2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Multiple Depth/Presence Sensors: Integration and Optimal Placement for Human/Robot Coexistence

Fabrizio Flacco and Alessandro De Luca

Abstract— **Depth and presence sensors are used to prevent collisions in environments where human/robot coexistence is relevant. To address the problem of occluded areas, we extend in this paper a recently introduced efficient approach for preventing collisions using a single depth sensor to multiple depth and/or presence sensors. Their integration is systematically handled by resorting to the concept of image planes, where computations can be suitable carried out on 2D data without reconstructing obstacles in 3D. To maximize the on-line collision detection performance by multiple sensor integration, an off-line optimal sensor placement problem is formulated in a probabilistic framework, using a cell decomposition and characterizing the probability of cells being in the shadow of obstacles or unobserved. This approach allows to fit the optimal numerical solution to the most probable operating conditions of a human and a robot sharing the same working area. Three examples of optimal sensor placement are presented.**

I. Introduction

When humans and robots share the same working area, safety is the primary issue of concern [1]. While potential injuries of unexpected human-robot impacts can be limited by lightweight/compliant mechanical design of the manipulator [2] and collision reaction strategies [3], preventing collisions in a dynamic and largely unpredictable environment relies on human-phriendly motion planning [4] and, primarily, on the extensive use of exteroceptive sensors [5].

Different types of sensors can be used to detect objects in the environment, including proximity sensors, laser scanners, single and stereo cameras, or a combination thereof (e.g., PMD cameras). One can categorize all these sensors, and their way of use, either as *depth* or *presence* sensors, the latter providing only a binary information in the sensed direction. Data processing and computational issues depend on the type of applications, such as surveillance, object reconstruction, or collision prevention.

In [6], based on a single depth sensor, a novel approach has been proposed for characterizing on line the configurations of a manipulator that are most dangerous for collision with a human operator moving in its workspace. Collision is prevented by commanding the robot motion via repulsive forces from such configurations, thus achieving safe human-robot coexistence. The originality of the method is that it processes sensed data on a (virtual) 2D image plane without the need of a 3D reconstruction or approximation of the human, as opposed to other approaches for detecting the presence or estimating the depth of objects in human-robot interaction, e.g., [7], [8].

When using a single depth or presence sensor to monitor the environment, the main problem is the lack of information on the occluded areas behind the sensed obstacles. As a consequence, a conservative estimation for collision detection needs to include the occluded area as part of the obstacle, resulting in a (pseudo) obstacle that is bigger than its real dimension. Using multiple sensors that observe the scene from different point of views is the natural solution to reduce occlusions, though requiring a suitable sensor integration strategy to limit the computational burden. Multiple presence sensors have been used to detect collision between known and unknown obstacles, fusing only data from vision cameras [9] or also from force/torque sensing [5]. In [10], multiple depth images are elaborated to calculate a conservative 3D approximation of all detected obstacles.

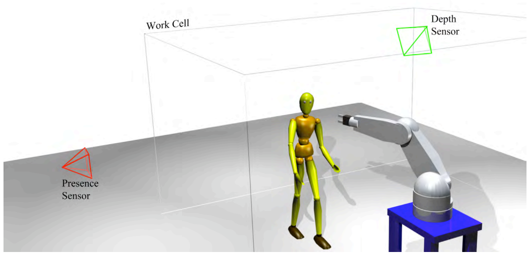

Fig. 1. A work cell with a robot manipulator and a human monitored by one presence and one depth sensor

The problem of optimal placement of sensors becomes even more relevant in the presence of sensing devices of different type, with the need of trading off information gains and additional costs. Finding an optimal sensor placement can be regarded as an extension of the *art gallery problem* originally posed by Victor Klee in 1973, which consists in finding the minimum number of cameras covering a given space, and is a prototypical problem for automated surveillance applications, see, e.g., [11]. The minimum number and optimal 2D placement of vision systems for total coverage is introduced in [12], using directional and omnidirectional cameras and in the presence of static obstacles. The 3D version of the same problem is tackled in [13] by a voting scheme and a greedy heuristics. When the number of sensors is given a priori, the problem reduces to finding what is their best placement. Examples include the optimal camera placement for monitoring a specified mobile robot trajectory [14], the optimal placement for minimizing the error on triangulation of fixed 3D features [15], the minimization of features error while considering performance degradation

The authors are with the Dipartimento di Informatica e Sistemistica, Università di Roma “La Sapienza”, Via Ariosto 25, 00185 Rome, Italy (e-mail:{fflacco,deluca}@dis.uniroma1.it).

978-1-4244-5040-4/10/$26.00 ©2010 IEEE

due to occlusions [16], and the use of networked cameras for totally covering specified regions in the work cell [17].

In this paper, we extend the work in [6] so as to include both depth and presence sensors and their integration for human-robot collision prevention. Moreover, we formulate the optimal sensor placement problem in a probabilistic framework. The working area is decomposed in cells, and the probability that a cell belongs to the shadow of obstacles or is unobserved by the sensors is derived. These probabilities are computed starting from the presence or depth maps of obstacles on their image plane, including the robot manipulator maps, and then integrating over all sensors. A cost function depending on the pose parameters of the sensors is defined and minimized numerically. The formulation is especially useful for providing sensor placement solutions that take into account different operative conditions, weighting cells in the working area according to the probability that the human or the manipulator will occupy them.

The paper is organized as follows. Section II provides a general description of the problem, introducing the notation used in the paper. In Sect. III the collision prevention approach proposed in [6] is recalled, and then extended to the consideration of multiple sensors of the depth or presence type. Section IV formulates in detail the optimal sensor placement problem and the associated probability computations. Finally, Section V presents the obtained numerical results for three optimal sensor placement examples of increasing complexity.

II. Problem Description

With reference to Fig. 1, define the *Work Cell* WC as a parallelepiped containing a number of static obstacles, the workspace of a fixed-base manipulator, and the typical workspace of a moving human operator. The work cell is monitored by p presence and d depth sensors. A coordinate frame is associated to each sensor, and we assume that all frames have been calibrated w.r.t. a global reference frame.

A. Sensor Modeling

We model each of the depth and presence sensors as a virtual camera, with the sensed information stored in its *Image Plane* IP. Using the so-called pinhole camera model, each projection ray passes through a point called *focal center* $\mathbf{F_c}$ and assigns its associated data to a point on the IP. The geometry of this ray projection is given by the sensor *projection matrix* $\mathcal{P}$, which contains extrinsic and intrinsic parameters. The extrinsic parameters are organized in an homogenous transformation matrix $\mathcal{E}$ from the global to the sensor frame. The intrinsic parameters are contained in a matrix $\mathcal{K}$ that projects a Cartesian point in the sensor coordinate frame to a point on its IP. If $\mathbf{X}$ is a point in the global frame, the corresponding point x on the IP is

$$\mathrm{x} = \mathcal{P}\mathbf{X} = \mathcal{K}\mathcal{E}\mathbf{X}. \qquad (1)$$

Indeed, all Cartesian points $\mathbf{X}$ laying on the same projection ray yields the same value x. Moreover, the *depth* of a Cartesian point with respect to a sensor is given by

$$\mathrm{depth}(\mathbf{X}) = \|\mathbf{F_c} - \mathbf{X}\|. \qquad (2)$$

A presence or depth sensor is characterized by an IP size, i.e., the dimensions of the array of pixels of its spatial discretization, and by a field of view (a solid angle). For a depth sensor, a range (between a minimum $\rho_{\min}$ and a maximum $\rho_{\max}$) and a depth resolution are also specified. Depth and presence sensors differ by the type of information stored in the pixels of the IP. The IP of a presence sensor contains a boolean information: a pixel is TRUE if the corresponding projection ray intercepts an object, FALSE otherwise. We call the IP content of a presence sensor the *Obstacle Presence Map* O_{PM}. To obtain the O_{PM} map, one can use background subtraction [9] or motion flow techniques on camera images. Instead, a pixel in the IP of a depth sensor contains the distance between the focal center and a detected object along the corresponding projection ray (or simply the *depth*). For an empty ray, the value of the corresponding pixel is set to $\rho_{\max}$. We call the IP content of a depth sensor the *Obstacle Depth Map* O_{DM}. The O_{DM} map can be computed using a stereo vision system, laser beams, PMD cameras, or integrated stereo vision cameras. Both maps O_{PM} and O_{DM} may be a function of time t in case of moving objects (i.e., the human operator).

B. Work Cell Analysis

Using depth and presence sensors it is possible to characterize which Cartesian points of the WC may belong to an obstacle. For a presence sensor, if a Cartesian point is detected as a part of an obstacle then all the points along the corresponding projection ray are considered part of the obstacle. Similarly, for a depth sensor, all the points along the corresponding projection ray with depth larger than the sensed value are considered as part of the obstacle.

Limiting ourselves to Cartesian points belonging to the work cell, the following standard definitions are recalled. The *Gray Area* GA of a sensor is the set of all Cartesian points in WC that are occluded by obstacles. The points of the GA that do not belong to the union O of real obstacles are called *Shadow Obstacle* SO. A *Pseudo Obstacle* PO is the union of O and SO[1]. Finally, the *Dark Area* DA of a sensor is the set of Cartesian points in WC that are not in its field of view while the *Free Area* FA of a sensor is the set of points in WC that are detected as certainly free of obstacles. Summarizing, at any time t, it is:

$$\begin{aligned} WC &= FA(t) \cup PO(t) \cup DA(t) \\ PO(t) &= O(t) \cup SO(t) \\ FA(t) \cap PO(t) &= \emptyset \\ FA(t) \cap DA(t) &= \emptyset \\ PO(t) \cap DA(t) &= \emptyset. \end{aligned}$$

[1] We shall regard the sets GA and PO as equivalent. They differ slightly only in the case of depth sensors, where the points $\mathbf{X}$ of the *surface* of the sensed obstacles belong to PO but not to GA.

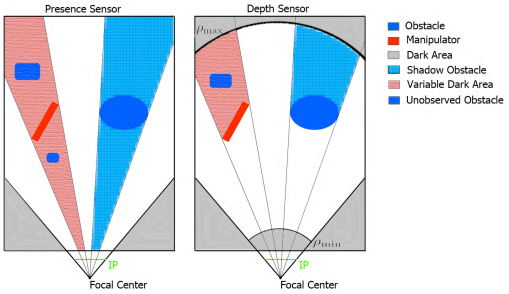

Fig. 2. Sketch of decomposition of a 2D WC into regions performed by a presence (left) or a depth sensor (right). A moving body (manipulator) is added here, and the (fixed) dark area DA is expanded with a variable part (VDA), possibly containing unobserved obstacles UO, see Sect. IV

For the two types of considered sensors, a pseudo obstacle is characterized via the 2D information on the IP (see Fig. 2). For a presence sensor, it is

$$PO_P(t) = \{\mathbf{X} \in WC : O_{PM}(\mathcal{P}\mathbf{X}, t) = \text{TRUE}\}.$$

For a depth sensor, it is

$$PO_D(t) = \{\mathbf{X} \in WC, \text{with}\, \rho_{\min} \le \text{depth}(\mathbf{X}) < \rho_{\max} : \\ O_{DM}(\mathcal{P}\mathbf{X}, t) \le \text{depth}(\mathbf{X})\}.$$

For illustration, consider the WC in Fig. 3, where a single (human) obstacle is monitored by two sensors. Figure 4 shows the O_{PM} map and the PO_P obtained using only the presence sensor, while Figure 5 shows the O_{DM} and PO_D for the depth sensor. The real obstacle in the WC is displayed with darker intensity. When integrating multiple sensors, it should be expected that all pseudo obstacles can be reduced to real obstacles. Apparently, the way to obtain an optimal sensor placement is to minimize/eliminate the dark areas and the shadow obstacles, as discussed in Sect. IV.

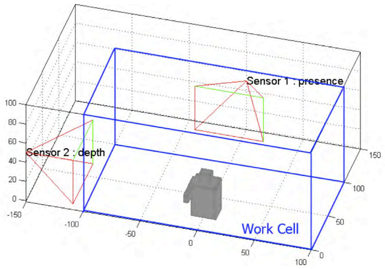

Fig. 3. Example of a work cell (left) monitored by two sensors (right)

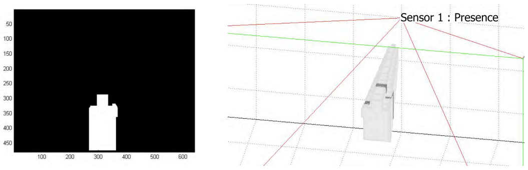

Fig. 4. The O_{PM} map obtained with the presence sensor (left) and the associated PO_P (right) for Fig. 3

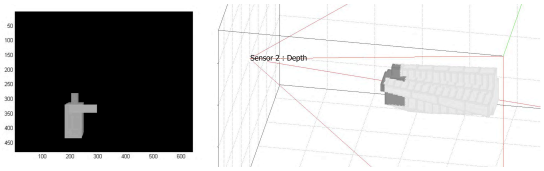

Fig. 5. The O_{DM} map obtained with the depth sensor (left) and the associated PO_D (right) for Fig. 3

III. Collision Prevention

Based on the approach presented in [6] for determining configurations of the manipulator to be considered dangerous for collision with a pseudo obstacle (a so-called *pseudo collision*) when using a single depth sensor, we include first the case of a presence sensor and then the integration of multiple sensors.

A. Manipulator Presence and Depth Maps

The robot manipulator is represented by a 3D geometric model that can be created using CAD programs or using simple primitive shapes such as cylinders or spheres[2]. For a given manipulator configuration $\mathbf{q}$, we can compute the position of all its Cartesian points using the manipulator direct kinematics and the geometric model. Applying the homogenous transformation $\mathcal{E}$, we obtain the coordinates of the manipulator points in the sensor frame.

The *Manipulator Presence Map* $M_{PM}(\mathbf{q})$ is obtained by projecting these points on the presence sensor IP using its intrinsic transformation $\mathcal{K}$. Since the norm of the position vector of a point in the sensor frame represents its depth, associating this information pixel by pixel in the depth sensor IP provides also the *Manipulator Depth Map* $M_{DM}(\mathbf{q})$. In particular, for a pixel $\mathbf{x}$ associated to Cartesian points $\mathbf{X}$ laying on the ray $\mathcal{P}\mathbf{X} = \mathbf{x}$ that *does not* intercept the robot body $RB \subset \mathbb{R}^3$, we set

$$M_{PM}(\mathbf{x}, \mathbf{q}) = \text{FALSE} \quad \text{and} \quad M_{DM}(\mathbf{x}, \mathbf{q}) = 0.$$

On the other hand, for a pixel $\mathbf{x}$ associated to Cartesian points $\mathbf{X}$ laying on the ray $\mathcal{P}\mathbf{X} = \mathbf{x}$ that *does* intercept the robot body, we set

$$M_{DM}(\mathbf{x}, \mathbf{q}) = \max_{\substack{\mathbf{X} \in RB \\ \mathcal{P}\mathbf{X} = \mathbf{x}}} \text{depth}(\mathbf{X}).$$

It should be noted that the manipulator presence and depth maps can be computed off-line for all manipulator configurations, being the relative pose between each sensor and the manipulator known and constant.

B. Estimation of Collision Configurations

Once the manipulator presence and depth maps have been obtained, the collision between a manipulator configuration and the pseudo obstacle can be easily estimated. With reference to Fig. 6, consider now both the presence and the depth maps of the manipulator and of the observed pseudo

[2] A small increase in robot dimensions allows more robust computations at the cost of some conservativeness.

obstacle. At a given time t, a generic configuration $\mathbf{q}$ of the manipulator is in pseudo collision if the following relations hold in the respective IPs. For a presence sensor,

$$\exists\, \mathbf{x} \in \mathrm{IP}_P : \{M_{PM}(\mathbf{x}, \mathbf{q}) \text{ AND } O_{PM}(\mathbf{x}, t)\} = \text{TRUE}. \quad (3)$$

For a depth sensor,

$$\exists\, \mathbf{x} \in \mathrm{IP}_D : M_{DM}(\mathbf{x}, \mathbf{q}) \geq O_{DM}(\mathbf{x}, t). \quad (4)$$

The collision tests are reduced to a trivial comparison between two-dimensional matrices. However, the inclusion of the robot in the WC should be handled with care, because the maps O_{PM} and O_{DM} will contain also the projections of the manipulator. The shadow of the observed manipulator would be evaluated as an obstacle, resulting in (false) self-colliding configurations at/close to the current robot configuration. The handy solution is not to scan those pixels in the maps during the collision test. This pixel omission produces what we call the *Variable Dark Area* —see Sect. IV.

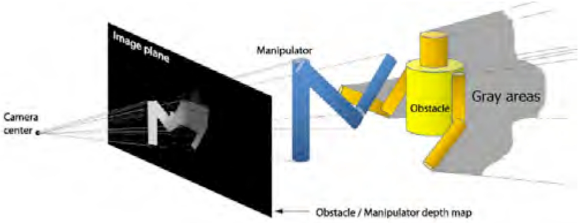

Fig. 6. For a single depth sensor, possible collisions are evaluated by comparing in the IP the obstacle map O_{DM} and the manipulator depth maps $M_{DM}(\mathbf{q})$ for a number of configurations $\mathbf{q}$ close to the current one [6]

C. Multiple Sensor Integration

Each sensor checks the collision between the robot manipulator and the pseudo obstacle, which is in general an overestimation of real *Observed Obstacles* $\widehat{O}$. This conservative estimation can be made more stringent by integrating the information of multiple sensors. Consider n sensors (of any type) that monitor the same work cell, so that n presence/depths maps are obtained with pseudo obstacles PO_i, $i = 1, \ldots, n$. The observed obstacles $\widehat{O}_i$ from the i-th sensor satisfy

$$\widehat{O}_i \subseteq PO_i, \quad i = 1, \ldots, n.$$

Moreover, it is

$$\bigcup_{i=1}^{n} \widehat{O}_i \subseteq \widetilde{PO} = \bigcap_{i=1}^{n} PO_i, \quad (5)$$

where $\widetilde{PO}$ is the pseudo obstacle obtained by integrating information from all sensors. While each sensor checks pseudo collisions using (3) or (4), the relation (5) leads to the simple rule for collision checking by sensor integration:

> *Collision with obstacles of a manipulator in the configuration* $\mathbf{q}$ *does not occur if at least one sensor has not detected a pseudo collision.*

Since $\widetilde{PO} \subseteq PO_i$, for all $i = 1, \ldots, n$, the evaluation of observed obstacles obtained by multiple sensor integration is more accurate than the evaluation by each single sensor.

For the example of Fig. 3, integrating the information from the two sensors gives the $\widetilde{PO}$ shown in Fig. 7. The shadow obstacle is reduced, while the isolated gray zone above the obstacle and the large one on the far right are due to the limited field of view of the two sensors. These zones could be shrunk with a better sensor placement.

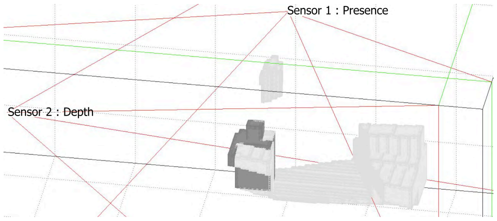

Fig. 7. The pseudo obstacle $\widetilde{PO}$ obtained integrating the two sensors of the example in Fig. 3

The analysis of the presence/depth maps for possible collisions proceeds as follows. Scan the first sensor map until a collision pixel is found. If no such pixel is found, then there is no collision. Else, move to the next sensor and repeat. Therefore, the worst case complexity grows linearly with the number n of sensors.

The above results extend the approach of [6], and can be directly used for improving the performance of the safe human/robot collision prevention scheme considered therein.

IV. Optimal Sensor Placement

To address the problem of optimal placement of multiple depth/presence sensors in the environment, we present a probabilistic approach based on work cell discretization.

Let the WC be decomposed in N regular elementary cells C_i, $i = 1, \ldots, N$. For each cell, we associate an a priori probability $P_O(C_i)$ to be part of an obstacle. These probabilities are assumed spatially independent and can be either set uniformly for all cells, or can be tailored based on previous experiments or knowledge about the environment. The probability to be an obstacle will be larger for cells where the human operator is expected to work. Cells belonging to known static obstacles, such as walls or tables, will have $P_O(C_i) = 1$.

Based on the geometry and placement of a single sensor, a cell occluded by other cells is in the gray area GA and then part of the pseudo obstacle for this sensor. We need actually to evaluate the probability $P_{SO}(C_i)$ that a cell is part of the shadow obstacle SO, i.e., part of the pseudo obstacle but not of the real obstacle. Thus, minimizing $P_{SO}(C_i)$ over all cells would give a good sensor placement.

Cells that are not observed are in the dark area of the sensor. It is useful to divide this area in two: the *Fixed Dark Area* is composed by cells that are not in the field of view of the sensor; the *Variable Dark Area* VDA is composed instead by cells that are behind the manipulator at a configuration $\mathbf{q}$. All those cells that are also part of a real obstacle will form an *Unobserved Obstacle* UO. Therefore, a better sensor placement should associate a lower probability $P_{UO}(C_i)$ to cells of an unobserved obstacle. Actually, cells in the dark area must be analyzed depending on the application of concern. For surveillance tasks, all cells in this area can be considered as part of the UO. For human/robot coexistence tasks, in order to ensure global human safety, cells in the fixed dark area must be considered in the first place as PO, while those in the variable dark area are counted as UO. In this way, in the collision prevention method proposed in [6] and extended in Sect. III, the manipulator will not be repulsed by its own shadow.

A. Cost Function

The cost function to be minimized for optimal sensor placement is defined as

$$J(\mathbf{s}) = \sum_{i=1}^{N} w_{SO}(C_i) P_{SO}(C_i) + k \sum_{i=1}^{N} w_{UO}(C_i) P_{UO}(C_i), \tag{6}$$

where the *sensor placement vector* $\mathbf{s}$ contains all the pose parameters of the sensors to be optimized. For instance, if we use $n = 2$ sensors, one to be placed on a vertical wall with constant orientation and another fixed in position but subject to a choice for its pan/tilt orientation, we will have a four-dimensional $\mathbf{s}$ with two linear and two angular variables. In (6), k is a positive factor used to *globally* determine the importance of covering the whole WC ($k \gg 1$, especially in surveillance tasks) with respect to minimizing detection of false collisions, and $w_{SO}(C_i)$ and $w_{UO}(C_i)$ are non-negative weights for *locally* handling cells in the two terms of the cost function. For example, we choose a large $w_{SO}(C_i)$ if we are mainly interested in detecting a potential collision in the particular cell C_i, or a large $w_{UO}(C_i)$ if this cell needs to be certainly covered by the sensors. For numerical purposes, the cost function (6) can be scaled by $1/N$.

Using (6), the optimal placement is obtained by solving the nonlinear optimization problem

$$\min_{\mathbf{s}} J(\mathbf{s}) \quad \text{s.t.} \quad \mathbf{A}\mathbf{s} \leq \mathbf{b}, \tag{7}$$

where the set of linear inequality constraints models the physical limits for the sensor poses (e.g., the range of angular values for a pan/tilt camera).

B. Computation of Probabilities

In order to evaluate the cost J in (6) for a given vector value $\mathbf{s}$, we need to compute probabilities of the cells in the WC according to their nature. Consider a single presence or depth sensor $sens$. When building the projection of cells on the IP through (1), one should take into account the finite size of pixels and cells. As a consequence, a single cell is projected onto more pixels and a single pixel contains the projection from multiple cells. In particular, with reference to Fig. 8, the projection rays to a pixel form a pyramid with tip on the focal center. All cells projected on the same pixel $\mathbf{x}$ will be ordered according to their (central) depth (2), by subdividing the pyramid in m *slices* $S_j^{\mathbf{x}}$, $j = 1, \ldots, m$. Each slice contains cells with the same range of depth; cells in $S_j^{\mathbf{x}}$ have a larger depth than cells in $S_k^{\mathbf{x}}$, for $j > k$.

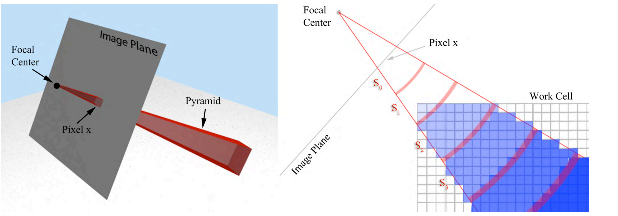

Fig. 8. Due to the finite size of a pixel, the rays of projection form a pyramid with tip on the focal center (left); a 2D view of the pyramid associated to a pixel and how cells are sorted in slices (right)

We will compute probabilities of slices rather than of single cells, and then assign these values to all cells in a slice. The probability of a slice to be part of an obstacle is given by

$$P_O(S_j^{\mathbf{x}}) = 1 - \prod_{i=1}^{N} A_j^{\mathbf{x}}(C_i) \overline{P_O(C_i)}, \tag{8}$$

where $\overline{P} = 1 - P$ and the binary indicator $A_j^{\mathbf{x}}(C_i)$ is 1 if cell C_i is projected on the jth slice of pixel $\mathbf{x}$ and 0 otherwise. Based on the $A_j^{\mathbf{x}}(C_i)$'s, it is useful to define also another binary indicator associated to $sens$:

$$E^{sens}(C_i) = \begin{cases} 1 & \text{if } C_i \text{ is projected on the IP of } sens, \\ 0 & \text{otherwise.} \end{cases}$$

For a presence sensor, a slice will contain cells that are in the gray area if at least one slice of the same pixel is an obstacle:

$$P_{GA}^{P}(S_j^{\mathbf{x}}) = \begin{cases} 1 - \prod\limits_{\substack{k=1 \\ k \neq j}}^{h-1} \overline{P_O(S_k^{\mathbf{x}})} & \text{if } j < h, \\ 1 & \text{if } j \geq h, \end{cases} \tag{9}$$

where h is the index of the first slice for which $P_O(S_h^{\mathbf{x}}) = 1$. Such index is needed because the techniques used to obtain the presence map (e.g., background subtraction) are not able to individuate static obstacles in the scene. Actually, when static obstacles are subtracted from the presence map, the slices beyond $S_h^{\mathbf{x}}$ should always be considered as obstacles.

For a depth sensor, a slice will contain cells that are in the gray area if the previous slice is not free, i.e., it contains an occluded cell which is part of the pseudo obstacle:

$$P_{GA}^{D}(S_j^{\mathbf{x}}) = P_{GA}^{D}(S_{j-1}^{\mathbf{x}}) + P_O(S_{j-1}^{\mathbf{x}}). \tag{10}$$

Note that when the manipulator is not considered, the probabilities for slices (cells) in the variable dark area will be $P_{VDA}^{P}(S_j^{\mathbf{x}}) = P_{VDA}^{D}(S_j^{\mathbf{x}}) = 0$.

C. Probabilities Including the Robot Manipulator

When a manipulator is included in the WC, a cell can belong to a pseudo obstacle, to the robot body RB, or to the free area. The probability $P_R(C_i)$ of a cell C_i to be part of the robot can be determined based on a frequency approach, either considering all robot configurations $\mathbf{q}$ to have the same probability or taking advantage of the known robot motion during a specific task. The probability $P_R(S_j^{\mathbf{x}})$ that a sensor slice belongs to the robot body is

$$P_R(S_j^{\mathbf{x}}) = 1 - \prod_{i=1}^{N} A_j^{\mathbf{x}}(C_i)\overline{P_R(C_i)}. \tag{11}$$

In our implementation, the robot manipulator images acquired by the sensor are subtracted respectively from O_{PM} and O_{DM}, in order to eliminate the trivial case of self-collision when using (3) or (4). Therefore, slices (cells) behind the manipulator are not considered as they will be part of the variable dark area.

For a presence sensor, a slice $S_j^{\mathbf{x}}$ before an obstacle contains gray area cells if no other slice contains robot cells and if there is at least a slice in the ray of projection containing an obstacle. The probability that the robot manipulator is not in the projection ray up to the h-th slice is

$$P_{NR}(S_j^{\mathbf{x}}) = \prod_{\substack{k=1\\k\neq j}}^{h-1} \overline{P_R(S_k^{\mathbf{x}})}. \tag{12}$$

Thus, the probability of slice $S_j^{\mathbf{x}}$ to be in the gray area is

$$P_{GA}^{P}(S_j^{\mathbf{x}}) = \begin{cases} P_{NR}(S_j^{\mathbf{x}}) \left(1 - \prod\limits_{\substack{k=1\\k\neq j}}^{h-1} \overline{P_O(S_k^{\mathbf{x}})}\right) & \text{if } j < h, \\ 1 & \text{if } j \geq h \end{cases} \tag{13}$$

and the probability that slice $S_j^{\mathbf{x}}$ is in the variable dark area is

$$P_{VDA}^{P}(S_j^{\mathbf{x}}) = \begin{cases} 1 - P_{NR}(S_j^{\mathbf{x}}) & \text{if } j < h, \\ 0 & \text{if } j \geq h. \end{cases} \tag{14}$$

For a depth sensor, a slice $S_j^{\mathbf{x}}$ contains gray area cells if a previous slice contains an obstacle and this obstacle is not subtracted due to the presence of the robot manipulator. The probability that the j-th slice contains an obstacle and no previous slices contains cells of the robot is

$$P_{ONR}(S_j^{\mathbf{x}}) = P_O(S_j^{\mathbf{x}}) \prod_{k=1}^{j-1} \overline{P_R(S_k^{\mathbf{x}})}. \tag{15}$$

Thus, the probability of slice $S_j^{\mathbf{x}}$ to be in the gray area is

$$\begin{aligned} P_{GA}^{D}(S_j^{\mathbf{x}}) &= P_{GA}^{D}(S_{j-1}^{\mathbf{x}}) + P_{ONR}(S_{j-1}^{\mathbf{x}}) \\ &\quad - P_{GA}^{D}(S_{j-1}^{\mathbf{x}}) P_{ONR}(S_{j-1}^{\mathbf{x}}), \end{aligned} \tag{16}$$

since the two considered events are not disjoint. Similarly, the current slice is in the variable dark area if a previous slice contains cells of the robot manipulator and none of the previous slices contain an obstacle. The probability that $S_j^{\mathbf{x}}$ contains cells of the robot and no previous slice contains an obstacle is

$$P_{RNO}(S_j^{\mathbf{x}}) = P_R(S_j^{\mathbf{x}}) \prod_{k=1}^{j-1} \overline{P_O(S_k^{\mathbf{x}})}, \tag{17}$$

and thus

$$\begin{aligned} P_{VDA}^{D}(S_j^{\mathbf{x}}) &= P_{VDA}^{D}(S_{j-1}^{\mathbf{x}}) + P_{RNO}(S_{j-1}^{\mathbf{x}}) \\ &\quad - P_{VDA}^{D}(S_{j-1}^{\mathbf{x}}) P_{RNO}(S_{j-1}^{\mathbf{x}}). \end{aligned} \tag{18}$$

D. Integration of Probabilities

Having computed the probabilities for all slices and all pixels of a given sensor, the relevant probabilities of the cells are obtained as

$$\begin{aligned} P_{GA}^{sens}(C_i) &= 1 - \prod_{j=1}^{m} \overline{P_{GA}^{sens}(S_j^{\mathbf{x}})} \\ P_{VDA}^{sens}(C_i) &= 1 - \prod_{j=1}^{m} \overline{P_{VDA}^{sens}(S_j^{\mathbf{x}})}, \end{aligned} \tag{19}$$

where products involve only factors with $A_j^{\mathbf{x}}(C_i) = 1$.

When using n sensors (p for presence and d for depth, $n = p+d$), the probabilities of each sensor must be integrated according to the general rule presented in Sect. III-C:

$$\begin{aligned} P_{GA}(C_i) &= \prod_{sens=1}^{n} P_{GA}^{sens}(C_i) \\ P_{VDA}(C_i) &= 1 - \prod_{sens=1}^{n} \overline{P_{VDA}^{sens}(C_i)}, \end{aligned} \tag{20}$$

where products involve only factors with $E^{sens}(C_i) = 1$. Equation (20) represents the probabilistic version of the following deterministic statements: *i)* a cell is in the gray area only if all sensors categorizes it in their gray area; *ii)* a cell is the variable dark area if at least a sensor categorizes it in its variable dark area.

Finally, if a cell C_i is not in the field of view of any sensor,

$$\sum_{sens=1}^{N} E^{sens}(C_i) = 0,$$

then it is part of the fixed dark area and we set

$$P_{GA}(C_i) = 1, \quad P_{VDA}(C_i) = 0. \tag{21}$$

We have now all ingredients for evaluating the probabilities that are used in (6) within the cost function J. A cell is part of a shadow obstacle if it is in the gray area but is not an obstacle:

$$P_{SO}(C_i) = P_{GA}(C_i)\overline{P_O(C_i)}. \tag{22}$$

A cell is part of an unobserved obstacle if it is in the variable dark area and is an obstacle:

$$P_{UO}(C_i) = P_{VDA}(C_i) P_O(C_i). \tag{23}$$

Furthermore, the cell weights appearing in J can take special values that depend on the probability of a cell to belong to the robot body. In fact, the main objective for human/robot coexistence tasks is to monitor and detect potential collisions in those cells of the WC that correspond to configurations that can be assumed by the manipulator. It is useful to set

$$w_{SO}(C_i) = w_{UO}(C_i) = P_R(C_i), \tag{24}$$

when $P_R(C_i) \leq 1 - \epsilon$, being $\epsilon > 0$ a suitable small parameter. On the other hand, if $P_R(C_i) > 1 - \epsilon$, we shall set $w_{SO}(C_i) = w_{UO}(C_i) = 0$, since such a cell will almost always be occupied by the manipulator.

In our probabilistic developments, we have considered that sensors are mutually independent. Without consideration of the pixel and cell finite size, this assumption holds unless two sensors have the same position. In practice, this assumption is valid if the sensors are not too close to each other. Although such sensor placement would certainly be not optimal, these situations can be avoided by adding a penalty term to J for sufficiently close conditions on the parameter values.

V. Examples of Optimal Sensor Placement

The presented theory and the optimal sensor placement method has been tested on different simulated environments. To this end, we have realized a simulator that includes the kinematics of an articulated manipulator, static obstacles, a human operator, and the extrinsic/intrisic projection characteristics of presence and depth sensors. For each of the N elementary cells C_i of the work cell WC, the probabilities $P_O(C_i)$ to be an obstacle and $P_R(C_i)$ to be part of the robot manipulator can set by the user.

For each problem of the form (7), the optimal solution was obtained using the pattern search algorithm of the Genetic Algorithm and Direct Search Toolbox of Matlab™, using a final tolerance of 0.01 for the solution $\mathbf{s}^*$ and without the need of the gradient of cost function J. We present three examples with a WC of size $270 \times 270 \times 300$ cm decomposed in $N = 836381$ cubic cells of 3 cm side. The IPs of all sensors have 640×480 pixels and depth sensors have 2 cm resolution. The robot is a $3R$ elbow-type manipulator.

A. One Sensor

In the first example, shown in Fig. 9, we have considered the two cases of using a single depth or a single presence sensor, which can be positioned on a circular support or radius 170 cm surrounding the WC. The optimization parameter is a scalar s, normalized between 0 and 1. The work cell contains a wall that may occlude the sensor. The cell probabilities associated to the manipulator are set considering all configurations in its joint ranges as equally probable. The cell probabilities associated to the moving obstacle (the human operator) are chosen larger slightly on the right of the center of the WC, decreasing linearly toward the edges (see the shade intensities in the right part of Fig. 9). The factor $k = 14$ was used in (6), after some trials.

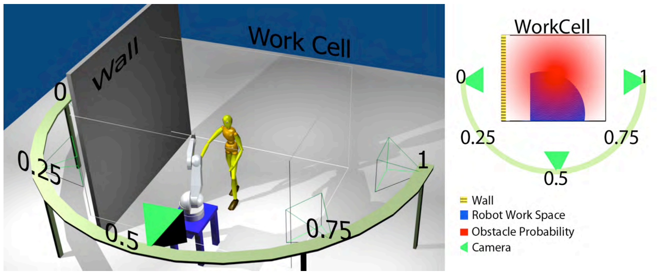

Fig. 9. Environment of the first example: 3D view (left) and schematic top view (right). The optimal solution for the depth sensor is shown

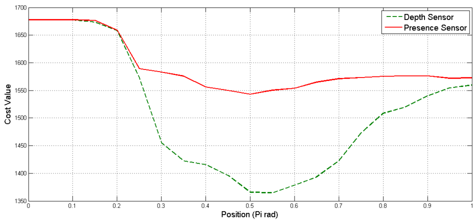

Fig. 10. Cost function for the first example

Figure 10 shows the costs J obtained with the presence or the depth sensor respectively, as a function of the normalized position of the sensor. It should be noted that both plots have the maximum at $s = 0$, due to the presence of the occluding wall. The cost function is always smaller for the depth sensor, as we may have expected from the poorer capabilities of the presence sensor. The optimal placement for depth sensor is at $s_D^* = 0.5781$. This position allows the best discrimination between cells typically occupied by the manipulator and cells mostly occupied by the human, according to the given probabilities. The solution was obtained after 16 iterations of the algorithm starting at $s_0 = 0$. The computational time was 85 min on a Intel Core Duo 2.4 GHz processor. Similarly, the optimal placement for the presence sensor is $s_P^* = 0.5$.

B. Two Sensors

In the second example, shown in Fig. 11, we consider two sensors. The depth sensor can be placed on a horizontal circular support, as in the first example, while the presence sensor is on a vertical circular support of the same size.

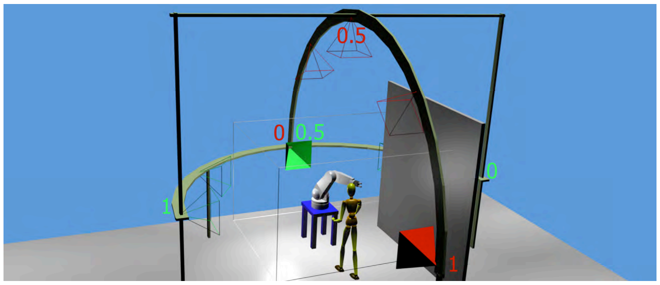

Fig. 11. Environment of the second example: two sensors are used (presence in red, depth in green) and their optimal placement is shown

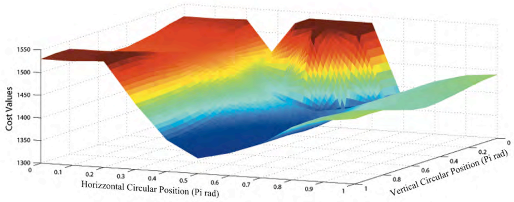

Fig. 12. Cost function for the second example

Figure 12 shows the 2D plot of the cost J. The highest saturated zones (in deep red) correspond to the wall occluding the depth sensor and to approaching a situation of sensor dependence, i.e., when $s_D = 0.5$ and $s_P = 0$, which is penalized in the cost function. Note that when the depth sensor is placed behind the wall, the value of J is the same as in the first example, with presence sensor at $s_P = 0.5$. The optimal placement is $\mathbf{s}^* = (s_D^*, s_P^*) = (0.5, 0.9688)$, obtained in 26 iterations from $\mathbf{s_0} = \mathbf{0}$ and 320 min time.

C. Three Sensors

In this last example, we consider one presence and two depth sensors that can be placed on limited square areas of 200×200 cm, each located on one of three orthogonal walls (see Fig. 13). An orientation constraint is added, so that the sensors focal axes point to $\mathbf{P_c} = (15, 35, 30)$ [cm] (the intersection of the normals to the centers of the feasible square areas) from any chosen position. The parameter vector is $\mathbf{s} = (x_1, y_1, y_2, z_2, x_3, z_3) \in \mathbb{R}^6$, and is initialized at $\mathbf{s_0} = (15, 35, 35, 30, 15, 30)$ [cm], i.e., with the presence sensor pointing downward vertically and the two depth sensors pointing straight horizontally. The human operator works mainly in front of the manipulator and this is taken into account in the probability distribution $P_O(C_i)$ of the cells. An occluding wall is also present. The optimal placement $\mathbf{s}^* = (-85, 134.96, 134.56, -70, 8.07, -69.98)$ [cm] is obtained after 568 iterations in about 5500 minutes running time, and is shown in Fig. 13.

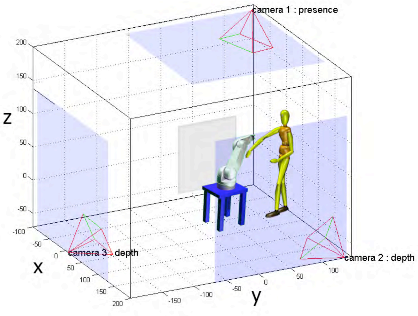

Fig. 13. Environment of the third example: three sensors are used (one presence and two depth) and their optimal placement is shown

VI. Conclusion

We have extended the method for checking potential collision configurations of a manipulator with a human and/or obstacles introduced in [6], including both presence and depth sensor types. A systematic procedure to obtain multiple depth/presence sensor integration has been proposed. The key aspect is that 2D computations are made in parallel on each image plane of the different sensors, and information is then fused in a straightforward way.

In order to maximize the sensor integration performance, an optimal sensor placement problem was formulated within a probabilistic framework and analyzed in detail. Based on a cell decomposition of the working area, the probabilities of having unobserved and shadow obstacles are determined in an incremental way and their weighted sum can be minimized numerically. The solution can be tailored to the specific tasks that the human operator and robot manipulator are expected to perform, in a probabilistic sense.

Results were presented for three representative environments with static obstacles, an articulated manipulator, and a human operator. The obtained sensor placements are significant, also from an intuitive point of view. The current bottleneck seems to be the large computational times required for solving off-line the optimization problem by standard numerical algorithms, in particular when the number of sensor placement parameters grows.

Acknowledgements

This work has been funded by the MIUR national project PRIN 2007 SICURA.

References

[1] A. Bicchi, M. Peshkin, and J. Colgate, "Safety for physical human-robot interaction," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 1335–1348.

[2] A. Bicchi and G. Tonietti, "Fast and soft arm tactics: Dealing with the safety-performance trade-off in robot arms design and control," *IEEE Robotics and Automation Mag.*, vol. 11, pp. 22–33, 2004.

[3] A. De Luca, A. Albu-Schäffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the DLR-III lghtweight manipulator arm," in *Proc. 2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006, pp. 1623–1630.

[4] E. Sisbot, L. Marin-Urias, R. Alami, and T. Siméon, "A human aware mobile robot motion planner," *IEEE Trans. on Robotics*, vol. 23, pp. 874–883, 2007.

[5] S. Kuhn, T. Gecks, and D. Henrich, "Velocity control for safe robot guidance based on fused vision and force/torque data," in *Proc. 2006 IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, 2006, pp. 485–492.

[6] R. Schiavi, F. Flacco, and A. Bicchi, "Integration of active and passive compliance control for safe human-robot coexistence," in *Proc. 2009 IEEE Int. Conf. on Robotics and Automation*, 2009, pp. 259–264.

[7] D. Ebert, T. Komuro, A. Namiki, and M. Ishikawa, "Safe human-robot-coexistence: emergency-stop using a high-speed vision-chip," in *Proc. 2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2005, pp. 2923–2928.

[8] I. Iossifidis and G. Schoner, "Dynamical systems approach for the autonomous avoidance of obstacles and joint-limits for an redundant robot arm," in *Proc. 2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006, pp. 580–585.

[9] D. Henrich and T. Gecks, "Multi-camera collision detection between known and unknown objects," in *Proc. 2nd ACM/IEEE Int. Conf. on Distributed Smart Cameras*, 2008, pp. 1–10.

[10] M. Fischer and D. Henrich, "3D collision detection for industrial robots and unknown obstacles using multiple depth images," in *Advances in Robotics Research: Theory, Implementation, Application*, T. Kröger and F. Wahl, Eds. Springer, 2009, pp. 111–122.

[11] M. Bodor, A. Drenner, P. Schrater, and N. Papanikolopoulos, "Optimal camera placement for automated surveillance tasks," *J. of Intelligent and Robotic Systems*, vol. 50, pp. 257–295, 2007.

[12] J.-J. Gonzalez-Barbosa, T. Garcia-Ramirez, J. Salas, J.-B. Hurtado-Ramos, and J. d. J. Rico-Jimenez, "Optimal camera placement for total coverage," in *Proc. 2009 IEEE Int. Conf. on Robotics and Automation*, 2009, pp. 844–848.

[13] E. Becker, G. Guerra-Filho, and F. Makedon, "Automatic sensor placement in a 3D volume," in *Proc. 2nd Int. Conf. on Pervasive Technologies Related to Assistive Environments*, 2009.

[14] S. Nikolaidis, R. Ueda, A. Hayashi, and T. Arai, "Optimal camera placement considering mobile robot trajectory," in *Proc. 2008 IEEE Int. Conf. on Robotics and Biomimetics*, 2008, pp. 1393–1396.

[15] G. Olague and R. Mohr, "Optimal camera placement to obtain accurate 3D point positions," in *Proc. 14th Int. Conf. on Pattern Recognition*, vol. 1, 1998, pp. 8–10.

[16] X. Chen and J. Davids, "Camera placement considering occlusion for robust motion capture," Stanford University, Computer Science Technical Report CS-TR-2000-07, 2000.

[17] Y. Xu, D. Song, J. Yi, and F. van der Stappen, "An approximation algorithm for the least overlapping *p*-frame problem with non-partial coverage for networked robotic cameras," in *Proc. 2008 IEEE Int. Conf. on Robotics and Automation*, 2008, pp. 1011–1016.