Visual Collision Detection for Corrective Movements during Grasping on a Humanoid Robot

David Schiebener, Nikolaus Vahrenkamp and Tamim Asfour

***Abstract*— We present an approach for visually detecting collisions between a robot's hand and an object during grasping. This allows to detect unintended premature collisions between parts of the hand and the object which might lead to failure of the grasp if they went unnoticed. Our approach is based on visually perceiving that the object starts to move, and is thus a good complement for force-based contact detection which fails e.g. in the case of grasping light objects that don't resist the applied force but are just pushed away.**

Our visual collision detection approach tracks the hand in the robot's camera images and analyzes the optical flow in its vicinity. When a collision is perceived, the most probable part of the hand to have caused it is estimated, and a corrective motion is executed. We evaluate the detection together with different reaction strategies on the humanoid robot ARMAR-III. The results show that the detection of failures during grasp execution and their correction allow the robot to successfully finish the grasp attempts in almost all of the cases in which it would otherwise have failed.

I. INTRODUCTION AND RELATED WORK

Grasping objects is an indispensable competence for humanoid robots. While grasp planning is a challenging problem that (for good reason) received and still receives a lot of attention, the actual execution of the planned grasps on a real robot frequently poses serious problems too. Those difficulties are due to imprecision in object localization, hand-eye calibration and execution of the planned grasping motion, as well as the planned grasps themselves which may sometimes be inappropriate. The authors in [1] and [2] have actually showed that the currently used grasp quality measurements often lead the grasp planners to solutions that are not reliable in the real world despite seeming good in the used mathematical models. The problem of grasp plans that are not or only approximately suitable arises in particular when no precise object model is available or an unknown object is to be grasped based on heuristics (like e.g. in [3] and [4]).

Visual servoing [5] is an important technique that helps to greatly reduce the effects of imprecise hand-eye calibration and inexact arm motion by localizing both hand and object in the same camera images. The position and orientation (pose) of the hand relative to the object is thus determined visually in the camera frame, and as long as the kinematic model is good enough to allow for an approximately correct motion, the hand can be visually guided towards the intended pose by continuous correction [6]. The degree of exactness that can be achieved using visual servoing is essentially limited by the precision of the perception components.

D. Schiebener, N. Vahrenkamp and T. Asfour are with the Institute for Anthropomatics and Robotics, High Performance Humanoid Technologies Lab (H^2T), at the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany. schiebener@kit.edu, vahrenkamp@kit.edu, asfour@kit.edu

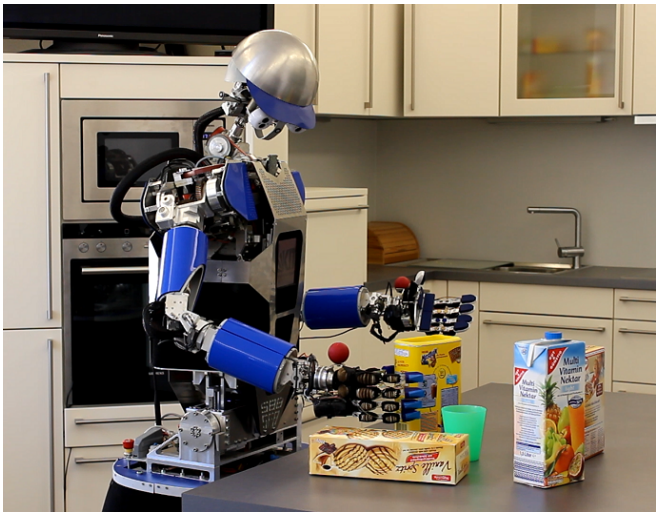

Fig. 1: The humanoid robot ARMAR-III grasping an object.

Thus, when we apply visual servoing, the remaining causes of imprecision are the object localization algorithm, the limited resolution of the vision system, the configuration of the fingers, and the grasp planner, especially when no perfect object model is available. In reality, these errors may be small but will always be present, and a frequent result is that the hand prematurely touches the object and moves it, which may cause the grasping to fail. Therefore, whenever the required accuracy of the grasp can not be guaranteed by the planning, perception and kinematic components, the robot should be aware of possible errors during the grasp execution and be able to detect and correct them.

Collision detection during grasp attempts has mostly been applied in the context of blind or reactive grasping, e.g. in [7], where objects from a box are grasped blindly. The torque detected by a force-torque sensor in the wrist of the robot arm is used to determine which finger touched the object and to correct the hand position accordingly. In [8], we reactively grasp unknown objects that have previously been segmented by vision and pushing actions. There, we use a force-torque sensor in the wrist, tactile pads in the fingers and the palm, and finger joint angle measurements to determine the contact location during the grasping approach and correct the hand position if necessary. In [9], tactile sensors in fingers and palm are used to adapt the hand position and finger configuration to the object pose and shape during the grasp

execution. In [10], the tactile sensors in the fingers are used to reactively adapt the finger configuration while closing the hand during grasp execution.

However, all approaches based on force or tactile feedback require that the object resists the robot hand sufficiently so that a force can actually be measured. For top-down grasps, this is usually unproblematic as long as the object is not too easily deformable, but when light objects are grasped from the side, the sensitivity of the currently available sensors is far from being sufficient. One way to circumvent this is to use proximity sensors as in [11], another way is to use visual information, which is what we propose in this work.

To the best of our knowledge, the only other attempt to visually detect collisions in the context of grasping is [12]. They obtain an RGBD point cloud from a static depth camera observing the scene which consists of a table surface with only the object on it and the robot arm, of which a geometric model is available. The arm is tracked in the depth image and the object is segmented by removing the table surface. When the object moves while the arm is near it, a collision is assumed to have occurred. The most probable part of the hand to have caused the collision is determined based on the geometric model. This information would allow to implement a reaction strategy, although this has not been done yet in that paper. It is not obvious though how this approach could be generalized to more complex scenes and a non-static camera.

Our approach is related to [12] in the sense that it is also based on the idea of visually detecting the motion of the object when a collision occurs. We took some inspiration from [13], where a static camera observes a scene in which the robot arm approaches an object and, in the moment it collides with it, causes a sudden spread of optical flow in the image area occupied by the object. In our case the situation is more complex though, as the camera is located in the robot head and moves during the execution of the grasp.

II. OVERVIEW

The execution of a grasp in general comprises the motion of the robot's arm, hand and fingers from an initial pose to a configuration in which the object is held firmly inside the hand. Collision-free motion planning in this high-dimensional space is challenging [14]. A common approach is to separate the grasp and the motion planning step by using precomputed grasp tables which are applied on localized object poses in order to allow for efficient processing. Such grasping pipelines (see e.g. [15] or [16]) usually comprise a motion execution component which is responsible for moving the end effector along a planned path.

Within this work, we assume that a grasping pose p_g together with a corresponding pre-grasp pose p_{pre} are available for the target object. Further, we assume that the straight trajectory between p_{pre} and p_g is collision-free. For our experiments, p_{pre} and p_g were defined manually, but in general this approach can seamlessly be integrated as a grasp execution module within the robot's grasping pipeline. In that case, the grasping poses will be computed by grasp planning components, and the pre-grasp pose is equivalent to a point on the approach trajectory. Fig. 2 shows a schematic overview of the proposed grasp execution procedure. During the critical last part of the approach, we continuously check for collisions and if one is detected, a corrective reaction is performed.

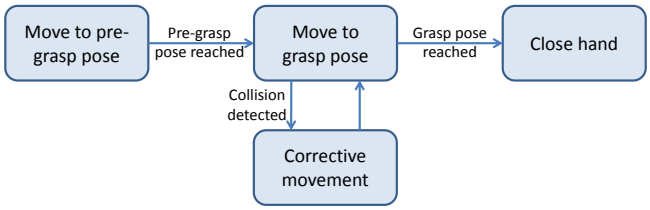

Fig. 2: Schematic overview of grasping with collision detection and correction: First, the robot moves the hand to a pre-grasp pose relative to the object. Then the actual grasp pose is approached while continuously checking for collisions. If a collision is detected, a corrective movement is executed and the grasp pose adapted. When the grasp pose is reached, the fingers are closed.

Fig. 3 shows the processes running during the critical approach phase. The hand is guided towards the grasp pose by visual servoing. At the same time, our visual collision detection continuously checks for indications that the hand has unintendedly collided with the object, in which case the approach is interrupted.

The details of the collision detection algorithm are explained in the following section III, and the different reaction strategies we implemented are described in section IV. The detection and the different strategies are tested on our robot and quantitatively evaluated in section V.

III. VISUAL COLLISION DETECTION

The main idea of our approach is to detect the motion of the object that is caused by the unintended collision with the hand. To this end, we track the hand in the camera images and observe the optical flow next to it in the direction in

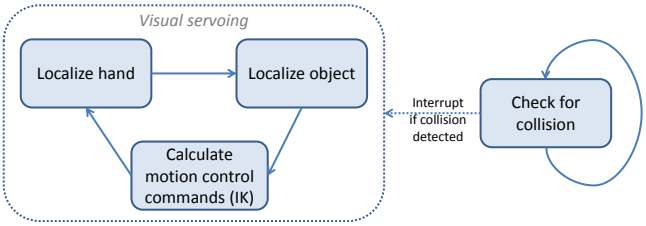

Fig. 3: Visual servoing with collision detection: The intended grasping pose relative to the object is approached using visual servoing, i.e. the robot continuously localizes object and hand visually, calculates the required relative motion of the hand in Cartesian space and an appropriate joint motion to realize it using inverse kinematics. Concurrently, it checks for collisions, and if one is detected, the approach is interrupted and a correction initiated.

which it is moving. The individual components are described in the following subsections.

A. Hand Tracking

Visual tracking of the robot's hand is necessary for both the visual servoing and the collision detection. For the visual servoing, we use a simple and very fast method in which we localize the red spherical marker fixed to the robot's wrist which can be seen in fig. 1 (see [6]). The orientation is obtained from the forward kinematics. For collision detection, we use a tracking algorithm based on the particle filter approach [17] which estimates the position, orientation and finger configuration of the hand. This comprehensive information is particularly important when trying to determine which part of the hand has collided with the object.

The end effector of our humanoid robot ARMAR-III [18] is a five-finger hand which is pneumatically actuated (i.e. with air pressure), has two DoF in each finger and one in the palm. Tactile sensor pads are installed in each finger tip and the palm, and a force-torque sensor in the wrist. However, these sensors are not used in this work[1]. The head is equipped with a stereo camera system.

The particle filter estimates the position and orientation of the hand, and a reduced set of the finger DoF, which results in a 12-dimensional state space. The particles are initialized with the position from the localization of the spherical marker, the orientation from forward kinematics, and the measured finger joint angles. In each iteration of the particle filter algorithm, the particles are perturbed by adding random Gaussian noise, and then the plausibility of the hand configurations defined by the particles is evaluated based on the current camera images. In the next iteration, particles are redrawn with a probability proportional to their rating, the relative hand motion since the last iteration is applied to them, random noise is added and they are evaluated again. Robustness of the tracking is enforced by only allowing particle configurations that are within an empirically determined interval around the configuration obtained from forward kinematics and joint value sensor readings.

The key component of the particle filter is the rating function which estimates for each particle s_i the conditional probability $p(z|s_i)$ that the input z (the camera images) was caused by the hand configuration defined by s_i. This probability is calculated based on five different cues, which are each determined in both of the stereo camera images. The first cue is $q_1(s_i) = 1/d$, where d is the distance between the positions of the red spherical hand marker in the model and in the camera images. The other four cues are based on the blue fingertips: They are projected into the images given the hand configuration of particle s_i. The cue $q_2(s_i)$ gives a rating proportional to the number of pixels in the area covered by the fingertips that have the correct color, $q_3(s_i)$ rewards if a large part of the area has the correct color[2]. The cue $q_4(s_i)$ checks for intensity edges in the image that correspond to those of the projected fingertip, and $q_5(s_i)$ takes the edge directions into account. The conditional probability of a given particle s_i is then

$$p(z|s_i) \;=\; \vartheta \, e^{\sum_{j=1}^{5} \omega_j \; q_j(s_i)}$$

where ϑ is a scaling factor and the ω_j are weights for the different cues.

On each pair of stereo camera images, two iterations of simulated annealing are performed to enhance the precision of the final localization result, which is the average of all particles weighted with their probability.

B. Optical Flow

Concurrently with the hand localization, we calculate the optical flow between the current camera image and the one taken at the last iteration of the collision check. The optical flow is determined using the algorithm proposed by [19] which is implemented in OpenCV. The idea of the algorithm is to approximate the neighborhood of each pixel by a quadratic function. If a quadratic function undergoes a translation, the displacement can be determined in closed form. By iteratively determining these translations first on a coarse and then on increasingly finer scales, larger displacements that exceed the direct neighborhood of the pixel can also be determined and refined. The algorithm provides a dense estimation of the optical flow between two images, although in larger monotone image regions it does not return any values. This is not a problem in our case, as we are interested in the image region around hand and object which offers enough visual information for the algorithm.

C. Collision Detection

In [13], the moment of the collision between robot arm and object is recognized by the fact that an area of significant optical flow appears next to the hand. In our case, the cameras are on the robot and moving with it during the grasp, and consequently there is optical flow throughout the whole image. Therefore we have to solve the more general problem of discovering if an object next to the hand moves in a way that is inconsistent with the rest of the scene. Note that for the static part of the scene, its projected motion is not equal throughout the image but depends on the distance to the camera.

To overcome this problem, we cluster the pixels of the camera image by their optical flow values. To this end, we apply x-means, a variant of k-means that automatically determines an appropriate number of clusters given a parameter that balances the number of clusters and their in-class variance [20]. The idea is to detect if there is a cluster of similar optical flow next to the hand which is different from the optical flow in the rest of the scene, which would indicate that an object is being moved by the hand.

[1]We used these sensors for reactive top-down grasping of unknown objects in [8].

[2]This way, we assure that the rating is not too good if the projected fingertips are extremely small or large, which would be the case if only one of the two criteria was used.

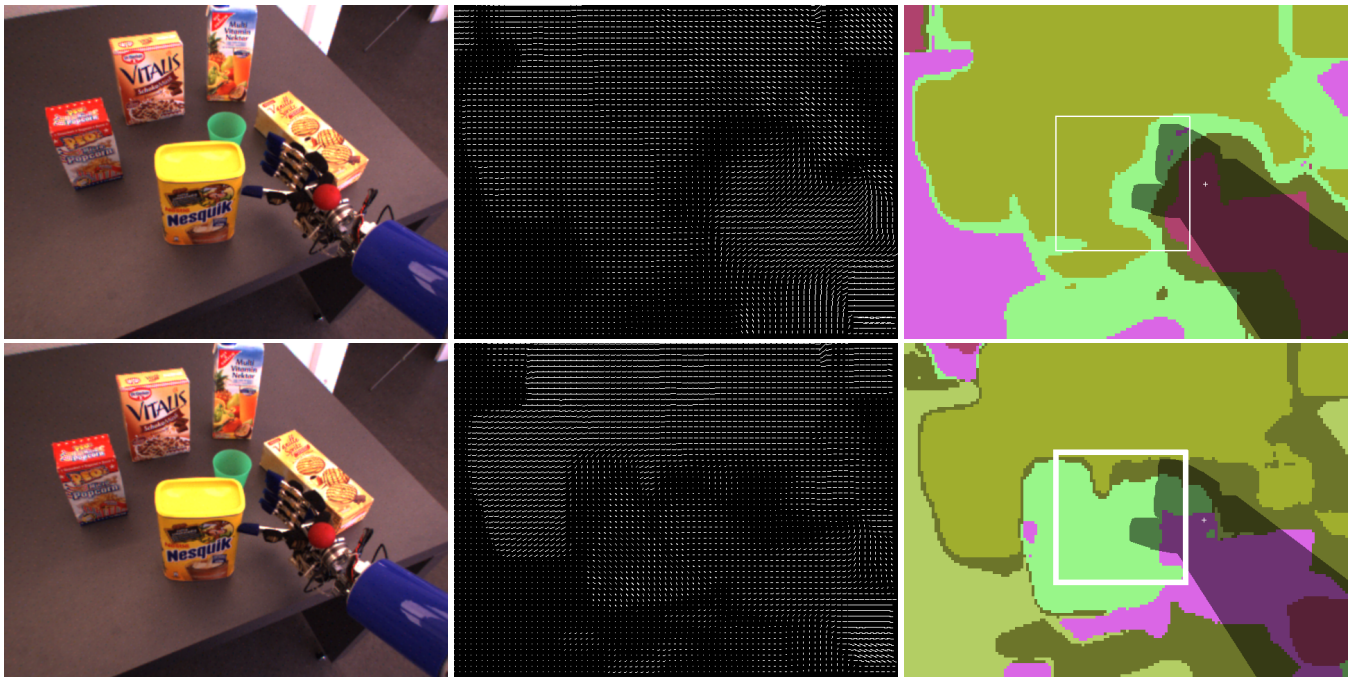

Fig. 4: Visual collision detection in the moment when the robot's hand touches the object: The left column shows the scene from the robot's cameras immediately before and after the collision. The central column visualizes the optical flow, the right column the clusters of similar optical flow, where each cluster has been marked with a distinct color. The darker area is occupied by hand and arm and therefore ignored. The white box marks the area next to the hand where we expect a possible collision to occur. If we find a cluster of optical flow that exists mostly within this area but not outside of it, this observation indicates that the hand collided with an object and caused it to move.

The image area that is checked for such an outstanding cluster is determined by taking the hand position, adding a translation into the direction into which the hand is currently moving, and projecting this point into the image. A quadratic area around that point which has roughly the size of the object is then analyzed[3]. For each cluster of similar optical flow c_i, we count the number n_i of pixels belonging to it in the whole image, and the number a_i of pixels belonging to it in the area in front of the hand. If for one of the clusters the ratio $\frac{a_i}{n_i}$ is more than 0.5, i.e. most of the pixels of the cluster occur inside that small area, this is a strong indication that this unique motion has been caused by an object that is being moved by the robot hand.

It is very probable (yet not certain) that the object will move in a similar way as the robot's hand and parts of its arm. Therefore the image area covered by hand and arm, which is determined based on the results of the hand tracking, is not taken into account when the values n_i and a_i are determined. Fig. 4 visualizes the optical flow, its clustering and the relevant image regions just before and during a collision.

Note that due to the restricted area in which we expect collisions to happen, individual motion in the background (which most of the time moves as a whole due to the camera's motion) can theoretically cause false collision detections, but only when it occurs within the image area next to the hand in the direction into which it is moving as described above.

[3]The size of the object in the image can be estimated from its model and the distance to the camera.

IV. CORRECTIVE REACTION

When the robot detects an unintended collision during the grasp execution, it should react in a way that allows to successfully complete the grasp. Optimally, the robot would have all relevant information about shape and pose of the hand and the object and could just re-plan a collision-free grasp trajectory. But obviously this information is not available, otherwise the collision wouldn't have occurred in the first place. Thus we have to use robust heuristics that can deal with incomplete and uncertain information and create a reaction that has a good chance of correcting the execution error that the robot committed.

A. *Collision Localization*

One piece of information that is necessary for a reasonable corrective reaction is which part of the hand collided with the object. A random change of the hand pose may sometimes be successful, but as shown in our experiments in section V the informed reaction strategies are clearly superior to random modifications of the grasp.

The information which part of the hand touched the object is immediately available when one uses tactile sensors, but if the collision was detected visually it has to be determined in another way. Although this depends on the kind of hand

that is used, one can assume that for a majority of grasping motions the fingers and in particular the fingertips are the primary causer of premature collisions. In the case of the conducted experiments, they are virtually always caused by the fingertips. We therefore obtain their positions from the hand localization and check which fingertip is closest to the object. This measurement is of course subject to errors in the perception of hand and object, but seemed to be always correct in our experiments.

B. Reaction Strategies

We implemented and evaluated different reaction strategies to correct the hand pose in the case of a premature collision. We limited ourselves to modifying the position and orientation of the whole hand, although we are aware that there are cases in which it would be necessary to correct the configuration of individual fingers.

The general reaction scheme is the same for all our proposed strategies: When a collision is detected, the hand retreats 2 cm into the direction that it came from with an absolutely straight motion to avoid disturbing the object any more. A corrective offset for the hand pose is calculated according to the respective strategy. The hand retreats another 2 cm during which half of the corrective offset is already applied, to make sure that the reaction has already taken effect before approaching the object again. The corrective offset is then permanently applied to the grasp pose definition. From that point on, the robot moves towards the intended grasping pose again as usual.

If the robot collides with the object again, another corrective reaction takes places. Thus, the corrective offsets add up, and the robot repeatedly tries to grasp and corrects the hand pose as often as necessary until the grasp is successful. In practice it would probably make sense that if the grasp doesn't succeed after a certain number of corrections, a totally different grasp is planned.

Within our reaction scheme, the key to a helpful correction is to determine an appropriate corrective offset. As a baseline, we implemented a strategy where the orientation of the hand is modified by a small random rotation. Such a purely exploratory approach would probably be the only possibility if there were no further information available about the collision, and there is a certain chance that the grasp will eventually succeed after one or more random modifications of the hand pose.

As in our case the information which finger collided with the object is available, we can determine a more constructive correction offset. The obviously useful kinds of motion are to either translate the hand into the direction of the finger that caused the collision, thus aligning the palm with the closest part of the object, or to rotate the hand such that the finger is turned away from the object, or a combination of both. We implemented all three variants and comparatively evaluate them in section V.

In the case of the hand of ARMAR-III, the thumb opposes the four other fingers, therefore we only need to distinguish whether the thumb or one of the other fingers collided with the object. Thus, our results can directly be transferred to simple grippers or precision grasps with two fingers. For more general hand configurations the reaction strategies have to be adapted to the individual hand geometry following the above principles.

TABLE I: Collision detection rate depending on the distance that the object has moved

2 mm	5 mm	10 mm
76 %	92 %	96 %

TABLE II: Collision detection rate depending on the angle between image plane and direction of movement (over a distance of 5 mm)

0°	45°	70°	90°
92 %	96 %	84 %	72 %

V. EXPERIMENTAL EVALUATION

We evaluated our approach on the humanoid robot ARMAR-III [18]. First, we tested the sensitivity of our visual collision detector by manually moving the object over a fixed distance while the robot hand was close to it. Table I shows the detection rate for the object to have moved, depending on the distance over which it moved. As can be seen, even if the object was shifted only by 2 mm, this is already detected most of the times, and when the translation is 5 mm or more the detection rate is clearly over 90%. The rare cases in which the object motion is still not detected when it moved more than 5 mm occur when less than half of the object lies within the observed area in front of the hand (see section III-C). We did not observe significant differences in the detection rate between the robot head being static or in motion, or when people were moving in the background.

One concern we had about our approach was whether it would be able to detect the object motion when it occured in the direction perpendicular to the image plane. As we are using optical flow, motions within the two dimensions spanned by the image plane should create the clearest signal, while e.g. a motion straight away from the camera would only cause the object to shrink in the image. Therefore, we tested moving the object by 5 mm in different angles relative to the image plane. The results can be found in table II, the experiment at an angle of approximately 0° is identical to the one for 5mm in table I. At 45° there seemed to be no difference, at an angle of around 70° the detection rate dropped slightly to 84%. This is the biggest angle that may occur in practice on our robot, as the tables etc. on which objects might be placed are lower than the cameras, therefore a motion on the table plane can never be exactly perpendicular to the image plane when the robot looks approximately towards the object. For the sake of completeness, we also tested a motion exactly into the depth direction and still obtained a reasonable detection rate of 72%.

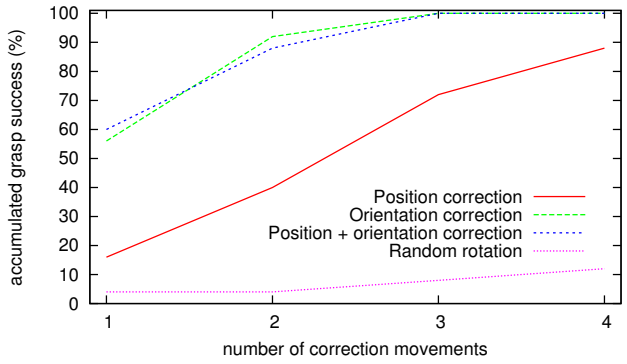

Fig. 5: Percentage of successful grasps after a certain number of correction movements, depending on the applied strategy. E.g., for the orientation correction strategy, in 56% of the cases one corrective motion was enough, another 36% of the grasps were successful after two corrections, and the remaining 8% of the attempts required three corrective movements.

In the actual grasping experiments, we had virtually no problems with undetected collisions. However, although the experiments above suggest that a very small motion is sufficient to detect the collision between hand and object, the objects were usually pushed a few centimeters. The two reasons for that are that our detection algorithm runs only at 2-3 frames per second on a standard PC due to the relatively high computational intensity of hand localization, optical flow calculation and clustering. More importantly, when a collision is detected, it takes some time until the hand actually stops moving forward. For these reasons, depending on the speed of the arm motion, the objects were usually pushed over 1-4 cm. Therefore, when for some reason the impact on the object position has to be kept minimal, the approach speed would need to be relatively slow, depending on the responsiveness of the used robot arm and, when this is very good, also on the available computational capacity.

Finally, we tested the performance of the overall system with the collision detection and the different reaction strategies we proposed in section IV-B[4]. We used five different test objects, and for each of them manually defined a grasp that seemed reasonable but failed in the real world when executed on the robot. We evaluated every reaction strategy by placing each of the five objects at five different reachable poses in front of the robot. We ignored grasp attempts that were immediately successful, so every strategy was tested with 25 grasps during which at least one collision with the object occurred.

Fig. 5 depicts the results of these trials. It shows how many of the grasping attempts had succeeded after a given maximal number of corrections. The baseline strategy where after a collision the hand hand pose was modified by a random rotation of 25° performs rather badly, as was to be expected. In only one case a single correction movement lead to a successful grasp, and another attempt succeeded after three and four corrections respectively.

The proposed strategies that take into account which finger seems to have caused the collision perform significantly better. The one where the position is modified by a translation of 25 mm towards the finger that caused the collision manages to successfully grasp the object after one correction in 16% of the cases, and in another 24% two corrective movements are sufficient. In 32% of the attempts three corrections were necessary. The overall success rate after at most four corrections is 88%, which is already quite an achievement regarding the fact that without the reactions all those grasps would have failed. In two of the remaining three cases the object was still grasped after further correction movements, but in one case it was finally pushed out of reach of the robot.

However, the two reactive strategies where the orientation of the hand was changed by 25° to turn the colliding finger away from the object, or the orientation changed by 15° and the position by 10 mm, turned out to be very successful. Both managed to grasp the object after one correction in about 60% of the attempts, and had an overall success rate of around 90% after one or two and 100% after at most three corrective movements.

We believe that the reason why the strategies that apply a rotational correction are so much more effective than the one that corrects only the position is that in our implementation of visual servoing, only the position of the hand is visually corrected, but its orientation is obtained from the forward kinematics of the robot. Thus, the orientation error during execution is much bigger than the position error. Additionally, when localizing an object, its position is usually determined more reliably than its orientation. Therefore, it is entirely possible that on other robotic platforms, or when the visual servoing can also correct the hand orientation, the comparison between the three strategies might turn out differently.

VI. CONCLUSIONS AND FUTURE WORK

We have presented an approach for visually detecting undesired premature collisions between a robot's hand and the object that is being grasped. The detection is based on analyzing the optical flow next to the hand in the direction into which it is moving, and detecting when the optical flow there is different from the rest of the scene, which indicates that the robot has caused the object to move.

We also proposed different strategies how to react to such a collision, which take into account an estimation of which finger has caused it and correct the hand position and/or orientation appropriately. Experimentally we showed that the detection works very reliably and that the proposed reaction strategies allow to correct a failed grasp attempt and virtually always conclude it successfully.

As the next step, we plan to complement this visual collision detector with classical tactile and force feedback sensors to cover both the cases in which the object is moved

[4]A video of the experiment is submitted with the paper, a high quality version can be found on https://www.youtube.com/watch?v=MkNIFWth5D4.

by the push and in which it resists it. How to combine these different sources in a constructive way to faster and more reliably detect collisions, and to determine the part of the hand that caused them, will be the central question here. This is of particular interest when executing more complex grasps or manipulative actions.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7 under grant agreement 270273 (Xperience).

REFERENCES

[1] J. Kim, K. Iwamoto, J. Kuffner, Y. Ota, and N. Pollard, "Physically based grasp quality evaluation under pose uncertainty," *IEEE Transactions on Robotics*, vol. 29, no. 6, pp. 1424–1439, 2013.

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

[3] G. Kootstra, M. Popovic, J. Jørgensen, K. Kuklinski, K. Miatliuk, D. Kragic, and N. Krüger, "Enabling grasping of unknown objects through a synergistic use of edge and surface information," *ijrr*, vol. 31, no. 10, pp. 1190–1213, 2012.

[4] D. Fischinger and M. Vincze, "Empty the basket - a shape based learning approach for grasping piles of unknown objects," in *iros*, 2012, pp. 2051–2057.

[5] J. Hill and W. Park, "Real time control of a robot with a mobile camera," in *The 9th International Symposium on Industrial Robots*, 1979, pp. 233–246.

[6] N. Vahrenkamp, S. Wieland, P. Azad, D. Gonzalez-Aguirre, T. Asfour, and R. Dillmann, "Visual servoing for humanoid grasping and manipulation tasks," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2008, pp. 406–412.

[7] J. Felip, J. Bernabe, and A. Morales, "Contact-based blind grasping of unknown objects," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2012, pp. 396–401.

[8] D. Schiebener, J. Schill, and T. Asfour, "Discovery, segmentation and reactive grasping of unknown objects," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2012, pp. 71–77.

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

[10] H. Dang and P. Allen, "Stable grasping under pose uncertainty using tactile feedback," *Autonomous Robots*, vol. 36, no. 4, pp. 309–330, 2014.

[11] K. Hsiao, P. Nangeroni, M. Huber, A. Saxena, and A. Y. Ng, "Reactive grasping using optical proximity sensors," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 2098–2105.

[12] J. Bernabe, J. Felip, A. P. del Pobil, and A. Morales, "Contact localization through robot and object motion from point clouds," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2013.

[13] G. Metta and P. Fitzpatrick, "Grounding vision through experimental manipulation," *Philosophical Transactions of the Royal Society: Mathematical, Physical and Engineering Sciences*, vol. 361, no. 1811, 2003.

[14] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Simultaneous grasp and motion planning," *IEEE Robotics and Automation Magazine*, vol. 19, no. 2, pp. 43–57, 2012.

[15] D. Chen, Z. Liu, and G. Wichert, "Grasping on the move: A generic arm-base coordinated grasping pipeline for mobile manipulation," in *European Conference on Mobile Robots (ECMR)*, 2013, pp. 349–354.

[16] T. Asfour, N. Vahrenkamp, D. Schiebener, M. Do, M. Przybylski, K. Welke, J. Schill, and R. Dillmann, "ARMAR-III: Advances in humanoid grasping and manipulation," *Journal of the Robotics Society of Japan*, vol. 31, no. 4, pp. 341–346, 2013.

[17] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[18] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "ARMAR-III: An integrated humanoid platform for sensory-motor control," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2006, pp. 169–175.

[19] G. Farneback, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis*, ser. Lecture Notes in Computer Science, J. Bigun and T. Gustavsson, Eds. Springer Berlin Heidelberg, 2003, vol. 2749, pp. 363–370.

[20] D. Pelleg and A. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in *Proc. 17th Int. Conf. Machine Learning*, San Francisco, CA, 2000, pp. 727–734.