by the system is used to improve grasping. The implementation on multiple platforms shows that the approach can contribute to the sensor skills of any robot or even enable the interaction of different robots on sharing and combining their knowledge about existing contacts. This opens the door to multi-robot scenarios, where contact hypotheses generators from different robots can be used together to share physical interaction information and localize contacts. Further research on the proposed contact hypotheses generators or the addition of more precise sensors will help to improve the overall performance of the system. Moreover, with a few modifications the framework can also be used to detect surprise (prediction and sensing mismatch) and enable low level reactive behaviours, internal model refinement or higher level reasoning.

ACKNOWLEDGMENT

The paper describes research carried out at the Institute for Anthropomatics and Robotics at the Karlsruhe Institute of Technology and at Robotic Intelligence Lab at Universitat Jaume I. The research leading to these results has

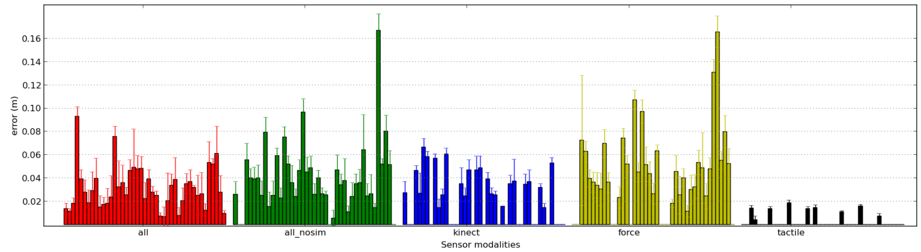
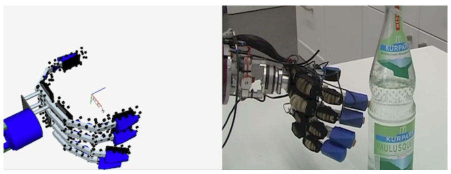
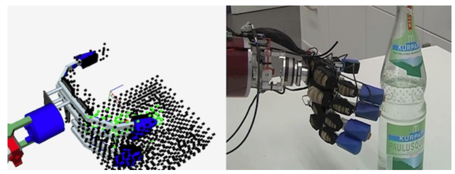


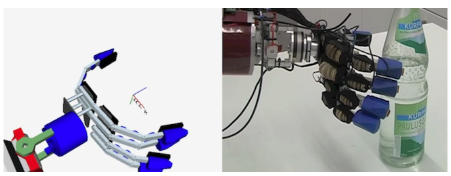
Fig. 7. Results for the touch experiments considering different sensor modalities. Results are grouped by sensor modality, each one has 43 slots, one for each touch performed, if the contact was detected a bar shows the ε and σ for that contact.



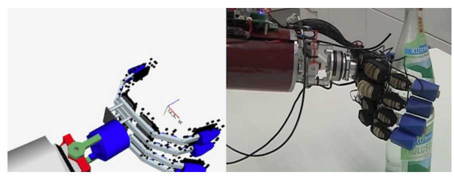
(a) Motion support hypothesis are generated while the hand moves.



(b) Contact detected. Force-torque hypothesis are fused with motion support hypothesis.



(c) Contact information is used to correct hand pose.



(d) Finally a good grasp is achieved.

Fig. 8. Execution on a real robot. Black voxels: Discarded hypotheses. Green voxels: Hypotheses after likelihood threshold, Red voxels: Contact detection output.

received funding from the European Union Seventh Framework Programme under grant agreement no. 270273 (Xperience), Ministerio de Economía y Competitividad (DPI2011-27846), Generalitat Valenciana (PROMETEOII/2014/028) and by Universitat Jaume I (P1-1B2011-54) and (PREDOC/2010/23).

References

[1] R. Johansson and J. Flanagan, "Tactile sensory control of object manipulation in humans," in *The senses, a comprehensive reference : somatotsensation Vol 6*, vol. 6, pp. 67–86, Elsevier, 1 ed., 2008.

[2] R. S. Johansson and J. R. Flanagan, "Sensorimotor control of manipulation," in *Encyclopedia of Neuroscience*, no. 8 in Encyclopedia of Neuroscience, pp. 593–604, Elsevier, 8 ed., 2009.

[3] L. Lacheze, Y. Guo, R. Benosman, B. Gas, and C. Couverture, "Audio/video fusion for objects recognition," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 652 –657, oct. 2009.

[4] J. Hackett and M. Shah, "Multi-sensor fusion: a perspective," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pp. 1324–1330 vol.2, 1990.

[5] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[6] I. Fermin, H. Okuno, H. Ishiguro, and H. Kitano, "A framework for integrating sensory information in a humanoid robot," in *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, vol. 3, pp. 1748 –1753 vol.3, 2000.

[7] W. Meeussen, J. Rutgeerts, K. Gadeyne, H. Bruyninckx, and J. D. Schutter, "Particle filters for hybrid event sensor fusion with 3d vision and force," in *Multisensor Fusion and Integration for Intelligent Systems, 2006 IEEE International Conference on*, pp. 518 –523, sept. 2006.

[8] M. Prats, A. P. D. Pobil, and P. J. Sanz, *Robot Physical Interaction through the combination of Vision, Tactile and Force Feedback - Applications to Assistive Robotics*, vol. 84 of *Springer Tracts in Advanced Robotics*. Springer, 2013.

[9] T. Ishikawa, S. Sakane, T. Sato, and H. Tsukune, "Estimation of contact position between a grasped object and the environment based on sensor fusion of vision and force," in *Multisensor Fusion and Integration for Intelligent Systems, 1996. IEEE/SICE/RSJ International Conference on*, pp. 116 –123, dec 1996.

[10] M. Meier, M. Schopfer, R. Haschke, and H. Ritter, "A probabilistic approach to tactile shape reconstruction," *Robotics, IEEE Transactions on*, vol. 27, pp. 630 –635, june 2011.

[11] P. Hebert, N. Hudson, J. Ma, and J. Burdick, "Fusion of stereo vision, force-torque, and joint sensors for estimation of in-hand object location," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 5935–5941, 2011.

[12] T. Asfour, K. Regenstein, P. Azad, J. Schroder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "Armar-iii: An integrated humanoid platform for sensory-motor control," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pp. 169 –175, dec. 2006.

[13] Y. Karayiannidis, C. Smith, F. Vina, and D. Kragic, "Online contact point estimation for uncalibrated tool use," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014.

[14] J. Bernabe, J. Felip, A. Del Pobil, and A. Morales, "Contact localization through robot and object motion from point clouds," in *Humanoid Robots, 2013 13th IEEE-RAS International Conference on*, Atlanta, oct. 2013.

[15] J. Felip and A. Morales, "Robust sensor-based grasp primitive for a three-finger robot hand," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 1811 –1816, 2009.

[16] K. Hsiao, S. Chitta, M. Ciocarlie, and E. Jones, "Contact-reactive grasping of objects with partial shape information," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 1228 –1235, 2010.