Intelligent Object Anchoring using Relative Anchors

Torsten Heyer
Institute of Automation
Otto-Hahn-Allee 1, NW1, D-28359 Bremen
University of Bremen, Germany
Email: theyer@iat.uni-bremen.de

Axel Gräser
Institute of Automation
Otto-Hahn-Allee 1, NW1, D-28359 Bremen
University of Bremen, Germany
Email: ag@iat.uni-bremen.de

***Abstract*—Object anchoring is a very important and useful concept, especially for robotic systems which use two different levels to represent objects, symbolic and sub-symbolic. The right sub-symbolic information acquired from sensors have to be combined with the symbolic description that points to the same physical object which is often a time-consuming and error-prone procedure, e.g. due to unsuccessful detection using machine vision. In this paper an extension of this anchoring concept is given using *relative anchors* which presents an intelligent object representation and allows the reduction of necessary machine vision operations. Also the inaccuracies given by the algorithms used for detection are taken into account and used to enhance the object anchoring. The concept is evaluated on a robotic system in a library scenario and the benefit is presented.**

I. Introduction

Mobile service robotic systems operate in unknown environments and have to work with a variety of different objects. Therefore objects with which the robot should operate have to be detected first, e.g. using machine vision. When the mobile robotic system should perform a desired task, e.g. grasp an object with an on-board mounted manipulator, the precise positions of objects must be known. If the robot changes its position when moving around, all objects need to be acquired again since the previous determined positions are no longer valid.

In order to make such system intelligent a proper strategy is necessary to handle, store and reuse the acquired information and to cope or compensate the inaccuracies coming from the acquisition algorithms. Often already known information can be used without being acquired again. Therefore a control strategy on a higher level then the task planing level is necessary which can reduce the often time-consuming and error-prone acquisition algorithms. One solution is *object anchoring*, introduced by Coradeschi et. al. [1]. They define object anchoring as the *process of creating and maintaining the correspondence between symbols and percepts that refer to the same physical object*. It is proved to be a reliable concept for systems working *with two different types of processes typically incorporated in autonomous systems that are designed to act in the real physical world: The type of processes that operate on the symbolic level and those that operate on the sub-symbolic level* [2]. When mobile robots with symbolic cognitive layer interact with different objects in the surrounding environment, an anchoring of these objects will be frequently needed [3].

The object anchoring concept introduced by Coradeschi et. al. were already presented in a series of papers (e.g. [1], [4]–[6]) and in [7] a good review of past and possible future trends in perceptual anchoring is given as well as other existing approaches to solve the anchoring problem. Also extensions of object anchoring were presented in the literature. In [3] an approach for recovery from perceptual failures, or more precisely anchoring failures is presented which creates hypothesis about world states and handles uncertainty in terms of probabilistic belief states. The authors in [8] create belief states about (temporally) unobserved objects which were previously observed by the robot and propose a *dynamic object store* to remember where the objects are or are likely to be.

In this paper a new extension of the object anchoring concept by Coradeschi is introduced using *relative anchors* which improves its use for mobile robots in unknown environments. It also takes into account inaccuracies coming from the machine vision algorithms and copes these influences partially. This concept reduces the number of necessary machine vision operations significantly and can be used for the robot to get to know and learn its environment itself. So the system can react in a flexible and intelligent manner.

The paper is organized as follows. First a short introduction to the object anchoring concept by Coradeschi is given. In Section II we give the mathematical description of our extension and explain the concept with several examples. In Section III we apply the presented concept to the robotic system FRIEND and evaluate its benefit in a library scenario. Finally, conclusions and outlook are presented in Section IV.

A. Object Anchoring

Here a short introduction to the object anchoring concept by Coradeschi is given and some definitions are repeated to understand the presented extension explained later which are based on these notations. The definitions and the example in this chapter are taken from [4].

The overall object anchoring concept with the presented extension is graphically displayed in Figure 1. The green highlighted classes belong to the original definition of the

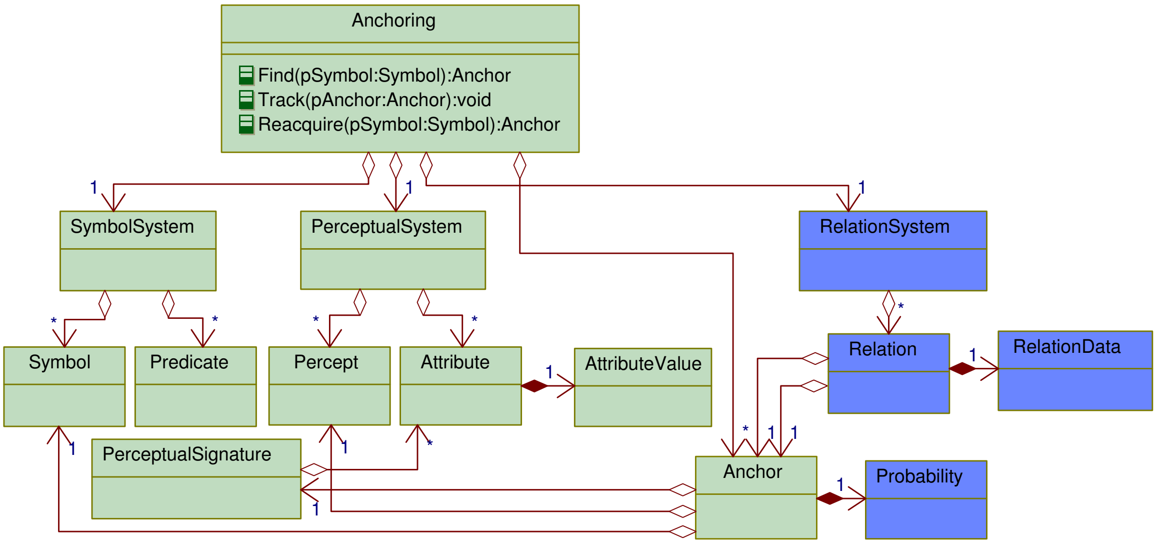

Fig. 1. Object anchoring overview: object anchoring by Coradeschi et.al. [4] (green) and new extensions (blue).

anchoring problem described in [4]. The elements that are time invariant are:

- A *symbol system* Σ including: a set $\chi = \{x_1, x_2, ...\}$ of individual symbols (variables and constants); a set $P = \{p_1, p_2, ...\}$ of predicate symbols; and an inference mechanism.
- A *perceptual system* Ξ including: a set $\Pi = \{\pi_1, \pi_2, ...\}$ of percepts; s set $\Phi = \{\varphi_1, \varphi_2, ...\}$ of attributes; and perceptual routines. A percept is a structured collection of measurements assumed to originate from the same physical object; an attribute φ_i is a measurable property of percepts with values in the domain $D(\varphi_i)$.

A *perceptual signature* $\gamma : \Phi \rightarrow D(\Phi)$ is a partial function from attributes to attributes values, which gives the values of the measured attributes of a percept. Then an *anchor* is any partial function α from time to triples in $\chi \times \Pi \times (\Phi \rightarrow D(\Phi))$. An anchor is a unique internal representation of an object o in the environment. At every moment t $\alpha(t) = \left(\alpha_t^{sym}, \alpha_t^{per}, \alpha_t^{val}\right)$ contains: a symbol meant to denote o; a percept generated by observing o; and a signature meant to provide the current (best) estimate of the values of the observable properties of o.

As an example two individual symbols $\chi = \{car1, car2\}$ are given, and the predicate set is $P = \{car, small, big, red, blue\}$. The precepts are the two image regions π_1 and π_2 at which the cars are detected. The set of attributes is $\Phi = \{type, colour, shape\}$. A perceptual signature could be γ_1

$$\gamma_1 : colour \mapsto (10, 1, 1) \text{ and } \gamma_1 : shape \mapsto (8, 4),$$

whereas the domain of *colour* is the set of triples of possible hue, saturation, and luminosity values; and the domain of *shape* is the set of pairs of possible length an width values. An anchor created at time t_1 connects these information and would then be $\alpha : t_1 \mapsto (car1, \pi_1, \gamma_1)$. In order for an anchor to satisfy its intended meaning, the symbol and the percept in it should refer to the same physical object.

An anchor α is *referentially correct* if, whenever α is grounded at t, then the physical object denoted by α_t^{sym} is the same as the one that generates the perception α_t^{per}. The anchoring problem, then, is the problem to find referentially correct anchors. In order to solve the anchoring problem for a symbol x three functionalities are used which seem to be sufficient to solve this problem in several domains.

- **Find()** to create a grounded anchor the first time that the object denoted by x is perceived.
- **Reacquire()** to update the anchor when the object is reacquired after some time that is has not been observed.
- **Track()** to continuously update the anchor while observing the object.

Since the subsymbolic description, i.e. the sensor data, is acquired by the robot, all objects are grounded w.r.t. the robot, which represents the world coordinate system. In the following these kind of anchors are called *absolute anchors* in contrast to the *relative anchors* introduced in the next chapter.

II. Relative Object Anchoring

A. Introduction and Definitions

When a mobile robot moves all absolute anchors created using Coradeschi's concept become obsolete, when the objects are not tracked. Then they need an update since the relative positions of the objects w.r.t. the robot are changed. So the objects have to be re-acquired and re-anchored, respectively. Since the relative positions between the objects are not changed in a fixed environment when the robot changes its position our approach anchors already grounded anchors between each other. Therefore the object anchoring concept by Coradeschi was extended by introduction of *relative anchors*. It is clear that machine vision algorithms yield no precise data and are influenced by detection inaccuracies depending on the algorithm used for the acquisition. For example objects can be detected very easy, reliable and with a high accuracy using markers (e.g. SIFT [9], SURF [10] or ARToolKit [11]). Low-level image processing algorithms like colour segmentation or edge detection are highly depending on varying illumination conditions which influences the quality of detection and subsymbolic information. But therefore the objects do not have to be prepared with artificial markers. Since no precise subsymbolic data is available the perceptual signature γ of a grounded anchor yields inaccurate object locations.

To consider these kind of uncertainty it is assumed that the determined location of an object o_i is distributed like a Gaussian distribution, i.e. the position of o_i is mathematically modelled by a Gaussian distributed random variable $O_i \sim N(\mu_i, \sigma_i^2)$. The mean μ_i is the precise object position and the variance σ_i^2 describes the degree of inaccuracy and the spreading of the distribution. The probability

$$P\left(|O_i - \mu_i| < \varepsilon\right) \tag{1}$$

decreases when the variance increases. The more reliable the machine vision algorithm is used for recognition of object o_i the lower the variance σ_i^2 is chosen for the corresponding random variable O_i. Therefore objects which are detected by artificial markers are modelled by a random variable with a lower variance than objects which are detected by low-level machine vision algorithms.

To perform a reliable object anchoring the probability in Equation 1 should be higher than a given level. This level depends on the accuracy necessary for grasping and manipulation. The value ε is a factor for the allowed inaccuracy of the subsymbolic information, i.e. $[\mu_i - \varepsilon, \mu_i + \varepsilon]$ is the interval for which a reliable object anchoring can be performed in the sense that the subsymbolic data is precise enough to be used by the robot. When the position of a book is acquired for example with a tolerance of one centimetre it also can be successfully grasped by a manipulator.

The originally definition of an anchor introduced by Coradeschi et. al. [4] is extended here by an accuracy component.

Definition 1. [Anchor] An *anchor* is any partial function α from time t to quadruples in $\chi \times \Pi \times (\Phi \rightarrow D(\Phi)) \times [0, 1]$. The components of $\alpha(t)$ are denoted by $\alpha_t^{sym}, \alpha_t^{per}, \alpha_t^{val}$ and α_t^{acc}, respectively.

The last component α_t^{acc} contains the accuracy of the anchors, i.e. how much the position of the corresponding object is reliable. When the robot moves around all grounded absolute anchors are lost since the subsymbolic information based on which the anchor was grounded, i.e. the position of the objects relative to the robot, is no longer valid. But in a fixed environment the objects do not change their position between each other and these relative information can be used to update the subsymbolic data of other objects. This leads us to the formal definition of a relative anchor.

Definition 2. [Relative anchor] A *relative anchor* is any partial function ρ from time t to triples in $A_{t_1} \times A_{t_2} \times [0, 1]$ with $t = \max\{t_1, t_2\}$, $t_1, t_2 \in \mathbb{R}$ and $A_t = \{\alpha_t\}$ the set of all at time t grounded anchors. The components of $\rho(t)$ are denoted by $\rho_t^{ref_1}, \rho_t^{ref_2}$ and ρ_t^{prob}. Then the two anchors $\alpha_{t_1} = \rho_t^{ref_1}$ and $\alpha_{t_2} = \rho_t^{ref_2}$ have a *relation* between each other.

Any relative anchor $\rho(t)$ grounded at time t contains the references to the two involved anchors between which the relation is created, called *reference anchors* of the relation, and a probability ρ_t^{prob} of the relation.

As seen in Figure 1 (blue highlighted classes) the model by Coradeschi is extended by a time-depending *relation system* Ψ including a set $R_t = \{\rho_1, \rho_2, ...\}$ of relations, consisting of the references to already grounded anchors and the necessary data which is given by the *relation signature*.

Definition 3. [Relation signature] The *relation signature* γ_ρ of a relative anchor $\rho_t = \rho(t)$ is a function from the two perceptual signatures of the involved anchors α_{t_1} and α_{t_2} to the relative spatial information

$$\gamma_\rho : (\Phi \rightarrow D(\Phi)) \times (\Phi \rightarrow D(\Phi)) \rightarrow D(\Phi)$$
$$\gamma_\rho(\alpha_{t_1}^{val}, \alpha_{t_2}^{val}) := dist_{spa}(\alpha_{t_1}^{val}, \alpha_{t_2}^{val}) \in D(\Phi_{spa})$$

whereas $\Phi_{spa} \subset \Phi$ contains all spatial attributes, e.g. position, orientation and location information, and $dist_{spa}(\cdot, \cdot)$ gives the relative location between the two involved objects.

The relation signature gives the relation data of two previous grounded anchors, i.e. where a specific object is placed w.r.t. to another one. So that a relative anchor can be created the two involved absolute anchors have to be grounded at that time, since their subsymbolic information have to be valid in order to create the data for the relation. When α_{t_1} is a relative anchor of α_{t_2} then also α_{t_2} is a relative anchor of α_{t_1} and

$$\gamma_\rho\left(\alpha_{t_1}^{val}, \alpha_{t_2}^{val}\right) = \gamma_\rho\left(\alpha_{t_2}^{val}, \alpha_{t_1}^{val}\right)^{-1}.$$

Relative anchors can be exist even when the involved objects are at the moment not anchored, since in a fixed environment their relative positions between each other remain. They can be used to re-anchor objects later without any re-detection. How this is done is discussed in details in the next chapter. It's clear that from time to time also relative anchors become obsolete and have to be deleted.

B. Use of relative Anchors

When the robot moves around, its position relative to all objects is changed. Therefore grounded anchors are no longer valid, marked as obsolete and have to be re-acquired at the next time. But relative anchors are still valid since the object positions relative to each other were not changed. When an object which has relative anchors is re-acquired by the robot its relative anchors can be used to update the absolute anchors of objects to which relations exist. This procedure reduces the often time-consuming and error-prone image processing algorithms which are necessary for a robot to detect objects in the

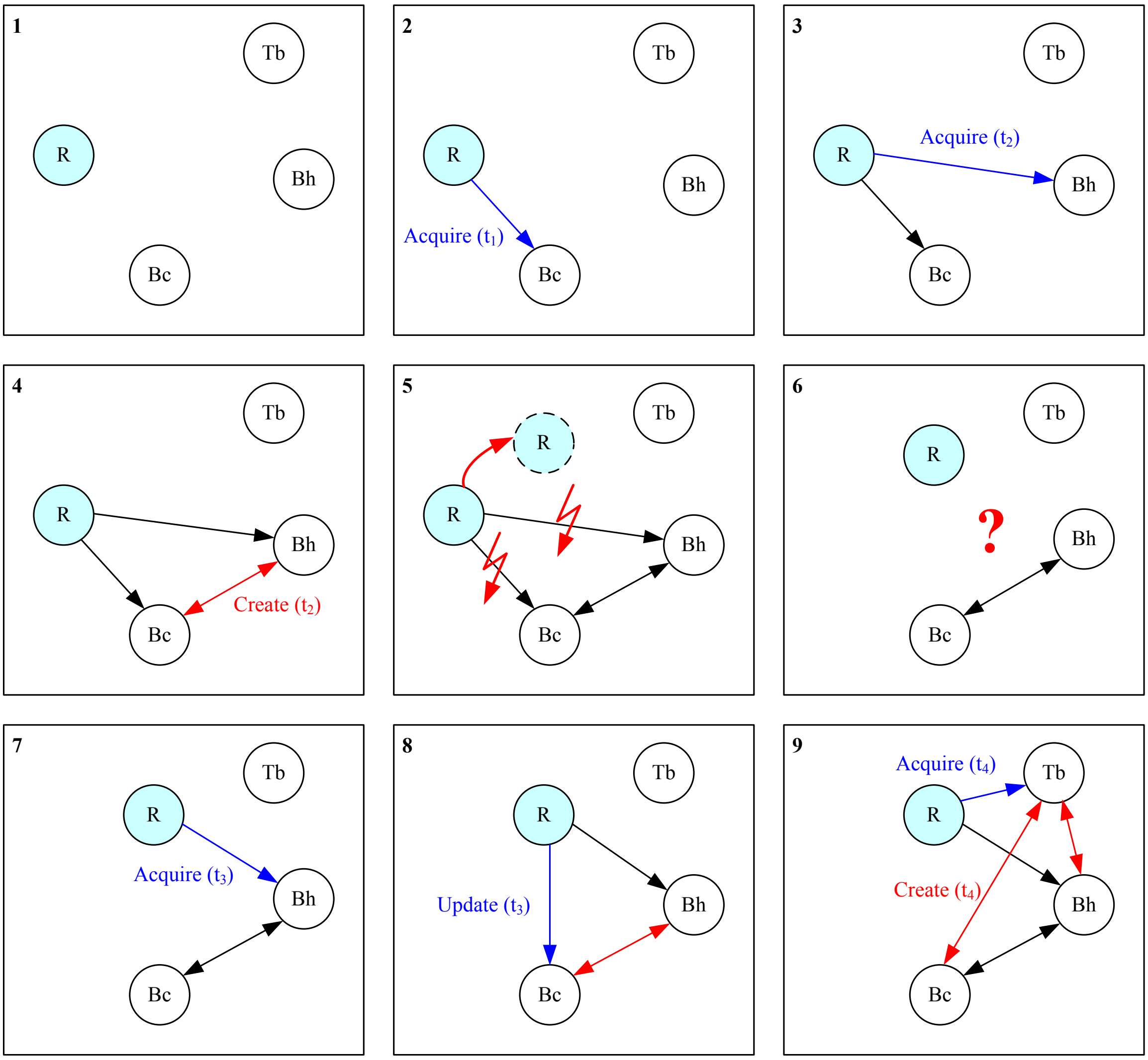

Fig. 2. Grounding and use of relative anchors in a library environment with a robot (R), a book cart (Bc), a book holder (Bh) and a table (Tb).

environment and relies on previous determined subsymbolic information.

Example. The grounding and update of the anchors, absolute and relative, is graphically displayed in Figure 2. We consider a library environment, reduced to a two-dimensional case, with a mobile robot (R) and different objects: a book cart (symbol: Bc, precept: π_1, symbolic description: γ_1, accuracy: q_1), a book holder (Bh, π_2, γ_2, q_2) and a table (Tb, π_3, γ_3, q_3) (Figure 2.1). The corresponding anchors $\alpha_{t_1}, \alpha_{t_2}$ of the first two objects were created and grounded at time t_1 and $t_2 > t_1$, respectively (Figure 2.2-2.3, blue arrows)

$$\alpha_{t_1} : t_1 \longmapsto (Bc.1, \pi_1, \gamma_1, q_1),$$
$$\alpha_{t_2} : t_2 \longmapsto (Bh.1, \pi_2, \gamma_2, q_2).$$

The only spatial attribute is here the position (x- and y-coordinate) w.r.t. the robot. When the book cart is detected and anchored at time t_2 the relative anchor ρ

$$\rho : t_2 \longmapsto (\alpha_{t_1}, \alpha_{t_2}, q_{12})$$

is grounded calculating the relative location of the book holder w.r.t. the book cart (Figure 2.4, red arrow). It's clear that the robot must not change its position between time t_1 and t_2, otherwise the at t_1 grounded absolute anchor becomes obsolete due to the no longer valid position information and the relation ρ cannot be created. The relative anchor ρ is still valid when the robot moves in the library but all absolute anchors become obsolete (Figure 2.5-2.6). When the robot re-acquires the book holder at time t_3 (Figure 2.7) automatically the location of the book cart can be re-calculated using the relative anchor ρ (Figure 2.8). When another object is detected, e.g. a table at time t_4, the net of relations can be extended and new relative anchors to the already grounded anchors of the book cart and the book holder can be created (Figure 2.9).

As already mentioned also relative anchors may become invalid due to different reasons.

1) Object is manipulated: When an environmental object o is moved by robot's manipulator the absolute anchor α of this object is marked as obsolete, i.e. the anchor is deleted. All relative anchors in which α is involved are deleted since they are no longer valid due to the changed relative position to other objects. When object o is placed down again its new location is re-calculated from the end-effector frame, the object is re-anchored and the anchor α is updated. Afterwards new relative anchors to other grounded objects can be created using the new subsymbolic information.

2) Object movement unseen by the robot: When an object is moved not registered by the robot, e.g. by a person in the room, the location of the object assumed by the robot is wrong. This leads to an error during task execution, e.g. grasping or manipulation, or to a user abort interrupting the execution since an error is predictable. In such a case task execution is aborted and this object has to be re-detected first before the execution can be continued or started again. This case happens normally only for 'small' object like a bottle, a glass or a book, but not for big environmental objects like a refrigerator or a cupboard.

In order to decide whether a relative anchor has to be deleted the robot has to be aware of the environment and also the system itself. Therefore a context-aware module was designed which collects possible information like

- *Task planning and execution information*, e.g. when which task is started and whether it is successfully executed or maybe aborted.
- *Acquisition information*, e.g. when which object was acquired or when an acquisition failed.
- *Information from system*, e.g. when the robot is moved.

All this information are evaluated related to urgency and significance. Based on reasoning the system decides whether it is necessary to react and how. When for example an object was acquired this information is handed to the object anchoring module, an absolute anchor is grounded and possible relative anchors created. When the task execution is aborted by the user immediately all processes on the system have to be stopped. If the robot changes its position this is registered and the system knows that objects involved in the next task execution have to be re-acuqired first since the last subsymbolic data are no longer valid.

C. Advantages

One advantage of using relative anchors additionally to absolute anchors is that as already mentioned the often time-consuming and error-prone machine vision algorithms can be reduced significantly. But furthermore the accuracy of the objects can be increased when relative anchors are used instead of a new direct acquisition by the robot. When the robot would acquire all objects again and again when they are needed the acquisition is influenced by the inaccuracies given by the used algorithm and the variance of the random variable used for modelling.

When a relative anchor ρ is created between two objects o_1 and o_2 this relation has inaccuracies due to precision of image processing. When this relation is used later to update the subsymbolic data, e.g. of object o_2, the relation describes the distance not exactly. But when the corresponding subsymbolic data is used by the robot, e.g. to grasp a book from the book cart (object o_1) and place it down on the book holder (object o_2), inaccuracies can be detected and compensated. This can be done for example using an eye-in-hand camera mounted on the end-effector in order to monitor the grasping procedure and to adjust the movement of the end-effector. The relation ρ can be updated by the system itself and therefore the accuracy is increased, i.e. the variance is reduced. Using this strategy the acquisition error of object o_2 can be totally compensated by the robot. There is only an influence of the inaccuracies coming from the acquisition of object o_1, since object's o_2 is calculated depending on the previous acquired position of object o_1. Therefore the accuracy of object o_2 increases the more it is grounded using the relative anchor ρ, whereas the accuracy of object o_1 always depends on the constant variance σ_1^2.

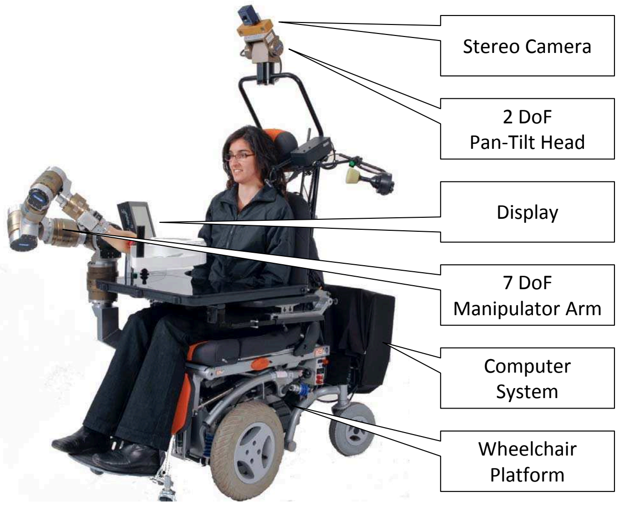

Fig. 3. The care-providing robot FRIEND (Photo: Frank Pusch/IAT).

When two objects are in relation to each other it does not matter which one of them is re-acquired and which one updated using the relative anchor. Based on the actual accuracies the robot can choose the object which can be detected with the highest accuracy. Then all others objects in relation to this one can be updated using the known relative anchors. And even when maybe the first trial to detect an object o_1 with anchor α_{t_1} fails due to unforeseen conditions another object o_2 with anchor α_{t_2} (the one with the secondary highest probability) which is in relation to α_{t_1} can be chosen to be detected first and then o_1's position can be updated using this relative anchor. Therefore unsuccessful detections do not result in a complete abortion of the execution. The system can use a related object which can be detected first before the other one can be updated using the relative anchor.

III. Application and Evaluation

The suggested extended object anchoring concept using relative anchors was implemented and tested within the

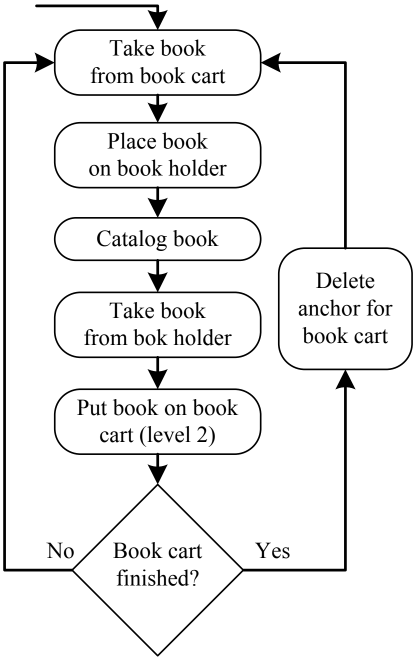

Fig. 4. State chart with overview about the book cataloguing procedure in the FRIEND library scenario.

software framework MASSiVE (*Multi-layer Architecture for Semi-autonomous Service robots with Verified task Execution*, [12]), tailored to the requirements of semi-autonomous and distributed systems, like the care-providing robot FRIEND (*Functional Robot arm with user-frIENdly interface for Disabled people*, [13], [14]) displayed in Figure 3. Due to the two-layered world model used in the based software framework MASSiVE the principle of object anchoring is ideally applicable here and was implemented including relative anchors.

As testing environment a library scenario is used in which a disabled person should work with FRIEND. The whole task of retrospective book cataloguing is split as show in Figure 4 in five sub-scenarios. In order to keep the used task planning, which is based on process-structures [15], [16], consistent and formally correct in each sub-scenario all involved objects have to be acquired again providing the correct data. For task execution a list of elementary operations (skills) is created which transforms the current state of the system into the desired target state and which can be executed autonomously by the system. These elementary operations consists of, e.g. machine vision algorithms (MV) to acquire objects [17], manipulative algorithms (MP) for trajectory planning or object grasping [18], [19] and user interactions (UI) [20]. The involved physical objects are similar as in the example in Section II a book cart (Bc), a book holder (Bh) and a series of books ($Bk.i$, $i = 1, ..., n$). A rough overview of the skills executed to perform the cataloguing of one book is following.

Task 1: Take book from book cart

1. MV::AcquireObjectBySCam(Bh): acquire book holder using stereo vision to have it as an obstacle for the trajectory planning of the manipulator.
2. MV::AcquireObjectBySCam(Bc): acquire book cart.
3. MP::CoarseApproachToObject(Bc): coarse approach to object using a 3D-model of the surrounding environment created from perceived information.
4. MP::FineApproachToObjectByHandCamera(Bc): fine approach to book on book cart using disparity and visual servoing by means of a hand camera. Book is grasped from the first shelf of the book cart.
5. MP::DepartWithObject(Bc, $Bk.i$): depart grasped book from book cart and move to free position in space.

User moves FRIEND in front of book holder.

Task 2: Place book on book holder

1. MV::AcquireObjectBySCam(Bh): acquire book holder using stereo vision.
2. MV::AcquireObjectBySCam(Bc): acquire book cart to have it as an obstacle.
3. MP::PlaceObjectOnPlatformByFTSControl($Bk.i$, Bh): place book on book holder using force-torque control.
4. MP::DepartFromObjectOnPlatform($Bk.i$, Bh): depart robot arm from book.

Task 3: Catalog book

1. MV::AcquireObjectBySCam(Bh): acquire book holder using stereo vision.
2. UI::OpenBook($Bk.i$): open book using an automated book holder.
3. UI::CatalogBook($Bk.i$): user catalogs the book and can if necessary thumb through the book.
4. UI::CloseBook($Bk.i$): close book.

Task 4: Take book from book holder

1. MV::AcquireObjectBySCam(Bh): acquire book holder using stereo vision.
2. MP::CoarseApproachToObject(Bh): coarse approach to book holder.
3. MP::FineApproachToObject(Bh): grasp book from the book holder and move manipulator to free position.
4. MP::DepartWithObject($Bk.i$, Bh): depart gripped book from book holder and move to free position.

User moves FRIEND in front of book cart.

Task 5: Put book on book cart

1. MV::AcquireObjectBySCam(Bc): acquire book cart using stereo vision.
2. MV::AcquireObjectBySCam(Bh): acquire book holder to have it as obstacle.
3. MP::PlaceObjectOnPlatformByFTSControl($Bk.i$, Bc): place book on second shelf of book cart.
4. MP::DepartFromObjectOnPlatform($Bk.i$, Bc): depart robot arm from book.

The amount of necessary machine vision operations is displayed in Table I. Hence without any object anchoring

	For first book	For all following books	Total for 20 books
Without OA	8	8	160
OA without CA	6	4	42
OA with CA	4	2	22

TABLE I
NECESSARY MACHINE VISION OPERATIONS WITHOUT USING OBJECT ANCHORING (OA), USING OA WITHOUT RELATIVE ANCHORS (CA) AND OA WITH CA.

concept for each book cataloguing process eight machine vision operations are necessary (1.1, 1.2, 2.1, 2.2, 3.1, 4.1, 5.1, 5.2). For a book cart with, e.g. 20 books at total 140 machine vision operations are necessary. When the object anchoring concept by Coradeschi is used this can be reduced significantly to 42. Here re-acquisition is only necessary at the beginning and after each wheelchair movement since the anchors then are deleted (for first book: 1.1, 1.2, 2.1, 2.2, 5.1, 5.2, and for all further books only: 2.1, 2.2, 5.1, 5.2). When the presented concept of relative anchors is used this amount can be additionally nearly divided by two. After the first recognition of both objects, the book holder (Bh, in 1.1) and the book cart (Bc, in 1.2), a relative anchor is created and thereafter only one of these two objects have to be acquired again when the robot moves (e.g. 2.1, 5.1).

When the book cart is finished and all books have been catalogued it is replaced by a new one. Since the relative anchor between book cart and book holder is not longer valid it is deleted. In the next iteration first both, book cart and book holder, have to be acquired again (see Figure 4) since the book cart was moved, and a new relative anchor between both objects can be created.

This shows that the presented extension of relative anchors used within the task planning and execution for service robotic systems can reduce the number of necessary machine vision operations significantly. And it is clear that this benefit increases the more object are involved in the task and the more complex the environment is.

To analyse the development of the accuracies, i.e. the development of the variance of the Gaussian distribution, a software simulation was used which models the two involved objects mathematically by Gaussian distributed random variables: O_1 for book cart and O_2 for book holder. Similar to the example in Figure 2 it is reduced to a two-dimensional case, where each object has only three degrees of freedom, namely x- and y-coordinate (in meters) and rotation (in degrees). The parameters for the distributions were chosen depending on distance to the robot, technique used for detection and real test results how reliable the object can be detected in the library environment:

$$\mu_{O_1} = [0.6, -0.8, 90], \qquad \sigma^2_{O_1} = [0.000225, 0.0004, 5]$$
$$\mu_{O_2} = [1.6, -0.2, 5], \qquad \sigma^2_{O_2} = [0.001, 0.0008, 5]$$

The mean values are the ideal position of the objects in the environment when no inaccuracies exist. The book cart is

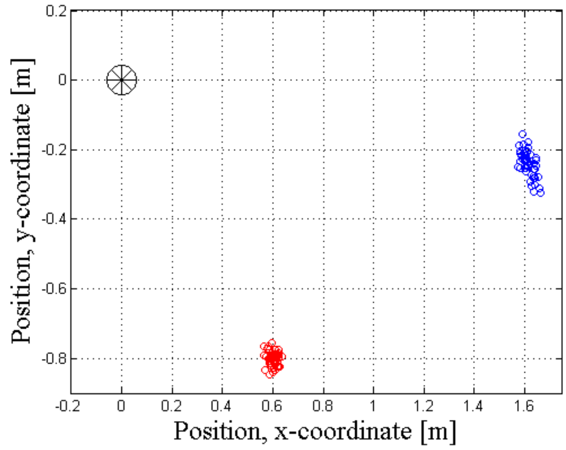

Fig. 5. Library environment with robot (position marked with star) and the randomly generated positions for the book cart (red) and the ones for book holder (blue) calculated using the relativ anchor.

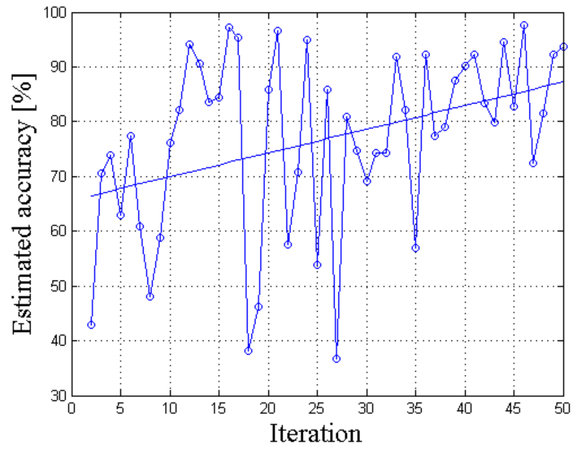

Fig. 6. Development of the accuracy for the book cart.

always detected directly by the robot, i.e. in each step based on the distribution randomly a possible location is generated. The book holder is acquired only in the first step directly by the vision system. Thereafter the location of the book holder is estimated using a relative anchor which is created in the first iteration. The estimated positions of the both objects are graphically displayed in Figure 5. Depending on the size of the variance the spreading varies for each object and each dimension.

The accuracy for the book cart remains always the same since it is reacquired and grounded in each iteration based on the given mean and variance. In Figure 6 the estimated accuracy of the book holder is displayed. Since the location of the book holder can be improved during manipulation in each step the accuracy is increased with the number of iterations, but it is also clear that a partially influence from the inaccuracies of the book cart detection remains, especially rotation inaccuracies when the distance between both objects is big.

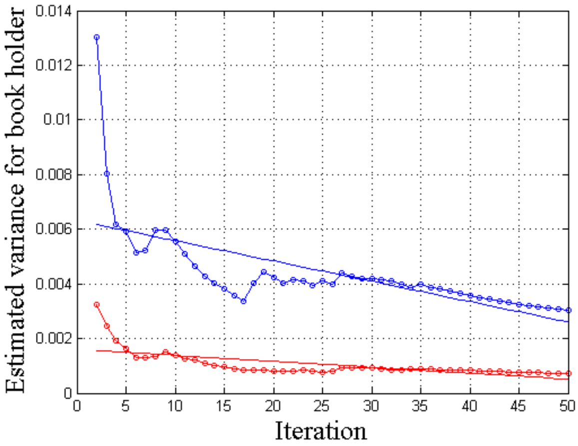

Fig. 7. Development of the variance for the object book holder: x- coordinate (red) and y-coordinate (blue).

That means that the variance of the distribution of O_2 becomes smaller with the number of iterations. The development of the variance is displayed in Figure 7, seperately for the x- and the y-coordinate, which was estimated using a χ^2-distribution.

IV. Conclusion

In this paper the use of relative anchors for object anchoring was presented as well as an extended object anchoring concept which takes into account inaccuracies with which objects are grounded. Especially for mobile systems this concept can reduce the number of necessary operations for object acquisition significantly as shown through the evaluation and the accuracy given by the machine vision algorithms can be skipped partially for the objects which are updated using these relative anchors. Since these operations are often time-consuming and error-prone the presented concept yields a benefit and the system is able to react in an intelligent manner when objects have to be acquired again and can use these relative anchors to learn environments itself.

This can be done best when as much as possible information can be collected and used for evaluation and reasoning. That is the system has to monitor the environment at the beginning of task planning and also permanently during task execution. In the future this module which is under development will be added to this object anchoring concept to increase the benefit and to create a platform for mobile robots to perform intelligent and reliable task planning and execution.

Acknowledgment

The authors would like to thank the Integration Agency Bremen which is part of Pension Office in Bremen, assigned to the Department of the Bremen Senator for Profession, Women, Health, Youth and Social Affairs, which funds the current project 'ReIntegraRob' and the State and University Library of Bremen with which the project is performed in co-operation.

References

[1] S. Coradeschi and A. Saffiotti, "An introduction to the anchoring problem," *Robotics and Autonomous Systems*, vol. 43, no. 2-3, pp. 85–96, 2003.

[2] O. Prenzel, "Semi-autonomous object anchoring for service-robots," in *B. Lohmann (Ed.), A. Gräser, Methods and Applications in Automation*. Aachen, Germany: Shaker-Verlag, 2005.

[3] A. Bouguerra, L. Karlsson, and A. Safiotti, "Situation assessment for sensor-based recovery planning," in *Proc. of the 17th European Conf. on Artificial Intelligence (ECAI)*, Riva del Garda, Italy, Aug. 2006.

[4] S. Coradeschi and A. Saffiotti, "Anchoring symbols to sensor data: preliminary report," in *Proceedings of the 17th AAAI Conference*, 2000, pp. 129–135.

[5] A. Loufti, S. Coradeschi, and A. Saffiotti, "Maintaining coherent perceptual information using anchoring," in *Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI05)*, Edinburgh, Scotland, 2005.

[6] M. Daoutis, S. Coradeschi, and A. Loufti, "Grounding commonsense knowledge in intelligent systems," *Journal of Ambient Intelligence and Smart Environments I*, pp. 311–321, 2009.

[7] S. Coradeschi and A. Loufti, "A review of past and future trends in perceptual anchoring," in *Frizche, P. (ed.), Tools in Artificial Intelligence, I-Tech Education and Publishing*, Vienna, Austria, 2008, pp. 129–135.

[8] N. Blodow, D. Jain, Z.-C. Marton, and M. Beetz, "Perception and probalistic anchoring for dynamik world state logging," in *Proc. of the 2010 IEEE-RAS Int. Conf. on Humanoid Robots*, Nashvilee, TN, USA, Dec. 2010.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, pp. 91–110, 2004.

[10] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *In ECCV*, 2006, pp. 404–417.

[11] "ARToolKit webpage," http://www.hitl.washington.edu/artoolkit/, [Online; accessed 04-January-2012].

[12] C. Martens, O. Prenzel, J. Feuser, and A. Gräser, "MASSiVE: Multilayer architecture for semi-autonomous service-robot with verified task execution," in *Proc. of the 10th Int. Conf. on Optimization of Electrical and Electronic Equipments (OPTIM)*. Brasov, Romania: Transilvania University Press, 2006.

[13] "FRIEND webpage," http://www.friend4you.eu/, [Online; accessed 04-January-2012].

[14] S. M. Grigorescu, T. Lueth, C. Fragkopoulos, M. Cyriacks, and A. Gräser, "A BCI controlled robotic assistant for quadriplegic people in domestic and professional life," *ROBOTICA, Cambridge University Press, UK*, 2011.

[15] O. Prenzel, A. Boit, and H. Kampe, "Ergonomic programming of service robot behavior with function block networks," in *A. Gräser (Ed.), Methods and Applications in Automation*. Shaker-Verlag, 2008.

[16] O. Prenzel, "Process Model for the Development of Semi-Autonomous Service Robots," Ph.D. dissertation, Bremen University, Institute of Automation, Bremen, Germany, 2009.

[17] S. M. Grigorescu, D. Ristic-Durrant, and A. Gräser, "ROVIS: RObust machine VIsion for service robotic system FRIEND," in *Proc. of the 2009 Int. Conf. on Intelligent RObots and Systems*, St. Louis, USA, Oct. 2009.

[18] C. Fragkopoulos and A. Gräser, "A RRT based Path Planning Algorithm for Rehabilitation Robots," in *Proc. of the 2012 Int. Conf. ISR/ROBOTIK*, Munich, Germany, Jun. 2010.

[19] C. Fragkopoulos and A. Gräser., "Sampling based path planning for high DoF manipulators without goal configuration," *IFAC World Congress*, 2011.

[20] O. Prenzel, C. Martens, M. Cyriacks, C. Wang, and A. Gräser, "System controlled user interaction within the service robotic control architecture MASSiVE," *Robotica, Special Issue*, vol. 25, no. 2, Mar. 2007.