

# Using Range Imaging Sensors with Color Imaging Sensors in Cognitive Robot Companions: A New and Simple Calibration Technique Based on Particle Swarm Optimization

*Jens Kubacki & Kai Pfeiffer*

*Fraunhofer Institute for Manufacturing Engineering and Automation (IPA),  
Nobelstrasse 12, 70569 Stuttgart, GERMANY*

## **Abstract**

A cognitive robot companion needs sensors in order to enhance its perceptual capabilities. Using different types of sensors in different mechanical settings there is a need to find suitable calibration models. Calibration can be performed between system components or relative to some global reference frame. In this paper we describe a method that can lead to useful calibration models in a generic manner. A particle swarm optimization algorithm is used to optimize a parameter set and suggest methods to reduce or to refine the calibration model iteratively. Relevant equations do not have to be reformulated as a system of equations and there is no principal restriction on the class of equations. The proposed method is applied to calibrate a new bi-modal sensor head in two different ways: directly between two sensors and to real-world coordinates.

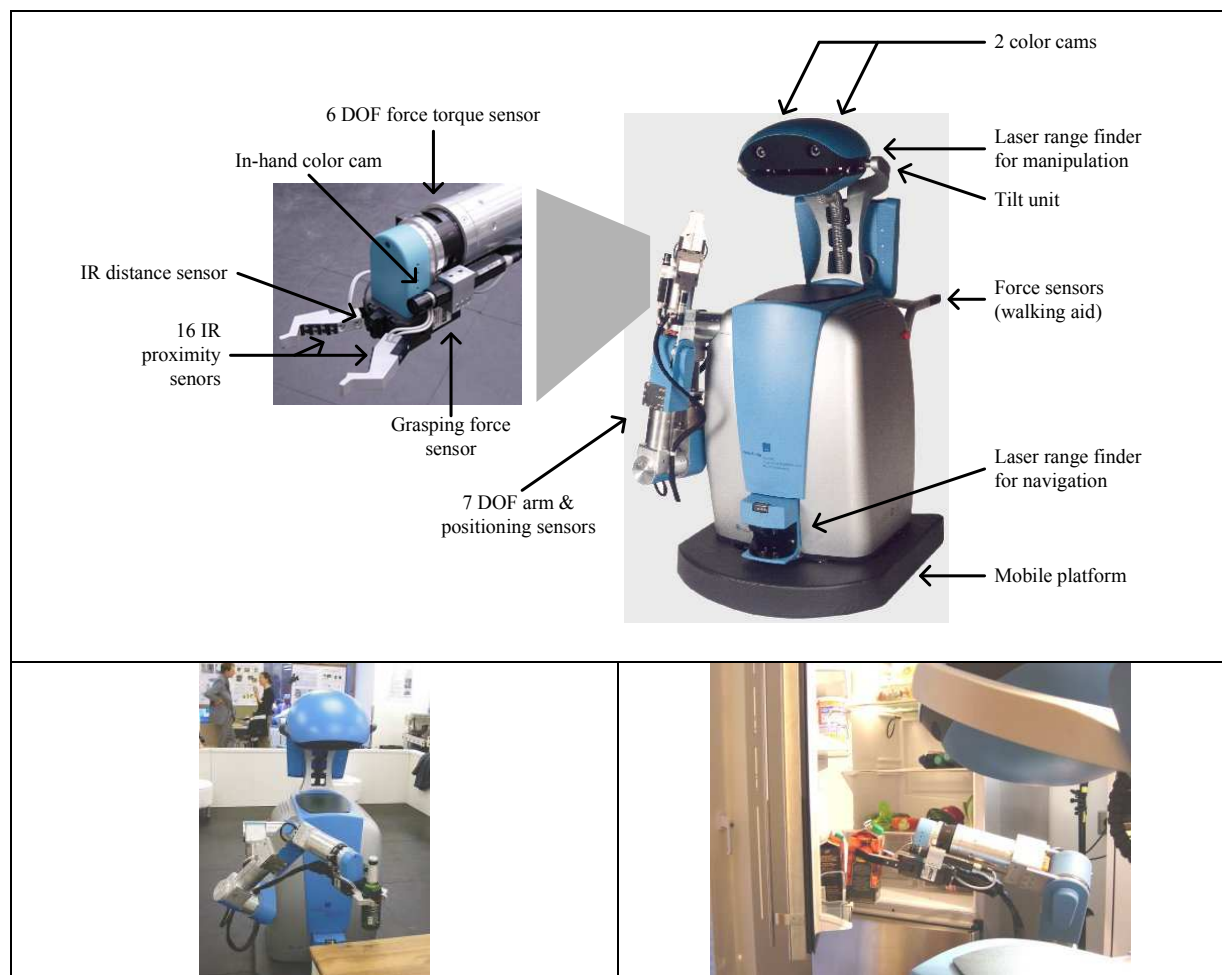
## **Keywords**

Calibration, particle swarm optimization, robot assistant, cognitive robot companion, perception, range imaging, automatic model selection.

## **1. INTRODUCTION**

As a successor of the idea of robot assistants as published in [1] in the European integrated project COGNIRON we develop methods and technologies for the construction of so-called cognitive robot companions. These artificial creatures are expected to evolve and improve their capacities on-line i.e. during operation and in close interaction with humans [2]. In order to be able to enhance their perceptual, manipulation and reasoning capabilities the robots need to acquire representations about tasks, objects, locations and humans. One pre-requisite that enables the development of robust recognition and task execution is the physical coupling of the robot with the environment in terms of sensors and actuators. See figure 1 (top) for an example of a modern sensor-actuator coupling with the environment. Focusing on the sensor side, new technologies can serve as enablers for the acquisition of representations and hence for the development of robust recognition and task execution. Therefore, new developments in the field of range imaging sensors are highly relevant for the design of cognitive robot companions. To use modern range imaging sensors and other sensors in these robots it is necessary to calibrate the sensors either directly to each other or relative to some common reference. Depending on the type of device, peripheral components, and the mechanical setting there is a special set of equations needed for calibration. We call this set of equations also *calibration model*. In order to find such model it is useful to develop methods that can lead to this set systematically, either manually or possibly with the aid of a computer.

After describing the state-of-the-art in chapter 2 we introduce a new calibration technique in chapter 3. A particle swarm algorithm is used to optimize a general parameter set. Relevant equations do not have to be reformulated into an equation system and there is principally no restriction on the class of relevant equations. In chapter 4 the implementation of the method is sketched and in chapter 5 experiments are conducted on two different calibrations related to a bi-modal sensor head. In chapter 5 also necessary equations for a specific set-up are derived. Chapter 6 contains results and future work and chapter 7 conclusions.



**Figure 1:** The Care-O-bot II (see [3]) as an example for a robot companion that needs to execute fetch-and-carry tasks. Top: The physical sensor-actuator coupling of the robot with the environment. Bottom left: The robot picks up a bottle of beer. Bottom right: the robot takes apple juice out of the fridge.

## 2. STATE-OF-THE-ART

### 2.1. Modern Range Sensors

One standard sensor device used in robotics is the color imaging sensor based on either CCD technology or CMOS technology. In addition to color information there are mainly three methods to produce range i.e. 'depth' images in robotics: (a) Using stereo-vision systems, (b) panning or rotating a 2d laser range finder, and (c) using 3d laser range finders.

Stereo-vision systems require for background structure in order to solve the correspondence problem. However, stereo-vision systems can work with high resolutions since they build on mature color imaging technology that has been driven by commercial applications.

2d laser range finders are standard components in the field of mobile robot navigation. They do not deliver a range image in one frame, but one 'line' of it reflecting the distances of a circular area in front of the sensor in one plane. If these sensors are panned or rotated it is possible to reconstruct a range image. Here a challenge is to synchronize the acquisition of single lines with the motion of the sensor. A drawback is the low effective frame rate as a result of this panning or rotation process. The Care-O-bot II shown in figure 1 utilizes a laser range finder which is integrated in the robot's head. Through panning of the head a 3d scene can be reconstructed. This scene is used for object localization and to plan and execute a collision-free trajectory for grasping.

3d laser range finders are also available on the market [4]. A drawback in mobile applications is the relative large volume of such sensors.

A promising new sensor technology is based on the time-of-flight measurement of light that returns on each photo element of an imaging chip after submission [5]. These sensors are very compact and deliver 3d range images in one frame. Already this technology has been shown to be suited for mobile robot navigation [6]. There are already a few sensors commercially available and others are in a prototypical state. See figure 2 for a rough overview.

Product							
Manufacturer	CSEM	PMD-Tec	PMD-Tec	Canesta	3DVSystems	3DVSytsems	Matsushita Electric Works
Reference	[7]	[8]	[9]	[10]	[11]	[12]	[13]
Pixel resolution	Up to 160x124	64x16	160x120	DP203, DP205, DP208: 64x64	752x582 (PAL) or 768x494 (NTSC)	510x492	128x123
Accuracy of one measurement	Down to 5 mm	> 6 mm	> 6 mm	> 6 mm	> 5 mm	> 3 – 5 mm	Not found
Frame Rate	Up to 30	Up to 50	Up to 10	Up to 15	Not found	Up to 60	Up to 15
Field of View	Range: 7.5m Angle: +/- 30° (diagonal)	Range: 7.5m Angle (HxV): 34° x 12°	Range: 7.5m Angle: 40°	Range: 5.0m Angle: 30°, 55°, 80°	Range: 3.5m Angle: 30°	Range: 2.5m Angle (HxV): 45° x 35°	Range: 7.5m Angle (HxV): 60° x 45°
Connection	USB 2.0	IEEE 1394a and Ethernet (IEEE 802.3u)	IEEE 1394a and Ethernet (IEEE 802.3u) and RS 232	USB 1.1	Not found	FireWire and RS 232	USB 2.0 and Parallel

**Figure 2:** Modern range imaging sensors that measure the time till re-arrival of a generated light pulse. These sensors are a considerable alternative to stereo-vision or mechanically animated laser range finders in robotics. Note that this is only a rough overview. The data printed in the table is drawn from the references given in the third row. Please refer to the manufacturers for details on a specific product.

We identify the following application areas in the context of cognitive robot companions relevant to modern range imaging sensors.

- Large scale geometrical modeling of space for navigation purposes (e.g. simultaneous localization and mapping)
- Reactive navigation competences: e.g. obstacle avoidance and person following
- Local scene reconstruction for object localization and collision-free grasping

- Geometrical human modeling for gesture and activity recognition
- Enhancement of color-based object recognition systems through range based segmentation

Note that this list is not meant to be complete. There may be more application areas in the context of robot companions for modern range imaging sensors. These applications we have extrapolated from our current activities in the field of constructing robot companions.

## 2.2. Calibration Techniques in Computer Vision

In computer vision the problem of calibration is usually split into the estimation of the *intrinsic* and the *extrinsic parameters* by solving a system of equations. The intrinsic parameters can include:

- Projection parameters of the pinhole model.
- Rotation and shift between the imaging chip and lens.
- Scaling parameter related to the imaging chip.
- Radial distortion of a real lens.

The extrinsic parameters include the 3d rotation and shift of the camera relative to some chosen world coordinate system.

To solve the calibration problem usually a system of equation is solved. A system of linear equations can be solved using standard methods. Nonlinear systems can be computed using the Gauss-Newton or Levenberg-Marquardt algorithms [16]. To cast complex models into a solvable system there is manual analysis work needed.

## 3. NEW CALIBRATION TECHNIQUE

### 3.1. Motivation

Developing robot companions we are often in a situation where we have to implement new sensors and/or actuators of various types in order to use them concurrently for recognition or task execution. Depending on the type of device and the mechanical setting of it relative to other components there is a special set of equations that is needed for calibration. Some devices need more complex models others may be modeled sufficiently with a reduced model (e.g. a camera equipped with some optics that does not lead to radial distortion). If e.g. different cameras are used in combination it is not clear which parameters are most relevant *between* them (e.g. equal lenses compensate for radial distortion when calibrating from one sensor into the other). Another issue is that for some tasks computationally cheap and thus fast methods are needed rather than very accurate ones. Furthermore, if the mechanical setting changes it should be possible to quickly adapt the calibration equations by either finding a new set of parameters or even by removing or adding terms. For these reasons, there is the aim to develop a method that allows finding a sufficient set of equations with minimal manual analysis effort possibly with the aid of a computer. Sufficient here means with acceptable (depending on the task) accuracy and generally low computational cost since the robot's on-board computers have limited capacity. Requirements for such a method are:

- Suited for learning from reference samples that can be produced by a simple set-up and procedure. In the best case autonomously by the robot itself.
- Independency of the class of equations that describe the problem.
- Simple combination of many equations possible.

- Easy editing/change of the model to be tested or even automatic selection of the equations needed.

We identify two types of calibration in the context of robot companions where such a method can help finding a sufficient calibration model:

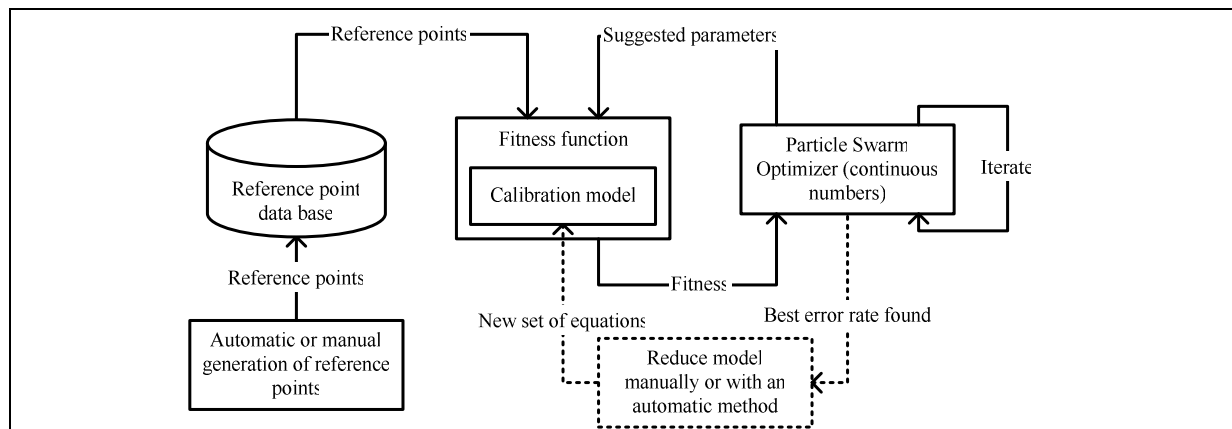
- Devices are calibrated pair-wise directly to each other i.e. calibration is performed from the readings of one device into the readings of another. We call this *direct calibration*.
- All devices can deliver their measurements relative a common reference frame (either attached to the robot or to a world-coordinate frame attached to some point in the environment). In this paper this is referred to as *real-world calibration*.

The first allows for fast search algorithms the use e.g. two sensors in conjunction in the same recognition algorithm. The second is suited for comparisons, fusions and communications of measurement results in a common format.

### 3.2. Approach

In order to find a simple-to-use method that supports the search for a set of equations that suffices the requirements stated in the last chapter the following method is proposed.

1. Generate a list of reference points by either: (a) designing a reference object and specialized detection programs to detect a point attached to this object in different sensors that need to be calibrated or by (b) generating a sample point list manually. Augment the list with real-world coordinates if needed. The reference points can also be generated by the robot autonomously.
2. Implement all available transformation equations needed to fully describe the problem as a chain of programming steps in their explicit format. The equations are given in the manuals of the sensors or in the literature of the relevant scientific field. Some equations (e.g. coordinate transformation between two systems) are described in general mathematics, robotics or computer vision literature.
3. Cast the problem into an optimization problem by utilizing a stochastic search optimizer as parameter estimator that iteratively suggests and improves a set of parameters. For improvement a fitness function needs to be defined that measures an error criterion of the specific parameter set on the sample list. In this paper we use a particle swarm optimization technique [14]. This technique does not depend on the class of equations i.e. it assumes no special properties of the functions involved.
4. Reduce or refine the set of equations (a) manually or (b) using some automatic scheme and repeat step 2 and step 3 with the new calibration model until a sufficient set of equations is found.



**Figure 3:** Sketch of the proposed method to derive sufficient calibration models. The model to be tested can be implemented in a simple explicit representation within a fitness function that is called by a particle swarm optimizer. The fitness function has access to the reference points to calculate an error measurement (fitness). The optimizer iteratively converges to a parameters set that gives a low error rate. The model can be refined or reduced manually or by some automatic method.

The particle swarm algorithm is used in this paper as a parameter estimator. The algorithm is described in detail in [14]. An advantage of this algorithm in comparison with other stochastic search algorithms, such as genetic algorithms is that it is simple to implement and that it is easy to use as an optimizer for problems in continuous numbers. In addition it can cope very well with multimodal maxima in the search space. Using this algorithm for our calibration procedure has the following advantages:

- It is still possible to get good results if parameters are not independent or even redundant.
- Not restricted to a special class of equations (e.g. the model equations can contain discontinuities)

In figure 3 is shown how we use the algorithm as parameter estimator with a data base of reference points in our current implementation.

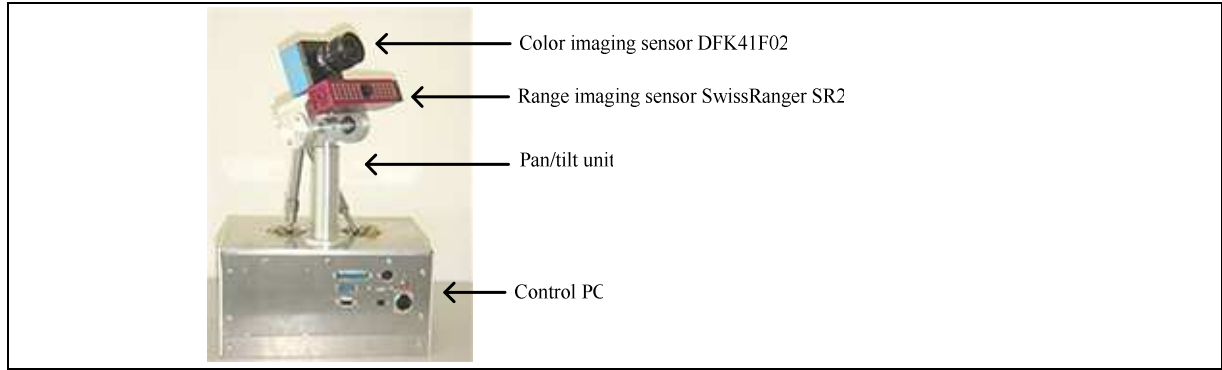
### 3.3. Example: Calibration of a Bi-Modal Sensor Head

As an example application for the proposed general approach to finding sufficient calibration models we chose the calibration of a bi-modal sensor head. The next generation of our robot companion (Care-O-bot III) is planned to incorporate a time-of-flight range imaging sensor [7] mounted next to a CCD color imaging sensor [15]. The basic hardware setting is shown in figure 4.

We apply both concepts to the bi-modal sensor head:

1. *Direct* calibration from the three readings (two image coordinates and range) of the range imaging sensor to the two image coordinates of the color imaging sensor
2. *Real-world* calibration from the three readings (two image coordinates and range) of the range imaging sensor to the x-, y-, and z-components in a chosen sensor reference frame.

The first strategy is aimed to allow for recognition algorithms that work in the range and color image concurrently and the second method allows to communicate recognition results to other parts of the system (e.g. for grasping an object).



**Figure 4:** The experimental set-up of the sensor head includes a range imaging sensor [7], a color imaging sensor [14], a pan/tilt unit, and a control PC.

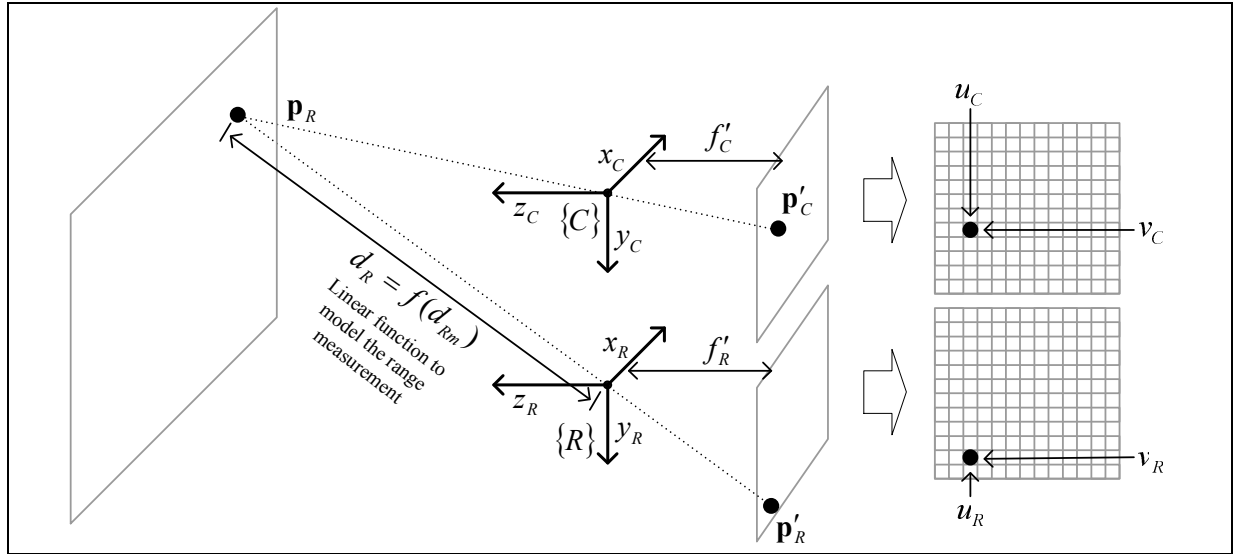
We describe the two types of calibrations formally as:

Direct Calibration:	$(u_R, v_R, d_R) \mapsto (u_C, v_C)$	(1)
Real-World Calibration:	$(u_R, v_R, d_R) \mapsto (x_R, y_R, z_R)$	(2)

where  $u_R$  and  $v_R$  are the discrete coordinates on the imaging chip of the range imaging sensor,  $d_R = f(d_{Rm})$  is the distance to the range imaging sensor that is a function of the distance reading  $d_{Rm}$  delivered by the sensor, and  $u_C$  and  $v_C$  are the discrete coordinates on the imaging chip of the color imaging sensor. Furthermore, the linear relationship termed *distance function* is adopted from [6]:

Distance function:	$d_R = c_{R1}(d_{Rm} + c_{R0})$	(3)
--------------------	---------------------------------	-----

We will now derive the equations necessary for the direct calibration and for the real-world calibration of the sensor head described in the chapter above. See figure 5 for the notations that are borrowed from [16] and [17]. As a reference frame we chose the frame assigned to the range imaging sensor  $\{R\}$ . The reference frame of the color imaging sensor is described with  $\{C\}$ . The point  $\mathbf{p}_R = (x_R, y_R, z_R)$  is projected onto the image planes of the two sensors.  $\mathbf{p}'_R = (x'_R, y'_R, z'_R)$  and  $\mathbf{p}'_C = (x'_C, y'_C, z'_C)$  describe the projections. The z-components  $z'_R$  and  $z'_C$  are fixed and equal to the corresponding focal lengths  $f'_R$  and  $f'_C$  of each sensor device. The x- and y-components  $(x'_R, y'_R, x'_C, y'_C)$  describe the 2d coordinates in the image planes of the sensors. On the left side in figure 5  $u_R, v_R, u_C,$  and  $v_C$  denote the discrete sensor readings resulting from the point's projections.



**Figure 5** Notation for the transformation equations of the sensor head used in this paper

Now let us derive the equations needed for the direct calibration and the real-world calibration. First we introduce the perspective projection (as described in e.g. [16]) of the range imaging sensor using the pin-hole camera model.

Projection range imaging sensor:	$x_R = \frac{x'_R}{\lambda_R}, y_R = \frac{y'_R}{\lambda_R} \text{ and } z_R = \frac{f'_R}{\lambda_R} \quad (4), (5) \text{ and } (6)$
----------------------------------	--

Here,  $\lambda_R$  is a scaling parameter which can be calculated for the range imaging sensor by:

Scale parameter range imaging sensor:	$\lambda_R = \frac{\sqrt{(f'_R)^2 + (x'_R)^2 + (y'_R)^2}}{d_R} \quad (7)$
---------------------------------------	---

Eventually, we assume scaling and shifting between the lens and the sensor chip as described in [15].

Scaling and shifting range imaging sensor:	$u_R = k_R x'_R + u_{R0} \text{ and } v_R = l_R y'_R + v_{R0} \quad (8) \text{ and } (9)$
--	---

For the direct and real-world calibration we want to derive  $(u_C, v_C)$  and  $(x_R, y_R, z_R)$  on the basis of  $(u_R, v_R, d_R)$ . Thus, we reverse equations (8) and (9):

Scaling and shifting range imaging sensor:	$x'_R = \frac{(u_R - u_{R0})}{k_R} \text{ and } y'_R = \frac{(v_R - v_{R0})}{l_R} \quad (10) \text{ and } (11)$
--	---

The real-world calibration can now be calculated by combining equations (4) to (7), (10), and (11). For the direct calibrations we need to introduce the relationship between the coordinate systems of frame  $\{R\}$  and frame  $\{C\}$ . We do this by introducing a coordinate transformation

Coordinate transformation:	$\mathbf{p}_C = \mathbf{R}_{CR}\mathbf{p}_R + \mathbf{o}_{CR}$	(12)
----------------------------	--	------

Where  $\mathbf{R}_{CR}$  is the rotation matrix and  $\mathbf{o}_{CR}$  the translation vector of  $\{C\}$  relative to  $\{R\}$ .

Now we can derive the projection model of the color imaging sensor:

Projection color imaging sensor:	$x'_C = \frac{x_C}{z_C}$ and $y'_C = \frac{y_C}{z_C}$	(13) and (14)
----------------------------------	---	---------------

Note, again we assume  $f'_R = 1$ . In coherency with the range imaging sensor we model scaling and shifting of the imaging chip relative to the lens.

Scaling and shifting color imaging sensor:	$u_C = k_C x'_C + u_{C0}$ and $v_C = l_C y'_C + v_{C0}$	(15) and (16)
--	---	---------------

We can now derive the full model for the direct transformation by combining equations (4) to (7) and (10) to (16).

## 4. IMPLEMENTATION

### 4.1. Detection of the Reference Object in Both Sensors

As a reference object we chose a red ball and a point associated to the centre of the cyclic region that the projection of the ball yields in the image plane of the corresponding sensor. This is rotationally invariant and can be detected easily in both sensors. The detection programs first generate binarized images and then searches for circular 'blobs' using a simple sub-procedure. In the color image only red areas are binarized. In the range image consecutive depth intervals are evaluated separately. The first (starting from the sensor and moving into the scene) interval that contains a circular blob is used for further processing. Note that this method is subject to errors since the detected points in the sensors are not located precisely on the same position.

### 4.2. Using Particle Swarm Optimization for Parameter Estimation and Model Search

The particle swarm optimization (PSO) algorithm is based on a directed random search in the parameter space. Please refer to [14] for a detailed description. A fixed-size group of so-called individuals searches for maxima. Each individual has access to a pre-defined set of neighbors in that it can see their fitness values and stores its own best-so-far fitness value. Using these inputs a velocity vector is generated and added to the current position. In high-dimensional parameter spaces the group behaves like a swarm that first moves to and then oscillates around potential maxima. In our calibration method the PSO repeatedly 'suggests' parameter vectors and gets a fitness value as feedback. The fitness value is in this paper defined as a negative quadratic error average between the estimated values using the suggested parameters and the real values. For our parameter estimation problem we enhanced the algorithm with the following features:

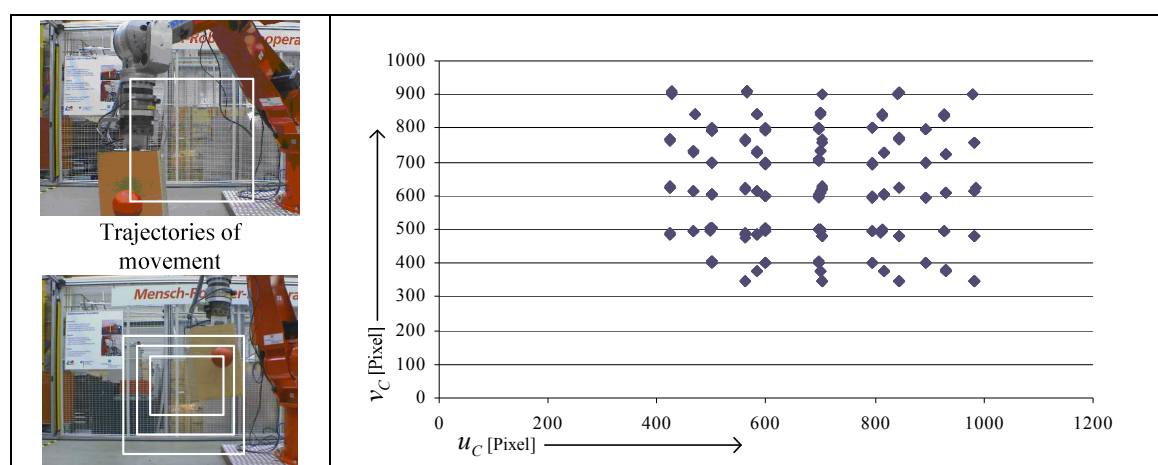
- Store the overall best-so-far individual to not lose a good solution that has been found already.
- Overwrite the initial values of the individuals with a pre-defined vector if there is an initial assumption of the parameters available.
- Initialize with a high number of individuals and take only the first few bests for further processing. This is to get better starting values as in the original version of the algorithm.
- Reduce the maximal velocity if there is no improvement of the overall best-so-far individual after a defined number of iterations. This is to find refined solutions if the algorithm oscillates around a potential maximum.
- An *incremental mode* that first optimizes the first parameter and then successively releases the remaining parameters. The starting parameters are set to neutral values.

The incremental mode is used to support to find simple calibration models automatically. First the optimizer finds a good solution to the problem by only estimating the first parameter of the first equation. The remaining parameters are set to a neutral value (i.e. to one in product terms and to zero in sum terms and so on). After a defined number of iterations the next parameter is released and the optimizer now varies both parameters. Then the next parameter is released and so on. While the equations are important to the quality of the model the user can observe a clear reduction of the error. If the error does not decrease anymore while new parameters are released one can assume that further parameters are not relevant for the problem that is defined ,by the reference points.

## 5. EXPERIMENTS

### 5.1. Real-World Calibration of the Range Imaging Sensor

The real-world calibration of the sensor head of the form  $(u_R, v_R, d_R) \mapsto (x_R, y_R, z_R)$  was conducted using our approach described in chapter 3. First, an industrial robot was moved through the scene of the range imaging sensor. This is shown in figure 6.



**Figure 6** Set-up and results of the generation of reference points using the range imaging sensor. The robot was moved along three consecutive rectangles (left). Right: The resulting points in the image of the range imaging sensor.

We set up a data based with 73 reference points:

$$\left\{ (u_{R1}, v_{R1}, d_{R1}, x_{R1}, y_{R1}, z_{R1}), \dots, (u_{Rn}, v_{Rn}, d_{Rn}, x_{Rn}, y_{Rn}, z_{Rn}) \right\} \quad (17)$$

where  $n = 73$  is the number of reference points.

The real-world calibration can be calculated by combining equations (4) to (7), (10), and (11).

The particle swarm algorithm was used to estimate four parameters:  $u_{R0}$ ,  $l_R$ ,  $v_{R0}$  and  $f'_R$ .

Since the scaling parameters and the focal length are not independent we set  $k_R = 1$ .

In order to use the particle swarm algorithm as fitness function was defined:

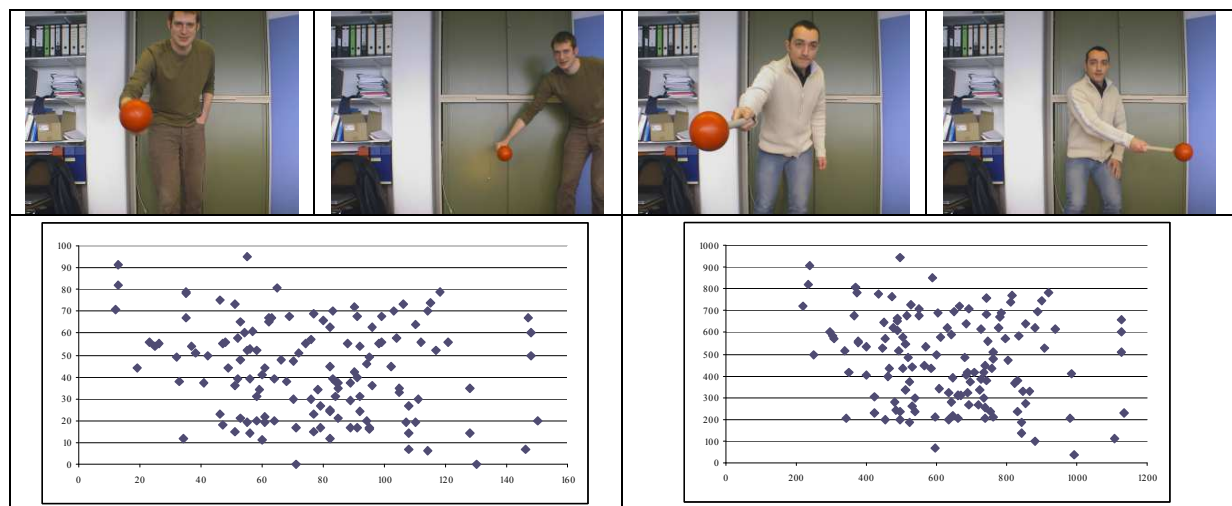
$$fit = -\frac{1}{3n} * \sum_{i=1}^{i=n} (x_R - \hat{x}_R)^2 + (y_R - \hat{y}_R)^2 + (z_R - \hat{z}_R)^2 = -err^2 \quad (18)$$

where  $(\hat{x}_R, \hat{y}_R, \hat{z}_R)$  are the estimated values using the suggested parameter set.

The resulting error on the reference data set was  $err = 44.52$  mm and is therefore higher than the error reported in [6]. It is not expected that the PSO algorithms missed a maximum point. Rather, there are errors expected in the detection algorithm for the reference object.

## 5.2. Direct Calibration of the Sensor Head

The direct calibration of the sensor head of the form  $(u_R, v_R, d_R) \mapsto (u_C, v_C)$  was carried out in the same fashion than the real-world calibration. However, here we not only searched for the parameters of a model. We also tried to derive a simplified model for the direct calibration. The reference points for the direct calibration do not need to be annotated with real-world coordinates. This is an advantage that can significantly simplify the generation of reference points. To demonstrate this we set up a data base through moving the reference object manually through the scene (see figure 7)



**Figure 7** Set-up for the direct calibration. Top row: example images from the color sensor. Bottom row: reference points detected in the range imaging sensor (left) in pixel coordinates and reference points detected in the color imaging sensor (right) in pixel coordinates.

We set up a data base with 134 reference points of the form:

$$\{(u_{R1}, v_{R1}, d_{R1}, u_{C1}, v_{C1}), \dots, (u_{Rn}, v_{Rn}, d_{Rn}, u_{Cn}, v_{Cn})\} \quad (19)$$

where  $n$  is the number of reference points.

As fitness function was defined:

$$fit = -\frac{1}{2n} * \sum_{i=1}^{i=n} (u_R - \hat{u}_R)^2 + (v_R - \hat{v}_R)^2 = -err^2 \quad (20)$$

where  $n = 134$  is the number of reference points and  $(\hat{u}_C, \hat{v}_C)$  are the estimated values using the suggested parameter set. Note that this fitness measure does not correspond to an error measurement in real-world.

We can construct the full model for the direct transformation by combining equations 4 to 14. Again we set  $k_R = 1$  and also  $f'_C = 1$  to account for the dependencies between the parameters. We also simplified the coordinate transformation by assuming  $\mathbf{R}_{CR}$  is the identity matrix. There were eleven remaining parameters to estimate:  $u_{R0}, l_R, v_{R0}, f'_R, k_C, u_{C0}, l_C, v_{C0}$  and the three components of  $\mathbf{o}_{CR}$ .

The particle swarm algorithm optimized all eleven parameters and achieved an error of  $err = 13.88$  pixels in the color imaging sensor. This is a good result since the range imaging sensor has 8 times more pixels in width and 7.74 times more pixels in height than the color imaging sensor. Thus the diagonal of 11.13 pixels would be an expected measure of this error which is given *a priori* through the different resolutions of both sensors and quantization errors. The specific values of the parameters are not published here since they depend on our specific mechanical setting. On other data sets we obtained better error rates around  $err = 9.5$  pixels in the color imaging sensor.

Next, we were interested in how our calibration method can be used to find a simplified set of equations that still yields an acceptable error on the reference points. Therefore, we started the particle swarm algorithm in the incremental mode. It could be observed that only releasing the first four parameters led to significant reduction of the error.

Thus, we implemented a presumably simplest version for the direct calibration, a linear relationship described by equation (15) and (16) with directly inputting  $u_R$  and  $v_R$  by assuming  $x'_C = u_R$  and  $y'_C = v_R$ . The best error rate achieved with this models was  $err = 15.82$  pixels in the color imaging sensor. An advantage of this model is that it can be reversed. Thus, it is possible to get the range value at a corresponding pixel in the color imaging sensor.

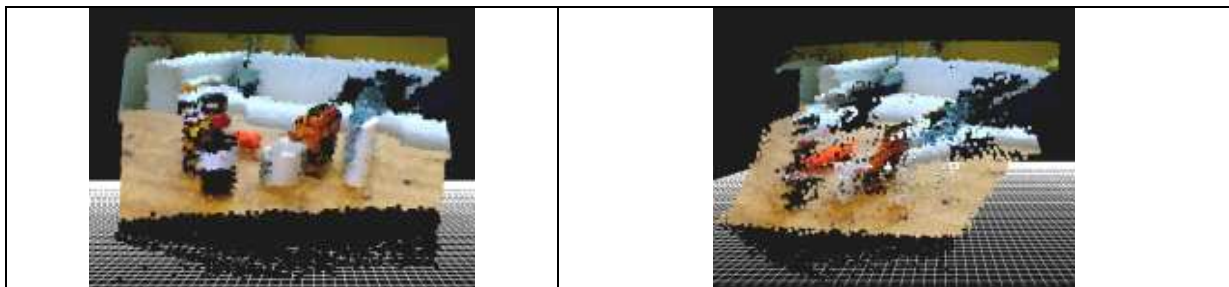
## 6. RESULTS AND FUTURE WORK

For our bi-modal sensor head we found calibration models of different accuracy and complexity. The real-world calibration with an error rate of  $err = 44.52$  mm needs to be improved. However, this is beyond the scope of this paper. One direct calibration model works with eleven parameters and achieves an error rate of  $err = 13.88$  pixels in the color image.



**Figure 8:** Left: The range image (with normalized histogram for illustration). Right: The color values for pixels in the range image generated with the direct calibration that uses eleven parameters. The black banner at the bottom is caused through the mechanical setting of the sensors.

Using our method with the particle swarm optimizer in incremental mode it was possible to reduce the model to a linear version with four parameters that is even reversible such that it can give the distance corresponding to a color pixel in the color imaging sensor. Combining the direct calibration and the real-world calibration we constructed a 3d scene which is shown in figure 9. When the scene is slightly rotated (figure 9, left) then errors can be detected.



**Figure 9:** Left: The scene of figure 7 in a 3d space. Right: the 3d scene slightly rotated. The errors are given by principle since the scene is only taken from one view.

In future we will further enhance the calibration models for the sensor head by using our proposed method and setting up high-quality reference point data bases. Furthermore, we aim to segment object appearances in the color image using range information. A first implementation is shown in figure 10. Here a 'blob' in the range image is detected first that corresponds to the position of the gripper of a robot. Then the coordinates of the bounding rectangle of this blob are transformed to the color image. The rectangle segments the color information of the object plus some disturbances from the background and the robot's gripper.



**Figure 10:** Left: The scene with the object attached to the robot's gripper. Centre: The segmented region by applying the linear direct transformation method. Right: The segmented region in large size.

The calibration method could potentially be extended to a fully automatic version. Here, an interesting problem lies in finding the order in which automatic methods 'try' different subsets of the equation to converge to a sufficient model.

Furthermore, there is current work on fusing different views of the scene into an improved 3d representation. First the coordinate transformation parameters from one view to the next are estimated then the scenes will be merged to get an improved scene.

## 7. CONCLUSION

The proposed method can easily be used to find suitable calibration models of different types incorporating different sensors. The method is simple since it uses the relevant equations explicitly and does not assume any special properties of the equations. This paper presented a scheme how the calibration model can be refined or reduced with minimal effort. Also a first attempt to reducing models automatically by using a particle swarm optimizer in an incremental mode that moves from simple to more complex models was presented.

We hope that this work can help others that use complex sensor settings or that it is may be one starting point into the wider field of automated model selection in the context of calibration problems.

## 8. ACKNOWLEDGEMENTS

The work described in this paper was conducted within the EU Integrated Project COGNIRON ("The Cognitive Companion") and was funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020.

## 9. REFERENCES

- [1] Hägele M., Neugebauer J., Schraft R. D. et al. (2001): From Robots to Robot Assistants. In: Proceedings of the 32<sup>nd</sup> International Symposium on Robotics in conjunction with IMS 2001 April 19 - 21, Seoul, Korea, Vol. 1, pp. 404 - 409
- [2] Raja C. (2004): The Cognitive Robot Companion and the European Beyond Robotics Initiative. In: 6<sup>th</sup> EAJ International Symposium "Living with Robots" 2004 October 4 and 5, Tokyo, Japan. <http://www.cogniron.org/Publications.php>.
- [3] Hans M., Graf B., Schraft R. D. (2002): Robotic Home. Assistant Care-O-bot. Past - Present - Future. In: IEEE Industrial Electronics Society u.a.: IEEE ROMAN 2002: Proceedings, 11<sup>th</sup> IEEE International Workshop in Robot and Human Interactive Communication, 2002 September 25 - 27, Berlin. Piscataway, NJ: IEEE, 2002, pp. 380 - 385

- [4] Perceptron Inc. (1993): LASAR Hardware Manual. 23855 Research Drive, Farmington Hills, Michigan, 1993.
- [5] Oggier, T. et al. (2003): An all-solid-state optical range camera for 3D real-time imaging with sub-centimeter depth resolution (SwissRanger™). In: Proceedings of the SPIE, Vol. 5249 No. 65, 2003. [http://www.swissranger.ch/pdf/CSEM\\_manuscript\\_5249-65.pdf](http://www.swissranger.ch/pdf/CSEM_manuscript_5249-65.pdf)
- [6] Weingarten J., Gruener G., Siegwart R. (2004): A State-of-the-Art 3D Sensor for Robot Navigation. In: Proceedings of IROS' 2004, Sendai, September 2004. [http://asl.epfl.ch/aslInternalWeb/ASL/publications/uploadedFiles/2004iros\\_weingarten.pdf](http://asl.epfl.ch/aslInternalWeb/ASL/publications/uploadedFiles/2004iros_weingarten.pdf)
- [7] Swiss Ranger SR2: CSEM SA. Swiss Ranger SR-2 Datasheet. <[http://www.csem.ch/corporate/Awards/p\\_531\\_SR-2\\_Preliminary-0355.pdf](http://www.csem.ch/corporate/Awards/p_531_SR-2_Preliminary-0355.pdf)>. Rev. 2005.07.22
- [8] PMD[vision] 1k-s (development kit) PMD Technologies: PMD Technologies. Datasheet PMD[vision]® 1k-s. <[http://www.pmdtec.com/inhalt/produkte/documents/PMDvision1k-S\\_000.pdf](http://www.pmdtec.com/inhalt/produkte/documents/PMDvision1k-S_000.pdf)>. ISSN: Rev. 2005.07.22
- [9] PMD[vision] 19k PMD Technologies: PMD Technologies. Datasheet PMD[vision]® 19k. <[http://www.pmdtec.com/inhalt/produkte/PMDvision19k\\_001.pdf](http://www.pmdtec.com/inhalt/produkte/PMDvision19k_001.pdf)>. Rev. 2005.07.22
- [10] Canesta (development kit): CanestaVision™ EP Development Kit. Datasheet with detailed specifications. <<http://www.canesta.com/downloads/11005-01%20rev%206%20EP%20DevKit%20Data%20sheet%20PSD.pdf>>. Rev. 2005.07.22
- [11] Zmini: 3DV Systems. Zmini datasheet. <<http://www.3dvsystems.com/products/Zmini/Zmini%20Data%20Sheet.pdf>>. Rev. 2005.07.22
- [12] DMC 100 3DVSystems: 3DV Systems. DMC100 System Specification. <<http://www.3dvsystems.com/products/dmc.html>>. Rev. 2005.07.22
- [13] Range Camera Prototype: Matsushita Electric Works, Ltd. Press release. <<http://www.mew.co.jp/e-press/2004/0402-01.htm>>. Rev. 2005.07.22
- [14] Eberhart, R., Shi, Y. and Kennedy, J.: Swarm intelligence. Morgan Kaufmann, 2001.
- [15] 1394 Imaging. Web page of the color imaging sensor DFK42F02: <http://www.1394imaging.com/products/cameras/dfk41f02/?sid=85d8622059239e8ffa36d8c8c25762c0>
- [16] Forsyth, D. A., Ponce, J. (2003): Computer Vision: A Modern Approach. New Jersey. Prentice Hall, 2003
- [17] Craig, J. J. (1989): Introduction to Robotics: Mechanics and Control. New Jersey. Addison-Wesley, 2<sup>nd</sup> edition, 1989
- [18] Vapnik, V. (1998): The Nature of Statistical Learning Theory. New York: Springer-Verlag, 1998.