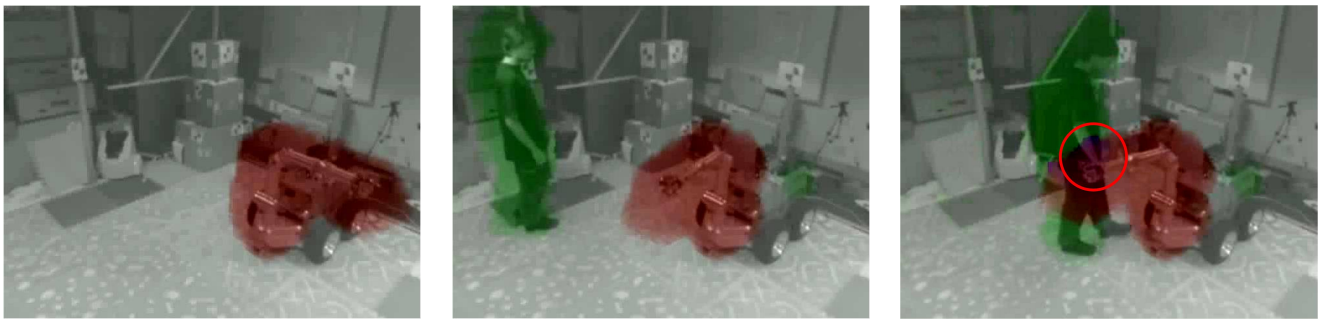
Sensor Fusion for Human Safety in Industrial Workcells

Paul Rybski[1], Peter Anderson-Sprecher[1], Daniel Huber[1], Chris Niessl[1], Reid Simmons[1]
{prybski,dhuber,reids}@cs.cmu.edu, {peanders,cniessl}@andrew.cmu.edu



(a) (b) (c)

Fig. 1: An example of our approach. (a) The workcell as seen by one of the 3D sensors. The red region indicates the adaptive danger zone surrounding the moving robot arm. (b) As the person enters the workcell, the green region indicates the adaptive safety zone surrounding the person. (c) When the person gets too close to the robot, the safety zone and danger zones intersect (shown in purple and highlighted with a red circle), and the robot automatically halts.

***Abstract*— Current manufacturing practices require complete physical separation between people and active industrial robots. These precautions ensure safety, but are inefficient in terms of time and resources, and place limits on the types of tasks that can be performed. In this paper, we present a real-time, sensor-based approach for ensuring the safety of people in close proximity to robots in an industrial workcell. Our approach fuses data from multiple 3D imaging sensors of different modalities into a volumetric evidence grid and segments the volume into regions corresponding to background, robots, and people. Surrounding each robot is a *danger zone* that dynamically updates according to the robot's position and trajectory. Similarly, surrounding each person is a dynamically updated *safety zone*. A collision between danger and safety zones indicates an impending actual collision, and the affected robot is stopped until the problem is resolved. We demonstrate and experimentally evaluate the concept in a prototype industrial workcell augmented with stereo and range cameras.**

I. Introduction

Current robotic manufacturing practices require that people be completely separated from active robots, which is typically achieved using fences or similar physical barriers. Since industrial robots can be large, fast-moving, and carry heavy or hazardous parts, a collision with a person could result in severe bodily injury or death.

While separation between robots and people ensures safety, the practice is inefficient for several reasons. Workcells may occupy large amounts of floor space due to the extensive workspace of the robot, even if only a small portion of that workspace is actually used. Any time material needs to be brought into or removed from the workcell, the robot must be halted while a person enters to deliver or retrieve the material. Perhaps most importantly, human/robot separation precludes activities involving robots and people working cooperatively. These limitations result in manufacturing processes that require more space, more robots, and more time than would be needed if robots and people could work together safely. Enabling robots and people to work safely in close proximity can increase the efficiency of robotic workcells. For example, if a worker needs to restock materials in a workcell, a robot could continue working in another part of the workcell without interruption. We envision a workcell where there is no need for safety fences.

This paper presents a sensor-based approach for ensuring the safety of people operating in close proximity of robots (Figure 1) [2]–[4]. Our approach uses the real-time fusion of multiple three dimensional (3D) imaging sensors to create a volumetric evidence grid representation of the space within the workcell [12], [13]. The occupied space is segmented into regions corresponding to background, robots, and people. Surrounding each robot is an adaptive *danger zone* that is based on the robot's position and trajectory. Similarly, an adaptive *safety zone* surrounds each person and follows them as they move about the workcell. This safety zone is "inflated" to be larger than the person to envelop them even when they are in motion. To ensure safety, regions that cannot be observed due to occlusion are also surrounded by safety zones. A danger and safety zone intersecting indicates that a robot and a person are too close and that a collision may be imminent. In such cases, the robot stops, or slows down, until the violation is cleared, for example by the person moving away.

Sensor-based safety in robotic workcells presents a number of challenges. Occlusions from equipment as well as

[1]The Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA, 15213

moving robots and people can prevent sensors from fully perceiving the space. The environments are complex and dynamic, and may contain previously unseen objects. Sensors must be carefully calibrated and synchronized to allow them to work together effectively. And, most importantly, the system must be extremely reliable, since a mistake can result in injury or death. The primary contributions of this paper are the development of an approach to sensor fusion that addresses these challenges and the implementation and evaluation of the complete safety system in a prototype industrial workcell.

II. Related Work

The safe interaction between humans and robots has been studied extensively since the early days of robotics [8], [10], [11], [15], [16]. Existing methods can be broadly classified into post-collision and pre-collision approaches [8]. Post-collision methods detect a collision as it occurs, and attempt to minimize the resulting damage. Methods in this category include detecting collisions through force and torque sensors on the robot [10], limiting forces through active or passive compliance [16], limiting the joint robot velocity so that the damage from a collision is acceptable [8], and cushioning the blow using padding on the robot. None of these methods actually prevent a collision, which reduces their usefulness in safety systems. Pre-collision approaches attempt to prevent collisions by detecting them in advance. These methods include proximity sensors either mounted on the robot [11] or in the environment [18].

Independently of robot safety applications, the problem of detecting and tracking people has been extensively studied in the computer vision community [5], [6], [14], [19], [20]. Vision-based methods using cameras work reasonably when people are well-separated, minimally occluded, and in neutral poses [5]. Pose estimation methods can address person detection when people are bending over or reaching out [6], [20]. These methods are not yet reliable enough for robot safety systems. Three-dimensional sensing can detect people in arbitrary poses, and the recently introduced Kinect system has proven to be fairly reliable for human pose estimation [19].

Most of the aforementioned methods have yet to be widely adopted by industry. Industrial robots usually achieve safety through separation, either through physical barriers (e.g., fences) or virtual barriers (e.g., laser-based light curtains). Advances in robotic controllers have enabled fine-grained programming of static safety regions, allowing closer human-robot interaction [1], [7]. The recently introduced SafetyEye system uses stereo vision to detect moving objects inside a safety region, reducing the need for fencing [18].

Our approach is most similar to the SafetyEye system, but is unique in several ways. We combine multiple sensors, which addresses the problem of occlusions; we utilize different modalities, which reduces sensitivity to limitations of a particular sensing modality; we explicitly model the background and the robot, which enables detection of people even in changing environments; and we construct dynamic danger zones based on the robots' position, trajectory, and capabilities, which provides a more precise boundary of the danger zone.

III. Sensor Fusion for Human Safety

In order for people to work safely in the proximity of industrial robots, their positions within the workcell must be constantly monitored, regardless of what they are wearing or doing. Full-field, 3D sensors, such as range cameras or stereo vision systems, are well-suited for detecting people in 3D space. Our approach employs multiple 3D sensors of different modalities placed strategically throughout the workcell. The sensors must be intrinsically calibrated so that the 3D data is geometrically accurate and extrinsically calibrated so that measurements from each sensor can be referenced in a single coordinate system.

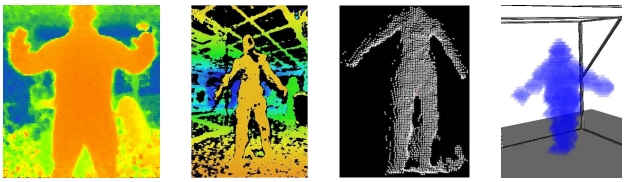
Our safety monitoring system begins by converting the data from each sensor into a probabilistic 3D evidence grid which represents occupied, unoccupied, and unknown regions [12]. The individual evidence grids from each sensor are fused together into a single evidence grid. Prior to workcell operations, with no people present and the robot in a known position, a fused evidence grid is generated and used to estimate a background model [4]. During operations, the robot's geometric model and kinematic parameters are used to build a posable, voxel-based robot model, which enables the system to remove the robot from the foreground.[1] Any remaining occupied foreground voxels are clustered into blobs, and blobs of sufficient size are considered to be potential people. Using the robot's known joint positions and velocities, which are assumed to be available, the robot is surrounded by a *danger zone*, which represents the maximum distance the robot can travel within one sensing cycle, plus the robot's stopping distance [3]. A somewhat larger *warning zone* provides additional security. Similarly, the volume representing each person in the workcell is expanded into a region called a *safety zone*, which is the maximum distance the person can move within a sensing cycle. If any part of a person's safety zone volume intersects with the robot's danger (warning) zone, the robot will be commanded to halt (slow) its motion. This enforced restriction on the robot's motion stays in effect until the safety violation is cleared when the person moves away. The next subsections describe each part of the system in more detail.

A. Sensors and Sensor Placement

Our system can be used with data from an arbitrary number and type of 3D imaging sensors. Our testbed uses stereo cameras and range cameras, each of which has complementary advantages and disadvantages.[2] Stereo sensors have the advantage that a variety of cameras, baselines, and lenses can be configured to achieve resolution and range accuracy as

[1] For simplicity, we describe the approach using a single robot. The extension to multiple robots, which we have also implemented, is straightforward.

[2] Other 3D imaging systems, such as structured light cameras or the Kinect sensor, could also be employed, with minimal additions to the system needed to convert their output into an evidence grid.



(a) (b) (c) (d)

Fig. 2: Range data from range cameras (a) and stereo vision systems (b) is first converted into point clouds (c) and then into probabilistic evidence grids (d), which are then fused together.

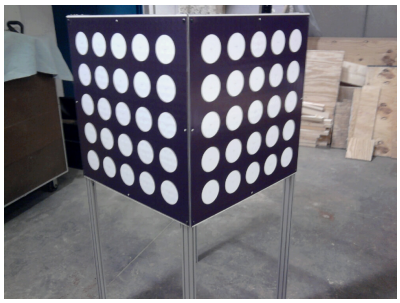


Fig. 3: A custom-built calibration cube is used to estimate the extrinsic parameters of the sensors.

needed for a given application. The sensing range is limited mainly by the acceptable level of range uncertainty, which increases quadratically with distance. Stereo requires good illumination and provides accurate range only in regions with sufficient texture. Repeated patterns can induce range estimation errors as well.

Range cameras (also known as flash lidars) are less sensitive to ambient light, since they are active sensors and use their own modulated infrared (IR) illumination. The cameras can measure range on featureless surfaces as well. On the downside, range sensors are often slower than stereo cameras and have difficulty imaging highly reflective or IR-absorbent surfaces because either no energy is returned or the detector is saturated. The sensors can also interfere with one another because stray returns from the IR signals of other sensors can cause spurious range measurements. Finally, like other active range sensors, range cameras suffer from the "mixed pixel" effect, which occurs when a single pixel images two surfaces located at different ranges (typically at object boundaries). These mixed pixels translate into phantom 3D points that appear where no surface actually exists. In our implementation, explicit mixed-pixel filtering and redundant sensors mitigate this problem.

To fully perceive a workcell, sensors must be placed at various locations around the space. The best sensor placement depends on the geometry of the environment, the number of sensors, their capabilities, and the task. Intelligent sensor placement is important, since regions that are unobserved due to occlusion or being outside every sensor's field of view could potentially contain a person. Sensors with overlapping fields of view help reduce the possibility of occlusion. We developed a simulation tool to quickly evaluate the quality of a proposed sensor placement. The tool allows placement and configuration of an arbitrary number of sensors and then casts rays – according to the sensor geometry – into a volumetric voxel grid, recording the number of sensors that observe each voxel. In practice, the intuitive notion of placing sensors high in the corners, pointing downwards, and oriented orthogonally or with opposing fields of view gives the best coverage and redundancy.

B. Sensor Calibration

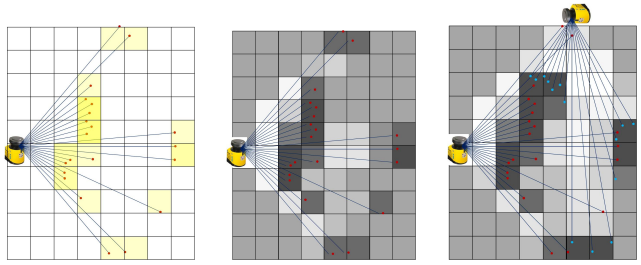
Each sensor must be intrinsically calibrated, and the collective sensors must be extrinsically calibrated with respect to the workcell's coordinate system. In our implementation, we use the factory calibration for the the sensors' intrinsic parameters. For extrinsic calibration, we estimate the relative pose between sensors using a custom-built calibration cube with a regular pattern on each face (Figure 3). During calibration, the cube is placed so that pairs of sensors can see at least one common face. Fiducials mounted on the floor are used to estimate the pose of the sensor network in the workcell's coordinate system.

In addition, the data acquisition of all sensors must be synchronized in order to fuse the data successfully. Without such synchronization, fast-moving objects would be misaligned between sensors, causing blurring when the data is fused. We have developed software and hardware approaches to improve overall sensor synchronization.

C. Evidence Grid Data Fusion

At each time step, data from each sensor is encoded into a 3D evidence grid (with a cell size of $10cm^3$) and combined with the evidence grids obtained at that time step from the other sensors (Figure 4). The fused evidence grid helps attenuate sensor noise by combining information and facilitates reasoning about the effects of occlusions that block a sensor's field of view, since any area unseen by the sensors could potentially contain a hidden person. The range measurements from a given sensor are first transformed into a point cloud using the sensor's intrinsic parameters. The details of this process are sensor-dependent, but are either well-known (for stereo vision) or straightforward (for range cameras). Next, the points in the point cloud are added to a 3D evidence grid [12]. The space within the workcell is discretized into a fixed-sized 3D grid of voxels ($10\mathrm{cm}^3$ in our implementation). Each voxel stores the log-likelihood of the probability that it is occupied. Cells are initialized to 0.5 probability, which represents the "unknown" state. Points are added to the evidence grid by tracing a ray from the sensor's center of projection through the workcell, applying a sensor-specific evidence model along the line. See [2] for details.

The individual evidence grids are fused together into a single, unified grid that extends over the entire volume of the workcell. Since each voxel stores the log-likelihood probability, the evidence in the grids can be combined simply by adding voxel values [12]. Evidence of occupancy from one sensor will strengthen evidence of occupancy from a second sensor (the same holds for unoccupancy). If one



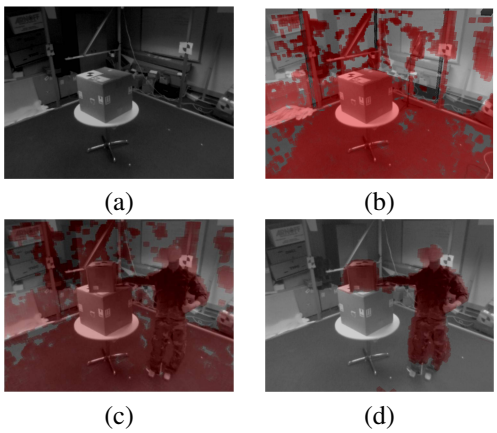
(a) (b) (c)

Fig. 4: A 2D illustration of the process of converting measurements into evidence grids. (a) The data points (red dots) are converted into voxels (yellow squares); (b) The resulting evidence grid, where the darker the square, the more likely that it is occupied (gray regions are unknown); (c) The fusion of multiple evidence grids reinforces the presence of obstacles (darkest regions).

sensor reports a value of occupancy but another reports a value of unoccupancy, the two values will tend to cancel out and shift the cell towards the unknown value since the data is otherwise inconsistent. For a given sensor, any cells that are occluded will remain set to the "unknown" state for that sensor and will have no effect on cells fused from other sensors. These evidence grids are re-created at each new timestep from new sensor data. Data from previous timesteps is not retained in the grids. For efficiency, we have added the option to fuse grids hierarchically and in parallel, first fusing groups of evidence grids, then fusing groups of groups, etc.

D. Background Subtraction

Our system uses a background model to represent aspects of the environment that are stationary and will not (or cannot) move (Figure 5(b)). The background model is created automatically upon initialization, the only requirement being that moving objects (e.g., people) must not be present. During operations, at each time step, the background model



(a) (b) (c) (d)

Fig. 5: The process of separating background from foreground. (a) The initially static background of the workcell containing objects that are expected to be immovable. (b) The red voxels show the background model learned by the system. (c) The full evidence grid after another box and a mannequin have been added. (d) The foreground evidence grid created by subtracting the background model from the full grid.

is subtracted from the fused evidence grid to determine the foreground voxels (Figure 5(d)). The foreground consists of all objects that entered the workspace since the background model was initialized. The space associated with the robot is treated separately, as described in the next subsection.

The most straightforward definition of background would be to use voxels in the evidence grid with sufficient evidence of being occupied. For safety purposes, however, we must also consider unknown regions. Some unobserved regions are perfectly safe. For example, the inside of a box cannot be observed by the sensors, yet it is unlikely that a person is hiding inside the box and will suddenly burst out and collide with a nearby robot. On the other hand, the region behind a screen might also be unobservable, but in this case, person could easily be standing within this occluded space. To address this problem, we developed the concept of *accessibility analysis* [4]. A region is considered accessible if a path exists between a known accessible region and an unknown region without passing through an occupied region. A closed box would be inaccessible, so the interior voxels would be considered part of the background. However, the area behind a screen would be considered accessible, and therefore part of the foreground, since one could access the occluded region by going around the side of the screen. Details of the approach can be found in [4].

E. Robot Modeling and Danger Zone Generation

The robot is not part of either the background or foreground – it is not static, but it also should not be treated as an unexpected object. In particular, voxels associated with the robot must be removed from the foreground; otherwise, "phantom" foreground objects would appear co-located with the robot, causing the robot to stop due to a false alarm.

Each robot is explicitly modeled using a voxel-based approximation. Since robots are articulated, each rigid part is modeled separately, which allows the model to be updated in real time based on the pose of the robot. The modeling process is performed once at initialization. The CAD models of each of the robot's parts are converted into triangular surface meshes. Next, we use a modified form of spatial occupancy enumeration via divide and conquer to convert the meshes into voxel grids [9]. In particular, we recursively subdivide each triangle into four using the midpoints of each edge until the triangle edge lengths are smaller than the target voxel grid cell size. In this way, a binary occupancy grid of the model can be computed directly from the vertices of the up-sampled triangle mesh. For improved model accuracy, we use a voxel grid with twice the resolution of the evidence grids. Assuming the input mesh is watertight, the resulting occupancy grid will be watertight (in a 6-connected sense), and we use a simple flood-fill algorithm to mark any interior voxels as occupied. The corners of the occupied voxels are then used to form point clouds for each robot part.

At runtime, the point clouds for each part are positioned according to the robot's pose, and a new, combined occupancy grid is generated representing the robot in its current position. This combined occupancy grid is overlaid with the

current fused evidence grid to subtract out voxels that are attributed to the robot. The same process is applied to tools or other payload objects that a robot may currently be holding.

The combined point-cloud model is also used for generating the danger zone surrounding the robot. At a given instant, the danger zone encompasses the region that the robot could occupy at any time in the next Δ_t seconds. The choice of Δ_t depends on the sensor framerate, latency of the sensing system, and time required to halt the robot (in our case, on the order of 400 ms, total). While a more accurate danger zone can be achieved using the robot's precise trajectory, such information is often difficult to obtain from commercial robots. Instead, we assume only that the current joint positions and velocities are provided, along with fixed accelerations.

We have developed an algorithm for estimating the danger zone for a multi-jointed manipulator in real time (Figure 6). The method, briefly summarized here (see [3] for details), can be applied to any robot with no cyclic kinematic chains. Given the current position and velocity and maximum acceleration of each joint, the range of possible joint positions that can be achieved in Δ_t seconds can be determined in closed form. Starting with the free end of each kinematic chain, the algorithm builds a sequence of volumetric "reachability" grids that store the minimum time for any part of the manipulator to reach each grid cell. A point-cloud model of the last link is swept through the range of possible values of the last joint, creating a reachability grid for that link. This grid is then converted into a point cloud (augmented with reachability times) and rigidly attached to the model of the previous link. The sweeping process is repeated down the kinematic chain to the robot's base, at which point the combined reachability grid represents the desired danger zone. The same process can be used to create a slightly larger *warning zone* as well.

Note that the dynamic danger zone is typically asymmetric, extending further in the direction the robot is currently moving. This means that it may be safe for a person to move immediately behind a robot even if it is not safe to be immediately in front.

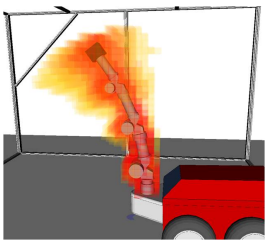


Fig. 6: Robot motion prediction. Joint positions and velocities are used to dynamically and efficiently predict the region the robot could occupy during a given time interval. The color of the voxel cloud indicates the minimum time for the arm to reach a given position, with red being the shortest and yellow, the longest.

F. Person Detection and Safety Zone Generation

Ideally, the safety zone surrounding a person would be computed in a manner analogous to the robot danger zone. However, people are not as predictable as robots, and estimating the detailed body pose of a person is a challenging problem (although [19] shows promising results). Instead, we create a safety zone simply by expanding the volume occupied by each potential person in the scene.

The system considers any connected group of foreground voxels of sufficient size to be a person. While it would be possible to explicitly recognize people, it is safer to conservatively assume that any large foreground object is a person, rather than to risk a missed detection and potential injury. Our approach can detect potential people regardless of their body pose or the direction they are facing.

The first step is to find all the connected components of the foreground voxels. Components with fewer than a threshold N_p voxels (10, in our case) are too small to be people, more likely to be noise, and are therefore discarded. The remaining components get surrounded by safety zones. Each safety zone represents space where a person could possibly move in Δ_t seconds, in any direction, where the maximum velocity of the movement is determined from existing safety standards, such as the R15.06 robotic industrial specification [17]. The safety zones are enlarged using morphological dilation to compensate for a person's potential movements in the space.

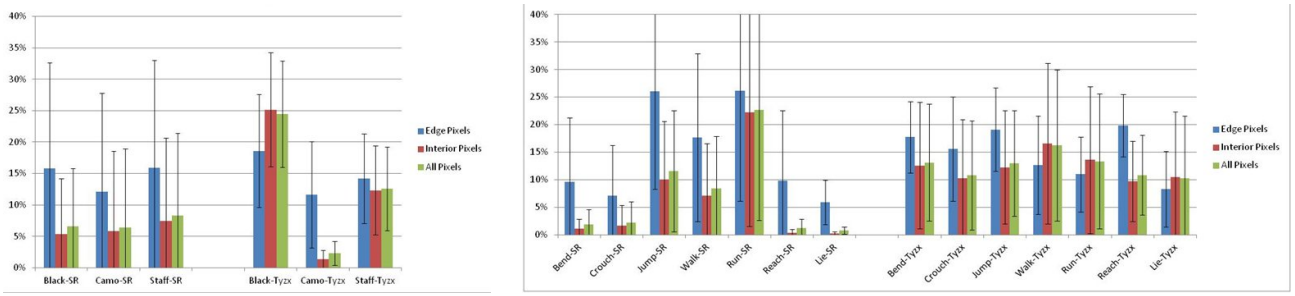
G. Collision Detection

The danger and warning zones of the robot and the safety zones of the people are updated at each time step as the robot operates and people move about the workcell. The zones are checked for collisions. If a safety and warning zone intersect, the robot is slowed down and the people in the workcell are alerted with a warning. If a safety and danger zone intersect, the robot is halted and a safety violation alert is sounded. When the person causing the warning or violation moves out of the way, the alarm ceases, and the robot resumes its normal operation. For handling multiple robots, we keep track of which warning/danger zone belongs to which robot, so that the system can slow/halt the correct robot and leave the others to continue working.

IV. Experimental Validation

We implemented and evaluated the proposed approach using a simplified industrial workcell testbed. The testbed consists of an aluminum cross-bar sensor frame – 4 m on a side and 2 m tall (Figure 5) equipped with two Swissranger SR4000s range cameras[3] and two Tyzx G3 EVS stereo cameras[4]. The sensors were mounted in the four upper corners of the frame to provide the widest possible overlapping coverage of the area. The testbed was equipped with two 2.67 GHz Intel Core i7 920 quad-core workstations running 64-bit Ubuntu Linux 10.04 LTS. All computers and sensors were connected through a wired gigabit Ethernet network.

[3]http://www.mesa-imaging.ch/
[4]http://www.tyzx.com/



(a) (b)

Fig. 7: (a) Percentage of missed pixels for each clothing type and sensor type, (b) Percentage of missed pixels for each action and sensor type. Error bars are one standard deviation from the mean.

Using this testbed workcell, we conducted experiments to characterize both the individual sensors and our overall safety monitoring system.

A. Individual Sensor Experiments

The goal of the first set of experiments was to determine how well each sensor type can detect people under different conditions. If a sensor cannot accurately detect a person under certain conditions, the system could potentially fail to detect the person, leading to an unsafe situation. We examined people performing a variety of actions, oriented in different directions, and wearing different types of clothing. For each type of sensor (stereo and range camera), we tested seven actions (crouching, jumping jacks, bending to touch toes, walking across workcell, running across workcell, reaching for an object, and lying down), two orientations (facing and perpendicular to the sensor), and three types of clothing (camouflage, Navy staff uniform (khaki top and black pants), and all black). We used the same person for each trial, and he performed the actions the same way each time, to the extent possible.

The images collected were then manually labeled to highlight the person. An analysis program then tallied the number of valid range measurements associated with each labeled pixel, both for the interior and perimeter of each person. Although it may be that some data was labeled as valid, but had the wrong range value, our experience with the sensors indicates that, except for mixed pixel effects along the perimeter, this is not much of a problem, in practice.

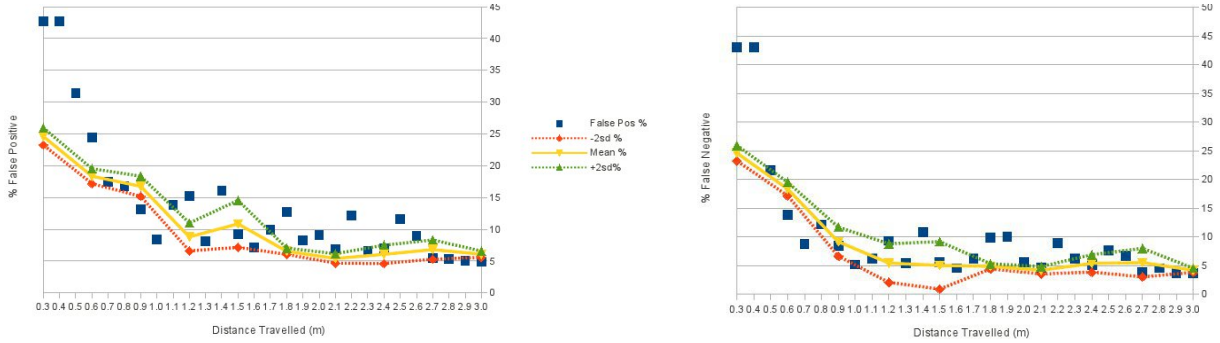
Figure 7 shows the results for just the perimeter (edge) pixels, just the interior pixels, and all body pixels. Figure 7a presents the data as a function of clothing type (with all actions aggregated), while Figure 7b presents the data as a function of action (with all clothing aggregated). The results indicate that the range camera (Swissranger) is fairly insensitive to clothing type, although it has fairly large variance. On the other hand, the stereo camera (Tyzx) is extremely sensitive to clothing differences. For camouflage clothing (Figure 8a), stereo actually performs significantly better than the range camera (Figure 8c), while for black clothing (Figure 8b) it performs much worse. The results also show that the performance along the perimeter is similar for both sensors, regardless of clothing type.

Analyzing the data by action (Figure 7b), we see that, for the range camera, slower motions (bend, crouch, reach, lie) are significantly better than faster motions (jump, run, walk), with running being by far the worst. This is not surprising, as the range camera has a relatively long exposure time, which exacerbates motion blur, which in turn adversely affects the performance of the range camera (Figure 8c). For the stereo, on the other hand, the exposure time is quite short, and so motion blur does not play as big a role (Figure 8a). Thus, for stereo, all the actions have similar rates of missed pixels (the faster motions are a bit worse, but not significantly so).

To more fully understand the distribution of missed pixels, we investigated the distribution of holes (connected clusters of missed pixels) in order to distinguish missing pixels that are primarily near the perimeter (e.g., missing heads or hands) from holes in the interior (e.g., caused by insufficient texture). Specifically, we calculated both the sizes of the holes and their distances from the perimeter of the body. The results (space precludes including the graphs) show that, for both sensor types, most of the holes are fairly small (1-3 pixels), although the tails of the distribution are quite large, with about 0.5% of the holes being larger than a third of the body size (see, for instance, Figure 8c). The range camera has over 90% of the holes touching the perimeter, and none more than 5 pixels from the perimeter, while the stereo has only 70% of the holes touching the perimeter, and 8% of the holes are more than 5 pixels from the edge, including 2.5% that are more than 15 pixels away. Such placement of holes is actually somewhat safer than those on or near the perimeter, since safety violations occur when the most distal part of

(a) (b) (c)

Fig. 8: Missed pixels (red) overlaid on image data for the jump action. (a) Camouflage and stereo (Tyzx), (b) Black and stereo (Tyzx), and (c) range camera (Swissranger).



(a) (b)

Fig. 9: Results of the fusion experiments. Percentage of false positives (a) and false negatives (b) as a function of distance along the track. Solid lines are the rates for the static experiments. Blue squares are the rates for the moving experiments.

the person intersects with a robot's danger zone. Interior holes will likely be surrounded by other parts that will make contact first, making interior holes less of a concern.

B. Fusion Experiments

The goal of the second set of experiments was to evaluate the quality of the fused evidence grid generated by multiple sensors. We tested the performance of the fusion algorithm in both static and moving cases. The experimental apparatus consisted of a box placed on a cart that could be pulled by an iRobot ATRV Jr.[5] mobile robot. A dedicated tracking system was developed that used a SICK LMS 219[6] planar LIDAR to estimate the position of the box. We chose to use a box target for these experiments because ground truth could easily be generated by rendering a simulated box in a virtual environment that could be voxelized and compared against the voxels returned by the sensors (Figure 10).

In the first experiment, ten sets of data were recorded where the box and cart were placed in a static position for 10 seconds. The cart and box were then moved 30 cm down the path. In the second experiment, the robot was commanded to pull the cart at 30 cm/s.

The results of both the static and the moving experiments are shown in Figure 9. Figure 9a presents the false positive results (perceiving voxels where there is, in fact, no box) and Figure 9b presents the false negative results (missing voxels that should be observed). We note that, due to discrete voxelization and errors in our position estimates, the "ground truth" may be off by a voxel. Thus, to calculate the numbers of false positives and false negatives, we used a distance-weighted exponential function to assign accuracy scores for both false positives and false negatives as follows:

$$Dist(v, G) = min(d(v, v')\forall v' \in G)$$
$$Score(v, G) = e^{ln(0.95)*Dist(v,G)^4}$$

Where the $Score$ function assigns a voxel a real value between 0 and 1, v is a voxel from a binary grid that is filled, and G is the set of all voxels. The scoring function chosen treats being one voxel away as 95% as good as being the same voxel, and drops off very rapidly after that.

To find the total detection rate, we iterate through the grid and calculate a score for each voxel detected and divide that by the maximum score possible, as follows:

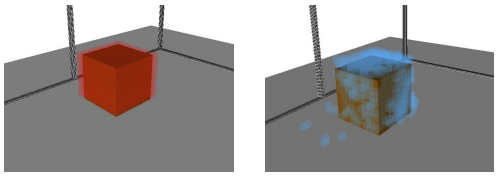
$$Acc(G_1 \rightarrow G_2) = \frac{\sum_{v \in G_1} Score(v, G_2)}{|G_1|}$$
$$FalsePositiveRate = 1 - Acc(G_{sensor} \rightarrow G_{model})$$
$$FalseNegativeRate = 1 - Acc(G_{model} \rightarrow G_{sensor})$$

False positives are obtained by comparing each voxel in the sensor grid against the nearest matching voxel in the model grid. False negatives are obtained by comparing each voxel in the model grid against the sensor grid. Each value is subtracted from one to obtain the failure rate.

As can be seen in Figure 9, the false positive and negative rates are highest when the cart is at the beginning of the track, since in that position, one side of the box cannot be seen. As the box position changes along the track, the error rates drop significantly until the entire box is in view and then hovers between 5% and 10%. The effect of a moving box can be seen as a slightly larger error, ranging from 5% to 15% for the false positive calculations but staying in the 5% to 10% range for the false negative calculations.

It is important to note that the evidence grid that we used in this experiment is only the base representation used by the overall system. Thus, even if there are holes in the middle or along the edges, they will likely be "filled in" by the

[5]http://www.irobot.com
[6]http://www.sick.com



(a) (b)

Fig. 10: (a) Ideal model box voxels in red generated in virtual environment. (b) Live sensor box voxels in blue projected into the virtual environment.

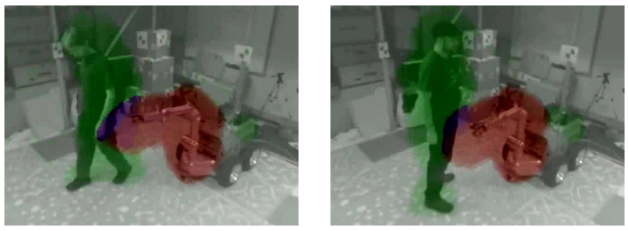


Fig. 11: Qualitative experiments demonstrating that the robot successfully halted when the human's safety region (green) intersected with the robot's danger region (red).

morphological operations that create the safety zones. Thus, in our evaluations with real people, we have found that given a reasonably-sized safety zone that takes into account RIA standards for human motion [17], the humans can be fully enclosed by the grid and thus protected at all times.

Finally, as illustrated in Figure 1(c) and Figure 11, we have performed numerous qualitative experiments where we tested the implementation by having different people walk about normally in the workcell in the presence of a robot. With our system running at approximately 10 Hz, we observed that the safety region successfully enveloped the people at all times while they were in the environment.

V. Summary and Future Work

This paper presented a real-time, sensor-based system that is intended to ensure the safety of people operating in close proximity to robots in industrial workcells. Our approach fuses data from multiple 3D sensors, of different modalities, into a volumetric evidence grid that is used to identify the locations of people and robots. Safety and danger zones surround the people and robots, respectively, and are each expanded to a size that will fully enclose them based on their maximum speeds and the cycle time of the system. Intersections between these two zones signal a possible impending collision, and the robots are commanded to slow and/or halt their motions, as needed. We have fully implemented this approach and have demonstrated its feasibility with a set of controlled experiments.

Currently, our implementation assumes that all foreground objects could potentially be people and extends safety regions around all of them, as this is the most conservative and safe option. For the future, we will seek to actively discriminate between human and non-human objects as well as to identify activities that the human is performing to improve system efficiency.

We believe that this type of technology will have a profound impact on how robots are integrated into factory settings. We look forward to a future where robots and people will work together, effectively and safely.

Acknowledgments

We would like to thank and acknowledge the following faculty, staff, and students for their contributions to this project: Sanjiv Singh, Kartik Babu, Seth Koterba, Thomas Koletschka, George Brindeiro, Greg Armstrong, and Brennan Sellner. This work was supported by the Office of Naval Research under contract #N00014-09-D-0584.

References

[1] ABB. Safemove white paper. Doc no. ROP 6904, http://www.abb.com, 2008.

[2] P. Anderson-Sprecher. Intelligent monitoring of assembly operations. Technical report, Carnegie Mellon University, The Robotics Institute, June 2011.

[3] P. Anderson-Sprecher and R. Simmons. Voxel-based motion bounding and workspace estimation for robotic manipulators. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, St. Paul, MN, USA, May 2012.

[4] P. Anderson-Sprecher, R. Simmons, and D. Huber. Background subtraction and accessibility analysis in evidence grids. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 2011.

[5] Ben Benfold and Ian Reid. Stable multi-target tracking in real-time surveillance video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3457–3464, Colorado Springs, CO, 2011.

[6] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *International Conference on Computer Vision (ICCV)*, 2009.

[7] S. Kock et al. Taming the robot: better safety without higher fences. Doc no. 9AKK105152A2830, http://www.abb.com., April 2006.

[8] J. Heinzmann and a. Zelinsky. Quantitative Safety Guarantees for Physical Human-Robot Interaction. *The International Journal of Robotics Research*, 22(7-8):479–504, July 2003.

[9] Y.T. Lee and A.A.G. Requicha. Algorithms for computing the volume and other integral properties of solids. i. known methods and open issues. *Communications of the ACM*, 25(9):635–641, 1982.

[10] Shujun Lu, Jae H Chung, and Steven A Velinsky. Human-Robot Collision Detection and Identification Based on Wrist and Base Force / Torque Sensors. In *International Conference on Robotics and Automation*, volume i, pages 3796–3801, April 2005.

[11] V.J. Lumelsky and E. Cheung. Real-time collision avoidance in teleoperated whole-sensitive robot arm manipulators. *IEEE Transactions on Systems, Man and Cybernetics*, 23(1):194–203, 1993.

[12] H. Moravec. Robot spatial perception by stereoscopic vision and 3d evidence grids. Technical report, Carnegie Mellon University, September 1996.

[13] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 116–121, 1985.

[14] Neeti A Ogale. A survey of techniques for human detection from video. Master's thesis, University of Maryland, 2006.

[15] Aslam Pervez and Jeha Ryu. Safe physical human robot interaction-past, present and future. *Journal of Mechanical Science and Technology*, 22(3):469–483, 2008.

[16] M H Raibert and J J Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems Measurement and Control*, 102(2):126–133, 1980.

[17] RIA/ANSI. *American National Standard for Industrial Robots and Robot Systems - Safety Requirements (RIA/ANSI R15.06)*. American National Standards Institute, New York, 1999.

[18] Pilz SafetyEye. http://www.pilz.com.

[19] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard, Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, CO, 2011.

[20] Y. Yang and D. Ramanan. Articulated pose estimation using flexible mixtures of parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, CO, 2011.