Human-robot interaction booth with shape-from-silhouette-based real-time proximity sensor

Patrick Dietrich[a,b], Florian Siegmund[a,b], Christian Bräuer-Burchardt[a], Stefan Heist[a], and Gunther Notni[a,c]

[a]Fraunhofer Institute for Applied Optics and Precision Engineering IOF, Albert-Einstein-Str. 7, 07745 Jena, Germany
[b]Institute of Applied Physics, Abbe Center of Photonics, Friedrich Schiller University Jena, Albert-Einstein-Str. 15, 07745 Jena, Germany
[c]Group for Quality Assurance and Industrial Image Processing, Department of Mechanical Engineering, Ilmenau University of Technology, Gustav-Kirchhoff-Platz 2, 98693 Ilmenau, Germany

ABSTRACT

Industrial robots have been an essential part of production facilities for many years. They allow fast and precise positioning of even the largest loads with very high repeatability. However, there are still many processes which are superiorly or more economically executed by humans. If a component requires several work steps, some of which are better suited to a robot and some to a human worker, cooperation between humans and robots would be beneficial. Due to the enormous power and speed of industrial robots, this poses a considerable risk to the worker. Therefore, tasks to be performed by humans and robots are usually completely decoupled in terms of space or time.

We suggest an approach, which allows a human worker to interact safely with a fast industrial robot. We achieve this by constantly monitoring the position of both robot and human and adjusting the robot's velocity according to its proximity to the worker. We present an interaction booth, which can be entered by a robot arm from the back and a worker from the front such that they can both access the machinery within. A multi-camera sensor, which is based on the shape-from-silhouette principle, constantly observes the booth to monitor its occupancy. We demonstrate that within 50 ms, our sensor can (1) detect a change in occupancy in the booth, (2) classify sub-volumes as "robot", "human", or "other object", (3) calculate the distance between human and robot, and (4) output this information to the robot controller. The working speed of the robot is then adjusted according to its distance to the worker.

Keywords: human-robot interaction, shape from silhouette, voxel carving, multi-camera sensor, real-time

1. INTRODUCTION

The interaction between fast and powerful industrial robots and human workers is challenging. In particular it is difficult to ensure the safety of the humans within reach of the robot due to the enormous forces the fast moving robot can exercise in a collision event. It is therefor necessary to absolutely avoid such collisions.

Today, tasks to be performed by humans and robots are usually completely decoupled in terms of space or time. Areas where robots work are closed for human workers. Entrances are guarded by safety light barriers or similar devices to detect when a human enters. While these devices can be considered safe in terms of reliability and measurement latency, they can only provide binary output, i.e. a human has entered or not. Therefor the reaction can also only be binary: robot running, or robot stopped. True interactions between robots and humans are ruled out by this approach.

Further author information: (Send correspondence to Florian Siegmund)
Florian Siegmund: E-mail: Florian.Siegmund@iof.fraunhofer.de, Telephone: +49 3641 807-268

Dimensional Optical Metrology and Inspection for Practical Applications XI, edited by Kevin G. Harding, Song Zhang, Jae-Sang Hyun, Beiwen Li, Proc. of SPIE Vol. 12098, 120980B
© 2022 SPIE · 0277-786X · doi: 10.1117/12.2617069

In contrast, our approach is to continuously monitor the positions of both the robot and humans in the same work space. We then adjust the working speed of the robot according to its proximity to the humans. The robot will eventually stop when it comes very close to a human, but before that it will gradually reduce its working velocity. While the position of the robot can be sensed with internal sensors, the position of the human workers must be determined by external sensors. Furthermore, these sensors must provide the positional information continuously and with a short time delay between sensing and information availability. Camera based sensors can determine a workers position in 2D. Combining several cameras into a multi-camera sensor can also provide 3D positional information.

Multi-camera systems have been used for many years for security applications, e.g. to monitor factory premises. The availability of inexpensive digital cameras makes it possible to generate data volumes that can hardly be monitored by individual employees. This has led to the need to partially or completely automate these monitoring tasks. Among other things, tracking people and objects through the viewing areas of multiple cameras is a major area of research. A review of the methods can be found in.[1] Closely related is the task of determining the position of people in large spaces captured by multiple cameras. A main application here are sporting events, such as football matches, where the positions of players on the pitch need to be determined automatically. Most of these methods have in common that they work in 2D: People or objects are detected on a common plane representing the ground, so that a map of positions is generated in top view. (e.g.[2,3]) A similar approach is to determine 2D positions within camera images stitched together using homographies, i.e., also in a plane (e.g.,[4]). While some of these methods are real-time capable, they are less suitable for workspace monitoring because of their 2D data representation.

In contrast, the *Shape From Silhouettes* method can be applied to generate a volumetric representation of objects whose silhouettes are detected in the camera images. If a voxel volume is used for data representation, the method is also called **Voxel Carving**. A review of the method can be found in.[5] In,[6] the method was used in conjunction with multiple cameras to capture actors in real-time at low resolution (64^3 voxels) in 3D and insert them into virtual scenes. In[7] another similar real-time capable method is proposed. These methods allow the fast measurement of the general shape of an object. The detailed 3D-geometry of the surface is not measured. In particular, the voxel carving method classifies parts of the monitored volume as occupied or not occupied.

2. METHOD

We divide a working volume with a regular grid of voxels. Each voxel is a cubic volume with a side length of 20 mm. Our first goal is, to classify each voxel into one of two categories: occupied, and empty. In Fig. 1, we show a simplified 2D example of correct and incorrect voxel classification.

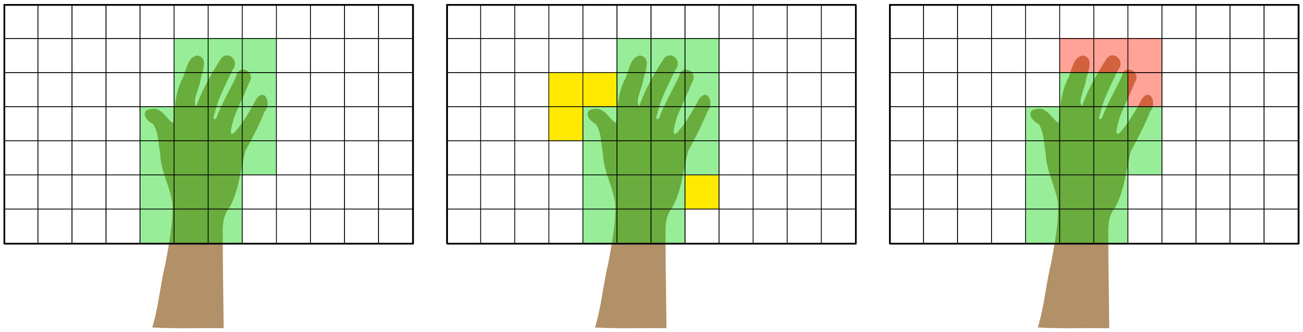

Figure 1. Principle of voxel classification. White voxels are classified as empty. Left: the green voxels have been correctly classified as occupied. Middle: the yellow voxels have been incorrectly classified as occupied. Right: the red voxels have been incorrectly classified as empty.

Every measurement system produces errors. For our case of work space monitoring some measurement errors can lead to severe safety issues, while others are less severe. In selecting a sensor system it is important to first categorize these errors:

1. **Voxels are incorrectly labeled as occupied** (Fig. 1 middle). This type of measurement error can lead to voxels not being used, although this would actually be possible. This can lead to slowdowns in the workflow or even to a standstill. However, unexpected collisions are not to be expected. This case is not a safety risk.

2. **Voxels are incorrectly labeled as empty** (Fig. 1 right). This type of measurement error can lead to collisions, e.g. when the robot enters a volume where a human is located, but which was incorrectly labeled as empty. A significant safety risk thus arises.

The great difference in the hazard resulting from these two types of measurement errors demands different prioritization: Type 2 must be avoided under all circumstances. Type 1, on the other hand, is tolerable to a certain degree in most applications. In case of doubt, a voxel must therefore always be marked as occupied. This must also be ensured in the event of an undetected (partial) failure of the measurement system. The voxel carving method can be implemented such that it fullfills these requirements (Fig. 2).

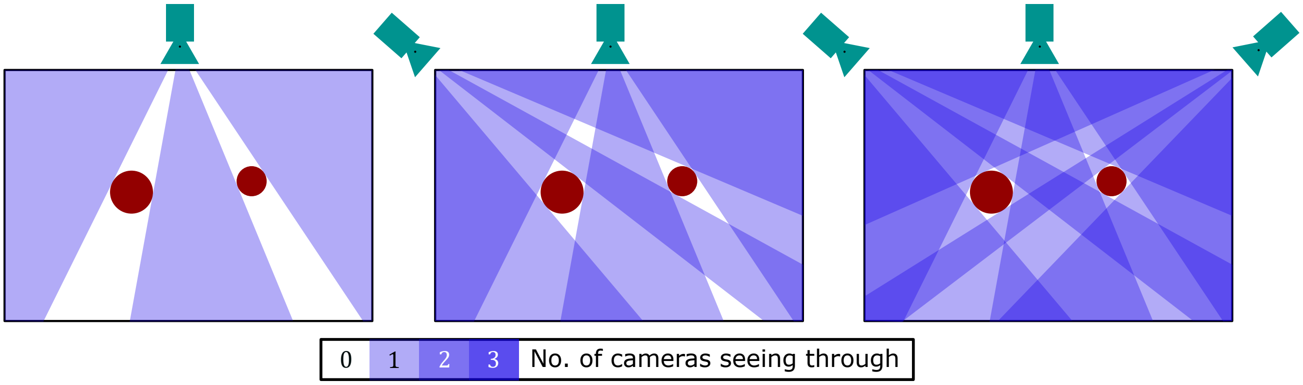

Figure 2. Measuring principle of our voxel carving sensor. Left: A measuring cell with two objects (red) is observed by one camera, the purple areas can be labeled as *empty* because the camera records the cell background there. Middle: the same volume is observed by two cameras. Right: observation with three cameras. All areas which are still white (or red) cannot be labeled as *empty* and are therefore labeled as *occupied*.

Each of the cameras captures a background image at system startup. With each consecutive exposure, the image then captured is compared with the background image. If a particular area within the new image has changed compared to the background, this means that there is an object between the background and the camera. If, on the other hand, the background is visible, there is no object in the line of sight. Voxels along it can then be marked as *empty*. The remaining voxels, are assumed to be *occupied*. Figure 2 shows an example of the principle.

Thus, it is never required to explicitly detect an object in the foreground. In terms of prioritizing measurement error handling (see above), this means:

- When a camera fails, fewer voxels are marked *empty*, more voxels are marked *occupied*. So there is no immediate safety risk.
- In case of sudden change in lighting conditions (e.g. due to lighting failure, light reflections, direct sunlight), in the worst case a change from the background image is detected, marking fewer voxels as *empty*, which leads to no immediate safety risk.

We have built an interaction booth, including a working volume of 1.2 m × 0.5 m × 0.5 m. The working volume can be entered by a robot arm from the back and a worker from the front. Both the worker and the robot can access the machinery within. The volume is constantly monitored by 4 cameras. From the camera images we first calculate an occupation model loosely based on the voxel carving method described in.[7] We then classify the voxels as worker or robot, based on their topological connection to the front or back of the working volume.

Afterwards, we calculate the shortest distance between robot and human as the shortest distance between the two voxel classes. This distance is quantized and output to the robot controller to adjust the robot's working velocity (the closer, the slower).

2.1 Experimental Setup

Two of the cameras are placed above the volume on the two short sides and two to the right and left of the measurement volume, respectively. The properties of the cameras are listed in table 1. The measurement cell is shown in figures 3 and 4.

Table 1. Technical data of the camera system.

Number of cameras	4
Sensor type	Global shutter with bayer color filter
Pixel resolution	832 × 632 Pixel
Focal lengths of the lenses	3.2 mm (lower cams) 4.8 mm (upper cams)
Framerate	70 Hz
Image integration time	10 ms
Trigger signal	Common hardware trigger for all 4 cameras

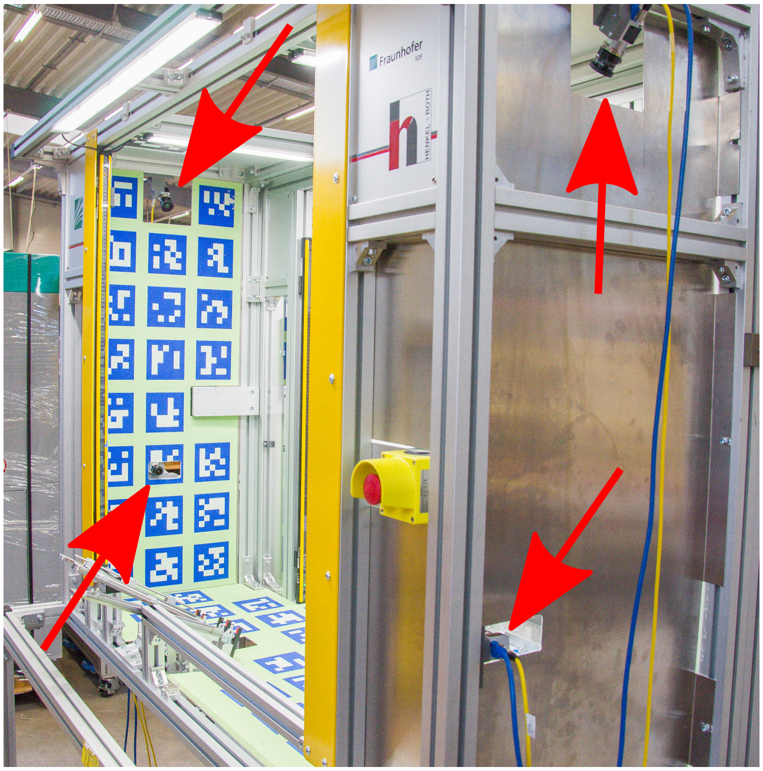

Figure 3. View from the right front into the measuring cell. The four cameras are marked with red arrows.

We chose light green as the background color with additional blue markers. These colors have the advantage that they contrast well with human skin of different types. In addition, the heterogeneity of the background ensures that if an object with background-like color and brightness is held in the measurement volume, confusion with the background occurs only partially, but not over the entire surface. The markers are used to check the camera position at power-on to ensure they have not moved since calibration as this would be a safety hazard.

2.2 Calibration

After fixing the cameras at their recording position, we measure their intrinsic and extrinsic projection parameters in a calibration process. For this process we follow the procedure described in.[8] The calibration process has to

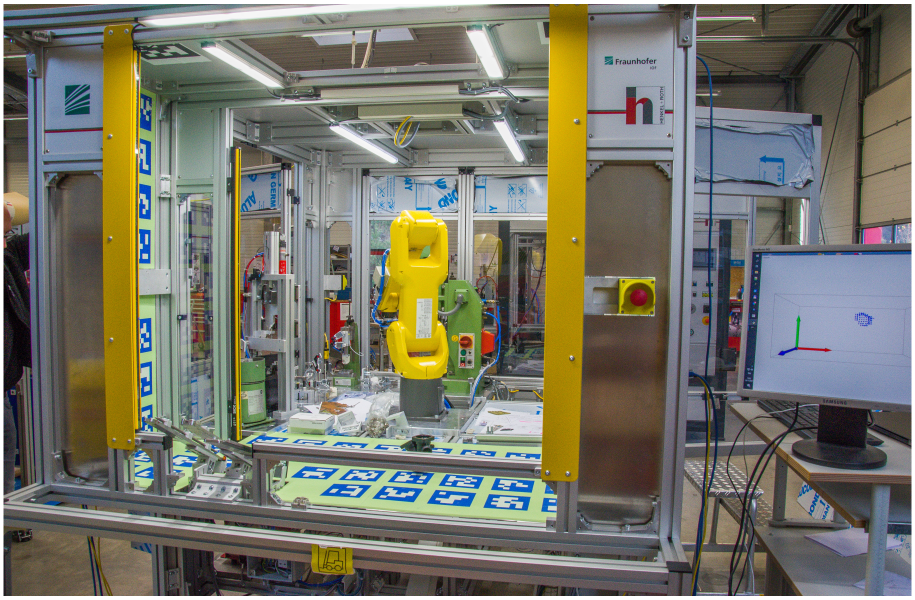

Figure 4. View from the front into the measuring cell. The industrial robot from Fanuc can be seen in the background.

be executed only once after installation or if a camera is moved, replaced or refocused.

From the calibration of the cameras, we know their projection functions relative to the coordinate system of the measuring volume. I.e. each 3D point in the working volume can be projected onto the sensors by means of the known projection functions of the cameras. Thus, for a given 3D point (X, Y, Z), the pixel coordinate (x_k, y_k) can be calculated for each of the cameras $k \in [1, 2, 3, 4]$. (If the 3D point is outside the area visible from the camera, (x_k, y_k) is outside the camera image.)

With these projection functions, we precompute a voxel to camera mapping for each camera: For each voxel center (X_v, Y_v, Z_v), the corresponding camera pixel (x_{vk}, y_{vk}) is computed for each of the cameras. We store this precomputed mapping as an array of pixel coordinates for each camera. I.e., we have four precomputed arrays each of which contains the pixel coordinates in one of the cameras for each voxel.

2.3 Data Processing

The data processing comprises the following 4 major data processing steps:

1. The separation of the camera pixels into *background* and *foreground*. Realized as a pure 2D calculation for each camera.
2. The assignment of foreground/background information to voxels inside the volume. Each voxel center that is mapped to a camera pixel classified as "background" is considered to be empty.
3. The classification of the occupied voxels into *human*, *robot* and *other object* based on their topoligical connection to the front or rear boundary of the measurement volume. Also, the shortest distance between all human and robot voxels is calculated.
4. Passing the distance information to the robot controller.

We realized the first two parts with a parallel program which uses one CPU core per camera. At the end of this parallel process, we create the voxel volume describing the volume occupancy. Then, we classify occupied voxels with a topological connection to the back of the volume as robot, and occupied voxels with connection to the front as human. After that, we calculate the shortest distance between any human voxel to any robot voxel.

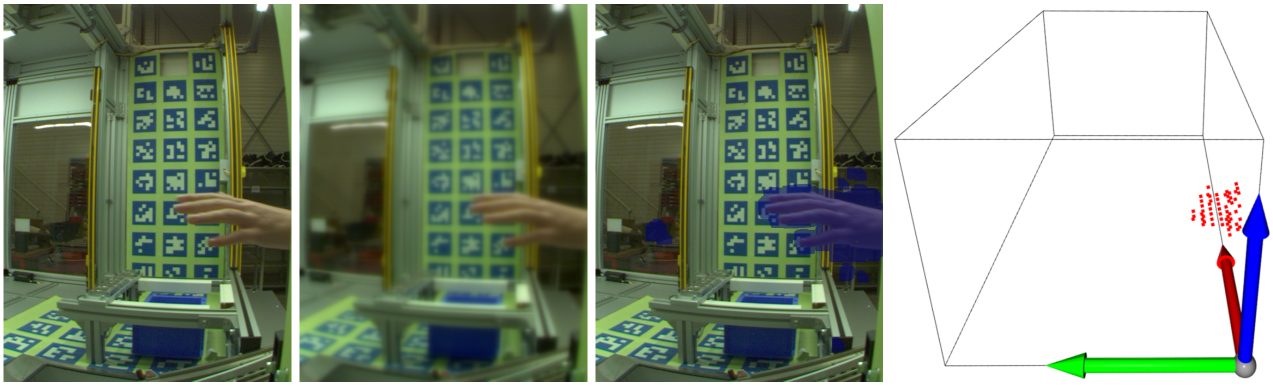

Figure 5. Separation of foreground and background. From left to right: original camera image, blurred camera image, foreground/background separation (foreground marked in blue), 3D volume resulting from the combined information of all four cameras (occupied voxels marked red).

Finally, this distance is sent via USB to a microcontroller board, which sends one of 8 possible binary signals to the robot controller. Each signal represents a quantized distance.

When the system is started, the measurement cameras each record a background image. To reduce image noise, this background image is calculated as an average of 25 individual recordings. In addition, we treat it with a spatial low pass filter (blur). We refer to this blurred background image as *reference image*. For each measurement, we then take one image per camera and treat it with the same blur as the background image. We refer to it as the *now-image*.

The blur has the effect that 1. camera noise is significantly reduced, 2. there are fewer false detections at texture edges, e.g. because the camera moves slightly due to vibrations of the cell, and 3. small details such as dust particles are removed from the image. Generally speaking, the blur filter improves the measurement robustness of the setup because it increases the signal to noise ratio. The signal is generated by objects with the size of a human finger or larger. Smaller objects can be ignored and thus be treated as noise.

We transform both the reference image and the now-image into the HSV color space. I.e., we dissect the images into *Hue* (color value on the color wheel), *Saturation* (color saturation) and *Value* (brightness). A change of the now-image compared to the background image is detected by pixelwise comparison of the individual components. In particular, we apply threshold values to the differences of individual components. E.g. for the saturation values in the background image s_b and in the now-image s_n, the threshold t_s is applied as $s_n - s_b > t_s$.

Since a certain change of the ambient light is to be tolerated, we apply only a coarse limit to the brightness, e.g. this can detect a black glove in front of the bright background. Beside this, two threshold values are applied to the color saturation and the color value difference, respectively. If one of the three threshold values is exceeded for a pixel, this pixel is classified as foreground, otherwise as background. We chose the threshold values in such a way that the human fingers and tools like screw drivers are safely classified as foreground while there are no or almost no false classifications as foreground.

After the binary foreground/background image is created, we expand the detected foreground area by five pixels in each direction using binary dilation. This further increases the robustness of the system as the volume which is not marked as empty in the following calculation step increases in size.

During measurement, we look up the foreground/background pixel value in the camera images for each voxel using the precomputed mappings. We do this lookup once for each camera and voxel. If a voxel is marked as background by at least one camera, it is considered empty. If a pixel is marked as foreground by each of the cameras, it is considered occupied. The right most image in Fig. 5 shows such a voxel volume. The voxel size was set to 20 mm × 20 mm × 20 mm.

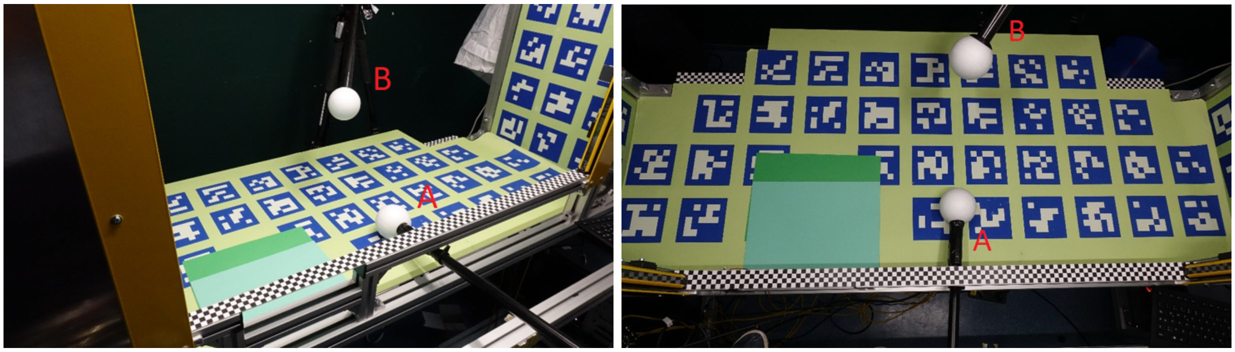

Figure 6. Photographs of the setup for the latency measurements. A) Tripod arm representing a human worker. B) Tripod arm representing the robot.

3. EXPERIMENTS AND RESULTS

3.1 Latency measurement

The latency which is relevant to the described safety application, is the time span between a change inside the measurement volume, e.g. the movement of a worker's hand, and the signal generated at the output of the measurement system, i.e. the output of the distance value to the robot controller.

We measured this latency with the following method. We positioned two tripod arms with styrofoam balls at their ends inside the measurement volume (see figure 6). The arm coming in from the back represents the robot, the one coming in from the front represents the human. As a result, the measurement system is then sending a binary signal to the robot controller indicating the specific distance between the two arms. We coupled this signal to the input of an oscilloscope. In addition, a brightness sensor was positioned at the edge of the measurement volume. The signal from this brightness sensor was connected to a second oscilloscope input.

Before the measurement, we turned off the light in the measurement cell, which resulted in the volume sensor marking all voxels as occupied because of the large difference in brightness from the reference image. The distance signal connected to the oscilloscope was therefor turned off by the measurement system. (Instead, the system sent out a different signal representing zero distance, which was not used for the experiment).

To start the latency measurement, we switched the lights back on. This first resulted in a signal from the brightness sensor and, after image acquisition and volume occupancy calculation, again in the output of the correct distance signal to the oscilloscope. Both signals were recorded with the oscilloscope and logged. Thus, the time difference between the two signals is an end-to-end latency which takes into account all partial latencies of the system. The measurement was repeated 50 times. The mean value of the measured end-to-end latencies was 43.5 ms, ranging from 32.9 ms to 53.2 ms.

3.2 Uncertainty of the distance measurement

To test whether the volume occupancy sensor can reliably estimate the minimum distance between human and robot in different areas of the measurement volume, we made the following experiment. Due to the selected voxel size of 20 mm, an average error of at least 40 mm is to be expected, whereby the distance should always be underestimated for safety reasons.

We placed a tripod arm with a Styrofoam ball representing a human arm at a fixed position from the front in the measurement volume. We placed a second similar tripod arm representing the robot arm from the back side of the volume, successively at 7 different positions. At each position, the distance between the two Styrofoam balls was determined with the volume occupancy sensor. Each of the 7 positions was repositioned a total of 10 times by hand, such that the sphere was approximately (but not exactly) at the same position again (estimated deviation about 5 to 10 cm). Thus, a total of $7 \cdot 10 = 70$ independent distance measurements were made.

In addition, we measured the correct distance between the two spheres in each case with a GOBO projector-based 3D sensor.[9,10] This sensor can measure the 3D geometry of objects with an accuracies in the order of 100 µm. For each measurement, we fitted two spheres to the 3D data from this sensor and calculated the distance between their surfaces. We consider these distances ground truth. The difference between the distance estimated by the voxel carving sensor and the ground truth for each of the measurements is depicted in figure 7.

The distance differences determined are all negative, i.e. the distances estimated with the voxel carving sensor are shorter than the ground truth. This is unproblematic with regard to the safety requirements.

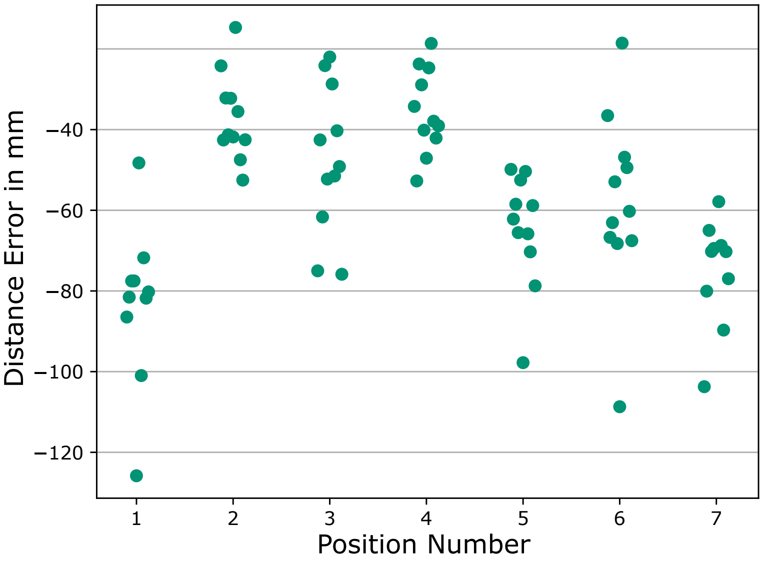

Figure 7. Distance error of the volume sensor: Distance between the two spheres determined by the voxel carving sensor minus ground truth sphere distance. The determined distances are always too short, never too long. (The dots representing individual measurements are slightly distributed along the x-axis to improve visibility.)

4. DISCUSSION

We successfully tested that the measured distances between robot and human worker are always shorter than the true distances. The reason is that the voxel carving method (as implemented here) marks voxels as occupied if they are not explicitly marked empty by one of the cameras. This can result in larger than actual volumes classified as human or robot, especially in the border regions where only one or two cameras observe the voxels. In addition, our experiments show that it is possible to send a new distance signal to the robot in less than 53.2 ms (43.5 ms average) after an actual change within the working volume has happened.

These two experimental results show that the concept which we presented here is suitable to prevent safety hazards in human-robot-interaction spaces.

5. CONCLUSION

In this contribution we applied the shape from silhouette method to ensure the safety in an interaction booth for the collaboration between a fast industrial robot and a human worker. Our multi-camera sensor continuously observes the working space and outputs an approximate distance between robot and human to the robot controller.

We chose a voxel carving based data processing method which inherently guarantees that the measured distance is never overestimated (but may be underestimated). We also validated this property experimentally.

In combination with the demonstrated fast end-to-end latency, these results are promising in every respect. In the future, we intend to re-implement the presented sensor on a real-time capable hardware, e.g. an FPGA. Further ideas include extending the system for entire working halls and taking into account semi-stationary objects within the working volume.

REFERENCES

[1] Wang, X., "Intelligent multi-camera video surveillance: A review," *Pattern Recognition Letters* **34**(1), 3–19 (2013). Extracting Semantics from Multi-Spectrum Video.

[2] Khan, S. M. and Shah, M., "Tracking multiple occluding people by localizing on multiple scene planes," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31**, 505–519 (3 2009).

[3] Sternig, S., Mauthner, T., Irschara, A., Roth, P. M., and Bischof, H., "Multi-camera multi-object tracking by robust hough-based homography projections," in [*2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*], 1689–1696 (11 2011).

[4] C, S. and Tieu, K., "Automated multi-camera planar tracking correspondence modeling," in [*2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*], **1**, I–I (6 2003).

[5] Cheung, K.-m. G., Baker, S., and Kanade, T., "Shape-from-silhouette across time part i: Theory and algorithms," *International Journal of Computer Vision* **62**, 221–247 (2005).

[6] Hasenfratz, J.-M., Lapierre, M., Gascuel, J.-D., and Boyer, E., "Real-Time Capture, Reconstruction and Insertion into Virtual World of Human Actors," in [*Vision, Video and Graphics*], 49–56, Eurographics, Elsevier, Bath, United Kingdom (2003).

[7] Cheung, G., Kanade, T., Bouguet, J.-Y., and Holler, M., "A real time system for robust 3d voxel reconstruction of human motions," in [*Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*], **2**, 714–720 vol.2 (6 2000).

[8] Zhang, Z., "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, 1330–1334 (Nov 2000).

[9] Heist, S., Lutzke, P., Schmidt, I., Dietrich, P., Kühmstedt, P., Tünnermann, A., and Notni, G., "High-speed three-dimensional shape measurement using GOBO projection," *Optics and Lasers in Engineering* **87**, 90 – 96 (2016). Digital optical & Imaging methods in structural mechanics.

[10] Heist, S., Dietrich, P., Landmann, M., Kühmstedt, P., Notni, G., and Tünnermann, A., "GOBO projection for 3D measurements at highest frame rates: a performance analysis," *Light: Science & Applications* **7**(1), 71 (2018).