# A Safe Fault Tolerant Multi-View Approach for Vision-Based Protective Devices

Antje OBER and Dominik HENRICH
Lehrstuhl für Angewandte Informatik III (Robotik und Eingebettete Systeme)
Universität Bayreuth, D-95440 Bayreuth, Germany
E-Mail: {antje.ober | dominik.henrich}@uni-bayreuth.de
http://www.ai3.uni-bayreuth.de/

## Abstract

*We present a new approach that realizes an image-based fault tolerant distance computation for a multi-view camera system which conservatively approximates the shortest distance between unknown objects and 3D volumes. Our method addresses the industrial application of vision-based protective devices which are used to detect intrusions of humans into areas of dangerous machinery, in order to prevent injuries. This requires hardware redundancy for compensation of hardware failures without loss of functionality and safety. By taking sensor failures during the fusion process of distances from different cameras into account, this is realized implicitly, with the benefit of no additional hardware cost. In particular we employ multiple camera perspectives for safe and non-conservative occlusion handling of obstacles and formulate general system assumptions which are also appropriate for other applications like multi-view reconstruction methods.*

## 1. Keyords

Safety, vision-based protective devices (VBPDs), fault tolerance, redundancy, replication, distance calculation, occlusion handling, occlusion masks, protection zone, multi-view reconstruction, human-robot cooperation

## 2. Introduction

An important application field of sensor technology for surveillance in industrial environments is safeguarding humans and machines during hazardous work processes by so-called *protective devices* [13] (equipment of machines). The aims are prevention of human injuries, material protection (*safety issues*) as well as detection of unauthorized accesses (*security issues*). In either case the protective device has to safely detect *unknown objects* intruding into a hazardous machinery area.

There exist different kinds of *certified safe* protective devices. Separating protective devices [2], e.g. protection doors, separate a hazardous area completely from access through physical barriers. Instead, non-separating protective devices use sensors to detect objects, e.g. pressure-sensitive devices [3] like shut-off mats or optoelectronic devices [14] like light barriers. These sensors deliver information about the state (occupied/non-occupied) of the so-called *detection zone*, which is the area that a sensor can monitor safely. With help of this information, the system can react (via a safety controller) in an appropriate way by causing the machine to slow down or to trigger an emergency shutdown.

The previously mentioned protective devices have limited dimensions of the detection zones and are not able to monitor larger 3D volumes that surround the complete machinery. Thus they are typically installed at entrances to areas of dangerous machinery. Due to the coarse approximation of the hazardous area (machinery volume) the machinery has to be stopped during operating mode more often, than actually necessary (*availability issue).* This is an important disadvantage for industry and can be avoided with the aid of recently developed camera systems, so-called *vision-based protective devices (VBPD)* [15]. These systems model *protection zones* (parts of the image) within the detection zone (complete image) and detect unknown objects, which might intrude into a protection zone via image processing algorithms e.g. change detection methods. An overview of change detection methods can be found in [5]. Another important benefit for availability arises as such system can be enabled to safely register when objects which have intruded into the protection zone are leaving the hazardous area again, so that the system can recover safely in an autonomous way. In comparison, the non-vision protective devices require manual systems reactivation by a user after a shutdown.

A certified VBPD is the product SafetyEYE [19], which consists of a tri-ocular stereo camera system that can be mounted on top of a working cell (top-down view). A drawback of the system is the limitation to one perspective which cannot handle optical occlusions in the scene e.g. through parts of machinery, so the system has to assume that all unknown objects are always visible within the monitored area. In result, this assumption does also mean that the system cannot sufficiently approximate *dynamic protection zones* to dynamic machinery such as industrial robots.

Existing *uncertified non-safe* camera systems offer promising improvements in handling occlusions by the use of multiple camera perspectives (multi-view). In [4, 8, 12] a dynamic protection zone is realized by a 3D hull of a robot for use in an image-based collision detection method of the robot with the environment for the goal of human-robot cooperation. Occlusions caused by obstacles in one camera are reduced with the help of epipolar geometry information from the other cameras. In [16] the collision detection has been replaced by distance computations which result in a shortest distance of all unknown sensor-detected objects to a protection zone. The general advantage of using distances is that the process of approaching objects can be recorded and evaluated for appropriate systems reaction such as speed regulation of a robotic arm. Occlusions have been examined in detail for multi-view voxel reconstruction methods for further improvement of the systems availability [17]. Also the use of time of flight cameras with polyhedron reconstruction methods has been considered for the regarded scenario [6].

Various other multi-view approaches cope with occlusions in 3D reconstruction methods, e.g. [9, 20]. However they do not cope with the regarded scenario of VBPDs (intrusion detection). Moreover they limit possible occlusions sizes, so that occluded objects are always partially visible in each camera [10] or assume occlusions conservatively as objects. Those approaches have a limitation in the approaches applicability, especially for our scenario.

Despite the obvious advantages for machinery availability, *multi-view VBPDs* have not been realized yet. One reason is the challenging requirement of safe object detection components (change detection methods) which still lead to availability issues due to changing backgrounds or illumination changes in the environment. Some different image processing approaches with assumptions, e.g. to the image background (static pattern) or to the objects appearance (special reflecting material) have been discussed in [7] to guarantee safe object detection. Though we are also working on that field, this paper does not discuss any change detection method. It focuses on how the information is processed after successful object detection.

Another reason for the non-existing multi-view VBPDs is the high software development effort to fulfill the special requirements of safe software design and implementation, so that no software fault can bring the protective device into an unsafe state. This can cause high economic cost, rising with the complexity of algorithms, in particular when reconstruction methods are employed.

However we believe that the largest barriers for industry are the economic cost of redundant hardware structures for safety reasons, especially for the camera sensors which lead to lower profitability for such complex multi-view systems. Thus, in this paper we come up with a solution that exploits the diverse perspectives of the cameras in order to handle occlusions, while providing at the same time a hardware fault tolerance of the distributed camera sensors at no additional hardware cost. We focus on a state of the art method for image-based distance calculation to unknown objects [16] for limitation of complexity though our approach can be transferred to other systems which use reconstruction methods as well. In comparison to [12] and [16] we formulate more general assumptions for safety distance calculations which are also valid for other scenarios.

In the following we present our idea of hardware fault tolerance in a multi-view system (Section 3). Then we give a detailed problem description of safe distance computation and occlusion handling (Section 4) and show the solution without and with occlusions by deriving necessary system assumptions (Section 5). Finally we present the safe functionality of our method in an experiment (Section 6).

## 3. Preliminary considerations

In order to realize protective devices with multi-view camera systems which handle obstacles, different ways exist to handle faults in the software or hardware. One preferable method is to prevent systems operation stops caused by failures (availability issue) of single system components through fault tolerant software and hardware structures. Focusing on the latter in this paper, these can be implemented to any form of hardware duplication like hardware redundancy (using one or more spares for a component which takeover when the component fails), hardware diversity (different implementations of the same specification) and replication (parallel processing of the same tasks and choice of the result on basis of a quorum). Such a fault tolerance enables the systems functionality, possibly on a reduced level, for a certain time until the system has detected the fault itself or by a user.

As we need multiple perspectives for the applicability in occluded scenes, by a duplication of each camera sensor, the economic cost would raise immense proportional to the amount of perspectives. Thus our approach is to use an overall replication for the whole system, which is implicitly given when the system consists of more than one camera. We determine a tolerance factor $\tau$ which considers that $\tau$ cameras may fail at any time. The factor is applied during the process of distance information fusion from all cameras. Although this approach rejects information at each time step, it has the advantage of handling $\tau$ errors no matter in which cameras they occur. In comparison, systems which realize redundancy through sensor duplication do only have a failure tolerance $\tau = 1$, which means, when both hardware devices of one sensor fail, the complete system fails, too. Moreover the same $\tau$ redundant cameras can be integrated in separated surveillance systems in order to further reduce economic cost as in industry it is common that several areas which have to be monitored are located very close to each other.

This approach ensures uninterrupted functionality of a camera system when hardware failures occur in up to $\tau$ cameras.

## 4. Problem description

Given a set of $n$ calibrated cameras (e.g. grayscale, color or depth camera) which synchronously capture images at times $t_i$ and a pinhole camera model, the monitored space $A_c$ in $\mathrm{R}^3$ of each camera $c \in \{1,\ldots,n\}$ is determined by a projection function $proj_c(x)$. This function produces a pixel $p$ in the camera for a point $x$ in $\mathrm{R}^3$ or an empty set, when the projected point does not hit the sensor area.

$$proj_c(x) = \begin{cases} p, \text{if } x \text{ is projected onto a pixel of } c \\ \varnothing, \text{otherwise} \end{cases} \qquad (1)$$

$A_c$ can then be described as the set

$$A_c = \{x \mid proj_c(x) \neq \varnothing\} \qquad (2)$$

The back-projection of all pixels $p$ from a camera $c$ into $\mathrm{R}^3$ results in a finite set of view pyramids $B_{p,c}$ with no gaps between them when assuming ideal pixels.

$$B_{p,c} = backproj_c(p) = \{x \mid proj_c(x) = p\} \qquad (3)$$

The view pyramids of different cameras intersect each other in $\mathrm{R}^3$ and subdivide the space into non-empty disconnected subspaces $S_i$ (Figure 1) (which are also called conexels [1]). Any two points $x$ and $y$ in $S_i$ are projected onto the same pixel in a monitoring camera $c$ or the empty set, in case the subspace is not monitored at all.

$$\forall c, \forall x \in S_i, \forall y \in S_i, x \neq y : (proj_c(x) = proj_c(y)) \qquad (4)$$

To complete the definition of $S_i$, any two points of different $S_i$ have to differ in at least one monitoring camera. Otherwise $S_i$ could be constructed which consist only of two elements.

$$\forall x \in S_i, \forall y \notin S_i, \exists c : (proj_c(x) \neq proj_c(y)) \qquad (5)$$

This definition also includes a specific subspace $S_\varnothing$ which needs not to be geometrically connected but has the property that it is not monitored by any of the $n$ cameras.

$$S_\varnothing = \{x \mid \forall c : proj_c(x) = \varnothing\} \qquad (6)$$

For the following approach only those subspaces $S_i$ are considered whose elements $x$ are projected onto each camera sensor, also called *common supervision space A.*

$$A = \{S_i \mid \forall x \in S_i, \forall c : (proj_c(x) \neq \varnothing)\} \qquad (7)$$

Within $A$ there exist protection zones $Z_j \subseteq A$ which are monitored completely by all cameras and which can be static or dynamic. We concentrate on a specific protection zone $Z$ because all following computation steps have to be computed identical for any protection zones $Z_j$.

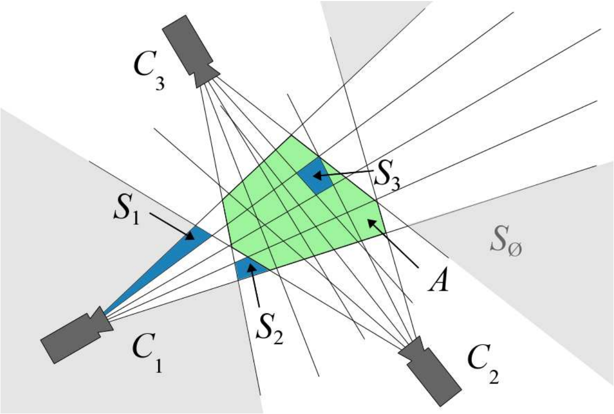

Figure 1: A schematic 2D illustration of subspaces $S_i$ which result from pixel back-projection into $\mathrm{R}^3$. The unconnected subspace $S_\varnothing$ (grey) is not monitored from any camera. The subspaces $S_c$ (blue) are seen from $c$ cameras. The monitored space $A$ (green) for the system contains subspaces which are completely projected into each camera.

Furthermore there exist occlusions in cameras caused by opaque static or dynamic obstacles $O_k$ with a modeled geometry and volume in $\mathrm{R}^3$, which is known at every time $t_i$, e.g. a robot, wall or table. It is not necessary to model static obstacles which are located behind $A$ from a cameras point of view.

$$\forall k \exists i : O_k \cap S_i \neq \varnothing \qquad (8)$$

If only one point of the volume that is occluded for a cameras view is projected on a corresponding camera pixel via (1), this pixel has to be determined conservatively as occluded pixel, for safety reasons. This results in a set of *occlusion pixels* $P_{\mathrm{occ},c}$ in each camera which can also be the empty set.

$$P_{\mathrm{occ},c} = \{p \mid \forall k \exists x \in O_k : proj_c(x) = p\} \qquad (9)$$

Projecting the occlusion pixels back into $\mathrm{R}^3$, all $S_i \subseteq A$ are assigned to adjacent groups, which differ in their degree of sensor visibility. The visibility encodes the number of cameras in which all points of $S_i$ are projected onto non-occluded pixels.

Though the real degree of visibility for a given $S_i$ can vary for different elements $x$, $y$ of $S_i$ the overall *visibility*

$V(S_i)$ is assumed conservatively.

$$V(S_i) = v \text{ with } \forall k, \forall x \in S_i, \forall y \in O_k : \\ (|\{c \mid proj_c(x) \neq proj_c(y)\}| = v) \tag{10}$$

There might exist opaque object volumes $U_m \subseteq A$ which are unknown a priori and therefore not modeled. They are detected by the sensors and do not intersect any obstacles.

$$\forall m \forall k : U_m \cap O_k = \varnothing \tag{11}$$

The visibility of objects is affected for every camera when located in occluded subspaces $S_i$ corresponding to Eq. 10. All parts of unknown objects which are projected onto non-occluded pixels for each camera form the set $P_{\text{unkn},c}$ of *unknown object pixels*. These pixel sets are constructed via a background subtraction algorithm from a set of synchronized camera images at times $t_i$. We assume an ideal background subtraction method.

$$P_{\text{unkn},c} = \\ \{\text{p} \mid \forall m, \exists x \in U_m : (proj_c(x) = p \wedge p \notin P_{\text{occ},c})\} \tag{12}$$

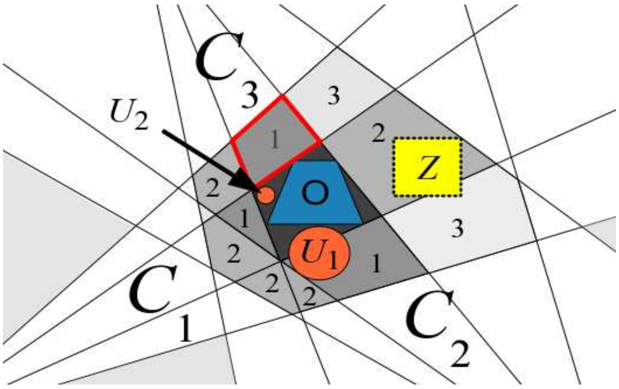

Figure 2: The visibility of the subspaces $S_i$ results from the back-projection of occluded pixels from $P_{\text{occ},c}$ from each camera $c$ in $\mathrm{R}^3$ which are produced by the occluding obstacle $O$ (blue). Shown also is an unknown object $U_1$ and a protection zone $Z$. The numbering of the subspaces represents the amount of cameras in which a corresponding subspace is visible. The visibility of the subspaces affects the visibility of the objects.

In Figure 2 the $S_i$ are grouped due to their degree of visibility $v$ in all cameras which is restricted by the dynamic obstacle $O$ (blue). To understand the figure it has to be noted that, given a dynamic obstacle, a camera is blind for the whole back-projection view pyramids of the occluded pixels, even if the subspace is lying in front of an obstacle from the cameras point of view (red bordered zone). This assumption has to be applied because we do not consider information about the appearance of known models in the images and therefore we are not able to distinguish between known dynamic objects and unknown objects in the process of background subtraction.

After discussion of the basic conditions we can formulate the goal: We are searching for the shortest fault safe distance $d_{\text{safe}}(\tau)$ to the closest unknown object $U_{\text{cl}}$ for a protection zone $Z$ (Figure 2 yellow) within $A$ by taking into account the discussed restrictions to the visibility caused by occluding obstacles. That safe distance has to be smaller or equal to the minimum distance between any points of $U_{\text{cl}}$ and the protection zone, also in case of $\tau$ failing cameras.

$$d_{\text{safe}}(\tau) \leq \min_{\forall p \in U_{cl}, \forall q \in Z} \{\|p - q\|\} \tag{13}$$

# 5. Approaches for safe distance calculation

After the detailed problem description for safe distance computation in the last section, we will develop a safe distance computation for all unknown objects to a protection zone for scenes without occlusions, only regarding the redundancy, in section 5.1. Via formulation of necessary system assumptions for occlusion handling, occluded scenes will be addressed in section 5.2.

## 5.1. Approach for non-occluded scenes

The real minimum distance $d_{\text{real}}$ between unknown objects $U_m$ and a protection zone $Z$ can have the maximum value $d_{\max}$ which is defined by the limits of the monitored area $A$ as well as the distance of the protection zones to these limits.

$$d_{\text{real}} = \min_{\forall m, \forall p \in U_m, \forall q \in Z} \{\|p - q\|\}, \text{ with } 0 \leq d_{\text{real}} \leq d_{\max} \tag{14}$$

As no occlusion exists in the scene, it is assumed that every point of unknown objects inside $A$ is projected onto a camera pixel resulting in an element of $P_{\text{unkn},c}$ in each camera via Eq. 1. We have to back-project them into $\mathrm{R}^3$ which results in back-projected view pyramids $B_{p,c}$ (Eq. 3) as we do not have information of the exact location of the objects within these pyramids. The back-projection pyramids from all elements of $P_{\text{unkn},c}$ include all points of the unknown objects $U_m$. Specifically, the union of all view pyramids of all cameras contains the complete object volumes $U_m$:

$$B_{united} = \bigcup_{\forall p \in P_{\text{unkn},c}} B_{p,c} \cap A \tag{15}$$

$$\forall m : (\overline{B}_{united} \cap U_m = \varnothing), \overline{B}_{united} \cap A \neq \varnothing \tag{16}$$

The shortest distance from a projection zone to a back-projection pyramid $d_{\text{back},p,c}$ (which can be seen as a kind of bounding box) is thus a lower bound of the distances to all points within the pyramid. In result, we get a safe lower distance $d_{\mathrm{cam},c}$ to all unknown objects for the camera $c$ by computing the minimum of the shortest distances to all pyramids $B_{p,c}$ from points of $Z$.

$$d_{\mathrm{back},p,c} = \min_{\forall b \in B_{p,c}, \forall q \in Z} \|b - q\| \tag{17}$$

$$d_{\mathrm{cam},c} = \min_{p \in P_{\mathrm{unkn},c}} (d_{\mathrm{back},p,c}) \tag{18}$$

$$d_{\mathrm{cam},c} \leq d_{\mathrm{real}} \tag{19}$$

There exist different methods for distance calculation in images when geometry of models is available through camera calibration. For instance in [16] an approach is presented in which the 3D models of a dynamic protection zone $Z$ (the robotic arm) is expanded with known radii until an intersection of the projected robot pixels with pixels of unknown objects is detected in one of the camera images. The radius reached before the intersection is then assumed as minimum distance. In [11] the distance is computed in $\mathrm{R}^3$ via simple vector algebra where the back-projected view pyramids $B_{p,c}$ are approximated by vectors through the pixel centers. The projection zones are modeled by spheres and the vectors to the sphere centers are chosen for efficient distance calculation. Various methods can be applied to generate safe back-projection pyramids $B_{p,c}$ which include all points of unknown objects (Eq. 15, 16). For instance, a solution for the latter method of distance calculation would be to subtract an offset from each $d_{\mathrm{back}}$ (Eq. 17) to take into account the angle between the vector through the pixel center and the pixel borders.

After such a safe computation of $d_{\mathrm{cam},c}$ for every camera $c$, the information of the cameras can be combined. Since all camera distances are each a lower bound of $d_{\mathrm{real}}$, without loss of safety, the best approximation can be reached by choosing the maximum.

$$d_{\mathrm{safe}} = \max_{c}(d_{\mathrm{cam},c}) \tag{20}$$

Now we want to consider hardware failures of $\tau$ cameras with $\tau \in \{0, ..., n-1\}$. Those failures may appear simultaneously caused by pixel errors or complete sensor dysfunctions. A safety issue for the system arises when the failure results in distances which are larger than $d_{\mathrm{real}}$. So we can formulate the following corruption function $d_{\mathrm{fail},c}$.

$$d_{\mathrm{fail},c} = \mathit{failure}(d_{\mathrm{cam},c}) > d_{\mathrm{real}} \tag{21}$$

Our approach for handling these possible failures is to assume the worst case in which all of maximal $\tau$ cameras fail and all these failures result in safety issues (Eq. 21). With help of a sorted list of all $n$ elements of $d_{\mathrm{cam},c}$ in ascending order, the $\tau$ biggest values are rejected and the selection of the distance at position $s_{(n-\tau)}$ ensures that $d_{\mathrm{safe}}(\tau)$ is a lower bound of $d_{\mathrm{real}}$.

$$L_{\mathrm{sorted}} = [s_1, ..., s_c] \tag{22}$$

$$d_{\mathrm{safe}}(\tau) = s_{(n-\tau)} \tag{23}$$

## 5.2. Approach for occluded scenes

In the following the influence of occluding obstacles on the safe distance calculation will be discussed using the given pixel sets of occlusions $P_{\mathrm{occ},c}$ and unknown objects $P_{\mathrm{unkn},c}$, before we present the equation for safe distance computation in the end of this section.

Detailed analysis of situations with occlusions showed that the difficulty of handling occlusions is based on the assumptions in Equation 16. This assumption can be transformed for all cameras into the following principle systems assumption.

$$\forall m : \left( \left| \left\{ c \middle| \forall x \in U_m : \left( proj(x) = p \wedge p \neq P_{occ} \right) \right\} \right| \geq 1 + \tau \right) \tag{24}$$

For each $U_m$ there always need to exist at least $1+\tau$ cameras which have the information of the complete object given as set of $P_{\mathrm{unkn},c}$, so that they can detect a distance which is safely below the real distance under the consideration that $\tau$ cameras might fail.

This principle assumption is explained by the examples in Figure 3 which shows a monitored space of three cameras as well as back-projected pixel sets (light red hatched and light blue). The unknown object $U_1$ is not visible for camera $C_2$ but can be completely observed from the other two cameras. Correctly, the both resulting distances are smaller than the real distance. The object $U_2$ is partially occluded in the cameras $C_1$ and $C_3$ and completely occluded for camera $C_2$. The corresponding distances are each bigger than the real distance from $Z$ to $U_2$, thus the resulting distance is not safe and the principle assumption from Eq. 24 is not fulfilled because of missing parts of the object in the corresponding pixel sets (light red hatched). The very small object $U_3$ represents a case which is even worse. The object is not visible for any sensor because of the occlusions. Therefore the principle assumption is not fulfilled at all.

A simple way to get all those problems under control would be to transform all occlusion pixels into object pixels for distance creation. Despite the guarantee for safe distances this approach is very conservative and would limit the usability of the system because of its imprecise distance approximation.

Better results are achieved by applying the following assumptions and processing steps which finally result in fulfilling the principle assumption from Eq. 24.

First of all it has to be ensured that objects do not get lost in all sensors simultaneously like $U_3$ in the situation in Figure 3. The problem is that there always exist occlusions around obstacles $O_k$ which are not visible by any sensor because of the visual hull [18]. Nevertheless, as we want to use the system for human or hand detection we can assume a minimum object size and shape that cannot hide completely in occluded volumes of visibility $v = 0$.

Now we consider situations as the one with $U_2$ in Figure 3, where the object is partially visible in some cameras. Concluding from Eq. 24 we make the assumption that all $S_i$ that intersect an object may be occluded in $\psi$ cameras at maximum. Then we can adapt Eq. 23 easily such that now the $\tau + \psi$ biggest distances of the sorted list are discarded and a safe distance results, as the influence of the occlusion is compensated.

While this assumption leads to a safe distance, in a real world environment it is impossible to ensure that all parts of an object may be visible in $n - \psi$ cameras (compare $U_2$ and $U_3$ in figure 3) all the time, which consequently would turn the system non-applicable in reality as this would only be fulfilled for unknown objects like $U_1$. So we relax this assumption by the definition of a parameter $\theta$, set to be equal to $\psi$ in value, which states that there may always exist $n - \theta$ cameras where the object is projected onto at least one pixel which is part of $P_{\text{unkn},c}$:

$$\forall m : (|\{c \mid \exists x \in U_m : (proj_c(x) \cap P_{\text{unkn},c} \neq \emptyset)\}| \geq n - \theta) \tag{25}$$

With this definition, it can happen that parts of the objects may be occluded in more than $\theta$ cameras. Having this more realistic assumption, we nevertheless need to ensure the semantics of $\psi$ for every $S_i$ to get a safe distance calculation using above equations. For this purpose we take advantage of the knowledge that at least parts of the objects are visible in some cameras (Eq. 25) and use image information to conclude the occluded parts to find a way that the semantics of $\psi$ are achieved by an image set that satisfies the semantics of $\theta$.

For a contiguous object $U_m$, the projection onto an image plane creates a connected pixel set $P_{\text{conn},c}(U_m)$.

$P_{\text{conn},c}(U_m)$ may intersect with $P_{\text{occ},c}$ as well as with $P_{\text{unkn},c}$. Three cases are possible: intersection of $P_{\text{conn},c}(U_\text{m})$ with only $P_{\text{occ},c}$ or only $P_{\text{unkn},c}$ or intersection with both. In the third case, there exists at least one pair of neighboring pixels with one pixel $p_{\text{occ}}$ in $P_{\text{occ},c}$ and the other pixel $p_{\text{unkn}}$ in $P_{\text{unkn},c}$ .These pixel pairs are called *contact pixel pairs* in the following.

Whenever a contact pixel pair is encountered, the connected subset of $P_{\text{occ},c}$ of which $p_{\text{occ}}$ is part of, is transferred into the $P_{\text{unkn},c}$ set. This is accomplished by flood-filling the connected pixel set, where $p_{\text{occ}}$ is located using some connectivity, e.g. 4-neighborhood. As a result of this, $P_{\text{conn},c}$ $(U_m)$ must completely be part of $P_{\text{unkn},c}$ afterwards for contiguous objects.

By this way, $P_{\text{conn},c}(U_m)$ completely belongs to either $P_{\text{occ},c}$ or $P_{\text{unkn},c}$ for each camera after processing all contact pixel pairs. Considering Equation 25, it follows that the number of cameras $c$ in which $P_{\text{conn},c}(U_m)$ belongs to $P_{\text{unkn},c}$ must be bigger or equal to $(n - \theta)$. This in effect means that then the projection of each and every volume $S_i$ in $U_m$ will be part of $P_{\text{occ},c}$ exclusively or $P_{\text{unkn},c}$ exclusively for a specific camera $c$. Thus all $S_i$ share the same set membership in each camera. And as the object as a whole projects completely into $P_{\text{unkn},c}$ for equal or more than $(n - \theta)$ cameras, this is true for each $S_i$. This way all $S_i$ fulfill the condition of $\psi$ which is what we intended.

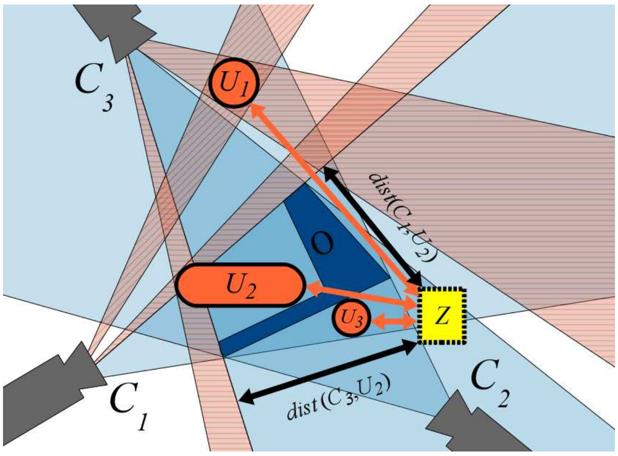

Figure 3: A schematic 2D illustration of camera-based distance measurements in the presence of occlusions for unknown objects $U_m$ to a protection zone $Z$. The occluding obstacle $O$ and related back projection pyramids are shown in blue.

After this treatment of partially occluded objects, the distance can be safely determined by adjusting Eq. 23.

$$d_{\text{safe}}(\tau, \psi) = s_{(n-(\tau+\psi))}, \text{ with } \psi = \theta \tag{26}$$

In the beginning of this section we discussed that the conservative approach to fulfill the principle assumption (Eq. 24) by using all pixels of unknown objects as well as of obstacles for distance calculation would lead to imprecise distance approximation which affects the availability. Nevertheless we can safely include this approach into the overall distance computation as it might lead to better results in some specific scene configurations of obstacles, cameras, etc. Therefore we have to apply Eq. 18 to all elements from object and occlusion pixels.

$$d_{\text{cam,cons},c} = \min_{p \in (P_{\text{unkn},c} \cup P_{\text{occ},c})} (d_{\text{back},p,c}) \tag{27}$$

The conservative distance computation for the whole system ends up in a distance $d_{\text{safe,cons}}(\tau)$ corresponding to Equation 23.

Finally we take the best result of both approaches to get the safe distance $d_{\text{safe,fused}}(\tau,\psi)$.

$$d_{\text{safe,fused}}(\tau,\psi)=max\left(d_{\text{safe,cons}}(\tau),d_{\text{safe}}(\tau,\psi)\right) \tag{28}$$

## 6. Experimental results

We implemented the fault tolerant distance calculation method for our multi-view system which uses a standard workstation and performs in real-time at 15 fps. All images have sizes of 80x60 pixels. In our first experiment, we use a simulated environment which is based on our real world system in order to avoid errors which arise from real change detection methods and in order to show our method in conjunction with a given ground truth, which would be more difficult to generate for the real setup. The ground truth was created through projection of simulated objects into the cameras resulting in ideal difference images.

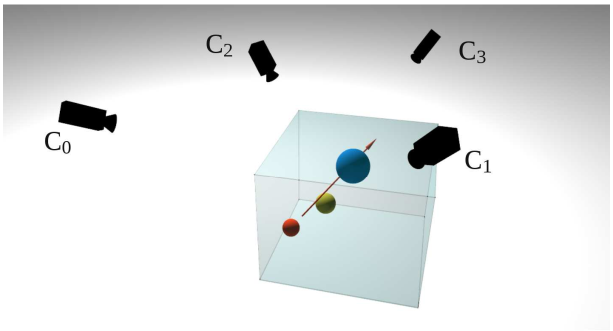

Figure 4: Visualization of the experimental setup of the simulation environment which contains an object (red sphere) that moves along the red arrow, an occluding obstacle (blue sphere) and a static protection zone (yellow sphere) to which the safe fused distance is calculated.

The simulated situation is shown in Figure 4. Four cameras ($C_0$ to $C_3$) are monitoring the common supervision space $A$, indicated by the transparent blue cube. A small red object of 300 mm in diameter is translated for a duration of approximately a hundred frames (10 sec) through space along a trajectory represented by the red arrow from the position (1000, 0, 0) mm to (0, 1000, 1000) mm. A static blue obstacle of 460 mm in diameter exists, that occludes the object completely at some positions along the trajectory in camera $C_1$ and also partly at various positions in the other cameras. A yellow sphere of 300 mm in diameter represents the protection zone to which distances are calculated for each set of images.

The experiment was performed several times with an increasing number of camera failures$\tau$. The results are shown in Figures 5 to 6. In the cases of $\tau = 1$ and $\tau = 2$, simulated failures were introduced around frame numbers 35 and 40 in marked cameras. The parameter $\theta$ was constantly set to 1. With $\tau$ ranging from 0 to 2, the main condition $n > \theta + \tau$ is ensured. The diagrams show the distances $d_{\text{real}}$ (black line) and the resulting safe distance $d_{\text{safe, fused}}$ $(\tau,\psi)$ (dotted black line) as well as all distances $d_{\text{cam},c}$ for the given cameras (different colored lines). To clarify the presentation, the conservative distances $d_{\text{cam,cons},c}$ for each camera are not shown. Since the occlusions are caused by a static obstacle (blue sphere), the corresponding distances are static, too, unless they get flood-filled when the object is intruding the occlusion. In that case, the distance to occluded areas increases to a local maximum because no occlusion pixels remain in an image with contact pixels pairs as all those pixels are converted into object pixels. The maximum of distances was limited to 1000 mm for all distances.

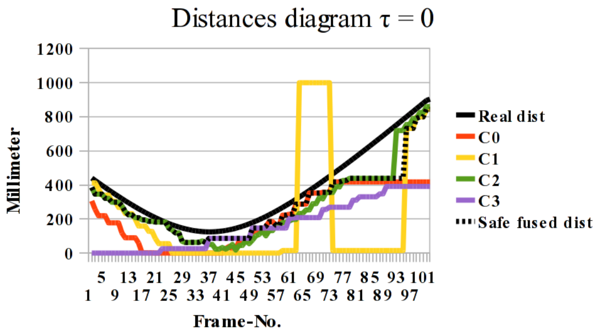

Figure 5: Distance diagram with failure parameter $\tau = 0$ showing the ground truth distance “Real dist” (black), the object distances $d_{\text{cam},c}$ for all cameras $C_0$ to $C_3$ (colored) as well as the safe fused distance $d_{\text{safe, fused}}$ $(\tau,\psi)$ (dotted black). The safe fused distance is always lower than the ground truth distance.

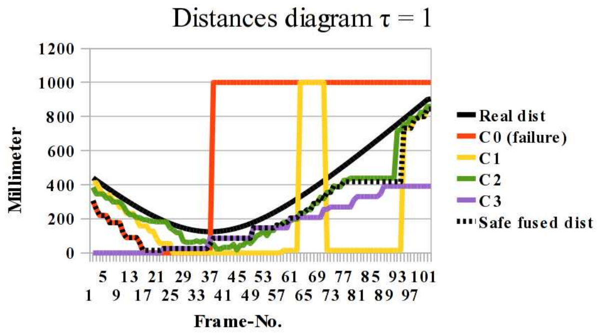

Figure 6: Distance diagram with failure parameter $\tau = 1$ showing the same distances as Figure 5. The safe fused distance is always lower than the ground truth distance though a failure is introduced to camera 0 from frame 35 to the end of the experiment.

In Figure 5 all cameras are performing distance computations without failure ($\tau = 0$). The object is occluded in camera $C_1$ completely in frames 65 to 75. The safe fused distance is always close to the real distance, and does never cross it, even in case of a single camera occlusion. The jaggedness of the lines originates from the limited resolution of the cameras, which effectively quantize the distance measurements coarsely. In Figure 6, a single camera failure is introduced to $C_0$ at frame 35. The effect of the failure parameter $\tau = 1$ in comparison to Figure 5 is a reduced safe fused distance for the first 35 frames. This reduction of the fused distance is even more evident in the scenario with the expected failure of two cameras, as presented in Figure 7.

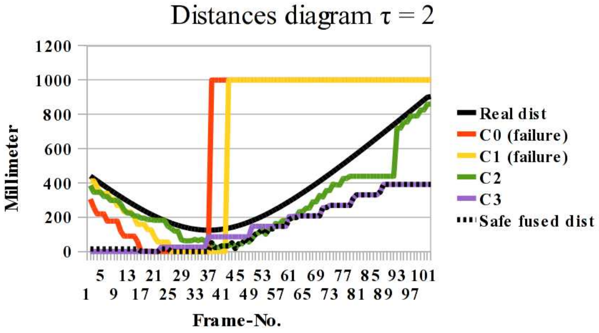

Figure 7: Distance diagram with failure parameter $\tau = 2$ showing the same distances as Figures 5 and 6. The safe fused distance is always lower than the ground truth distance though failures are introduced to camera 0 and 1 from frame 35 and 40 respectively, to the end of the experiment.

This example shows the downside of rejecting too many camera distances for safety issues which leads in the first frames to a fused distance close to zero. Nevertheless the resulting distance still remains safely below the real distance faced with the failure of the cameras. To improve distance computation this case, the total number of cameras could be increased.

The second experiment shows the distance calculation approach applied to our real world working cell which also monitors a common supervision space $A$ with four cameras ($C_0$ to $C_3$). The parameter $\theta$ was set to 1 after manual estimation of occlusion sizes. The experimental setup is visualized in Figure 8 which shows the view of camera $C_3$.

The setup consists of two static obstacles, which both produce occlusion pixels, on the right hand a machinery box (blue) and in the middle a robot which represents a static protection zone (yellow) to which the distances are calculated. The robot was not moved during the experiment while a person was walking through the cell along the trajectory as represented in Figure 8 (green line). The experiment was executed for duration of 255 frames at 15 fps (17 sec). As shown, the person surrounds the robot and performs a loop around the machinery box. A simple background subtraction algorithm is applied to all cameras for object pixel creation. We do not have a given ground truth for this experiment, but can show nevertheless the distance behavior for a real world scenario.

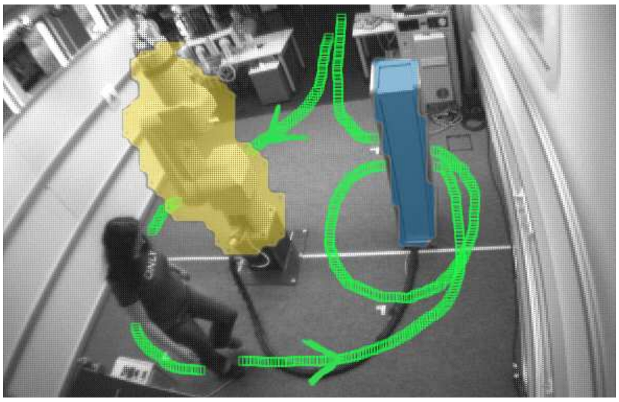

Figure 8: Visualization of the experimental setup of the real world working cell with two occluding obstacles, a static protection zone (yellow) which is represented by the non-moving robot model and a static machinery box (blue). A person is walking along the trajectory (green) which causes object pixel sets in the cameras due to a background subtraction algorithm to which the distances to the protection zone are computed.

Figure 9 represents analog to the first experiment the resulting distances $d_{\mathrm{cam},c}$ to the object pixels of the four cameras (colored lines) as well as the resulting safe fused distance $d_{\mathrm{safe,fused}}(\tau,\psi)$ (dotted black line). To understand the distances around 500 mm for the cameras $C_2$ and $C_3$ (yellow and light blue lines) it has to be mentioned that those cameras are watching more than just the common supervision space $A$ and therefore detect the person before entering and after leaving the scene. In comparison, the cameras $C_0$ and $C_1$ have views limited trough walls and cannot detect the person before entering $A$, this event can be detected in Figure 8 when the distances decrease beginning from the maximum distance, which was set to 2000 mm as minimum border of the working cell.

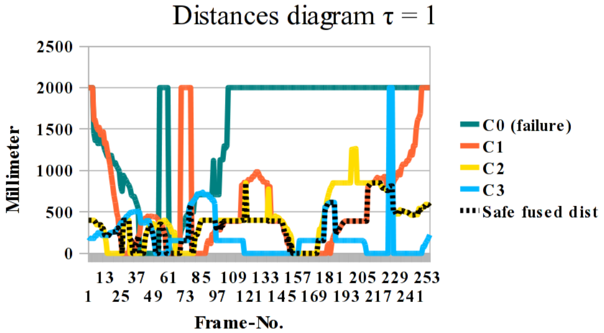

Figure 9: Distance diagram for the real world experiment with failure parameter $\tau = 1$ showing the object distances $d_{\mathrm{cam},\,c}$ for all cameras $C_0$ to $C_3$ (colored) as well as the safe fused distance $d_{\mathrm{safe,\,fused}}(\tau,\psi)$ (dotted black). A camera failure of $C_0$ is introduced from frame 95 to the end of the experiment.

Also, when the person leaves the scenario, the distances rise up to the maximum again (just for camera $C_1$). Similar to the first experiment, a single camera failure of $C_0$ was introduced at frame 95. The trajectory of the person enters several times the protection zone from the cameras points of view, which results in distances equal to zero in Figure 8. The occlusion handling with the parameter $\theta$ worked well, because the person is only invisible in $\theta$=1 cameras at one time while surrounding the occluding machinery box and the occluding robot. If the person was occluded simultaneously in more than $\theta$ cameras the fused distance would not be safe any longer.

## 7. Conclusions

We have presented a vision-based surveillance approach that exploits multiple camera perspectives for distance computations to achieve increased distance accuracy in 3D and fault tolerance at the same time. Additionally the system is able to cope with occlusions of the detected objects by known modeled obstacles which provide applicability to more (industrial) scenarios. The correct handling of occlusions and failures is proven under certain assumptions. These assumptions are represented by the occlusion parameter $\theta$, which has to be user-given or automatically determined. The overall system is performing in real-time on a standard workstation. Experimental results show effects of camera failures on the safe fused distance and call for further examinations concerning loss of availability and applicability in different industrial scenarios.

## 8. Ackknowledgments

The authors would like to thank Elan Schaltelemente GmbH & Co. KG for the financial support in the SafetyVision project as well as Thorsten Gecks for his comments to this work.

## 9. References

[1] J. R. Casas, and J. Salvador. Image-based Multi-view Scene Analysis using ‘C*onexels*’. The 1st HCSNet Workshop on the Use of Vision in HCI (VisHCI’06), pp 19-28, Canberra/Australia, November 1-3, 2006.

[2] DIN EN 953. Safety of machinery – Guards – General requirements for the design and construction of fixed movable guards. 2009

[3] DIN EN 1760-1. Safety of machinery – Pressure sensitive protective devices – Part 1: General principles for the design and testing of pressure sensitive mats and pressure sensitive floors. 2009

[4] D. Ebert, D. Henrich. Safe human/robot cooperation: image-based collision detection for industrial robots. IEEE International Conference on Intelligent Robots and Systems (IROS’02), Lausanne, September 30 – October 4, 2002.

[5] S. Y. Elhabian, K. M. El-Sayed, and S. H. Ahmed. Moving Object Detection in Spatial Domain using Background Removal Techniques – State-of-Art. Recent Patents on Computer Science, pp. 32-34, Vol. 1, No. 1, 2008.

[6] M. Fischer, D. Henrich. 3D Collision Detection for Industrial Robots and Unknown Obstacles using Multiple Depth Images. German Workshop on Robotics (GWR’09), Braunschweig/Germany, June 9-10, 2009.

[7] F. Gardeux. Detection of persons based on digital visions. Hygiene et Securite du Travail, 208:61-74, 2007.

[8] T. Gecks, D. Henrich. Multi-Camera Collision Detection allowing for Object Occlusions. 37th International Symposium on Robotics (ISR’06)/4th German Conference on Robotics (Robotik’06), Munich/Germany, May 15 – 17, 2006.

[9] L. Guan, J.-S. Franco, and M. Pollefeys. 3D Occlusion Inference from Silhouette Cues. The 20th IEEE Conference on Computer Vision and Pattern Recognition (CVPR’07), Minneapolis/USA, June 17-22, 2007.

[10] L. Guan, S. Sinha, J.-S. Franco, and M. Pollefeys. Visual HullConstruction in the Presence of Partial Occlusion. 3rd International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT’06), Chapel Hill/USA, June 14 – 16, 2006.

[11] D. Henrich, M. Fischer, T. Gecks, S. Kuhn. Sichere Mensch/Roboter-Koexistenz und Kooperation. ROBOTIK 2008, Munich/Germany, June 11-12, 2008.

[12] D. Henrich, T. Gecks. Multi-camera Collision Detection between Known and Unknown Objects. 2nd ACM/IEEE International Conference on Distributed Smart Cameras – (ICDSC’08), Stanford/USA, September 7–11, 2008.

[13] IEC 60204-1. Safety of machinery – Electrical equipment of machines – Part 1: General requirements. Edition 5.0, 2005.

[14] IEC 61496-2. Safety of machinery – Electro-sensitive protective equipment – Part 2: Particular requirements for equipment using active optoelectronic protective devices (AOPDs). Edition 2.0, 2006.

[15] IEC/TR 61496-4. Safety of machinery – Electro-sensitive protective equipment – Part 4: Particular requirements for equipment using vision based protective devices (VBPD). Edition 1.0, 2007.

[16] S. Kuhn, T. Gecks, and D. Henrich. Velocity control for safe robot guidance based on fused vision and force/torque data. IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Heidelberg/Germany, September 3-6, 2006.

[17] S. Kuhn, D. Henrich. Multi-View Reconstruction of Unknown Objects within a Known Environment. 5th International Symposium on Visual Computing (ISVC’09), Las Vegas/USA, November 30 – December 2, 2009.

[18] A. Laurentini. The visual hull: A new tool for contour-based image understanding. In Proc. 7th Scandinavian Conference on Image Analysis, pp. 993-1002, 1991.

[19] Patent DE 10 2006 057 605 A1. Verfahren und Vorrichtung zum Ueberwachen eines dreidimensionalen Raumbereichs. PILZ GmbH & Co. KG, 2006, http://www.safetyeye.com.

[20] A. Schick, R. Stiefelhagen. Real-Time GPU-Based Voxel Carving with Systematic Occlusion Handling. The 31st Annual Pattern Recognition Symposium of the German Association for Pattern Recognition (DAGM’09), pp. 372-381, Jena/Germany, September 9-11, 2009.