A Novel 3D Camera Based Supervision System for Safe Human-Robot Interaction in the Operating Room

P. Nicolai, J. Raczkowsky, and H. Wörn
Karlsruhe Institute of Technology, Karlsruhe, Germany
Email: philip.nicolai@kit.edu

***Abstract*—In anticipation of upcoming technological advances in the operating room, it is necessary to already give thought to how humans and robots can safely interact and cooperate in the operating room of the future. In this paper, we present a supervision system, consisting of seven 3D cameras, and the according shape cropping algorithm, which allows verifying the correct setup of surgical robots, detecting potential collisions between robots and their surroundings as well as monitoring the correctness of the robots' motions. The system has already been successfully implemented, set up and evaluated.**

***Index Terms*—robot assisted surgery, surgical robots, human-machine interaction, scene supervision**

I. Introduction

The most well-known surgical robotic system today is the DaVinci® system by Intuitive Surgical [1], which is used for Minimally Invasive Robotic Surgery (MIRS). It consists of a patientside cart including multiple robotic arms that each holds a surgical tool or the endoscope. The medical instruments and the endoscope are teleoperated from the surgeon console. Seen from the technological point of view, the Da Vinci system is rather limited: It is large and heavy, there is no internal or external collision detection for the robotic arms and – by design – no possibility for autonomous motions. As the advances in robotic research since the introduction of the first DaVinci system are manifold, there are many possibilities for upcoming new surgical robotic systems to overcome these limitations. One example for a system that aims to compete against the Da Vinci is the MiroSurge system by German Aerospace Institute (DLR) [2]. It features three small robotic arms which can be attached to the operating room (OR) table, provide force-torque sensing and are fully actuated.

In the dawn of such new robotic surgical systems, it is necessary to already give thought to how humans and robots can safely interact and cooperate in the OR of the future: If robots and surgeons interact close to each other or even share the same workspace, do new safety hazards arise? How can riskless interaction be guaranteed? Can the robots' performance be monitored by an independent system? In this work, we present a supervision system based on 3D Time-of-Flight (ToF) cameras for the OR that aims to provide answers to these questions. The system is part of the OP:Sense research platform for new concepts in surgical robotics [3]. OP: Sense also integrates other camera subsystems for separate objectives that are not in the scope of this work.

The presented supervision system is not targeting one specific surgical robotic system. Instead, concepts are researched and implemented that can be applied to a wide range of robotic arms. While full integration has been performed with the DLR MiroSurge, using MIRO robots, and the KIT OP:Sense research platform, using KUKA LWR4 robots, the system is open to adaption to other systems based on single robotic arms. Examples for such systems are upcoming systems like the SOFIE system for minimally invasive surgery developed at the University of Eindhoven [4].

II. System Description

A. Hardware Setup

We use multiple ToF cameras to acquire 3D views of the scene. These cameras send out short, modulated pulses of infrared light which is reflected by the scene. After the reflections are captured by the cameras, a depth map can be calculated based on the phase shift between the original signal and the reflection [5].

The 3D camera system consists of six PMD S3 cameras with a resolution of 64 px x48 px which are ceiling-mounted around the region of interest (the OR table). In addition, one PMD CamCube 2.0 with a higher resolution of 204 px x 204 px is mounted directly above the situs. This relatively high number of cameras is based on the observation that persons who perform tasks at and around the OR table regularly block the field of view of one or more of the low resolution cameras. In the realized

Manuscript received September 3, 2014; revised December 1, 2014.
This work was funded by the European Commission's Seventh Framework program within the projects 'Patient Safety in Robotic Surgery (SAFROS)' under grant no. 248960 and 'Active Constraints Technologies for Ill-defined or Volatile Environments (ACTIVE)' under grant no. 270460.

doi: 10.12720/joace.3.5.410-417

spatial configuration, this is mostly compensated for by the overlapping fields of view of the cameras.

The PMD S3 cameras are connected to a dedicated 100 MBit network switch with a 1 GBit uplink to the server which performs scene reconstruction and analysis. The PMD CamCube is connected directly via USB.

Due to the low resolution of the cameras, traditional methods for extrinsic calibration between the cameras such as Zhang [6] are not applicable. Instead, the cameras are registered against an external optical tracking system. A custom registration pattern has been designed which includes large geometrical features with high contrast that are identifiable in the grayscale images provided by the PMD cameras. In addition, the pattern is equipped with retroreflective marker spheres so it can be tracked by the optical tracking system. While we are using an A.R.T. tracking system in a six camera configuration, registration could also be performed against other optical tracking systems that work with the standard NDI marker spheres.

In the OP:Sense setup, two light-weight robots LWR4 from KUKA are integrated which can be controlled by haptic input devices (for telemanipulation), by hand or in autonomous mode. The robots are mounted to a standard OR table with a special mount, resulting in a configuration which is similar to mounting them at the OR table rails. Standard laparoscopic instruments have been adapted and motorized for use with the robots. All experiments presented in this work have been conducted using this setup. In addition, the 3D camera system has been successfully integrated with the DLR MiroSurge System with three MIRO robots, but no qualitative measurements could be taken in this configuration due to time constraints.

The high-speed 2D camera "SpeedCam MacroVis" has been used in order to measure the delay of the camera system.

B. Software Implementation

For scene reconstruction and analysis the Point Cloud Library (PCL) [7] is used. The Robot Operating System (ROS) [8] serves as the communication framework which handles all data transfers of the point clouds acquired by the cameras.

As ToF cameras need to be synchronized in order to prevent crosstalk and erroneous measurements, a triggering scheme has been implemented. The PMD S3 cameras are triggered time-multiplexed in two groups of three cameras each. In each group, cameras are set to use different modulation frequencies. The PMD CamCube is triggered in between the PMD S3 groups.

The actual pose of the robot based on its internal sensors is used continuously in the Shape Cropping algorithm. It can be provided to the supervision system in the form of full poses for all links via remote procedure calls or be calculated by the supervision system itself based on the kinematic model of the robot (Denavit-Hartenberg parameters) and its current joint values.

We pre-process all data provided by the 3D cameras only by filtering potential "flying pixels". Flying pixels result from erroneous measurements which incorporate sensor readings from different distances, resulting in noisy measurements. As they occur at the edges of objects, we apply edge detection filters to the depth map to detect and remove potential flying pixels as shown in Fig. 1.

Figure 1. Filter chain for flying pixel removal; left: output of combined Sobel filter applied to depth map; *center*: binary filtering mask based on Sobel output; right: final depth map after filtering

After filtering flying pixels, the point clouds of all cameras are transformed into a common coordinate frame and merged into one point cloud. As this point cloud now consists of all valid measurements of the scene taken by the different cameras at one point in time, we will further call it the virtual scene.

III. Safe Zone Concept and Shape Cropping

The virtual scene, which has been set up as described above, can now be analysed for different potential safety risks. For this purpose, a virtual safe zone is constructed around each robot. The safe zone represents a volume around the robot in which no object or person should be present because of the risk of an immediate collision. Fig. 2 shows a schematic view of the safe zone in detail (left) as well applied to the whole robot (right). In both views, the inner green segment corresponds to the robot outer hull. All points inside this hull, displayed in light green in the left figure, are considered part of the robot surface itself and therefore non-colliding points. The red segment corresponds to the safe zone around the robots; points inside this region are considered to belong to objects which might soon cause a collision. The grey points outside the safe zone correspond to objects that do not currently pose a risk of collision. For all experiments presented here, the "thickness" of the safe zone was set to 10 cm.

The safe zone concept is implemented using the so-called Shape Cropping algorithm (see Fig. 3). The idea is closely related to popular collision tests such as axis aligned bounding boxes or oriented bounding boxes. While these are mostly used for pure virtual collision testing, e.g. CAD-model against CAD-model, Shape Cropping adapts the concept to using real world data, e.g. live data from PMD cameras. As Shape Cropping deliberately uses only simple geometrical constraints, it is applicable to all kind of devices that provide 3D images. It does not require a high spatial resolution for feature detection or RGB information for colour-based segmentation. This is especially important in the OR where robots will likely be covered in sterile drapes, making their shape more indistinct.

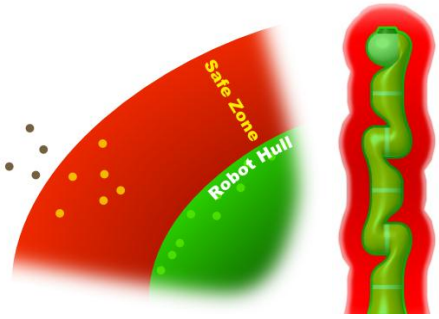

Figure 2. Left: Schematic view of the safe zone; right: Illustration of the safe zone concept applied to the whole robot

For each link (starting by the robot basis):

1. *Calculate segment pose and crop bounding box from full scene*
2. *Crop outer shape of safe zone from resulting data*
3. *Segment inner robot shape from resulting data*
4. *Analyze contents of safe zone and inner robot shape, store number of inliers and outliers for current segment*
5. *Merge all points not inside the safe zone back to scene*

Calculate evaluation criteria over all segments: total number of outliers, ratio of outliers to inliers, average number of outliers per segment

Figure 3. High-level overview of the shape cropping algorithm

IV. Safety Features

In addition to the filtering of flying pixels, the output of each camera is continuously checked for consistency with the sensor parameters. Examples for possible inconsistencies are measurements outside the range of the sensor specifications or discrepancies in the real and expected delay between triggering the camera and arrival of the acquired image. If an inconsistency is detected, the measurements of the according camera are excluded from the virtual scene and therefore not used for any subsequent calculations.

A. Robot Localization for Setup Verification

The available workspace of robotic arms in a fix environment depends on the kinematic properties of the robot (e.g. number and layout of joints), obstacles present as well as the position of the robot in the scene. In laparoscopic procedures, a careful pre-operative planning of the trocar points and the according robots' positions can significantly improve the fidelity of the instrument tips inside the patient. In contrast, bad positioning of the robot in relation to the trocars can prevent the surgeon from performing the desired tasks due to lack of reachability and so–in the worst case–require repositioning of a robotic arm during the intervention.

In order to verify the correct positioning of the robotic arms according to a pre-operative plan, we propose an automatic localization of all robots present at the OR table to recognize an incorrect setup before the procedure is started.

For this reason, we have developed and integrated two different methods for locating the robot in the virtual scene: A passive localization approach without movement of the robotic arms which relies on known scene geometry and an active localization approach for which each robotic arm has to perform a motion sequence. In both cases, the initial detection is later refined by an optimization procedure. Both methods can be used with any kind of 3D camera, provided that the scene is acquired from distributed points of view.

B. Passive Localization

The first method is based on known scene geometry which can be employed to minimize the search space in which the robots can be located. In the current implementation, it is applicable for robots that are attached to the OR table such as the DLR MiroSurge or the OP:Sense research platform. The detection of the base position of the robot is performed in three steps:

1) Localization of a landmark

Using the shared scene representation based on all 3D cameras (virtual scene), the landmark is detected using an applicable method. In this work, we use the OR table as a landmark. The implementation with which the experiments were carried out under the assumption that, in order to yield the best visibility of the situs, the OR table is placed axis-aligned to the camera system. The system has since been be extended to allow for arbitrary orientations of the OR table.

2) Segmentation of region of interest

Based on the pose of the landmark determined in step one, the regions of interest (ROI) on are segmented from the scene. As both the MiroSurge system and OP:Sense system use robots which are mounted to the sides of the OR table, the regions of interest span about 20 cm on either side of the table.

3) Localization of robot base

To identify the robot bases in the ROIs segmented in step two, Euclidean Clustering based on Kd-trees is performed on both regions (see [9]). The extracted clusters are projected onto the XY-plane of the OR-table and analysed for instances of the cross-section of the outline of the robot base link. Based on the reprojection and the known height-difference between OR table and robot base, the robot base pose is calculated. This step is repeated for all extracted clusters.

C. Active Localization

For scenarios in which no information/landmark about the location of the robot in the scene is available, we developed an active localization method which is based on spatial change detection. Spatial change detection is a technique already applied in various fields [10]-[12]. This localization method requires that there is a method of controlling the robot by commands sent from the supervision system. No real-time or full control is necessary; a predefined motion which can be started by the supervision system is sufficient. In the following it is assumed that the robot arms start in their home position,

assumed to be (0 °, 0 °, 0 °, 0 °, 0 °, 0 °[, 0 °]) in joint space, which corresponds to being in an upright pose. If a specific joint is mentioned, this corresponds to the LWR4.

1) Spatial change detection of moving robot

The robot performs motions of 7.5 °in its second joint in the range of [-22.5 °, 22.5 °]. After each motion, a new octree representation of the scene is calculated and compared to the previous one. The differences are added to a point cloud which stores the spatial change of the scene.

2) Localization of robot base

Using principal component analysis (PCA), the two major components of the accumulated point cloud are calculated. The first component corresponds to the upward axis along the robot in its home position. Based on the shape of the accumulated point cloud, the known geometric properties of the robot and the location of the accumulated point cloud, the position of the robot can be estimated.

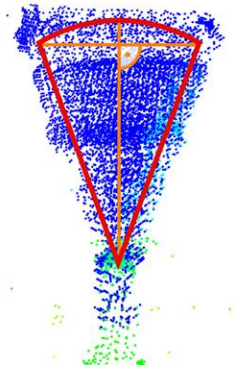

Figure 4. Point cloud of spatial change detection for robot localization

Fig. 4 shows the resulting point cloud overlaid with the geometric properties of the robot's motion (red) and the first two principal components (orange).

D. Localization Optimization

Both localization methods detect the robot using coarse spatial measurements. To refine the pose acquired by either of these methods, the Shape Cropping algorithm can be used to transform the localization into an optimization problem. Therefore we define the following quality criteria based on the number of points of the virtual scene that are inside the robot shape (inliers) or the safe zone (outliers): the absolute number of outliers for the whole robot, the ratio between outliers and inliers and the average number of outliers per segment. To find the optimal position of the robot, an inverse hill-climbing algorithm is used which evaluates possible robot locations inside a fixed range against these criteria.

E. Collision Estimation

To ensure that the surgical personnel and the robot can safely operate in the same workspace, possible collisions between the robot and the surgeon have to be detected in advance and, if possible, be avoided. Using the supervision system, a safe zone around the robot can be constructed as described above. By detecting intrusions into this zone before actual collisions occur, the supervision system can issue a warning or alter the robot's motion path to avoid the collision.

F. Static and Dynamic Collision Checks

When performing collision checks, we differentiate between static and dynamic configurations of the robots. Situations in which the robot is not moving or is only performing slow and/or small motions are regarded as static configurations. An example is a robot arm equipped with a laparoscopic instrument that performs only relatively small motions most of the time. If a robot performs faster motions covering a larger distance, we call it a dynamic configuration. One example is an autonomous motion to a predefined target pose, e.g. in order to position laparoscopic instruments close to a trocar for subsequent hand-guided insertion by the surgeon.

The Shape Cropping algorithm used for calculating the safe zone is based on the current pose of the robot which geometrically consists of the poses of the different segments of the robot. These are calculated based on the current joint values of the robot and therefore provide a real-time representation of the robot pose. While the Shape Cropping algorithm could be applied directly to static configurations, this is not feasible for dynamic configurations because the virtual scene is slightly delayed compared to the real environment. If the latest available robot pose is used for the collision check in combination with the latest available virtual scene, data from two different time steps are compared to each other. The robot sensed by the camera would "trail behind" the real robot, leading to an incorrect segmentation of the robot from the virtual scene and in turn resulting in violations of the safe zone.

This is solved by introducing a ring buffer in the internal robot representation. It stores the received or calculated segment poses for an amount of time equal to the delay of the PMD cameras. As all data in the supervision system feature a timestamp, which is set to the acquisition time of the camera in case of point cloud messages, we can perform Shape Cropping based on consistent data: The latest virtual scene (for which the safety zone shall be computed), the pose of the robot hull from the same time, and the pose of the safety hull based on the latest segment pose data. This way, the safety zone is moving slightly in front of the robot while the robot hull correctly segments the "delayed" robot as sensed by the PMD cameras.

G. Supervision of Robot Pose Correctness

As the method for collision estimation takes into account information from both the internal sensors of the robot and the 3D scene representation, it inherently performs a consistency check between these data sources. If the segment poses are reported incorrectly by the joint encoders, the robot hull calculated by the Shape Cropping algorithm does not match the real shape of the robot. This results in numerous violations of the safe zone: The robot calculated based on incorrect segment poses virtually "collides" with the robot as sensed by the PMD cameras.

V. Experiments and Results

A. Camera System

1) Scene update rate

Table I shows the results of analyzing the achieved update rate of the full scene as well as the PMD S3 groups. While the minimal acquisition time and the maximal acquisition time are heavily influenced by the first measurements, the standard deviation of less than one millisecond over 5.000 respectively 10.000 measurements shows the accuracy of the camera system timing.

TABLE I. Frame Rate Analysis for PMD Camera Subsystem

	Full scene	S3 group
Frame rate	18.25 fps	9.12 fps
Minimal acquisition time	23 ms	78 ms
Maximal acquisition time	85 ms	141 ms
Standard deviation	0.9 ms	0.7 ms
# of frames analyzed	10.000	5.006

2) Delay of the camera system

To accurately determine the camera delay, a high-speed camera (SpeedCam MacroVis) was used that filmed both the real scene (with distinctive events performed) and the virtual scene displayed on a screen. In order to prevent synchronization problems, the scene was captured with a 240 Hz which is a multiple of the computer screen update rate of 120 Hz. Several sequences were captured by the high-speed camera with timestamps imprinted in every frame. This allowed for parsing the video stream afterwards and annotating the times where corresponding events happen in both scene representations.

As the PMD cameras are triggered in two groups, two delays have been annotated for each event: the time when the event was observed by the first camera group and the time when the event was observed by the second camera group. The average delay before an event is picked first was measured to be 175 ms while the second measurement delay was 244 ms. This corresponds to the frame rate of the S3 cameras which take approximately 70 ms for acquiring a new frame. In comparison, the delay of a distributed camera system with multiple MS Kinect 3D cameras (which is also integrated into the OP:Sense platform) was measured to be 966 ms.

3) Robustness

To assess the robustness of the system to technical faults, adverse events concerning the connectivity of the cameras were simulated and the response of the system observed. As all cameras are controlled by separate threads and operate independently of each other, single PMD cameras can be physically disconnected without influencing the functionality of the other cameras. Even disconnection of multiple PMD S3 cameras during operation did not affect the performance and frame rate of the remaining ones. If a camera is disconnected and does not transmit new information, the previous information is removed from the scene immediately and not used for further processing. The same result has been obtained by disconnecting the PMD CamCube which is connected by USB.

As the connection to the PMD S3 cameras as based on TCP/IP with a fix IP address configuration for each camera, it is even possible to disconnect a camera by pulling the plug, observe the subsequent removal of the respective data from the virtual scene, and reconnecting the camera. After an initialization delay of about 2 seconds, the camera restarts operation and is subsequently included in the virtual scene again.

B. Supervision of Robot Pose Correctness

1) Robot localization

For evaluation of the accuracy of the robot localization by PMD cameras, a ground truth was established by localizing the robot's base pose with a tracked pointer each time the robot was moved. In order to assess the repeatability, each localization was performed two times. If a significant difference was measured, three additional measurements were taken at the same position.

2) Passive localization

To evaluate the maximum achievable accuracy of the passive localization, the OR table was placed in different positions in which the robot's base was partially visible by the PMD CamCube located above the OR table. As all tests were performed with a grid size of 0.5 cm, the theoretical upper limit for accuracy was 0.25 cm per axis. Table II shows the results of the measurements, split into the error along each (global) axis as well as total distance between real and measured location. As can be seen, the passive localization can achieve an accuracy of up to 9.5 mm. The small average error on the x-axis is explained by the fact that in the experimental setup, the x-axis corresponds to the long sides of the OR table (to which the robots are attached), which is optimally covered by multiple PMD cameras.

TABLE II. Accuracy of Passive Robot Localization

	avg. error	min. error	max. error
x-axis	0.28 cm	0.1 cm	0.8 cm
y-axis	0.75 cm	0.1 cm	1.3 cm
z-axis	0.63 cm	0.4 cm	0.9 cm
Euclidean distance to correct location	1.12 cm	0.95 cm	1.58 cm

3) Passive localization

The same methodology as above was used for measurement of the accuracy of the active localization. Table III shows the results achieved by active robot localization. As can be expected, the error range without optimization is significantly higher than in the case of passive localization, due to the fact that active localization uses no a-priori knowledge, e.g. landmarks.

TABLE III. Accuracy of Active Robot Localization

	avg. error	min. error	max. error
x-axis	1.32 cm	0.2 cm	3.0 cm
y-axis	0.54 cm	0.0 cm	1.6 cm
z-axis	3.37 cm	0.4 cm	8.7 cm
Euclidean distance to correct location	3.97 cm	1.02cm	8.8 cm

4) Localization optimization by shape cropping

The Shape Cropping algorithm can be used to further refine the results of passive and active robot localization. For the experiments, localization optimization was performed with three iterations with different ranges (9 cm / 3 cm / 1 cm) and different step sizes (3 cm / 1 cm / 0.3 cm).

Figure 5. Different configurations of the robot used for evaluation of the Shape Cropping position optimization, left to right: "elbow" joint upright (configuration I), "elbow" joint in a 45 °position (configuration II) and "elbow" joint angled sideways (configuration III)

For evaluation of the optimization quality, the real position of the robot was again determined using the A.R.T. optical marker-based tracking system and stored as the ground truth vector t_g. The initial guess was simulated by adding random distances between -10 cm and 10 cm to each element of t_g. For each of three different exemplary configurations of the robot (see Fig. 5), 50 initial guesses were created and optimized using the quality criteria defined above.

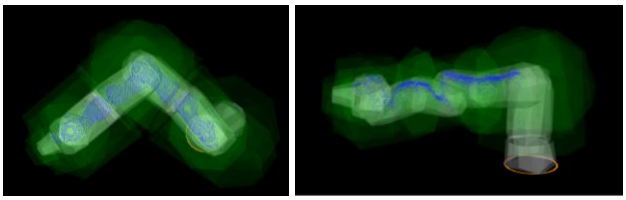

Figure 6. Top and side view of the robot in the first configuration with robot hull in grey surrounded by the green safe zone and inliers displayed in blue

Fig. 6 shows an optimal localization in which all acquired 3D points belonging to the robot are inside located the robot hull. While this is the goal of the optimization criteria, it is not feasible in most situations due to reflections and camera noise.

TABLE IV. RESULTS OF APPLYING THE LOCALIZATION OPTIMIZATION TO RANDOMLY DISTRIBUTED SIMULATED INITIAL GUESSES FOR THREE DIFFERENT ROBOT CONFIGURATIONS

	avg. error of initial guess	avg. error of final solution	avg. improvement
Configuration I	6.07 cm	1.23 cm	4.84 cm
Configuration II	5.65 cm	0.57 cm	5.09 cm
Configuration III	5.82 cm	1.47 cm	4.35 cm

Table IV shows the obtained results. The average improvement is ~ 4.8 cm with a maximum improvement of between 12.8 cm and 14.4 cm (depending on configuration). While the overall accuracy is similar to the one achieved by passive localization, it clearly surpasses the results of performing active localization without further refinements. For this reason, the supervision system optimizes robot poses which were determined by active localization as default.

5) Consistency checks

To evaluate the sensitivity of the consistency check for the robot pose, the LWR4 was positioned in a non-colliding pose without any objects close by. The streaming of joint data was interrupted so the robot hull and safe zone stayed in a fixed pose (corresponding to the last received joint values). For each joint, the robot was moved until a collision was detected. The delta between the original and the colliding joint value was annotated and the joint positioned back to its original value before testing the next joint.

As expected, variations of the first joints were detected quickly: Collisions were detected after rotating either of the first two joints by 2 °. This is due to the fact that small variations in the first joints cause big variations in the position of the end effector due to the law of the lever. In contrast, differences in the last joints were only detected from 7 ° upwards. Fig. 7 shows the detection of an incorrectly calibrated robot joint. Naturally, the precise results will differ between different types of robots based on their kinematics.

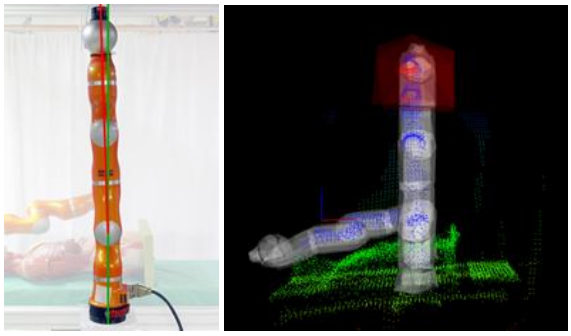

Figure 7. Inconsistency between joint angles and real position of the robot; left: view of the real robot with a deviation of 2 °in the second joint, right: inconsistency detected by collision check

C. Collision Detection and Avoidance

1) Static configuration

Evaluation of the collision detection between the non-moving robot and an object entering the safe hull around the robot was performed using two different parameters. The distance between robot and object at which the incoming collision was detected was annotated as well as the time between detection of a potential collision and the collision itself. The high-speed camera was used again to measure the time delay; detected collisions were signaled by switching the color of the outer hull of the colliding segment to red.

In a first set of experiments, the robot was approached with a hand moving in at approximately 20 cm/s. This corresponds to a careful movement, e.g. if the person is aware of possible, unexpected motion events in close proximity or is performing a delicate task. In 15 events, the average time between detection of the impeding collision and the hand touching the robot was 460 ms with an average distance of 7.5 cm between the hand and the robot.

Fig. 8 depicts a scene in which the robot is touched by a hand and a collision is detected.

In a second set of experiments, a far higher hand movement speed of about 90 cm/s was employed to test the systems response to motions of average and higher speed. This speed relates to quick, purposeful motions in which a person aims to e.g. hand something over to another person. Again, 15 events were annotated and analyzed for timing and distance. As a result, 14 out of 15 times the violation was detected, but only after the hand arrived at the robots surface. On average, the delay between the collision and the detection was 236 ms.

As a result, the system's ability to detect collisions by incoming objects depends on the approach speed of the object. For slow movements, such as careful motions or involuntarily shifting of the body closer to the robot, this allows to apprehend possible collisions and act accordingly.

2) Dynamic Configuration

In order to test the ability to stop a moving robot before it collides with an obstacle, a fixed trajectory was set which the robot executed continuously with a Cartesian speed at the end effector of 15 cm/s. If a violation of the safe zone around the robot was detected, the robot was automatically stopped by a signal from the supervision system and the distance between the object and the robot was measured. Objects positioned in the robot's path included a human hand, plastic boxes and OP-cloth.

In all test cases, the collision was detected successfully. The average measured distance amounts to 7 cm. This safety margin is high enough to allow different strategies for reacting to the sensed potential collision, e.g to not stop the motion abruptly, but gradually slowing down and still stopping before a collision occurred.

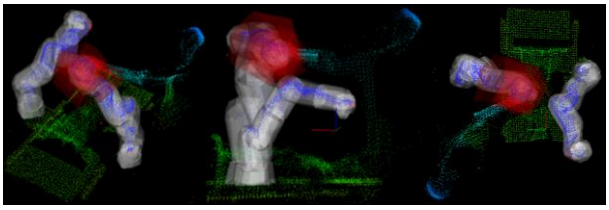

Figure 8. Different views of a person touching the end effector of a robot. White shapes: inner robot hull, dark blue points: inliers of the robots, red shapes: safe zone of the colliding segments

VI. Discussion

In this work, we have presented a 3D camera supervision system for safe usage of robots in the OR and human-robot cooperation. The fundamental idea is the construction of a safe zone around the robot, which is implemented by Shape Cropping and enables various safety features.

The achieved results show that the camera system achieves an update rate of 18 Hz with a very small delay and proves robust against technical failures of single components. The system achieves a precise localization of the robots, which enables an automated verification of the correct robotic positioning before the intervention starts. When a robot is moving, impeding collisions can be estimated and avoided. Faults of the robotic system such as miscalibration or incorrect movements can be detected.

As the system is designed in a modular fashion, it is easily upgradeable with new components. Up-to-date 3D cameras such as the Kinect 2 or the Bluetechnix Argos 3D P100 offer a significantly higher resolution and/or a higher frame than the PMD cameras which are currently integrated in the system. Integration of these new cameras is expected to boost the systems' performance by further increasing the accuracy and decreasing the delay of the camera system. This would be beneficial for collision estimation with fast-moving objects, but could also enable tracking of medical instruments which is currently not feasible due to the low resolution.

Apart from technical upgrades of the cameras, we are currently extending the system with Augmented Reality capabilities in order to project information such as the current safe zone directly into the scene. Furthermore, the virtual scene data will be integrated into the trajectory planning of the OP:Sense platform. We also aim to implement Dynamic Shape Cropping, which adapts the shape of the safe zone depending on the current motion and velocity of the robot.

References

[1] G. Guthart and J. Jr. Salisbury, "The intuitive telesurgery system: Overview and Application," in *Proc. IEEE International Conference on Robotics and Automation*, 2000, pp. 618–621.

[2] U. Hagn, *et al.*, "DLR MiroSurge: A versatile System for research in endoscopic Telesurgery," *International Journal of Computer Assisted Radiology and Surgery*, pp. 183-193, vol. 5, 2010.

[3] P. Nicolai, *et al.*, "The OP: Sense surgical robotics platform: First feasibility studies and current research," *International Journal of Computer Assisted Radiology and Surgery*, pp. 136-137, vol. 8, 2013.

[4] L. J. M. Bedem, "Realization of a demonstrator slave for robotic minimally invasive surgery," Ph. D. dissertation, Univ. of Eindhoven, Netherlands, 2010.

[5] A. Kolb, E. Barth, and R. Koch, "ToF-sensors: New dimensions for realism and interactivity," presented at the IEEE Workshop on Computer Vision and Pattern Recognition, Anchorage, AK, June 23-28, 2008.

[6] Z. Zhang, "A flexible new technique for camera calibration," in *IEEE Transactions on Pattern Analysis* and *Machine Intelligence*, IEEE Press, vol. 22, 2000, pp. 1330-1334.

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

[8] M. Quigley, *et al.*, "ROS: An open-source robot operating system," presented at the IEEE International Conference on Robotics and Automation Workshop on Open Source Software, Kobe, JP, May 12-17, 2009.

[9] R. B. Rusu, "Semantic 3D object maps for everyday manipulation in human living environments," Ph.D. dissertation, Univ. of Munich, Germany, 2010.

[10] M. Törmä, P. Härmä, and E. Järvenpää, "Change detection using spatial data problems and challenges," in *Proc. IEEE International Symposium on Geoscience and Remote Sensing*, 2007, pp. 1947–1950.

[11] G. Wu, H. Chen, and L. Cao, "Temporal and spatial change detection for scientific data set stream," in *Proc. IEEE International Conference on Audio Language and Image Processing*, 2010, pp. 618-622.

[12] H. Andreasson, M. Magnusson, and A. Lilienthal, "Has something changed here? Autonomous difference detection for security patrol robots," in *Proc. IEEE International Conference on Intelligent Robots and Systems*, 2007, pp. 3429-3435.

P. Nicolai was born in Heidelberg, Germany in 1983 and holds a diploma in informatics by the Universität Karlsruhe (TH), Germany, since 2010. He is currently working as Member of Scientific Staff of the Intelligent Process Control and Robotics Lab (IPR), Institute for Anthropomatics and Robotics (IAR), Karlsruhe Institute of Technology (KIT) in Karlsruhe, Germany. He was leading researcher on operation room supervision in the European project SAFROS and is currently working on his Ph.D. thesis in the field of safe robot assisted surgery. Mr. Nicolai has been awarded the best paper award at the conference of the German association for computer- and robot-aided surgery (CURAC) in 2010 and was finalist for the best video award at the IEEE International conference on Robotics and Biomimetics 2011.

Dr. J. Raczkowsky is academic director of the medical research group at the Intelligent Process Control and Robotics Lab (IPR), Institute for Anthropomatics and Robotics (IAR), Karlsruhe Institute of Technology (KIT) in Karlsruhe.

Prof. H. Wörn is chair of the Intelligent Process Control and Robotics Lab (IPR), Institute for Anthropomatics and Robotics (IAR), Karlsruhe Institute of Technology (KIT) in Karlsruhe.