Velocity control for safe robot guidance based on fused vision and force/torque data

Stefan Kuhn, Thorsten Gecks and Dominik Henrich

Lehrstuhl für Angewandte Informatik III (Robotik und Eingebettete Systeme)
Universität Bayreuth, D-95445 Bayreuth, Germany
E-Mail: {stefan.kuhn|thorsten.gecks|dominik.henrich}@uni-bayreuth.de
http: //ai3.inf.uni-bayreuth.de/

Abstract - **We present a method for securing guided robot motions in terms of human/robot cooperation. For this, we limit the maximum allowable velocity of the robot based on the distance to the human or to the next obstacle and generate the effective velocity using guidance informations provided by the interacting human. Therefore, we fuse the two heterogenous data types of a *camera* and a *force torque sensor*. The cameras are used to monitor the robot's workspace applying a difference image method. Given this obstacle information, distances are calculated between the robot and humans or objects in the environment respectively. The distance within each image is determined via an extended difference image method. The distances acquired from each camera are fused to approximate the real robot to object distance within the workspace. This distance regulates the maximum allowable velocity of the robot. The force/torque sensor provides the guidance information, i.e. amount, direction of the force and moment. This information is used to generate the robot's movement taking the maximum allowable velocity into consideration.**

Index Terms – human/robot cooperation, industrial robot, workspace supervision, difference image method, heterogeonous multisensor fusion, vision, force torque sensor

I. Introduction

Humans are skilled at complex tasks and are able to react very flexibly to unknown situations. Industrial robots are very strong, fast, persevering and accurate. In close collaboration, the skills of both can be combined.

Close collaboration means that human and robot together perform a task at the same time and at the same location. In particular, it also means that the human should be able to get into contact with the robot, for example to guide it, to transport and position workpieces with its assistance, or to guide motions within a manufacturing process.

Besides the pure guidance/cooperation intention of the human, which has to be transformed to a robot motion, safety – in particular of the human – is the most important aspect for example when transporting high loads that can cause enormous joint torques and thus are considered dangerous.

Safety means first of all that unintended contacts of the robot with the environment have to be strictly avoided. In [3] contacts are classified into the categories task, control, and collision contacts. Task contacts are needed to fulfill a given task and control contacts are needed to control the robot for example via guidance. The collision contacts cover all unintended contacts between the robot and the environment and have to be avoided.

An other aspect of human/robot cooperation is a suitable, intuitive robot behavior. A robot performing a motion with high velocity near the human is not tolerable and causes stress situations.

In Section II, we give a short overview on the state of the art regarding human robot cooperation systems, in particular the aspects of guidance and safeguarded robot motions. Section III contains the description of the proposed safeguarded robot guidance system, followed by an outline of experiments to be conducted (Section IV) and a conclusion (Section V).

II. State of the art

In the past, different approaches for human/robot cooperation have been developed. Basically two relevant sections can be identified: Generating robot movements by recognition of the human's intention, in particular via force torque sensor data, and safeguarding robot movements in terms of collision detection. Most of the research done in this area focusses on either problem, but not on the combination of both.

For example, some camera-based approaches exist, which only safeguard free robot movements. An early approach, which falls into this category is [1]. Collision avoidance is achieved by a difference image method. The robot performs motions within three different velocity levels. If there is no human present within the robot's workspace, the robot is allowed to perform its tasks with maximum velocity, else with a reduced velocity. If the human gets too close, the movement is stopped. The system presented in [5] is similar to the previous one. Only two velocity levels are implemented – normal velocity and stop. A major difference to [1] is that evasive movements are realized and thus the number of stops is reduced.

Another system is described in [11] and [14]. It is designed for the assembly of small parts. Again, cameras are used to monitor the workspace of the robot and to detect the human. The ergonomic aspects of the robot movement was the main research focus. The robot's current velocity and accelleration override is determined by the distance to the human and the angle between the robot's and the human's movement. Guided robot motions are not considered.

Besides the camera-based approaches, laser scanner techniques are used to detect the human within the workspace of the robot. In [12] the human is approximated by a 3-dimensional cylinder based on the acquired 1½-dimensional distance data. The distance calculation between this cylinder and the robot affects the maximum allowable velocity of the robot. Another feature is the guidance of the robot's TCP within tube-shaped regions inside the workspace. The combination of both is not detailed.

An approach working with short-range distance data is described in [7]. The robot is covered with proximity and haptic sensors underneath a skin made of elastic material. Also the motor currents are measured. Three different velocity levels are used. If no human is detected by the proximity sensors, the robot is allowed to perform its task with maximum velocity, else the robot velocity is reduced. If the haptic sensors indicate a contact, the robot is stopped. Guidance of the robot is only mentioned as an example of use. In that case, the robot runs with reduced velocity.

The system described in [8] implements a zero-gravity behavior as a guidance method. In that mode, the robot is completely passive and appears to be weightless. It is argued, that this behavior is inherently safe for the human and thus no additional safeguarding is needed. It is not clear, if workpieces can be transported and positioned with this behavior. Apart from this, certain guided cooperation types like assisted manufacturing processes are not feasible because the robot cannot apply any force on workpieces when in guided mode. Another implemented behavior is called *impact force control*, which causes a stop of the robot upon contact with an obstacle if the contact forces derived from the motor currents get higher than a specified value. In [13] the ergonomy of robot guidance is considered. The guidance information provided by the human operator via the force/torque sensor is interpreted depending on velocity. In case of high velocities, the hybrid controller switches to velocity control, while low velocities cause switching back to position-controlled behavior. Safeguarded guidance is not considered.

In Table , the mentioned approaches are classified by the type of movement and by the type of reaction to security violations. Conventional robots in industrial environments can not be guided and do not avoid dangerous situations for human operators.

In this paper, we present a robot guidance technique in combination with a camera-based distance-controlled robot velocity. Therefore, we fuse the camera and force/torque sensor data to derive the actual robot speed. As part of a human/robot cooperation concept described in [9], this paper resides in the section of guided motion with velocity control as safety strategy.

TABLE I: Related Work

		Motion Context	
		Free motion	Guided motion
Safety Context	None	Conventional Robot Systems	[8], [13]
	Stop	[1], [4], [7], [8]	[7]
	Velocity Control	[1] (3-stepped) [4] (2-stepped) [7] (3-stepped) [12]	This work

Fusion of camera and force/torque sensor data has also been investigated the field visual servoing. Several research groups have focussed on improving visual servoing tasks (like welding along a given contour) by fusing tool position information obtained from the camera data with contact force information from the force/torque sensor to follow the given contour on unknown surfaces [10, 15] or to improve speed and accuracy of force following a contour by predicting the tool center point trajectory [2]. However, as visual servoing is focussed on control of robot motions and not on safeguarding guided motions, approaches in this field do not solve the problem addressed in this paper. Most of all, as these approaches only supervise the robots tool center point, they can not assure a safe interaction of humans and robots for the entire robot body.

III. System Concept

This section is divided into four subsections. Beginning with a description of the distance calculation on a one-camera basis (subsection A), the subsections B and C describe the fusion of the distance data of multiple cameras regarding obstacle occlusions. The subsection D explains the fusion of vision and force/torque data for velocity regulation.

A. *Single Camera Distance Calculation*

In this section, the calculation of the distance between the robot and an obstacle is described based on a difference image. The difference image is derived by the difference of a reference image showing an empty workspace and a current image of the workspace including humans and obstacles. The geometry and the position of the robot are known. Regarding the obstacle, only the difference pixels in the difference image are known. As shown in Figure 1, obstacles that are lined up behind each other in a row from the camera's perspective are projected onto the same area within the difference image.

As no depth information is given only, the calculation of a lower bound of the minimal distance the obstacle might have with the associated projection is possible. Nevertheless, calculating this lower bound is sufficient for providing safety in human/robot cooperation. Using more than one camera as described in the following section can provide a

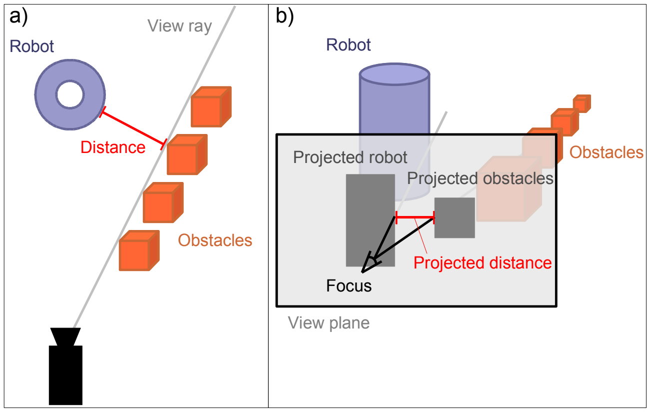


Fig. 1. a) Obstacles lined up behind each other in a ray of sight. Both robot and obstacles are surveilled by a camera. b) Obstacles are projected onto the same area within the camera's view plane. Thus, calculating a distance within the difference image method can only reveal a lower bound of the minimum distance that the obstacle might have.

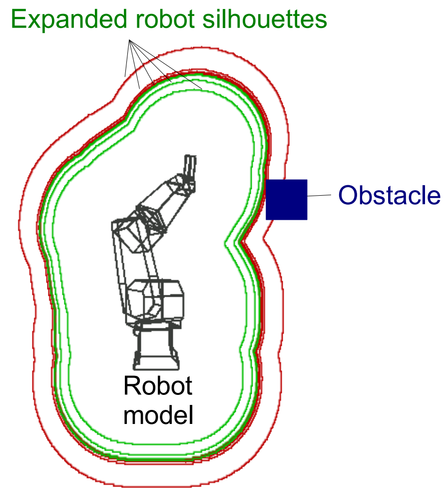


Fig. 2. The figure illustrates the robot model, an obstacle and the generated expanded shillouettes within a binary search.

more accurate distance calculation by resolving the 3-dimensional position of the obstacle.

Since the geometry of the robot is known, it can be expanded in space at a specified configuration. As a result, the projected area of the expanded robot model in the difference image increases and intersects with the projected obstacle at a particular expand radius. To determine the lower distance bound, a binary search can be applied as illustrated in Figure 2. In the figure, the shape of the robot model is illustrated at different expansion radii. The resulting silhouettes are projected onto the difference image. If the obstacle pixels are located inside a silhouette, the expand radius has to be decreased by half, else the expand radius has to be increased by half. After a defined number of steps, the caculation stops and the lower distance bound between robot and obstacle is approximately determined.

B. Multi-camera obstacle reconstruction

The following descriptions are based on the scenario illustrated in Figure 3a containing a robot volume R and an obstacle volume O. The work space is surveilled by two cameras. Figure 3c presents a view of the scene as seen by Camera 0, maybe resembling a scene with a tall mobile robot and a small human standing beneath.

In this scenario an obstacle could be occluded by the robot volume. This cannot be detected by the difference image method used here and thus has to be resolved by taking advantage of different views of the same scene from other cameras. Basically, the robot generates foreground pixels in the difference image. To detect distances to obstacle, foreground pixels generated by the robot have to be eliminated from the difference image first. Therefore a virtual robot model is projected into each camera producing a set of pixels labelled robot covering all foreground pixels caused by the robot.

Next, we need to restore obstacles in front of or behind the robot that may be occluded using the obstacle information from the other cameras. The basic assumption is that an obstacle cannot be occluded by the robot from more than θ camera perspectives. In this scenario, θ is set to 1. To determine the position of occluded obstacle volumes that might exist in front of or behind the robot volumes, we use an epipolar line method as described in [4, 6]. For each pixel labelled robot in Camera 0 (Figure 3c), the corresponding epipolar line in Camera 1 is checked for intersection with pixels identified as obstacle. If there is any intersection, the robot pixel in Camera 0 might correspond to an obstacle and is thus labelled *pseudo-obstacle* pixel. This same procedure is repeated for Camera 1, to determine the occluded obstacles within this camera.

For arbitrary scenarios, using C number of cameras in total, a robot pixel is labelled pseudo-obstacle, if more or equal to $(C - \theta)$ other cameras contain an intersection between the epipolar line corresponding to the robot pixel and an obstacle pixel. Applying this insight yields the difference image shown in Figure 3d. The resulting difference image contains pixels with three different labels: *empty*, *obstacle* and *pseudo-obstacle* as described in [6].

Pixels labelled (pseudo) obstacle can be backprojected resulting in cones within the robot workspace. Intersecting these cones with backprojected cones from other cameras results in intersection volumes. In Figure 3b, these volumes carry the labels O, P_R, P_0 and P_1. In the following, we will use this notion of intersection volumes, although the backprojection is not calculated explicitely, because it is not necessary for the described algorithm.

The volumes P_0 and P_1 could contain obstacles leading to the same reconstruction of occluded volumes as long as the projection of these obstacles does not exceed the area covered by the projection of the obstacle volume O within each camera. Providing safe obstacle distance calculation thus requires correct calculation of distances to these volumes, too.

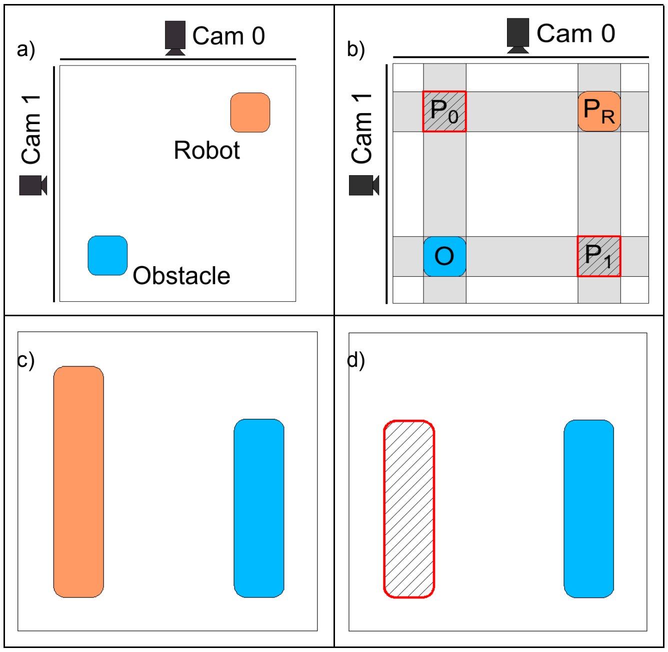


Fig. 3. Difference image generation and reconstruction of occluded obstacles. Figure a and b give us a top views of the scene surveilled by two cameras. Figure b contains a view on the backprojection cones from the robot and the obstacle (parallel projection is used for clarity). Figure c and d show the view on the scene from the perspective of Camera 0. Figure c is the original view and d is the view with pseudo-obstacle pixels (hatched area).

C. Multiple Camera Distance Calculation

As described in subsection A, the (expanded) robot volume R at its current position is projected into each camera. The resulting pixel set is tested for intersection with obstacle and pseudo-obstacle pixels. Thus, for each camera, two kinds of image-based distances are calculated: o_i is the shortest distance to pixels labelled obstacle and p_i is the shortest distance to those labelled pseudo obstacle. So, as C is the number of cameras, $2C$ distances exist in total, ranging from zero distance to some arbitrarily big distance in case that no respective pixel type intersection exists within the image. In principle, the obstacle actually closest to R not necessarily contributes to the $2C$ shortest image-based distances, since objects more distant to the robot test volume in reality may appear closer to R in the camera images.

Without loss of generality, we assume that a volume labelled V_0 has the shortest distance to the currently tested robot volume R. In a first approach, we can determine this distance by treating obstacle and pseudo-obstacle pixels in the same way, such that within each camera only the shortest distance to any of the obstacle or pseudo obstacle pixel is calculated. This can be achieved by calculating the minimum of o_i and p_i for each camera. Afterwards, the maximum of all camera-specific distances results as the shortest distance to V_0. Thus the shortest distance is:

$$dist = \max_i \left(\min \left(o_i , p_i \right) \right) \tag{1}$$

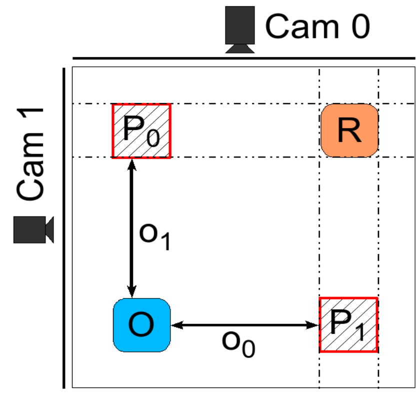


Fig. 4. Situation leading to zero distances in pseudo object distances p_i to the robot volume R.

Outline of a proof: The image distance of any projected obstacle volume to the projected robot volume R is always less than it's real world distance for all possible camera positions. It may be possible, that from different perspectives other volumes have a smaller distance within the image, but the minimum distance in a camera will never be bigger than the real world distance to the closest obstacle, so that the maximum of minimum distances within all cameras is less or equal to the real distance of the closest volume.

If Equation (1) is applied to the test situations in Figure 4, it will output the distance between the robot volume R and the intersection volume P_R as shortest distance, which is zero. The volume P_R is part of the robot volume R. R results from the intersection of the backprojected cones of the projected real robot volume. Thus R is always a superset of the real robot volume and thus could contain obstacles in reality. However, if obstacles exist within the volume R, these obstacles would be invisible in all cameras. This is a contradiction to the choice of the parameter θ, which requires any obstacle to be occluded in at most θ cameras. It is assumed that $\theta < C$, which is necessary for reasonable system operation. As a conclusion we can state that P_R can never contain an obstacle as this volume is a subset of R.

P_R results from the intersection of backprojected cones of only pseudo-obstacle-type pixels. In this way it is distinct from all relevant intersection volumes possibly containing obstacles, because those intersect with the cones of at least $(C - \theta)$ backprojected obstacle pixels.

Based on the insight that the distance calculation to P_R is based on the p_i only, we ignor the information from these distances and calculate the resulting minimum distance from the o_i only, to obtain a distance that better approximates the distance to a the relevant intersection volumes. Nevertheless, taking only the maximum of all o_i would result in the wrong distance, as the obstacle closest to the robot may be occluded by the robot in some camera images. In that case, the o_i is misleadingly calculated to a further distant object. In the following, we will deduce a rule to safely determine the minimum obstacle distances from the o_i only.

V_0 has at least $k = (C - \theta)$ corresponding distances o_i (regardless of wether they are calculated or not), as it is represented by pixels labelled obstacle in at least k cameras. As we are searching for the maximum of the calculated distances for volume V_0, we can just sort the distances o_i from all cameras in ascending order resulting in a sorted sequence $[s_1, ..., s_C]$. If we pick the distance s_k from the beginning of the sorted sequence, this distance is always less or equal to the real distance for V_0.

$$dist = s_k \qquad (2)$$

Proof: We need to distinguish two cases:

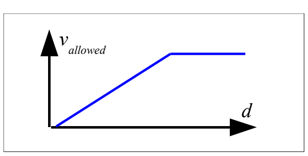
1. The first $[s_1, ..., s_k]$ are all distances to V_0. As s_k is associated with V_0 it is less or equal to the real distance to V_0.
2. Less than k members of the set $(s_0 ... s_k)$ are associated with V_0. This means that o_i associated with other V_j are smaller than the o_i associated with V_0 and were sorted in front of them. Thus s_k is even less or equal to the maximum o_i from V_0 and thus is less or equal to the real distance.

As we have seen in the previous paragraphs, both Equations (1,2) deliver a safe approximation of the minimum distance. Thus, also the maximum of both (Equation 3) is a safe approximation, but can lead to increased robot mobility, if one of the distances is not zero, while the other one is.

$$dist = max\left(s_k, \max_i\left(min\left(o_i, p_i\right)\right)\right) \qquad (3)$$

D. *Fusion of camera and force/torque sensor data*

The camera sensor system and the distance fusion has been described in the previous section. The force-following method of [13] is applied to determine the motion based on the data of the force/torque sensor. But in contrast to [13], the motion is also limited by a maximum allowable velocity based on the calculated distance within the difference images. This interrelation is proposed in [9] within a human/robot cooperation concept and can be expressed by the distance velocity diagram:



The maximum allowable velocity can be maintained by either limiting or scaling the velocity according to the measured obstacle distance (Figure 5). When applying the limiting method, the maximum allowable velocity is proportional to the measured obstacle distance. The robot will never exceed this value regardless of the force applied by the operator. When applying the scaling method, the measured distance is multiplied with the user-applied force both values. From a practical viewpoint, the scaling method seems

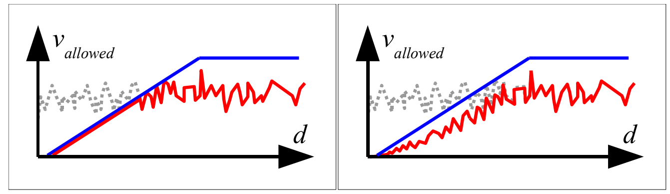


Fig. 5. Illustration of two different methods for maintaining the maximum allowable velocity (in blue). The red curve below the maximum allowable velocity represents the resulting robot velocity. The dotted curve represents the unmodified force-determined velocity. Diagram a) illustrates the limiting method, diagram b) illustrates the scaling method.

to be more ergonomic, as it behaves more smoothly on obstacle approach.

If the distance calculation includes the entire robot body, a guidance would be impossible since the robot stopps because the human operator gets too close to the robot. Therefore in [9], an adaptable area of surveillance is proposed. To achieve this, the area surrounding the point of human contact is excluded from the distance calculation. The distance calculation is then restricted to red hatched area in Figure 6.

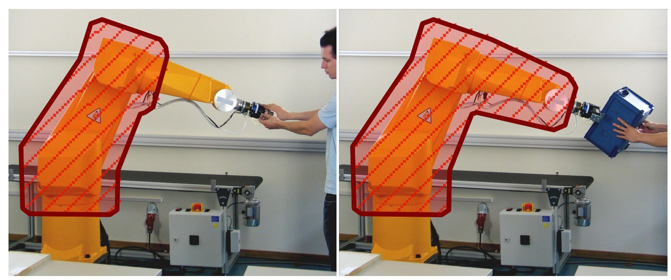


Fig. 6. The adaptable area of surveillance is neccessary to enable robot mobility in different guidance scenarios. The distance calculation is restricted to this area.

IV. Experiments

Our experimental setup comprises a Staeubli RX130 robot, firewire color cameras with VGA resolution and a force/torque sensor mounted at the robot's wrist. The image classification into foreground and background pixels is done on a seperate PC for each camera (AMD Sempron 3000+ Processor with 512 MB RAM). The classified images are gathered on a single PC (AMD Athlon 64 X2 Dual Core Processor 3800+ with 2GB RAM). Combined with the user-commanded speed from the force-follower the robot can reach up to 0.25 meters per second.

A. *First Experiment*

In the first scenario, the robot is guided backwards by an operator in the direction of a second human, who is representing a dynamic obstacle to be avoided. The robot must not collide with this human, but instead reduce its velocity based on the calculated distance right until a stop, even if the operator insists on guiding the robot towards the obstacle (Figure 7).

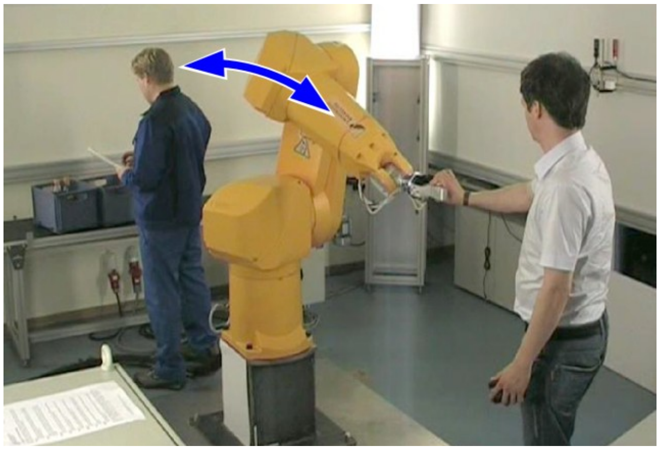


Fig. 7. First experiment scenario: The operator moves the robot towards a worker. The velocity decreases. After a short stop, the operator moves the robot back to the original position. The velocity increases.

The realized prototype utilizes three cameras running at 7.5 Hz frame rate. The image-based distance calculation is performed at a resolution of ca. 5 mm within a range of 0 to 650 mm. This created 50% load on one core of the processor. The robot speed is controlled by the detected distance and ranges from a maximum speed at a distance of 400 mm to zero speed at 104 mm. Figure 8 illustrates the utilized robot models. For velocity control the limiting method is used.

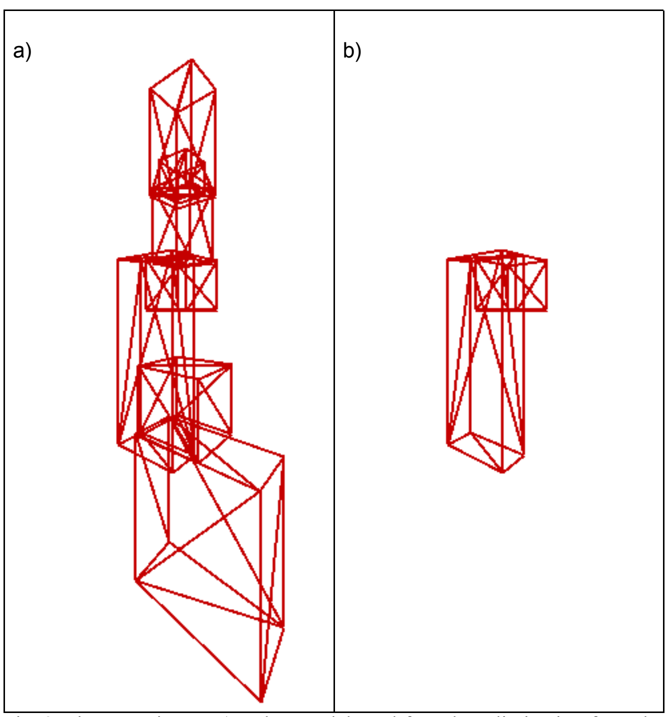


Fig. 8. First experiment: a) Robot model used for robot elimination from the computed difference image (see also Section III Subsection B).
b) Expandable robot model which defines the adaptable area of surveillance used to determine obstacle distances.

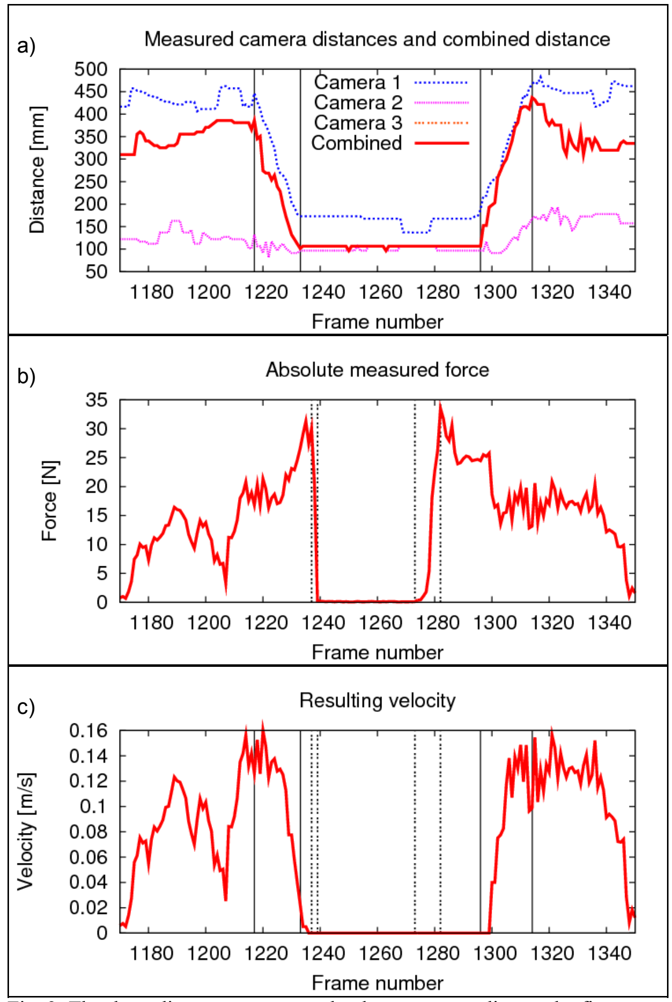


Fig. 9. The three diagrams represent the data corresponding to the first experiment. Diagram a) shows the measured distance of each camera and the calculated combined distance. In diagram b) the absolute force applied by the user is shown. Diagram c) displays the robot velocity calculated from the fused sensor data.

Figure 9 shows the results for the first scenario in several diagrams. An occlusion threshold of $\theta = 1$ is used. The graphs are synchronized on the frame number and subdivided into several sections marked by vertical (dotted) lines. The robot velocity diagram is about 3 frames out of sync because of several delaying factors: At first, the calculated distance is median-filtered (filter-length 3) to extract outliers. Based on the calculated distance, a velocity command is send to the robot. On the controller, this command is fused with the velocity set by the force following module. This resulting velocity is then send back to the PC and recorded.

In the first section up to frame 1218, the distance stays at a high level and thus the robot speed follows the operator-applied force. Beginning with frame 1218, the distance reduces until a minimum distance is reached at frame 1233. Conversely, the force applied by the operator still increases, as he notices a resistance. Then the operator stops pushing as the robot stops in between frames 1238 and 1240 to

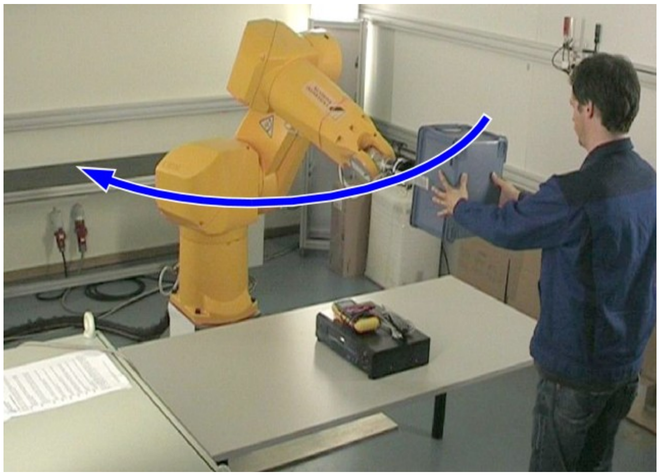


Fig. 10. Second experiment scenario: The operator moves a workpiece above an obstacle.

check for the human obstacle obviously inhibiting movement in the desired direction. As the obstacle can not move away, the operator starts pulling the robot backwards beginning with frame 1272. Again, the force increases very fast, as the robot velocity still is zero, as the human obstacle is still at close proximity. Then the operator signals the human obstacle to move away a little bit to release the robot, which starts at frame 1296. With increased distance the robot velocity is getting more and more proportional to the operator-applied force.

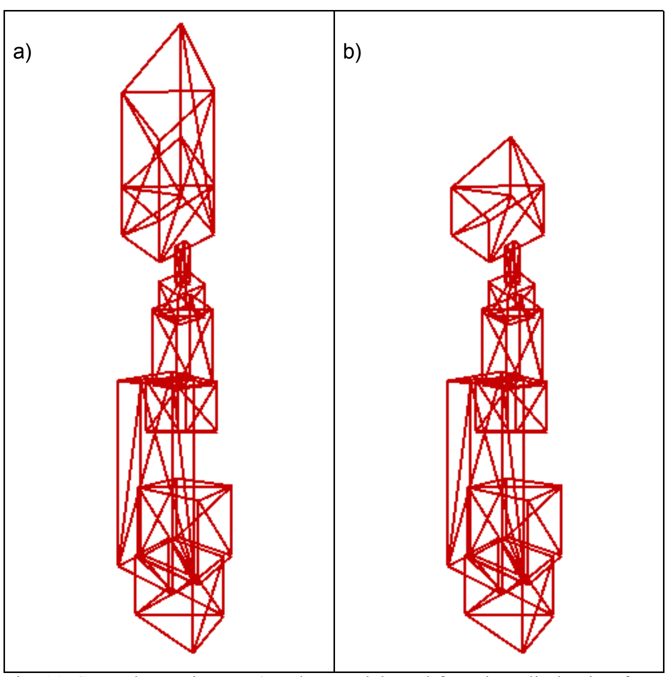


Fig. 11. Second experiment: a) Robot model used for robot elimination from the computed difference image (see also Section III Subsection B).
b) Expandable robot model which defines the adaptable area of surveillance used to determine obstacle distances.

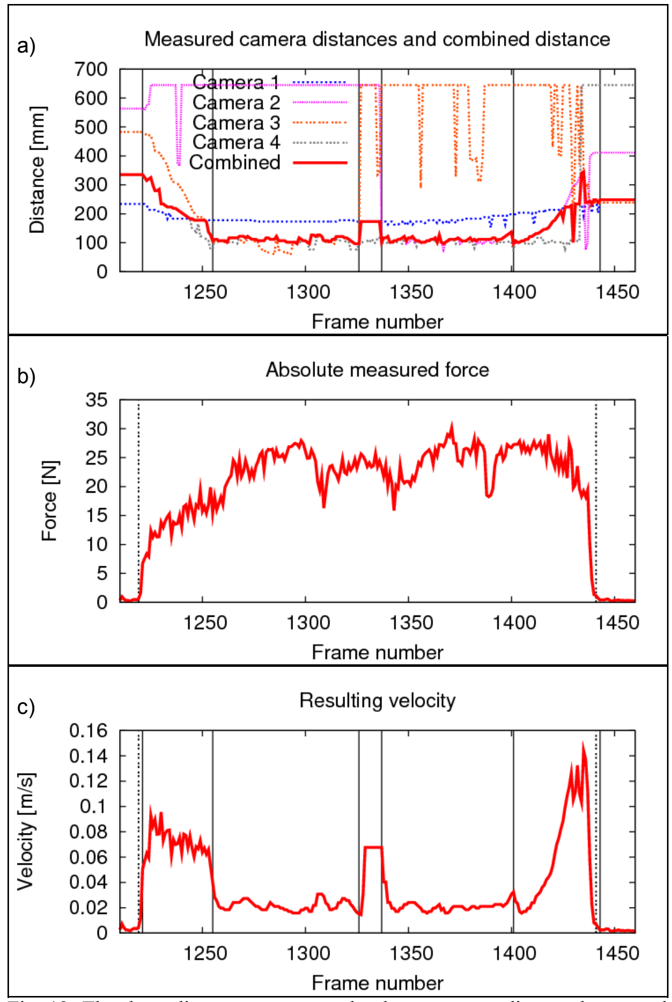


Fig. 12. The three diagrams represent the data corresponding to the second experiment. Diagram a) shows the measured distance of each camera and the calculated combined distance. In diagram b) the applied force is shown. Diagram c) display s the resulting velocity.

B. *Second Experiment*

In the second scenario, the robot is guided along an obstacle in close proximity and controls its velocity accordingly (Figure 10).

The experimental setup consists of four cameras running at a frame rate range of 4 to 7.5 Hz. All other parameters are equal to those used in the first experiment, except the minimum/maximum distance for speed regulation, which now ranges from 32 to 300 mm. Calculated distances, forces and robot velocities are shown in Figure 12. In this scenario, the scaling method is applied for velocity regulation. The parameter θ is set to two, because the camera arrangement could lead to occlusions in more than one camera. This parameter typically increases as the number of cameras increases. The robot models used for this scenario are illustrated in Figure 11. The robot model used for foreground pixel elimination (Figure 11a) contains a box at its tool center point that exceeds the actually gripped box in volume. This is to ensure that the operator touching the box is not included in distance calculation concerning the parts of his body that residue within this box (typically his fore-arms and hands).

At frame number 1219, the user starts to guide the robot and the applied force increases. Concurrently the distance to obstacles placed on a table in front of the worker decreases from frame 1221 to 1255, which causes the robot to slow down accordingly. The minimum distance then stays constant while the object passes by the obstacle on the table. Nevertheless, the combined distance is less than the real distance, which is due to the higher parameter θ. By the time the workpiece attached to the guided robot is approximately centered above the obstacles on the table, these objects are occluded within two cameras watching the scene from above. At that point the true minimum distance results from distance calculation causing the robot speed to jump to a higher level from frame 1326 to frame 1337.

Beginning with frame number 1401, the distance to the obstacles increases again and the robot speed regulates proportionally as the user applied force stays at a hight level. As the force decreases at frame 1441, the speed of the robot follows this preset.

V. Conclusion

We presented the fusion of the vision and force/torque sensor data for securing guided robot motion in terms of human/robot cooperation. The vision sensor provides the data to apply a difference image method. In order to calculate minimum distances between robot and object, the known robot model is expanded until it intersects with an object in the difference image. Then the distance information from multiple cameras is combined to increase the calculated distance accuracy even in case of object occlusions. Therefore a threshold value, which indicates the maximum number of cameras that may not see the object caused by occlusions at a time, is used. The determination of this threshold value may be automized in future work. The calculated distance is used to define a safe maximum velocity of the robot, which can be determined by limiting the velocity or by scaling it. The general motion generation is done by a force-following method, which is bounded by the safe maximum velocity. An adaptable surveillance area is used to enable robot mobility.

References

[1] Baerveldt A.-J.: „Cooperation between Man and Robot: Interface and Safety“ In: IEEE International Workshop on Robot and Human Communication, Tokyo, Japan, September 1 -3, 1992

[2] Baeten J., Schutter J.D.: "Hybrid Vision/Force Controlat Corners in PlanarRobotic-Contour Following", IEEE/ASME Transactions on mechatronics, Vol. 7, No. 2, June 2002

[3] Burghart C., Yigit S., Kerpa O., Osswald D., Woern H.: „Concept for Human Robot Co-operation Integrating Artificial Haptic Perception“. Proceedings of IAS-7, Marina del Rey, USA, 03. 2002, p. 38-45.

[4] Ebert D., Henrich D.: „Safe Human-Robot-Cooperation: Image-based collision detection for Industrial Robots" In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1826-1831, Lausanne, October 2 - 4, 2002

[5] Ebert D.: “Bildbasierte Erzeugung kollisionsfreier Transferbewegungen für Industrieroboter“ PhD Thesis, Informatics Faculty, University of Kaiserslautern, Germany, 2003

[6] Gecks T., Henrich D.: "Human-robot cooperation: Safe Pick-and-Place Operations", 14th IEEE International Workshop on Robot and Human Interactive Communication,August 13-15, 2005, Nashville, Tennessee USA

[7] Heiligensetzer P.: „Sichere Mensch-Roboter Kooperation durch Fusion haptischer und kapazitiver Sensorik“, Dissertation, University of Karlsruhe, 2003

[8] Heinzmann J., Zelinsky A.: „Building Human-Friendly Robot Systems“, In: Proceedings of International Symposium of Robotics Research ISRR'99, Salt Lake City, USA, 9-12 October, 1999

[9] Henrich D., Kuhn S.: „Modeling Intuitive Behavior for Safe Human/Robot Coexistence and Cooperation“, In: IEEE International Conference on Robotics and Automation, Orlando, Florida, USA, 15-19 May 2006

[10] Hosoda K., Igarashi K., Asada M.: "Adaptive Hybrid Control for Visual Servoing and Force Servoing in an Unknown Environment", IEEE Robotics and Automation Magazine, Vol.5, No.4, pp.39-43, 1998

[11] Schulz O.: „Image-based 3D-surveillance in Human-Robot-Cooperation“, In: Modern Trends in Manufacturing. Second International CAMT Conference, Wroclaw, Poland: Oficyna Wydawnicza Politechniki Wroclawskiej, 2003, 20-21 Feb. p.327-34

[12] Som F.: „Sichere Steuerungstechnik für den OTS-Einsatz von Robotern“, In: 4. Workshop für OTS-Systeme in der Robotik (Sichere Mensch-Roboter-Interaktion ohne trennende Schutzsysteme), Stuttgart, 2. Nov. 2005

[13] Stolka P., Henrich D.: „A Hybrid Force Following Controller for Multi-Scale Motions“, In: SYROCO 2003, 7th International Symposium on Robot Control, 1-3 September 2003, Wroclaw/Poland

[14] Thiemermann S.: „Direkte Mensch-Roboter-Kooperation in der Kleinteilmontage mit einem SCARA-Roboter“, Dissertation, University of Stuttgart, 2005

[15] Xiao D., Ghosh B., Xi N.,Tarn T.J.: "Sensor-Based Hybrid Position/Force Control of a Robot Manipulator in an Uncalibrated Environment" IEEE Transactions on control systems technology, Vol. 8, No. 4, July 2000