

Collision Avoidance of Robots by Artificial Force Field Around Geometric Primitives Using Two Range Image Sensors

Christian Thormann and Alexander Winkler^(✉)

Hochschule Mittweida, University of Applied Sciences, Mittweida, Germany
{christian.thormann,alexander.winkler}@hs-mittweida.de
<https://www.inw.hs-mittweida.de/webs/winkler3>

Abstract. This article presents an approach to collision avoidance of industrial robots. It is based on artificial potential fields or force fields generated by virtual charges which are placed on the obstacles. The obstacles are detected by range image sensors. To avoid fault detections e.g. as a result of occlusions and distortions two cameras are used to supervise the robot work space. An algorithm is worked out which eliminates the distortions in the camera images. After this, a method is developed to calculate a 3D image of the environment. For fast computation of the artificial forces the obstacles are classified into geometric primitives. The virtual charges are placed in 3D on the objects. From the charge positions and their potential value an artificial force can be calculated which acts on the end-effector. The force corrects the primary planned robot path to avoid collisions with the obstacles.

Keywords: Collision avoidance · Depth image · Sensor fusion · Virtual force field

1 Introduction

The development of robot work cells without safety fences [6] and the coming up of human-robot cooperation or coexistence in shared workspaces make collision avoidance of robots to a very important issue. In the case that humans are present in the robot workspace their safety can be guaranteed by observing the distance between robot and human and reduced velocity of the robot motion, which are supervised by safety controller. Another possibility is the limitation of the dynamic power of the robot to 80W and its static force to 150N.

However, collisions between robots and humans are only one hazard. It seems to be probably that frequently the human is the source of danger. He or she modifies the robot workspace by putting and taking objects. So a dynamic workspace occurs, which increases the risk of collisions. For that reason, the focus of this paper is not to avoid collisions between robots and humans. Instead of that, collisions with lifeless objects should be prevented by modifying the original planned

robot path. For such a scenario it is not necessary to equip the robot system with special safety controllers and safe sensors.

In this paper obstacles should be detected by range image sensor which is a part of the Microsoft Kinect RGBD camera. The Kinect camera is a popular sensor often used in robotics since some years. Something about its use can be found in [1]. To get a full 3D image and handle occlusions and distortions two cameras are used to observe the robot work cell. Every camera provides 30 fps high definition color picture and 30 fps depth image with a resolution of 512×424 . Measuring range of depth image is between 0.5 and 4.5 m. In [4] detailed investigations on fusion of multiple Kinect images can be found. However, in this article we use only a very simple algorithm. In [5] the 20-joint model of the human provided by multiple Kinect cameras directly is used for data fusion. An algorithm of collision detection for robots using multiple depth sensors is presented in [2]. In this article after detection of obstacles and potential collisions robot should not be stopped. It should be continuing its task. Thereto the robot path has to be modified to avoided collisions.

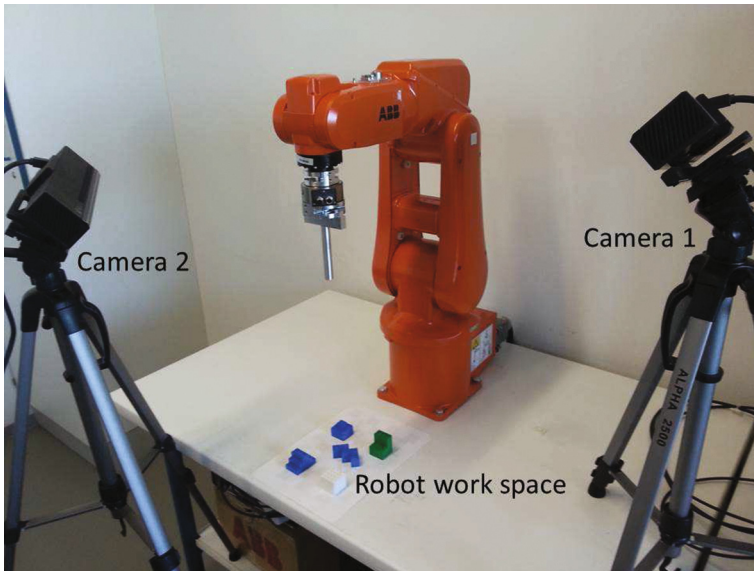


Fig. 1. Robot work cell observed by two range image sensors.

For this purpose the experimental setup which can be seen in Fig. 1 is configured. Besides the Kinect cameras, the robot work cell furthermore consists of a low payload six-axis articulated robot of type ABB IRB 120. It is controlled by IRC5-Compact robot controller. On the table where the robot is mounted, some obstacles are placed which consist of Lego bricks. The RGBD cameras are connected to a PC, which performs the fusion of the images and the detection of

the obstacles. Furthermore, the PC computes the positions of so called artificial charges. They lead to a virtual force which is used for robot path modification. The virtual force is also calculated on the PC.

This paper is organized as follows: First, in the next section a simple algorithm is presented which performs the fusion of the depth images provided by the two range image sensors. Thereafter, obstacles can be detected regarding the resulting 3D image. In Sect. 3 an approach is worked out for placing virtual charges on and in the obstacles. For this purpose, the objects are fractionized and assigned into geometric primitives. In Sect. 4 collision avoidance is verified on a practical experiment using the method of artificial force field generated by the virtual charges. This approach is also shortly described. Finally, in the last section the conclusion is given.

2 Obstacle Detection Using Two Depth Images

As already mentioned obstacles in the robot work space should be detected using two depth images collected from different perspectives. In comparison to approaches which using only one camera, fault detection as a result of occlusions and distortions should be reduced.

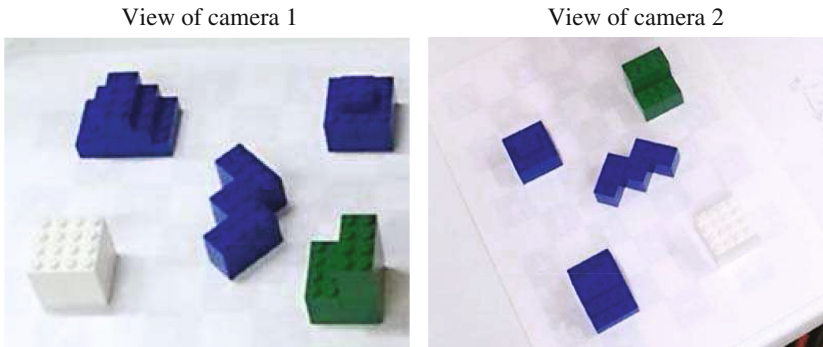


Fig. 2. Robot work cell with obstacles viewed from different perspectives. (Color figure online)

Figure 2 show clippings of the color pictures of the two Kinect cameras which are located on different positions. In the example five objects are placed in the robot workspace. The corresponding depth images from the range sensors in the Kinect cameras can be seen in Fig. 3. Both of them have the same coordinate frame. For verification of the algorithms, the contours of the obstacles are included manually in the depth images. They are represented by the red lines. As it can be seen, the real obstacles are shifted with respect to the depth images. Additionally, in parts objects are not detected completely by the range sensors

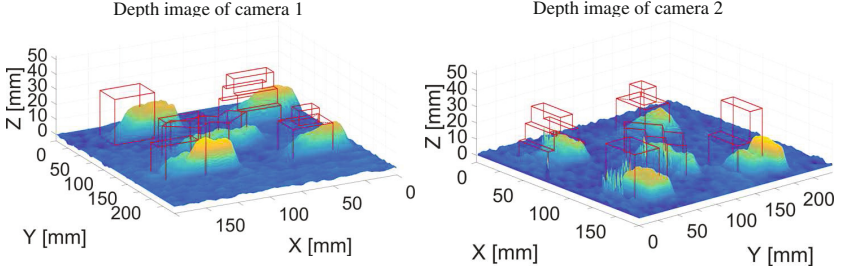


Fig. 3. Depth images of the robot work cell from different perspectives. (Color figure online)

which may be a consequence of occlusions. Now the task is to merge this two depth images to get a full 3D image of the obstacles in the robot work space.

The received pictures are distorted as a result of the perspective of the camera. Using the following algorithm the distortion can be compensated. Figure 4 shows the situation of a two dimensional point represented by vector $\mathbf{P}_0 = [x_0 \ z_0]^T$. It should be rotated around the angle α . The result is point \mathbf{P}_1 . Its coordinates can be calculated as follows:

$$\mathbf{P}_1 = [x_1 \ z_1]^T = \left[x_0 + z_0 \tan \alpha \quad \sqrt{z_0^2 + (z_0 \tan \alpha)^2} \right]^T \quad (1)$$

In the case that the point $\mathbf{P}_0 = [x_0 \ y_0 \ z_0]^T$ is given in three dimensional space, the rotation have be performed around the two angles α and β . Now point \mathbf{P}_1 yields to:

$$\mathbf{P}_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} x_0 + z_0 \tan \alpha \\ y_0 + h \tan \beta \\ \sqrt{h^2 + (h \tan \beta)^2} \end{bmatrix}, \quad (2)$$

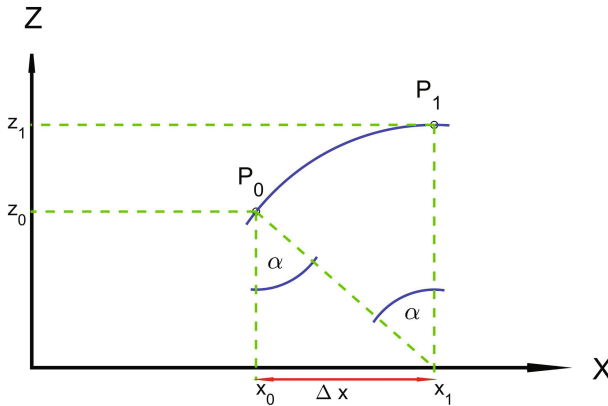


Fig. 4. Shifting of a two dimensional point.

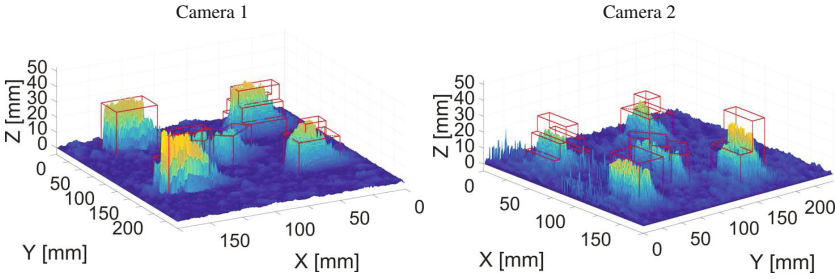


Fig. 5. Adjusted depth images.

where

$$h = \sqrt{z_0^2 + (z_0 \tan \alpha)^2}. \tag{3}$$

The angles α and β depend on the positions of the cameras. Its values are determined empirically. In the scenario which is shown in Fig. 1 the angles are $\alpha_1 = 18^\circ$, $\beta_1 = 57^\circ$, $\alpha_2 = -50^\circ$ and $\beta_2 = -25^\circ$ for camera 1 and camera 2, respectively. The results of the algorithm are shown in Fig. 5. As it can be seen, the distortion could be removed successfully. The edges which can be viewed by the cameras are now conform with the objects.

The resulting depth images are represented by matrices \mathbf{M}_1 and \mathbf{M}_2 . They can be merged to matrix \mathbf{N} as follows:

$$\mathbf{N} = \gamma_1 \mathbf{M}_1 + \gamma_2 \mathbf{M}_2, \tag{4}$$

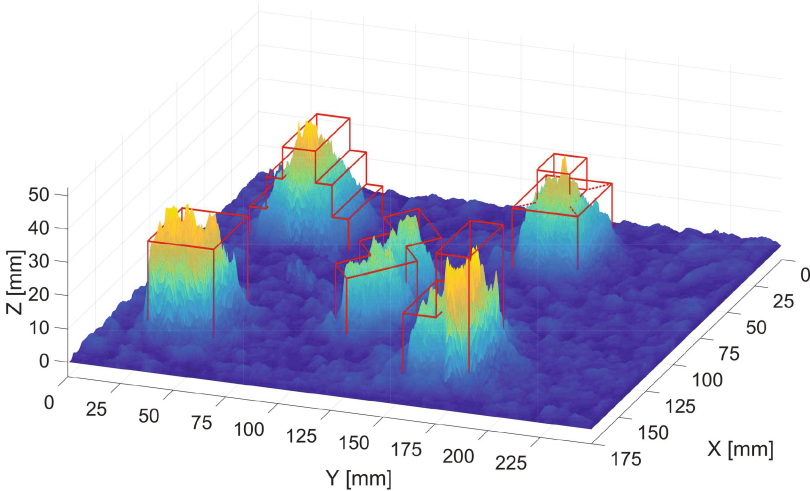


Fig. 6. Merged depth image of the example.

where γ_1 and γ_2 are correction values. They are necessary to compute the proper height of the obstacles. The values depend on the position of the according camera. In our scenario these values are $\gamma_1 = 1.1$ and $\gamma_2 = 1.4$. The final depth image of the example can be seen in Fig. 6. Here, the obstacles are successful detected.

3 Placing of the Virtual Charges

In the next step the detected obstacles in the robot work space should be supplied with virtual charges. In [8] an approach was presented, which places the charges on the whole surface of the object. For the sake of decreasing the computation time during real time path generation the number of the virtual charges should be reduced. For this purpose a new algorithm will be used.

The obstacles are fractionized with respect to the z-axis of the robot world frame. So for every obstacle we get some 2D binary images. Their number depends on the distance of factorization. Here, this distance is chosen to 10 mm. The objects found in the binary images should be assigned to geometric primitives like line, square, circle, etc. For this purpose from binary objects some characteristic features have been determined. First, the object is completely enveloped by a rectangle. Its width and high are denoted to w and h , respectively. The feature aspect ratio ar represents the maximum of the relationship between h and w :

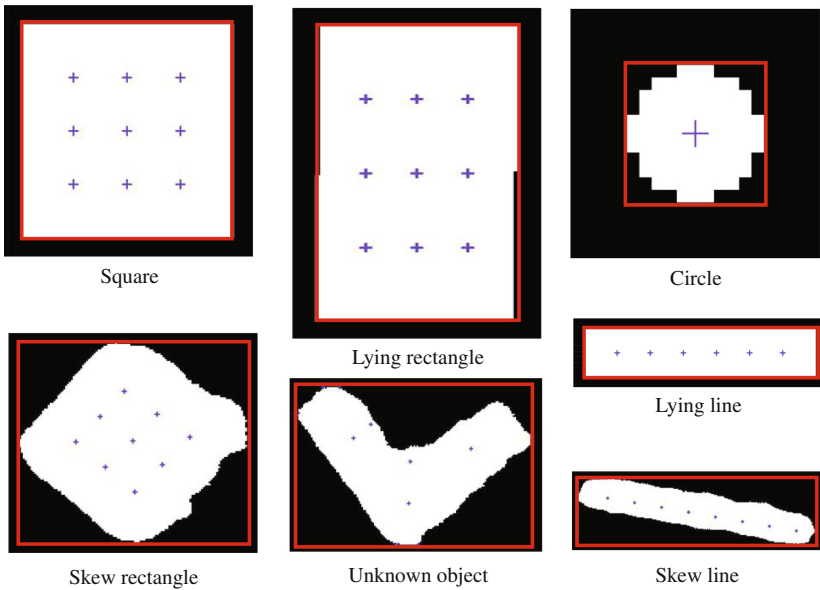


Fig. 7. Geometric primitives with enveloped rectangles and charges.

$$ar = \max\left(\frac{h}{w}, \frac{w}{h}\right). \tag{5}$$

Another feature is pixel ratio pr :

$$pr = \frac{\text{number of pixels which are assigned to the object}}{\text{total count of pixel within the rectangle}}. \tag{6}$$

For the purpose of further placing of the virtual charges also the center of the rectangle and the extreme positions where the object touch the rectangle are required. With the values of features ar and pr the objects can be classified into geometric primitives by comparing the features with different thresholds.

Placing of artificial charges depends on the kind of geometric primitive of the obstacle or rather the geometric primitive of the according layers. Figure 7 show the used selection of geometric primitives, the enveloped rectangles and the positions of charges.

As already mentioned the distance between the layers of the obstacles is 10mm in z-direction. For the example used in this paper they can be seen in Fig. 8. The algorithm assigns every layer to one geometric primitive. Thereafter, virtual charges can be placed corresponding to the rules presented in Fig. 7. They can be also seen for the particular layers in Fig. 8.

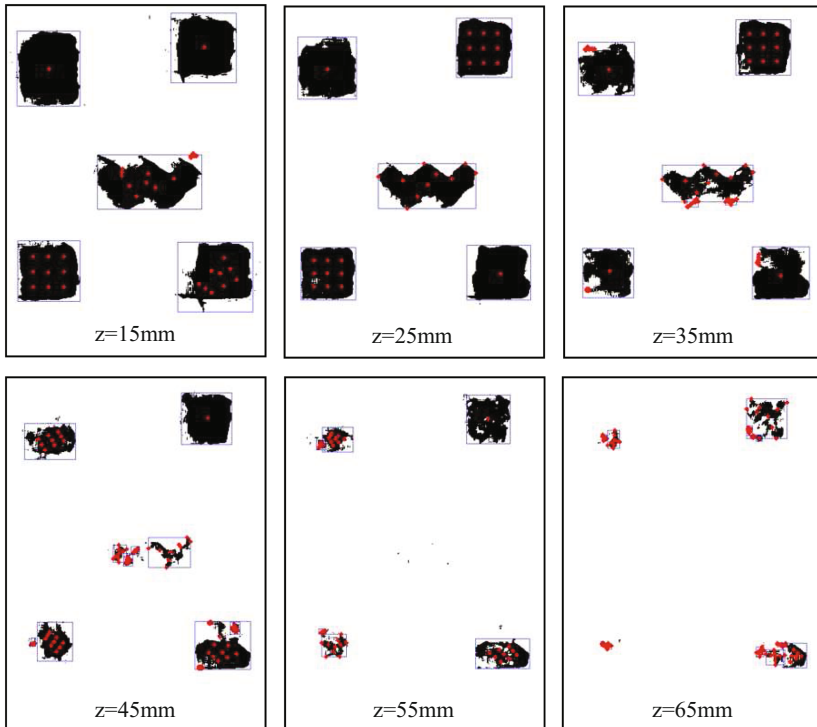


Fig. 8. Layers of the obstacles with placed virtual charges.

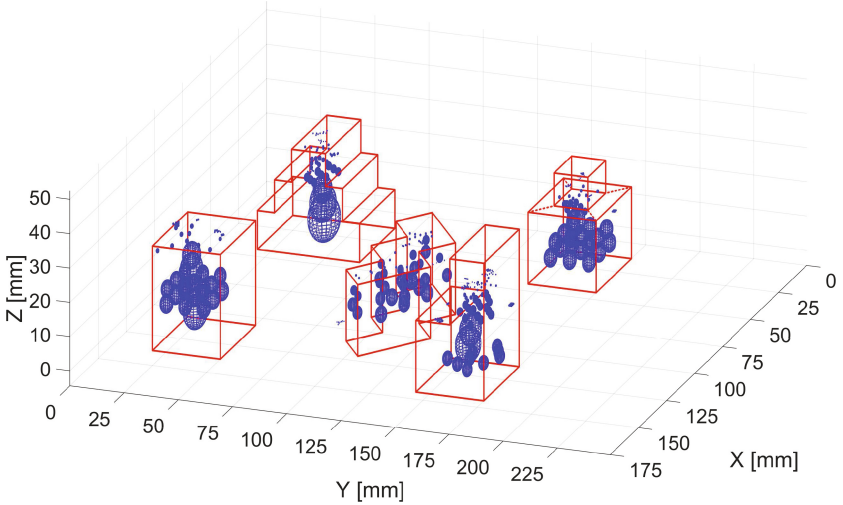


Fig. 9. Virtual charges placed in and on the obstacles.

At least, the 3D positions of the charges in the robot workspace show Fig. 9. The diameter of every charge illustrated in this figure is proportional to its potential Q_i .

4 Collision Avoidance

The approach of artificial potential fields is a well known method for path planning of mobile or stationary robots [3]. This idea can be also used for collision avoidance. The artificial force emitted by the obstacle influences the robot motion to bring the manipulator arm away from it. For real time generation of the virtual force field the approach of artificial charges was proposed [7].

Here the generation of the force field is performed by the following algorithm: The number of detected obstacle layers is denoted as c . Every object layer gets its total potential value T_i . It is a function of the area of the enveloped rectangle A_i , already explained in the previous section, and the pixel ratio pr_i of the object layer with respect to the rectangle:

$$T_i = A_i pr_i^3 = w_i h_i pr_i^3. \quad (7)$$

The value of T_i is split to all charges of the object layer which leads to charge potential Q_i :

$$Q_i = \frac{T_i}{n_i}, \quad (8)$$

where n_i is the number of charges on the according obstacle layer. Force function \mathcal{F}_i for every obstacle layer is chosen to reciprocal quadratic relationship between force and $\|\mathbf{p} - \mathbf{e}_{ij}\|$:

$$\mathcal{F}_i(\|\mathbf{p} - \mathbf{e}_{ij}\|) = \frac{Q_i}{\lambda \|\mathbf{p} - \mathbf{e}_{ij}\|^2}, \quad (9)$$

where $\|\mathbf{p} - \mathbf{e}_{ij}\|$ represents the distance between the robot end-effector and the according charge. Force function can be tuned by factor λ . The virtual force vector \mathbf{F} of all charges acting on the robot tool located at \mathbf{p} can be finally computed using the principle of superposition:

$$\mathbf{F} = \frac{1}{\lambda} \sum_{i=1}^c \sum_{j=1}^{n_i} \left(\frac{Q_i}{\|\mathbf{p} - \mathbf{e}_{ij}\|^2} \frac{\mathbf{p} - \mathbf{e}_{ij}}{\|\mathbf{p} - \mathbf{e}_{ij}\|} \right). \quad (10)$$

Positions of the virtual charges are presented by vector \mathbf{e}_{ij} , where i is the index of the obstacle layer and j is the index of the charge. For practical reasons and to reduce the computational time it may be suitable to limit the range of influence of a charge.

Now an example of collision avoidance is presented: In its task the robot should move along a programmed path of meander form which is located in x and y -direction of the robot world frame. Permanently, from the current end-effector position \mathbf{p} and the positions of the virtual charges \mathbf{e}_{ij} the virtual force vector \mathbf{F} is calculated according to (10). It should modify the robot path by vector $\Delta\mathbf{X}$. For this purpose, e.g. by simple spring behavior can be used which is a kind of mechanical impedance:

$$\Delta\mathbf{X} = \mathbf{k}\mathbf{F} \quad \text{with} \quad \Delta\mathbf{X} = [\Delta x \quad \Delta y \quad \Delta z]^T, \quad \mathbf{F} = [F_x \quad F_y \quad F_z]^T. \quad (11)$$

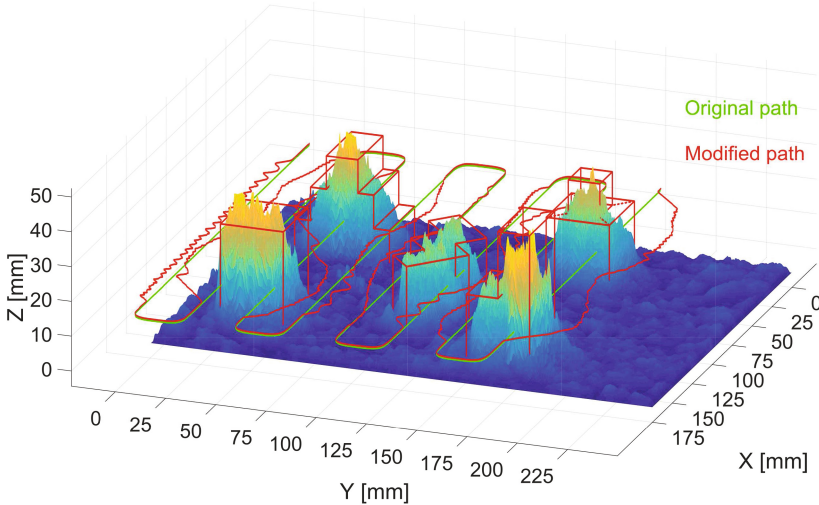


Fig. 10. Original and modified robot path as a result of virtual force emitted by obstacles.

The parameters in matrix \mathbf{k} can be understood as spring constants. The matrix \mathbf{k} should have diagonal form:

$$\mathbf{k} = \text{diag}(k_x, k_y, k_z). \quad (12)$$

Path correction is performed in path coordinate frame. This function is supported by the used ABB robot controller.

For the selected test scenario the original and the modified robot path can be seen in Fig. 10. Thus it appears that obstacles located at the robot path are circumnavigated and so collisions are successfully prevented. For better presentation here the parameter k_z is set to zero.

5 Conclusion

In this article an approach was presented which computes a 3D image from the depth images of two Kinect cameras. The cameras are used to observe the work cell of an industrial robot from different perspectives to detect obstacles. With two cameras fault detections as a result of occlusions and distortions can be avoided.

Collisions between the robot and the obstacles should be prevented. For this purpose the method of the artificial force field was used. In this paper an algorithm was proposed, which classifies the sections of the objects into geometric primitives with the objective of effective placing of virtual charges in and on the obstacles. The charges are used to compute the artificial force field. A repulsive force acts on the robot and corrects its preliminary path to move the robot end-effector away from the obstacles. The absolute value and the direction of the force vector were computed in real time.

All presented approaches were verified by practical experiments. For this purpose a suitable test scenario was developed.

References

1. El-Iaithy RA, Huang J, Yeh M (2012) Study on the use of microsoft kinect for robotics applications. In: Proceedings of the IEEE/ION position, location and navigation symposium, pp 1280–1288
2. Fischer M, Dominik H (2009) 3d collision detection for industrial robots and unknown obstacles using multiple depth images. In: Kröger T, Wahl FM (eds) Advances in robotics research. Springer, pp 111–121
3. Khatib O (1986) Real-time obstacle avoidance for manipulators and mobile robots. *Int J Robot Res* 5(1):90–98
4. Lenz C, Grimm M, Röder T, Knoll A (2012) Human-robot-workspaces fusing multiple kinects to survey shared human-robot-workspaces. Technical report, Technische Universität München
5. Morato C, Kaipa KN, Zhao B, Gupta SK (2014) Toward safe human robot collaboration by using multiple kinects based real-time human tracking. *J Comput Inf Sci Eng* 14(1):011006

6. Som F (2006) Innovative robot control offers more operator ergonomics and personnel safety. In: Proceedings of joint conference on robotics - 37th international symposium on robotics and 4th German conference on robotics. VDI Verlag
7. Winkler A, Suchý J (2009) Intuitive collision avoidance of robots using charge generated virtual force fields. In: Kröger T, Wahl FM (eds) Advances in robotics research. Springer, pp 77–87
8. Winkler A, Suchý J (2012) Dynamic collision avoidance of industrial cooperating robots using virtual force fields. In: Proceedings of the 10th international IFAC symposium on robot control, pp 265–270