Tackling Multi-sensory 3D Data Acquisition and Fusion

Tim Bodenmüller, Wolfgang Sepp, Michael Suppa, and Gerd Hirzinger

Abstract— The development of applications for multi-sensor data fusion typically faces heterogeneous hardware components, a variety of sensing principles and limited computational resources. We present a concept for synchronization and communication which tackles these challenges in multi-sensor systems in a unified manner. Here, a combination of hardware synchronization and deterministic software signals is promoted for global synchronization.

Patterns of event-driven communication ensure that sensor data processing and evaluation are not bound to runtime constraints induced by data acquisition anymore. The combination of unified range and pose data description, event-driven communication, and global synchronization allows to build 3d sensing applications for various tasks. The proposed concept is implemented and evaluated for a variety of applications based on the DLR Multisensory 3D-Modeller. Extendability to other range and pose sensors is straightforward.

I. INTRODUCTION

The generation of three-dimensional models of arbitrary objects and environments is highly requested in the domain of robotics and beyond. Work cell exploration, collision avoidance, and grasp planning rely on task specific 3d models in the same way as workpiece inspection, rapid-prototyping, and the design of scenes for the games industry. The continuous trend to more detailed and realistic models pushes the requirements to more accurate, faster and easier 3d acquisition of real objects processes further.

The limitations of a static single range-sensing device to acquire complete models are obvious. First, these devices sample the target object geometry only from a particular view and second, its sensing ability depends on specific surface properties such as reflectivity, color, texture or smoothness.

A common way to enable the automatic combination of different views is to measure the sensor pose by external devices such as robots and tracking systems synchronously to the acquisition of range data. Further on, the sensor pose can be estimated automatically from intensity or range image data which is still an ongoing research topic. Sensor limitations are eluded on the other hand combining different sensors or sensing principles in one device. Such multi-sensory devices are predestined for the concurrent acquisition and fusion of complementary range information.

In practice, the integration of range with position sensing devices shows to be a recurrent and tedious process, due to the heterogeneity of hardware components, the diversity

Fig. 1. Diversity of range sensors: SICK LMS 200 (http://www.sick.com), Z+F Imager 5006 (http://www.zf-laser.com), Point Grey Bumblebee 2 (http://www.ptgrey.com), SR-3000 (http://www.swissranger.ch), 3D Scanners ModelMaker Z (http://www.3dscanners.com), Steinbichler Comet IV (http://www.steinbichler.de), DLR-LRS (http://www.dlr.de/rm), PMDTec PMD[vision] 19k (http://www.pmdtec.com).

of sensing principles and sensing algorithms, as well as arbitrary application requirements. Moreover, the variance of both, cycle time of sensors and computational complexity of applications has to be taken into account. In particular, range image sensors cause high computational load and require high bandwidth. In this paper, we promote a flexible concept for the efficient integration of the variety of range and position sensing devices for heterogeneous and distributed 3d applications.

II. RELATED WORK

In the field of robotics several concurrent frameworks allowing for sensor integration exist. The most common tools for low-level control are Matlab-Simulink[1], Labview[2], and Ptolemy II[3]. These frameworks support synchronous data flow models, where execution time of the components is idealized and where all input data is processed. Usually, all parts are executed synchronously at a constant rate and the interface is not bound to a special application domain.

Based on these concepts, various frameworks, especially for mobile robotics, have been developed. Early work of Pack et al. [1] proposes an actor-oriented framework where interfaces and control are based on multi-valued or fuzzy-logic. No assumptions have been made about the model of concurrency. Waarsing et al. [2] present a framework for multi-sensory, multi-actuator systems which conform to dynamic data flow models. Here, the programmer decides whether

[1] http://www.mathworks.com/

[2] http://www.ni.com/labview/

[3] http://ptolemy.eecs.berkeley.edu/ptolemyII/

signals are processed synchronously or asynchronously. Nevertheless, a distinct actor is designed to control the order of execution within the network. The framework does not specify any interfaces. The MCA architecture of Scholl et al. [3] addresses the definition of a common interface for actors which can be connected hierarchically. The input and output interface are not sensor specific and are restricted to vectors of floating point values. Another popular open source framework is Player/Stage of Vaughan et al. [4]. This architecture focuses on the definition of abstract device interfaces for sensors and actuators but is not optimized for high cycle rates or high bandwidth. Recently, Tessier et al. [5] propose a dynamic data flow architecture being aware of the synchronism of sensor readings. This purpose is achieved by saving the date of incoming sensor data.

III. MODEL OF 3-D ACQUISITION AND FUSION

The proposed concept addresses the communication problem of multi-sensory networks for 3d sensing. In contrast to real-time frameworks used in robotics data processing is split into synchronous low-level data acquisition and possibly asynchronous range data handling.

In detail, the concept considers three system layers as depicted in Fig. 2. On the hardware layer, all components are globally synchronized. A bridge marshals hardware specific communication to event-driven communication on the sensor layer. Moreover, the bridge ensures concurrent and generic access to hardware components. Separate unified streams of pose sensor and range sensor data are gathered and merged on the application layer according to application specific demands.

In the paper first a generalization of the joint hardware/software synchronization of Suppa et al. [6] is presented (see Sec. IV). On top of global synchronization at hardware level a event-driven framework in accordance with [2], [3], [5] is build which offers the possibility to integrate time consuming processes for range data evaluation (see Sec. V). Further, a simple formalized interface to arbitrary range and pose sensors is proposed (see Sec. VI). Finally, an imple-
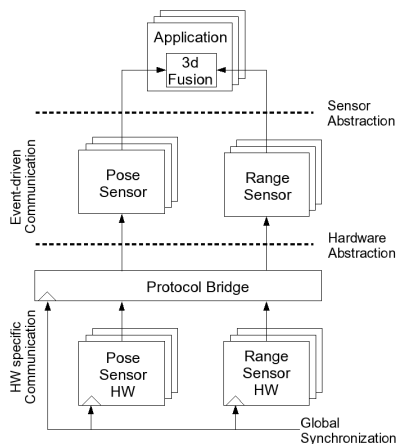
mentation of the above concepts is given (see Sec. VII) and 3d sensing applications are outlined, showing the flexibility and simplicity of the concept (see Sec. VIII).

IV. GLOBAL AND LOCAL SYNCHRONIZATION

In contrast to closed loop control of actuated robots only the acquisition of sensor data has to be globally synchronized while the following processing steps are not time-critical and can occur asynchronously from the sampling process.

We achieve decoupling of real-time and non real-time processing by *labeling* sensor readings with the timestamp of a *global* clock. When data from different sensors have to be related to each other then information is again *locally* synchronized over the assigned timestamp. The requirement occurs as soon as range data is merged either temporally using timestamps or spatially using timestamps and pose information.

Global synchronization is achieved through combined hardware-software synchronization. Here, synchronization of a component is either obtained through periodic, electrical synchronization pulses (h-synced) or through periodic software messages on a deterministic bus (s-synced). A dedicated node, the so called sync bridge, is required to generate software synchronization messages contemporaneously to the hardware synchronization signals. The concept, illustrated in Fig. 3, allows to integrate devices in the system which are able to be synchronized either on a low hardware-level (e.g. cameras and laser modules) or on a software-level (e.g. proprietary pose sensing devices). The key is to finally label measurements with the timestamp of the global clock transmitted over a deterministic bus. Thereafter, the measurements can be transmitted over any kind of deterministic or non-deterministic connection.

The data received from the components can finally be any of the following types: h-synced, s-synced or not synced at all. A protocol bridge which marshals between hardware specific protocols and generic sensor events furthermore labels the primer and latter sensor readings taking specific latencies into account (cf. Fig. 2).

V. SENSOR COMMUNICATION SEMANTICS

The semantics of communication between data sources and data sinks are mainly responsible for the performance of the overall system. The semantics determines how and when sources and sinks interact. In contrast to generic communication networks 3d sensing systems have several concurrent demands on the type of interaction. While collision avoidance

Fig. 2. Overview of the communication concept for 3d data acquisition and fusion.
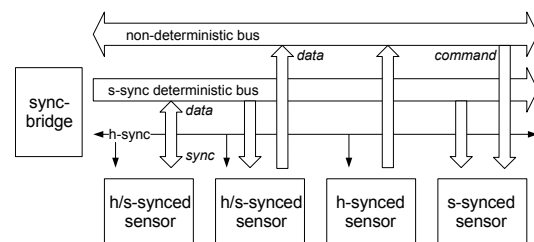

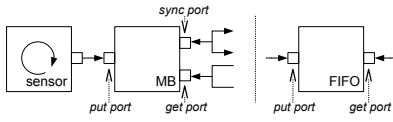Fig. 3. Hardware and software synchronization.

Fig. 4. Communication with Multi-Buffers (MB) (left) and First-In-First-Out buffers (FIFO) (right).

applications for instance rely on a low latency of the sensor signal, computationally expensive 3d modeling applications would request for buffered access to sensor data. Therefore, in this paper three distinct communication patterns for event-driven communication between data sources and sinks are promoted.

The key concept of the architecture is the implementation of the data interfaces by active Multi-Buffers (MB) as a generalization of double and triple buffering (see Fig. 4). The buffer provides two ports on the application side, one for synchronization and the other for non-blocking read access. This component enables to couple or decouple the sensing process from the application process by choosing an appropriate connection. Note, that a single entry of the buffer is generally not copied but *shared* by multiple concurrent reading processes. Three types of communication pattern are possible, that are

- synchronous,
- asynchronous buffered, and
- asynchronous unbuffered

communication (Fig. 5). Synchronization is achieved when the multi-buffer notifies the application of a new entry. When this call is used on the application side to immediately gather and process new sensor readings then all the computation lie within one execution path.

Alternatively, the sensor data sources and sinks can be decoupled with the asynchronous, buffered communication pattern by pushing the sensor readings into a First-In-First-Out (FIFO) buffer. The application actively gathers the sensor data and processes it as soon as computation resources become available. This type of coupling ensures maximal, lossless throughput, especially for applications with varying
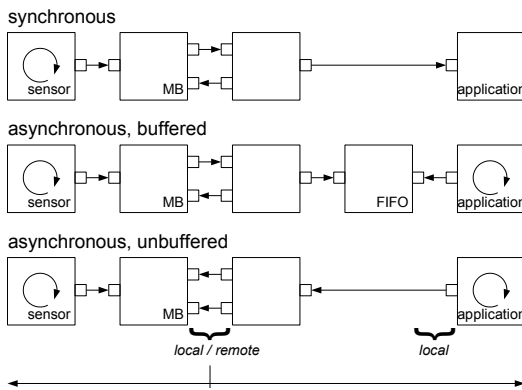

Fig. 5. Patterns of event-driven communication and coupling between sensor and application.

processing time or delayed processing.

The third possibility for decoupling sensor and application is given by the asynchronous, unbuffered communication. In contrast to the other two communication types, the application is not forced to process all sensor data. Instead, the process requests sensor data *on demand* getting the most up-to-date measurement. This pattern allows to incorporate computationally expensive processes and applications which sporadically demand for sensor data.

VI. SENSOR INTERFACE

Unified interfaces to range and pose sensors facilitate the development of 3d sensing applications. They reduce the implementation effort and minimize errors due to the use of different formats, units, representations, etc. In detail, a generic range data description has to strike a balance between offering raw, uncalibrated, sensor specific data and providing a set of 3d points, i.e. the information to be transferred must be chosen judiciously. Here, the data is represented by a calibrated range image in combination with a unified mapping to Cartesian coordinates. Thus, applications are able to use complete range data information but are not burdened by complicate, sensor-specific conversion rules.

A. Generic pose data description

The pose measured by a robot, a tracking system, or a registration algorithm is a rigid motion transformation. Unlike the variety in system specific representations of such transformations, here we rely on the representation as homogeneous matrix

$$^{world}\mathbf{T}_{local} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0'} & 1 \end{pmatrix}. \tag{1}$$

The range sensor pose is mandatory for 3d data fusion, i.e. global alignment of range data. Since in general, the range sensor pose is not directly measurable an additional constant transformation $^{local}\mathbf{T}_{sensor}$ is needed, which relates the measured coordinate frame (*local*) to the range sensor frame (*sensor*). This additional transformation is estimated during system calibration and is specific to a particular combination of range sensor and pose sensor.

B. Generic range data description

All sensors are identified to provide grid-ordered measurements with respect to a local sensor coordinate system dependent on the measurement principle. Hence, the sensor is assumed to sample its environment equidistantly on a two-dimensional grid or that the measurements can be transformed respectively. This assumption includes stripe-sensors and touch-probes as a special subset of 2d grid-ordered samples (either one-dimensional or single value). Further, a grid value directly represents a metric distance and not disparities for instance.

Let $\mathbf{I}_{dist}$ denote a range image of size $N_u \times N_v$ containing single distance values d_{ij} correspondingly to the i-th row and j-th column. In the following a generic and simple calculation rule for the mapping $\mathbf{I}_{dist} \to \mathrm{R}^3$, i.e. for conversion into
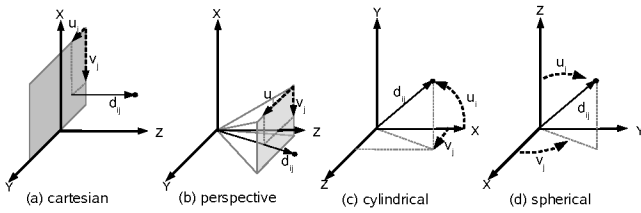
Fig. 6. The four geometric types and its representation of a range measurement in Cartesian space.

Cartesian coordinates and global alignment of range data is introduced.

As the grid-ordering of the range values depends on the measurement principle of the sensor, the sensor geometry is taken into account. Most sensors are identified to measure relative to one of four basic geometric types: *Cartesian*, *perspective*, *cylindrical* or *spherical* (see Fig. 6). Examples of sensors fitting to these types can be found in Table I.

Let $\mathbf{p}_{raw}(i,j) = (u(i), v(i), d_{ij})^T$ denote the location of a distance value in the sensor-specific coordinate system, with the grid locations,

$$
\begin{align}
u(i) &= u_0 + i \cdot \Delta u; \quad i \in \{0, 1, \ldots, (N_u - 1)\} \quad (2) \\
v(j) &= v_0 + j \cdot \Delta v; \quad j \in \{0, 1, \ldots, (N_v - 1)\} \quad (3)
\end{align}
$$

defined by an offset u_0, v_0 and a sample width $\Delta u, \Delta v$. The mapping into Cartesian sensor space $\mathbf{p}_{raw}(i,j) \mapsto \mathbf{p}_{sensor}(i,j)$ is calculated in accordance with the geometric type of the sensor coordinate system (see Table I).

Using the calculation rules above a minimal description for range data is identified. It consists of the following values:

- the synchronization cycle timestamp (cf. Sec. IV)
- the geometric type
- the range image size N_u and N_v,
- the offset u_0, v_0, and sampling width Δu, Δv,
- the range image $\mathbf{I}_{dist}$,

Finally, a point $\mathbf{p}_{sensor}(i,j)$ in sensor coordinates is aligned to global space by the concatenation of range data, calibration, and pose for the same global time according to

$$
\hat{\mathbf{p}}_{world}(i,j) = {}^{world}\mathbf{T}_{local} \cdot {}^{local}\mathbf{T}_{sensor} \cdot \hat{\mathbf{p}}_{sensor}(i,j), \quad (4)
$$

TABLE I
The equations for the calculation of $\mathbf{p}_{sensor}(i,j)$ depending on the geometric type.

geometric type	$\mathbf{p}_{sensor}(i,j) =$	example
Cartesian	$\begin{pmatrix} u(i) \\ v(j) \\ d_{ij} \end{pmatrix}$	Laser-pointers, touch-probes
perspective	$\begin{pmatrix} u(i) \cdot d_{ij} \\ v(j) \cdot d_{ij} \\ d_{ij} \end{pmatrix}$	PMD[vision] 19k, Bumblebee 2
cylindrical	$\begin{pmatrix} d_{ij} \cdot \sin u(i) \\ v(j) \\ d_{ij} \cdot \cos u(i) \end{pmatrix}$	SICK LMS 200, DLR-LRS
spherical	$\begin{pmatrix} d_{ij} \cdot \sin v(i) \cos u(j) \\ d_{ij} \cdot \sin v(i) \sin u(j) \\ d_{ij} \cdot \cos v(i) \end{pmatrix}$	Z+F Imager 5006

where $\hat{\mathbf{p}}$ denotes the homogeneous coordinate of a 3d-point $\mathbf{p}$. Note, that this final alignment is assigned to the application level and not to sensor level.

VII. IMPLEMENTATION

In the following, the above concepts are implemented on the basis of the DLR Multisensory 3D-Modeller.

A. 3D-Modeller Hardware

The DLR Multisensory 3D-Modeller is a highly integrated, multi-sensory device consisting of stereo cameras, a rotating laser-scanner module, and laser-line modules. The system implements three different range sensors: the Laser Range Scanner (LRS) [7], the Laser Stripe Profiler (LSP) [8], and the Stereo Camera Sensor (SCS) [9]. A detailed description of the hardware components and range sensing principles can be found in [10].

The system is used hand-guided or mounted on a robot. At the moment, coupling to four different pose sensor is implemented: a smARTtrack1 optical tracking system[4], a FARO GoldArm passive manipulator[5], a KUKA industrial robot[6], and the DLR light-weight robot. The image-based sensors are registered to the pose sensors using the CalDe/CalLab toolbox[7]. Given the calibration, it is possible to combine the range data from the different sensors.

B. Synchronization

The hardware synchronization is implemented using a 25Hz video clock signal, and for software synchronization a CAN bus is used, sending a consecutive counter as timestamp. The 3D-Modeller is connected via Firewire to a host PC, the so-called *Sensor PC*. The latter runs both, the hardware abstraction and the sensor services. Currently, a system with a Intel Dual P4 3.06GHz processors, running a standard Linux 2.6, is used.

C. Sensor Communication

The communication model introduced in Section V is not restricted to a specific mechanism of transportation. We implemented the communication channels using a combination of POSIX shared memory and POSIX 1003.1b semaphores for local interprocess communication and Remote Procedure Call (RPC) interfaces or the XIRP protocol[11] for inter platform communication. While RPC is based on the platform independent data representation XDR[8] the XIRP protocol is based on XML for the specification of a standardized sensor interface for robot control [11]. All of these mechanisms are hidden to the programmer and allow to setup distributed concurrent systems.

The communication channels are used to transfer raw sensor data from the protocol bridge to the range sensors as well as to transfer pose data and generic range data

[4]http://www.ar-tracking.com
[5]http://www.faro.com
[6]http://www.kuka.com
[7]http://www.dlr.de/rm/callab
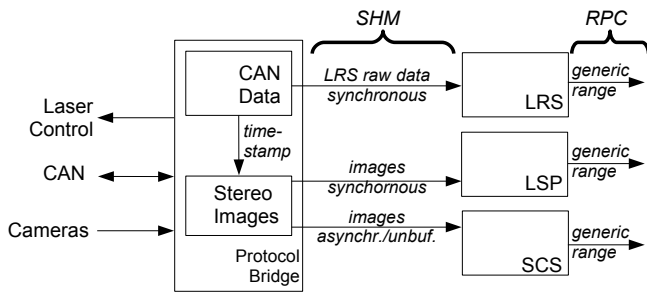[8]eXtended Data Representation

Fig. 7. Communication patterns and channel implementation used for the connection of protocol bridge with range sensors and with the application.

to the application. Efficiency is guaranteed using shared memory communication for the former kind of connection and platform distribution is ensured using the channel implementation over RPC for the latter kind of connection as illustrated in Fig. 7. Range sensors such as the LRS and LSP are connected to the protocol bridge using the synchronous communication pattern. The computationally far more expensive stereo range scanner SCS is decoupled from the native cycle frequency of the camera using the asynchronous, unbuffered communication pattern.

D. Sensor Communication Performance

The models of communication of Sec. V allow for a lean implementation using the mechanisms of POSIX conform systems as mentioned above. That is, the communication channels keep the overhead low despite the flexibility with respect to the number of users in the system. Fig. 8 illustrates the superiority of communication using shared-memory multi-buffers compared to copy based point-to-point communication. Both methods perform equally for small packet sizes (16 bytes) achieving approx. a latency of $25\mu s$ and a maximal cycle rate of 22kHz on a Dual P4 3.06GHz. The gain in performance of shared multi-buffer communication becomes obvious for packets as large as a color PAL image (1320k bytes). The cycle time for multiple
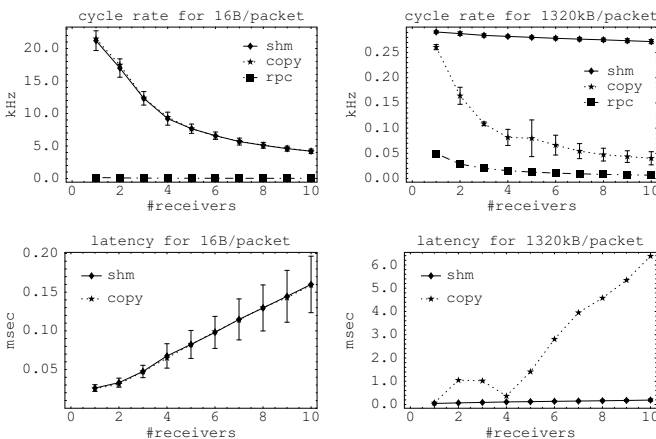

Fig. 8. Performance measurements for communication based on shared-memory read, shared-memory copy read, and remote procedure calls. Top: Maximally achievable cycle rate. Bottom: Average latency at maximally achievable cycle rate.

copy operations is naturally limited by the maximal memory bandwidth. Shared access to images instead, achieved by temporarily locking the buffer, is independent of the number of users, keeps latency low and cycle rate high.

Obviously, inter-platform communication cannot achieve a comparable performance. The system designer can however opt between locally low communication overhead and remote computational resources.

VIII. APPLICATIONS

The concepts described in this paper favor the interchangeability of range and pose sensors and leave the decision on which sensors to use and how to connect to a sensor to the application. This allows arbitrary complex, heterogeneous applications to simultaneously use a set of sensors. With the synchronization concept, an application is able to merge range and pose data on demand by local synchronization, i.e. a comparison of timestamps. In the following, three applications using the sensors of the 3D-Modeller are mentioned.

A. Online Surface Triangulation

In this application, real-time 3d modeling of smaller objects like busts or technical parts is performed, using an arbitrary combination of range and pose sensor. The used online surface reconstruction algorithm [12] incrementally generates a 3d model of an object from a stream of 3d points. Concurrently, single camera images are gathered and mapped onto the surface. Moreover, the live image of a camera can be underlaid to the 3d model using direct access to the hardware abstraction layers. In Fig. 9 a screen-shot of the process with augmented live image is shown.

As model generation and visualization are computationally expensive processes on remote systems, the asynchronous, buffered communication is used for both pose and range data transmission, assuring no data is lost due to delayed processing. In contrast, the camera images are transferred in asynchronous, unbuffered mode, skipping images if the visualization is too slow. The communication for this application is illustrated in Fig. 10.

Fig. 9. Online triangulation with live camera stream augmentation.

B. Work Cell Exploration

In this application, the environment of the robot (the physical space) is only partially known, i.e. there exist unknown regions to be explored.

The exploration algorithm incrementally gains information about the surroundings analyzing range information from different sensors attached to the robot TCP. It chooses sensing locations, so-called *Next Best Views*, based on the specific stochastic sensor model and accessibility [13]. The unified range sensor interface enables to switch between sensors while exploring. It also allows for simultaneous use of multiple pose sensors like robot manipulators concurrently to the optical tracking system. As processing is slower than the sensor cycle time[9] only the newest sensor readings are used for the exploration step. Therefore, an asynchronous, unbuffered communication is chosen (see Fig. 10).

C. Object Recognition

The concept has been adopted to the implementation of a object recognition task with the 3D-Modeller mounted on the DLR humanoid robot JUSTIN [14]. Here, range measurements from a table top scene are acquired with the SCS and provided to the object recognition modules. Pose information is concurrently gathered from a soft synchronized pan-tilt unit and linked to the correspondingly timestamped range data. The fused sensor readings allow to map the range measurements to the robot centered coordinate frame. Thus, the 3d location of the recognized bottles and glasses can be directly used to command the robot arm and hand in a dynamic look-and-move approach.

Due to the slow, computationally expensive process of object recognition, an asynchronous, unbuffered transmission of range sensor readings is appropriate.

IX. CONCLUSIONS

3d modeling applications with mobile, hand-guided or robot-controlled range scanning devices exhibit hard synchronization constraints. Moreover, it challenges software development due to the variety of range sensor devices, sensing principles, and the concurrent access to possibly large amounts of data.

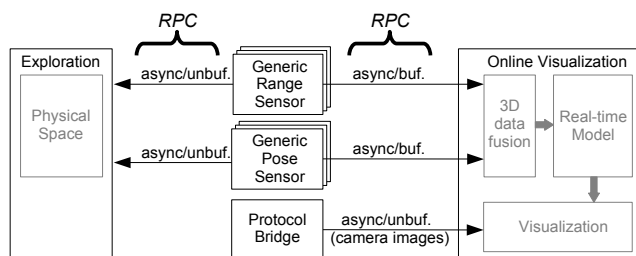[9]A typical work cell has the size of $4000 \times 4000 \times 2000$mm with 20mm resolution

---

This paper proposes a unified communication concept for multi-sensory devices, which addresses the constraints of 3d scanning. First, combined hardware/software synchronization is promoted which allows to process data asynchronously on non-real-time operating systems. The communication semantics allows for synchronous and asynchronous coupling depending on the application requirements. Moreover, it ensures low communication overhead for concurrent access to large data sets, e.g. image data. Finally, a unified sensor interface with a generic data description is introduced which represents a large step towards a standardized interface for 3d sensing, as it handles most types of range and pose sensors.

X. ACKNOWLEDGMENTS

REFERENCES

[1] R. Pack, D. Wilkens, and K. Kawamura, "A software architecture for integrated service robot development," in *IEEE Conf. On Systems, Man, and Cybernetics*, 1997.

[2] B. Waarsing, M. Nuttin, and H. V. Brussel, "A software framework for control of multi-sensor, multi-actuator systems," in *11th Int. Conf. on Advanced Robotics*, 2003.

[3] K. Scholl, J. Albiez, and B. Gassmann, "MCA - an expandable modular controller architecture," in *3rd Real-Time Linux Workshop*, 2001.

[4] R. Vaughan, B. Gerkey, and A. Howard, "On device abstractions for portable, resusable robot code," in *Proc. of the Int. Conf. on Intelligent Robots and Systems*, 2003.

[5] C. Tessier, C. Cariou, C. Debain, C. R., F. Chausse, and C. Rousset, "A real-time, multi-sensor architecture for fusion of delayed observations: Application to vehicle localisation," in *9th Int. IEEE Conf. on Intelligent Transportation Systems*, 2006.

[6] M. Suppa and G. Hirzinger, "A novel system approach to multisensory data acquisition," in *8th Conf. on Intelligent Autonomous Systems*, 2004.

[7] F. Hacker, J. Dietrich, and G. Hirzinger, "A laser-triangulation based miniaturized 2-d range-scanner as integral part of a multi-sensory robot-gripper," in *EOS Topical Meeting on Optoelectronic Distance/Displacement Measurements and Appl.*, 1997.

[8] K. Strobl, W. Sepp, E. Wahl, T. Bodenmüller, M. Suppa, J. Seara, and G. Hirzinger, "The DLR multisensory hand-guided device: The laser stripe profiler," in *Proc. of Int. Conf. on Robotics and Automation*, 2004.

[9] H. Hirschmüller, "Stereo vision in structured environments by consistent semi-global matching," *IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[10] M. Suppa, S. Kielhöfer, J. Langwald, F. Hacker, K. H. Strobl, and G. Hirzinger, "The 3d-modeller: A multi-purpose vision platform," in *Int. Conf. on Robotics and Automation*, 2007.

[11] M. Gauss, F. Dai, F. Som, U. E. Zimmermann, and H. Woern, "A standard communication interface for industrial robots and processor based peripherals - XIRP," in *37th Int. Symposium on Robotics*, 2006.

[12] T. Bodenmüller and G. Hirzinger, "Online surface reconstruction from unorganized 3d-points for the DLR hand-guided scanner system," in *Proc. of 2nd Symposium on 3D Data Processing, Visualization, Transmission*, 2004.

[13] M. Suppa, P. Wang, K. Gupta, and G. Hirzinger, "C-space exploration using sensor noise models," in *Proc. of Int. Conf. on Robotics and Automation*, 2004.

[14] C. Ott, O. Eiberger, W. Friedl, B. Bäuml, U. Hillenbrand, C. Borst, A. Albu-Schäfer, B. Brunner, H. Hirschmüller, S. Kielhöfer, R. Konietschke, M. Suppa, T. Wimböck, F. Zacharias, and G. Hirzinger, "A humanoid two-arm system for dexterous manipulation," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2006.

Fig. 10. Communication to the online modeling, visualization, and exploration applications.