Multi Kinect People Detection for Intuitive and Safe Human Robot Cooperation in the Operating Room

Tim Beyl
Philip Nicolai
Jörg Raczkowsky
and Heinz Wörn
Institute for Process Control and Robotics
Karlsruhe Institute of Technology
Karlsruhe, Germany
Email: tim.beyl@kit.edu

Mirko D. Comparetti
and Elena De Momi
Politecnico di Milano
Department of Electronics, Information and Bioengineering
NeuroEngineering and medicAl Robotics Laboratory
Milan, Italy

***Abstract*—Microsoft Kinect cameras are widely used in robotics. The cameras can be mounted either to the robot itself (in case of mobile robotics) or can be placed where they have a good view on robots and/or humans. The use of cameras in the surgical operating room adds additional complexity in placing the cameras and adds the necessity of coping with a highly uncontrolled environment with occlusions and unknown objects. In this paper we present an approach that accurately detects humans using multiple Kinect cameras. Experiments were performed to show that our approach is robust to interference, noise and occlusions. It provides a good detection and identification rate of the user which is crucial for safe human robot cooperation**

I. Introduction

In the past years medical robotics has reached a point where minimally invasive tele-operated systems are not just enough. Current robotics research focuses more on flexible assistant systems, flexible robotics or semi-autonomous systems. Those systems have in common that the usage has to be as intuitive as possible. A future goal in research should be to have robots that the surgeon and the personnel does not have to pay direct attention to. More specific, the system has to behave like the surgeon expects it to behave, yet to ensure a safe usage of the system especially in the case conventional serial or parallel robots are used.

A Common advantage of these systems such as the DLR Miro [1], the LARS robot [2] and research systems using the KUKA LWR 4 [3] is the applicable workspace and high payload of these robots. This offers new applications like Ultrasound probe steering, open surgery with increased accuracy and a applicable workspace, orthopedics interventions using milling devices and saws, as well as the field of minimally invasive robotic surgery that is widely explored by Intuitive Surgical [4] and the University of Washington [5]. However, compared to specially designed kinematics that move only around a remote center of motion, these kinds of robots need to perform larger movements in order to move the tooltip to the desired position. Requirements like remote center of motion for minimally invasive robot surgery can be solved in hardware with special designed robots. Also with special designed robots, safety systems are less important due to smaller movements. Systems using multi-purpose robots have to cope with this issues in software using special supervision systems.

978-1-4799-2722-7/13/$31.00

Our approach aims at solving the problem of safety with multi purpose robots inside of the operating room. Moreover it offers the base for a comprehensive operating room supervision that can also be used for specially designed robots in order to improve the intuitiveness and the efficiency of the system. Use cases of the system are simple high speed collision avoidance and on-line path planning based on low resolution point clouds, workflow detection focused on the operating room environment, probabilistic and rule-based inference from the perceived situation and workflow based control of the operating room and the surgical robots.

To the best of the authors knowledge there is no publication about similar approaches yet. However the problem of multi depth camera room supervision for other applications is tackled by [6] who developed a multi-kinect system for interaction with the environment and the past self, [7] who are focusing on the "The room is the computer" and the "Body as display" approach as well as [8] who are tackling the problem of interference induced by using multiple structured light Primesense (Primesense, Tel-Aviv, Israel) devices. Other approaches like [9] and [10] focus on multi-Kinect people tracking and dynamic scene reconstruction from asynchronous depth cameras.

Our system is composed of multiple Photonic Mixer Devices that are dedicated to low-latency high speed scene supervision. These cameras are used for collision avoidance. Additionally four Microsoft Kinect cameras are used that offer larger resolution having the drawback of more latency introduced into the processing chain. Our supervision system is developed in the context of OP:Sense [11] which is mainly dedicated to safe and intuitive workflow controlled human robot interaction in the operating room as well as exploring new applications for the operation and has recently been ported to ROS [12] and OROCOS [13]

The following article describes how we tackled the processing of the high amount of data coming from the Kinect cameras, registration issues between the cameras and describes our approach to find corresponding users in the camera's field

of view. Finally an approach to detect robots in the scene and a CUDA [14] based distance calculation for point clouds is described.

II. Framework

The OP:Sense system is composed of two KUKA lightweight robots (KUKA, Augsburg, Germany) that form the core of the project, a ceiling mounted camera rig holding an ART tracking system, PMDtec (PMD technologies, Siegen, Germany) as well as Kinect cameras, custom built attachable surgical instruments, milling devices, an ultrasound device, a high precision Stäubli (Stäubli Internationl AG, Freienbach, Switzerland) RX90 robot, a special endoscope steering robot and an endoscope. The software framework is based on OROCOS for low level real time robot tasks such as inverse kinematics computation or interpolation, as well as ROS for data processing, acquisition and processing. On top of that, a custom built framework based on a java implementation for ROS is being developed in order to cope with probabilistic, rule- and workflow-based control of the complete system.

Every Kinect as described in [15] acquires 3-channel 8 Bit RGB images with a resolution of 8 bit per channel as well as a 11 Bit depth map with the same resolution. The capturing frequency is 30Hz which results in a data rate of around 40MB/s This requires the use of a dedicated USB host controller per Kinect. In order to build up a scalable supervision system with a flexible number of camera, we outsourced the data capturing to small AMD based PCs that provide the images from up to two connected Kinect cameras via Ethernet and ROS. This allows for an theoretically unlimited amount of used cameras. The proposed algorithm runs on-line with up to four cameras when using an Intel Core i7 3770.

The Kinect cameras and the processing chain introduce a notable time delay described in results into the system. In order to cope with fast movements and time critical tasks such as collision detection and avoidance the system is supported through low delay, high speed PMD cameras. Other objects that are too small for being tracked via a marker less systems, such as surgical instruments, can be equipped with optical markers tracked via an ARTtrack system.

The coordinate frames of the independent camera systems are registered to each other to acquire a multi-modal scene representation. The layout of the supervision system can be seen in Fig.1 A close to reality setup with KUKA lightweight robots inside the OR can be seen in Fig.2

III. Methods

The proposed approach is based on OpenNI and ROS. The number of used Kinect cameras can be parametrized at the start up of the system. The detection and correspondence matching as well as distance computation works as follows

A. Registration of Kinect cameras with respect to a reference frame

As Kinect features a depth camera as well as a RGB camera with known transformation between the cameras, colored point clouds can be acquired. Using this relationship between the cameras a checkerboard based algorithm is used

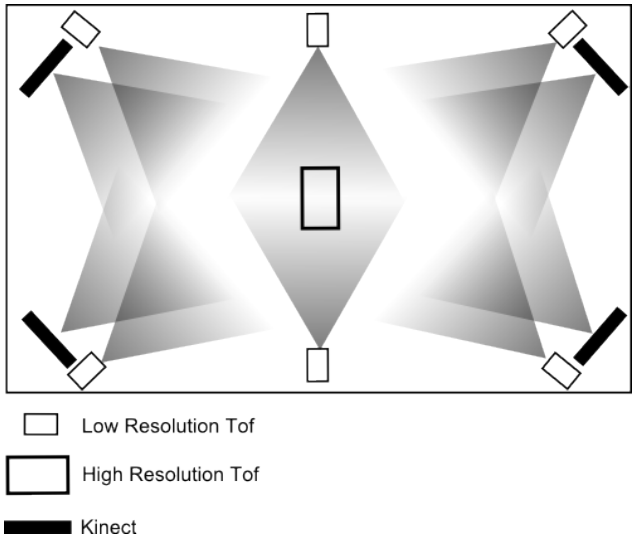

Fig. 1. Layout of the Camera rig from a bird's view. High Resolution (204x204 pixels) ToF is pointing downwards all other cameras are pointing towards the floor with an angle between 30-45 degrees compared to the floor

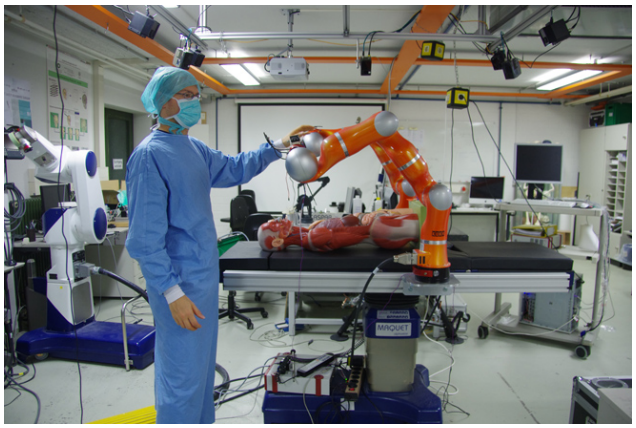

Fig. 2. Setup in the IPR lab showing the surgeon, the operating table and the minimally invasive robotics setup

for the registration of the system. For maximum accuracy both cameras are intrinsically calibrated and afterwards extrinsically calibrated to each other. One of the four Kinect cameras is the reference camera K_{ref} and we perform a pairwise registration between every remaining camera (K_1, K_2 and K_3) and K_{ref}. The following steps are performed in order to find the transformation

1) The checkerboard is placed inside the field of view of both cameras. OpenCV [16] checkerboard detection is applied to both RGB images
2) For every RGB pixel a depth pixel is captured at the position of each detected corner.
3) In order to be more robust to noise, average filtering over 60 depth frames is performed. The corresponding depth values at the position of the checkerboard corners are stored as references.
4) The correspondences are used to estimate a transformation between the camera frames using PCL [17]. The estimation is based on incrementally building a covariance list and the means of the correspondence list. The

rotation is estimated using eigenvalue decomposition of the covariance. The translational component is estimated using the means of the point sets similar to what happens in a standard ICP [18] iteration.

B. Detection of a reference plane within the reference frame

Both the correspondence detection for users in different cameras and the robot base detection depend on knowledge about the plane equation that describes the floor of the room the system is used in. The plane is needed for detecting the robot at the OR table and to project the centroids on. A common method to detect a plane is to use a RANSAC algorithm [19] in combination with a plane model. We used the RANSAC plane estimator that is shipped with PCL. RANSAC delivers a plane equation of the dominant plane inside a point cloud. The plane equation is stored for future processing

C. Robot Base detection

In the medical scenario to which the approach of this paper is applied, the robots are attached to an operating room table. After the OR table is moved to its position, which may vary between different executions, the position and orientation of the robots has to be determined. First, we detect the plane and shape of the OR table using a constrained RANSAC. Different constraints such as the probable table height and angle against the reference plane are taken into consideration. As the robots are attached to the table on side-rails, their orientation with respect to the table is known whereas their position is variable. We initially locate the base of the robots by a simple circle fitting restricted to the possible locations along both rails. Using this base position, the robots' virtual models are fitted into the scene to determine their correct 6D pose. As operating rooms and the workflows may vary (stationary base, movable operating table), the approach has to be adapted to the current scenario. For adapting to the real world, other robot detection algorithms have been implemented that use model fitting or spatial change detection.

D. Human detection using OpenNI

OpenNI and NITE are frameworks for the development of 3D sensing technologies using the Primesense sensor that is built into the Kinect camera. For a single Kinect camera it offers user detection and full body tracking without the requirement of a calibration pose. We use OpenNI for user detection in every single Kinect camera. Up to 16 users can be detected in the field of view of a single Kinect camera. We extract the pixels of every single user and add them to an empty depth map that is used for future processing. The result is as set of n depth maps per camera where n is the number of users detected in the camera frame. Every depth map holds the depths of a single detected user. This is repeated in every iteration and triggers the execution of the complete processing chain. Fig.3 shows a user detected by two Kinect cameras in different positions

E. Removal of noise pixels in depth image

A common problem of PMD and Kinect sensors is noise at the edges inside an image. Using PMD cameras, the term "jumping pixels" is commonly used for pixel that change position from foreground to background and vice versa. However Kinect shows a behavior where several pixels at corners lie between foreground and background. This noise makes it hard to perform a robust brute force distance computation and affects the computation of centroids. We use a two step noise removal process. In the first step the morphological erosion filter is applied to each depth image holding a user. This removes pixels at the corners of these images eliminating most of the noise.

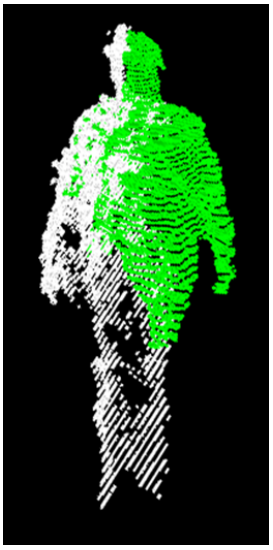

Fig. 3. User detected by two cameras

F. Point cloud computation for every detected user

Using the known focal length of the Primesense sensors lens, the resolution of the sensor and the function that maps depth maps to meters, one can compute a point cloud from every depth map.

G. Denoising in Point clouds

The resulting point cloud is still not completely free of noise. Especially artifacts that occur during movements cannot be completely removed by the erosion operation. In order to cope with the remaining noise statistical outlier removal is being performed. The used algorithm is described in [17] and uses k-neighbors of a point to find inliers and outliers. Points that are closer to the mean μ of the k neighbors than 1.0*σ (standard deviation) of the k-neighbors are considered as inliers. The result is a point cloud representing a single user in a single camera with almost no visible noise at the edges. This provides the optimal base for centroid computation, sensor fusion and distance computations.

H. Finding correspondences based on centroids

In order to determine corresponding users the centroid of the point cloud representing the complete user is computed. The centroid of a point cloud can be computed by summing up the translational components of every point in the scene on its own. Dividing the resulting vector through the number of

points gives the centroid of a point cloud like shown in eq.(1)

$$c = \begin{bmatrix} \frac{p_{1,x}+p_{2,x}+...+p_{n,x}}{n} \\ \frac{p_{1,y}+p_{2,y}+...+p_{n,y}}{n} \\ \frac{p_{1,z}+p_{2,z}+...+p_{n,z}}{n} \end{bmatrix} \quad (1)$$

where p_i is a point in the point set.

In our approach the centroids are used for the fusion of user point clouds. The point clouds are transformed into the reference frame K_{Ref}. Afterwards corresponding users are found using the Euclidean distance between every computed centroid. If the Euclidean distance is below a certain threshold, the users in the clouds are considered to be the same user. A matrix holding the correspondences (in the following: correspondence matrix) is being computed that is used for concatenation of the point clouds. However the centroid is a measure that is not very robust to movements of the user. Bending of the users or articulation of a joint causes the centroid to move. As the field of view of the cameras is different, partial occlusions are possible and only parts of a user may be in the fov of a single camera. The movement of a centroid in a single camera may result in a distance between corresponding centroids that is above the defined threshold which results in false negative detection of correspondences.

A solution is to enlarge the threshold for correspondence detection which results in higher false negative detection. We tackle the problem using the fact that the erosion operation from the removal of noisy points does not only remove noise but also useful information. The impact on image parts representing limbs is higher compared to the impact on abdominal and thoracic regions. The cause for this is that in a 2D view (depth image) of the users more edges are visible for limbs compared to the upper body parts. The erosion operation removes pixels at every edge and therefor stronger affects the limbs.

The reduction of limb points moves the centroid more into the center of the body of the detected user as long as the users body is visible in the viewport of the camera. The overall number of Points inside thorax and abdomen exceeds the number of the points of the limbs which additionally helps to pull the centroid towards the center of mass of a user. However the problem of the movement of the centroid through bending of the upper body is not solved. We reduce the problem to a two dimensional one by projecting the centroids to the floor plane which was detected in a prior step. This eliminates errors along the longitudinal axis of the user and helps for cases where only parts of a user are visible in a single camera image.

The computation of the distance between the centroids is performed on this projected centroids and the final correspondence matrix is being stored.

An example for a case where projection to the floor helps is, if a users complete body is visible in one of the cameras and only an arm is visible in one of the other cameras. In the 3 dimensional case the centroid of the user in the full body cloud is close to the center of mass of the user, but in the cloud representing only the arm is close to the center of mass of the arm. Projection to the floor in this case eliminates all errors along one of the three axes and allows for lower thresholds.

Projection can simply be performed by computing the normal of a plane (in our case the floor) and moving the centroid along this normal until the plane is being crossed.

Anyway, a high threshold for correspondence estimation is necessary in order to reduce false positive detection rate as the view-ports of the Kinects are different which results in partial point clouds of the user. As the part of the body behind the surface seen by the Kinect cannot be captured the centroid is biased along the axis between the observed user and the Kinect towards the Kinect camera. In order to cope with registration errors and the aforementioned problem a threshold of 0.6m has been chosen for our experiments but the absolute minimum for acceptable correspondence estimation has not been evaluated yet.

I. Concatenation of point clouds per user

Using the correspondence matrix we perform a lookup on the computed point clouds of the users detected in different cameras. The point clouds are concatenated into a single larger point cloud which results in overlapping regions being represented by points of several cameras and several viewpoints and regions that are only represented by the points of only one camera.

Regions with overlapping point clouds introduce redundancy into the concatenated point cloud. In order to reduce the size of the point cloud and to remove redundancy we downsample the point cloud using a grid. This reduces the amount of points in areas where points are dense and does not reduce areas with a lower number of points. The downsampled point cloud is being used for distance computations and situational inference in subsequent steps of the process

J. Distance computation on GPGPU

In order to get a measure about distance between humans, robots, as well as humans and robots we implemented a CUDA based algorithm that operates on point clouds without using any shape information. Collision checks with meshes are not included but in the target scenario there is no need for precise collision checking. As described before the robot base is detected in the scene using point clouds. After successful detection we rely on the CAD model of the robot and position encoders of the robots joints. We update the CAD model using Denavit-Hartenberg forward kinematics and the measured joint angles. From the CAD model we extract all vertices dropping the triangles. The vertices of a CAD model can be considered to be a point cloud. Both, the CAD point cloud and the point cloud from the processing chain are being loaded into CUDA pinned memory in every updated cycle. Finally the cloud is uploaded to the graphics ram of a NVIDIA Geforce GTX480. A kernel on the NVIDIA GTX480 computes eq.(2)

$$d(p,q)^2 = ||q-p||^2 \quad (2)$$

which is the squared euclidean distance. We use a brute force method to compute all squared euclidean distances between the source and target cloud. Afterwards we compare all squared euclidean distances on the graphics adapter. The points for which the squared euclidean distance is minimal, are closest in the point clouds. We pass back the smallest value together with point indices of the points in source and target cloud

to the main process where the square root for this value is being computed on the CPU. This gives a measure of the closest distance between two point clouds and is being used as simple collision avoidance in a first step. Two thresholds have been defined. If the distance is below threshold 1 a warning is displayed to the user. If the distance is below threshold 2 the robot can be stopped in order to avoid hazardous situation like collisions. The use of thresholds describing safe and unsafe regions removes the requirement of performing a collision check in our scenario. Small distances between a user and the robot can be considered unsafe and uncomfortable for a user, especially in the case of surgical robots where the robot acts as an assisting system. The brute force approach only works if the data is nearly free of outliers, as those are considered as user points as well as useful information. If outliers and noisy pixels cannot be completely removed the quality of the distance computation decreases resulting in a higher false positive detection rate of hazardous events. Fig.4 shows the running system.

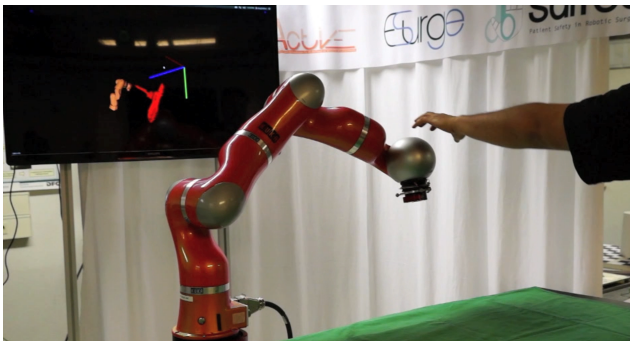

Fig. 4. Complete system during run time Foreground: supervised scene Background: Scene representation with user detected being to close to the robot

IV. Results

We measured the frequency of the received image data collected and transmitted via Ethernet by the mini computer using built-in ROS mechanisms. The resulting frequency without any computation was 25Hz. The accuracy evaluation for the checkerboard based registration has been performed using 12 poses for the checkerboard in a volume of 1.8m x 1.0m x1.5m. The data has been statistically analysed using a Kruskal Wallis method to show the registration error of the pairwise registered Kinects. In the following, Kinect pairs of a source and the reference camera are called pair.

TABLE I. Median values in mm and quartile range for the accuracy evaluation of each pair of Kinect™ cameras.

Variable	Median Value	1st quartile	3rd quartile
Pair 1	19.8392	12.7431	28.2076
Pair 2	26.8867	19.9021	34.8021
Pair 3	26.6838	20.2448	35.1427

In Table I, the median values and the inter-quartile ranges of the accuracy evaluation are reported. The statistical analysis showed that the results for pair one are significantly different from the other two pairs. Fig.5 shows the results of the accuracy evaluation for each pair of Kinect cameras in all the tested positions.

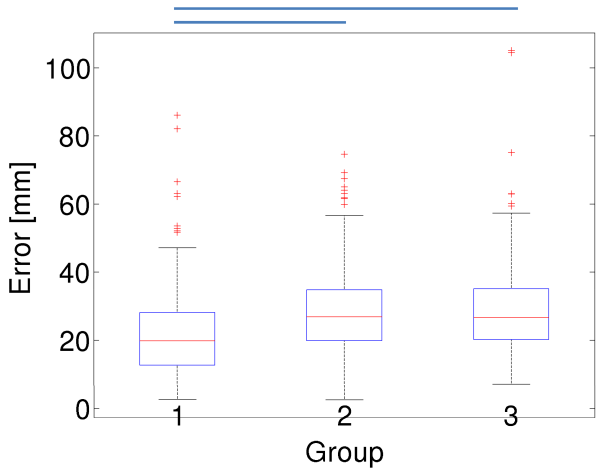

Fig. 5. Boxplot of the error for each pair of Kinect in each position. The horizontal bars shows the statistical difference between the positions, if present.

When introducing the detection and fusion chain performance measurements show that an Intel Core i7 3770 can process up to four Kinect cameras using the proposed approach.

Our approach accurately removes noise pixels at the edges of detected users but also removes some useful information as well. We did not observe any pixels not connected to a user during our experiments which made distance computation without knowledge about the objects possible.

We did not yet include algorithms that rely on the combined point clouds. This means that no information about allowed false positive and false negative detected correspondences is available. In the future we are going to fuse tracking data collected from multiple trackers based on our approach to built an accurate skeleton model. The false positive and false negative detection rates are correlated via the threshold for correspondence estimation. The higher the threshold, the lower the false negative detection rate, but the higher the false positive detection rate.

We measured the delay between the actual action inside the scene and the time until the data has been completely processed using a high speed camera that was able to see both the screen and the scene. The delay has been extracted by measuring the time between the action being performed and the monitor showing the result. The delay is about 950ms im average using Kinect cameras and transmission via ROS and Ethernet.

V. Discussion and Conclusions

We propose a novel approach for fusion of detected users inside 3D depth maps as well as a registration method for RGB-D sensors. The approach is scalable and robust to interference introduced by the use of multiple Kinect cameras and does not need models to compare the captured data with. Therefore the approach is highly flexible, can run with different detection algorithms and works also for other objects than humans and other RGB-D cameras than the Kinect.

However several issues have not been tackled yet. The approach shows a registration error that makes it usable

for high level scene supervision for workflow detection and for inferring about the situation but is not suited for high precise measurements of body area, volume or exact distance calculation. Additionally the removal of noisy pixels increases the model quality but removes also part of the surface around an object being detected. The use of better sensors like precise stereo cameras could reduce the error but introduces problems like lighting and the need for stereo calibration. The delay between the action being performed and the scene being completely processed is rather high. This is not a problem for inferring about long lasting processes like workflow detection in the operating room but makes the use of faster systems like our PMD system necessary for time critical tasks like collision avoidance. In order to reduce the threshold for correspondence estimation, a shape based model for computing a point that is closer to the center of mass of a user compared to the centroid may help.

Future work will use the fusion approach to fuse joint angles as well. The correspondence matrix will be used to determine skeleton tracker observing the same user. We plan to implement a fusion approach for skeleton tracking based on fitness values for every single tracker. An important research topic is to include situation based information into the process. Measuring distance to infer about hazardous situation is not enough as in cooperative control modes that may occur during operations the robot is allowed to be in contact with the surgeon. In order to further improve the detection rate we plan to introduce additional tracking algorithms into the processing chain which enables us to combine detected users from both algorithms. This approach very likely reduces the false negative detection rate of the complete system.

The final output data will be used to infer about the current situation in the operation, to switch between workflow steps and will form the base for probabilistic and rule-based control of the overall system. The target system is a complete integrated neurosurgical platform developed in the scope of the ACTIVE project. Here several robots work close together with humans and share the same workspace. The system is intended to follow the workflow during the operation and provide safe and intuitive human robot interaction during the whole intervention.

ACKNOWLEDGMENT

This research was funded by the European Commissions 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. The authors thank the EU for its financial support.

REFERENCES

[1] R. Konietschke, U. Hagn, M. Nickl, S. Jorg, A. Tobergte, G. Passig, U. Seibold, L. Le-Tien, B. Kubler, M. Groger, F. Frohlich, C. Rink, A. Albu-Schaffer, M. Grebenstein, T. Ortmaier, and G. Hirzinger, "The dlr mirosurge - a robotic system for surgery," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, 2009, pp. 1589–1590.

[2] J. A. Cadeddu, A. Bzostek, S. Schreiner, A. C. Barnes, W. W. Roberts, J. H. Anderson, R. H. Taylor, and L. R. Kavoussi, "A robotic system for percutaneous renal access," *J. Urol.*, vol. 158, no. 4, pp. 1589–1593, Oct 1997.

[3] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schaeffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, and G. Hirzinger, "The kuka-dlr lightweight robot arm - a new reference platform for robotics research and manufacturing," in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, 2010, pp. 1–8.

[4] G. Guthart and J. Salisbury, J., "The intuitivetm telesurgery system: overview and application," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 1, 2000, pp. 618–621 vol.1.

[5] B. Hannaford, J. Rosen, D. Friedman, H. King, P. Roan, L. Cheng, D. Glozman, J. Ma, S. Kosari, and L. White, "Raven-ii: An open platform for surgical robotics research," *Biomedical Engineering, IEEE Transactions on*, vol. 60, no. 4, pp. 954–959, 2013.

[6] Y. Lou, W. Wu, H. Zhang, H. Zhang, and Y. Chen, "A multi-user interaction system based on kinect and wii remote," in *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*, 2012, pp. 667–667.

[7] A. D. Wilson and H. Benko, "Combining multiple depth cameras and projectors for interactions on, above and between surfaces," in *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, ser. UIST '10. New York, NY, USA: ACM, 2010, pp. 273–282. [Online]. Available: http://doi.acm.org/10.1145/1866029.1866073

[8] F. Faion, S. Friedberger, A. Zea, and U. Hanebeck, "Intelligent sensor-scheduling for multi-kinect-tracking," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 3993–3999.

[9] L. Zhang, J. Sturm, D. Cremers, and D. Lee, "Real-time human motion tracking using multiple depth cameras," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 2389–2395.

[10] M. Nakazawa, I. Mitsugami, Y. Makihara, H. Nakajima, H. Habe, H. Yamazoe, and Y. Yagi, "Dynamic scene reconstruction using asynchronous multiple kinects," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, 2012, pp. 469–472.

[11] P. Nicolai, T. Beyl, H. Monnich, J. Raczkowsky, and H. Worn, "Op:sense - an integrated rapid development environment in the context of robot assisted surgery and operation room sensing," in *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, 2011, pp. 2421–2422.

[12] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009.

[13] H. Bruyninckx, "Open robot control software: the orocos project," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 3, 2001, pp. 2523–2528 vol.3.

[14] M. Garland, S. Le Grand, J. Nickolls, J. Anderson, J. Hardwick, S. Morton, E. Phillips, Y. Zhang, and V. Volkov, "Parallel computing experiences with cuda," *Micro, IEEE*, vol. 28, no. 4, pp. 13–27, 2008.

[15] Tim Beyl and Philip Nicolai and Jörg Raczkowsky and Heinz Wörn, "Ein Kinect basiertes Überwachungssystem für Workflowerkennung und Gestensteuerung im Operationssaal," in *Tagungsband der 11. Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie e.V. (CURAC)*, 2012.

[16] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, "A brief introduction to opencv," in *MIPRO, 2012 Proceedings of the 35th International Convention*, 2012, pp. 1725–1730.

[17] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[18] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 2, pp. 239–256, 1992.

[19] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692