Published in 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego/USA, Oct 29 – Nov 2 2007

Using Maps from Local Sensors for Volume-Removing Tools

Philipp J. Stolka and Dominik Henrich

Lehrstuhl für Angewandte Informatik III
(Robotik und Eingebettete Systeme)
Universität Bayreuth, D-95447 Bayreuth, Germany
{philipp.stolka, dominik.henrich}@uni-bayreuth.de

Abstract – Currently industrial robotic systems employ almost exclusively global maps for navigation purposes, if any. Additional information – intra-process, spatial, current, and persistent sensor data – is useful to cope with uncertainty, measurement errors, and incompleteness of data. We propose to augment robot world models by using local sensors (which provide data from a local ε-environment) and build precise maps from local sensors, with force and audio classification in orthopedics applications with a medical robot system (RONAF) as an example. Improving precision of this map-building is presented both for data localisation and data insertion.

Keywords: ***surgical robotics, local sensors, mapping, precision, calibration, navigation, milling***

I. Introduction

Many robotic applications rely on navigation to perform their tasks. With preprogrammed motions evoking the impression of autonomous activity, and with reasoning providing a certain level of insight, often a robotic system embedded into an actual environment can only execute complex tasks when it has knowledge of its surroundings. This knowledge can be grounded in information from sensors, which in turn is stored in maps to allow the system to access previously acquired knowledge for reasoning purposes.

One can clearly differentiate two kinds of situations: the map can be available before process execution starts, or it has to be built, completed or maintained during execution. Furthermore, two conceptually different types of sensors can be used: *global sensors* which collect data from a large area and return it with associated position information (the data is embedded into a coordinate system, e.g. in radar imaging), and *local sensors* that are more restrained in their range – they collect only data from the immediate neighbourhood of the sensor interaction location.

Currently, stationary robots – including both industrial and medical robotic systems – and to some extent mobile robots almost exclusively employ static maps from global sensors for navigation purposes. These maps can be built from radar, laser, or ultrasound imaging (*US*), from cameras or camera arrays, from computer tomography (*CT*) or magnetic resonance tomography (*MRT*), and many other sources. While a global map facilitates the planning process, it may have insufficient quality in terms of precision, timeliness etc. Although a large body of works concerning the *simultaneous localization and map-building* (*SLAM*) problem for mobile robots exists, the case is different for stationary robots. We aim to augment the world model for these systems by implementing map-building and navigation based on local sensors – which are able to capture information only directly from the sampling location environment, and which would otherwise be lost to the process control – and present the medical robot system RONAF as an example for an orthopedic application (Figure 1, [3], [6]) in *computer-/robot-assisted surgery (CAS/RAS)*.

This paper gives an overview of the state of the art motivating further work (Section II). Local sensors and maps from local sensors are introduced with a discussion of their performance and limits (Section III). Based on a specific surgical robot system (Section IV), simulation and experimental results are presented (Section V). Finally, conclusions and promising further work are presented (Section VI).

II. State of the art

In this section, the results presented in this paper are motivated from the point of view of medical and especially surgical applications, such as the presented system itself. In CAS/RAS, stationary robotic systems widely and almost exclusively employ global sensors for navigation. This includes imaging sensors (CT, MRI, US) for preoperative

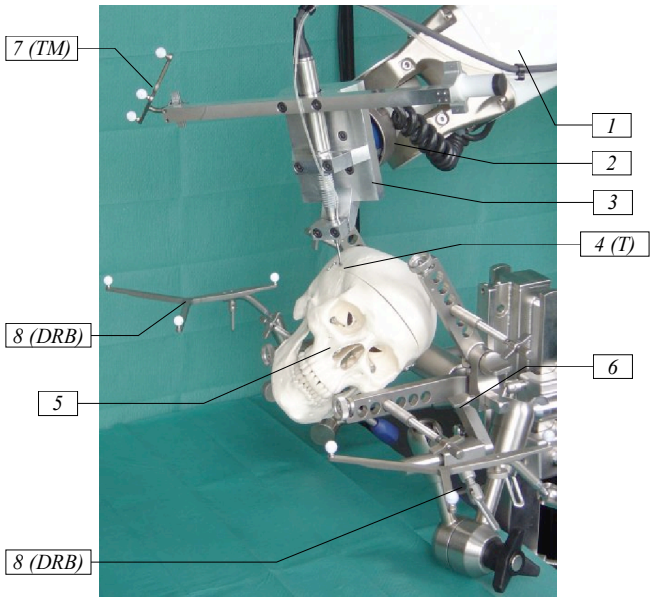

Figure 1: Setup of the RONAF system – robot arm (1) with force/torque sensor (2), tool holder (3), surgical miller tip (4), skull phantom (5), fixtures (6), and infrared-reflective markers of the tool (7) and the robot base (8)

planning and confirmation purposes as well as position sensors (encoders, trackers) for intraoperative use.

Local sensors, in contrast, are rarely used, e.g. force/torque sensors or proximity detectors which operate in control cycles. Prominent examples are the control of feed rate in milling applications based on force measurements [2], [5] or tool deflection correction [2], [12]. Another sensor (developed for the presented system) is a highly precise A-mode ultrasound scanner for thickness measurements [14], which can be used for bone volume reconstructions in conjunction with tracking. Laser surface scans and fluorescence measurements for tissue discrimination are other possibilities under early development [8]. For general industrial systems, local sensors include e.g. proximity detectors like capacitive sensors or PSD-based distance sensors.

In milling applications, the contact between miller and material provides visual, haptic, tactile, and olfactory clues to the current situation. However, they receive comparatively little attention. One area of research is use of force data for classification, e.g. the classification of surgeon skill based on force measurements at endoscopic grippers [10], where Hidden Markov Models prove useful for analyzing the low-frequency flow of data. Another area under development is the high-frequency analysis of vibration and sound [1], where microphones record audio samples during bone milling which are classified as different bone contact states through neural networks, support vector machines, or Hidden Markov Models. The use of 3D force and sound samples as input for contact state classification and mapping in milling applications has been presented in [11].

Tool deflection by deformation can have noticeable effects upon surgical serial manipulators. For simple setups, this can be approximated reasonably well by simple axis-independent linear models (for an overview cf. [12]; [2] presents another similar system). Similarly, faulty or insufficient tool calibration obviously directly affects localisation results.

No approaches seem to exist which aim at persistent use of and navigation based on local sensor data, and map-building is considered only insofar as tracked US or laser scans are used for reference purposes or registration control. Precision needs to be examined closer for maps built by local sensors. Thus, “local navigation” as defined in [6] promises a benefit unexplored in current systems. Of course such local sensor information can be used for sensor-based control as well as navigation, an approach which is described for the presented system e.g. in [3, 5].

III. Method

In this section, after a brief definition of important terms (Subsection A), the concept of local sensors is introduced (Subsection B). This is followed by the introduction of the map from local sensors concept (Subsection C) and a discussion of the limits of resolution thus achievable.

A. Definitions

In the context of this work, we define *sensors* as system components able to provide a feature vector C of the physical external environment or internal system state. A *map* M may store these features in an associated coordinate system, and provides *data entry* and *query functions* $b_M(C, t, P) = M$ and $q_M(P, t) = C(P, t)$, respectively, with P and t being positional and temporal constraints on the map interaction, and C a relevant feature vector. Furthermore, a robotic system needs to be able to *localize* the sensor with respect to some coordinate system by determining a coordinate vector of its own location, which is performed by measuring or receiving environment features supporting that task. An autonomous system can decide to perform *exploration* of the environment by roaming around while sampling features. If those features are saved in a map during exploration, this process turns into a more or less goal-oriented charting or *mapping* task. Finally, if a map exists and the robot system is able to perform localization with respect to the map, it can not only reason with that information use it for planning, but also execute motion plans and thus engage in *navigation*.

Table 1: Sensors currently in use for CAS/RAS

	CT	***MRT***	***US***	***Laser scans***	***Localizers***
spatial resolution (image)	high (3D)	high (3D)	medium (2.5/3D)	high (2.5D)	high
reliability	high	high	noisy imaging	easily disturbed	dep. on technology
discernible features	bone (+soft tissue)	soft tissue (+bone)	soft tissue (+bone)	surface feat. (+tissue)	no environm. information
latency	non-real-time	non-real-time	high (intermittent)	high (intermittent)	low
temporal resolution	n/a	n/a	very low	low (dep. on strategy)	high
sampling cost	high	high	rel. high	rel. high	low
patient stress	high (radiation)	high (time, acoustic)	low	low	none

B. Local Sensors

From the overview in Table 1, we can conclude that the sensors currently in use for CAS/RAS applications are either inconvenient to use (high sampling costs, patient stress), not dynamic (high latency), or have only limited capabilities (e.g. no imaging capability). Therefore it seems reasonable to look for sensors that provide dynamic data, have low sampling cost, and are able to detect relevant environment features. This is desirable for applications outside of CAS/RAS as well.

One way of receiving such information is to sample data which is already being generated in the process without additional procedures. We will call an appropriate

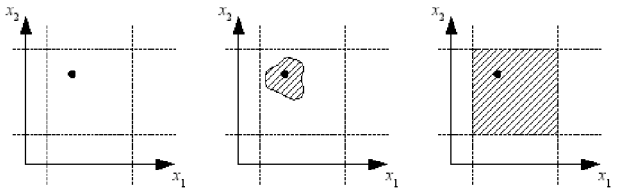

Figure 2: Comparison of the sensor resolution and range of local and global sensors (left: local with diffuse data, center: local with structured data, right: global; hatched area: sensor range where features could be differentiated)

sensor with the following *locality* characteristic a *local sensor*: The sampled local sensor data $C(P) \in \Re^{m \times n}$ is descriptive only of the immediate ε-environment of the *interaction location* $P \in \Re^6$ (the place the data is referring to, e.g. a tool tip position), although the actual sampling can take place elsewhere (in a microphone, a laser scanner etc.). Within this ε-environment, the sensor data can be structured (i.e. have image dimension[1] $m > 0$) or diffuse ($m = 0$; this does not need to imply that the sampled data originates from just one point).

This is contrasting with *global sensors*, which provide $C(P)$ for all $P \in \Re^6$ simultaneously[2]. This difference in the descriptive power of local and global sensors is shown in Figure 2, with the coordinates x_i denoting the image dimension (the information dimension n of their single sample elements is irrelevant for the definition of local sensors).

Certain kinds of local sensors can have additional properties:

- *Tool-mounted:* The normal tool interaction itself is generating the input data, so there is no additional, extrinsic measuring procedure necessary to sample data.
- *Tool-determined:* The spatial sensor range and the the set of discernible features directly depend on the tool used for interaction.
- *Dynamic:* As they interpret data originating from the process itself in real time, their latency can be kept to a minimum.

As a corollary from these latter properties, such *tool-based local sensors* incur only minimal additional stress and cost, as their input is a by-product of the process taking place anyway. Although their information dimensionality can be high (i.e. can be a high-dimensional feature vector per sample element), their image dimensionality is often zero, as the diffuse data is describing just the immediate tool position environment; that is, their data is usually not embedded into an associated coordinate system.

Two examples for tool-based local sensors are force sensors mounted between a robot and a tool (e.g. a miller) and audio sensors (microphones sampling process noise), which are shortly introduced here (cf. [Stolka05]), as they are used in the presented surgical system.

Force Sensor Data

Force/torque sensor data can monitor the low frequency system response. In the RONAF system, force data vectors $F(t_i)$ are sampled continuously at a varying rate of 25…200 Hz. Then, a feature vector $v(t_i) = (\mu, \sigma)_{|F(t_i)|}$ of the sliding average and standard deviation of absolute forces $|F(t_j)|$ over a period of $d = 0.1$ s is computed, and finally a Bayes classifier assigns system states. These features were chosen because of their relative invariance to "irrelevant" transformations (effects like changes in direction of milling motion or sample rate).

This classifier can be sensibly used to determine the miller operation and contact states C, especially for differentiation between milling in bone or soft tissue. However, as tool resonance is characterized by high frequency components, we cannot recognize it based on force measurements alone in this system.

Audio Sensor Data

Recorded sound from the process serves as a complementary high-frequency data source. In the presented system, audio data is sampled through a room microphone mounted on a tripod stand (44.1 kHz (mono)/16 bit, ~1 m off the region of intervention), feature extraction takes place (real Fast Fourier Transformation (FFT) with Blackman windowing [4]), and nearest-neighbor classification results with timestamps are transmitted to the main control system over TCP/IP. We classify robot motion and milling without contact, under bone contact, and with resonance. Thus, we can detect the operational state of the miller and the contact state C with a high sample rate (~86 Hz).

Table 2 shows how the presented local sensors satisfy the initially stated evaluation criteria.

Table 2: Presented local sensors (used for CAS/RAS)

	Force/Torque	***Audio***
spatial resolution	dependent of tool shape and temporal resolution	
reliability	*(?)*	*(?)*
discernible features	bone, dura mater, air; miller operation	bone, air, tool resonance; miller operation
latency	primarily network transmission	
temporal resolution	medium (varying sampling rate)	high (high sampling rate)
sampling cost	F/T sensor already present	microphone placement in Operating Room
patient stress	none (additional security)	none

1 The *image dimension* refers to the dimensionality of the whole sensor data sample (a finite set from an m-dimensional space), to be distinguished from the *information dimension*, referring to the dimensionality n of a single sample element (e.g. a sample pixel).

2 „All" and „simultaneously" are obviously to be understood within limits dictated by the sensors' time resolution and spatial range.

C. Maps From Local Sensors

Once suitable local sensors are available, their (soften freely available) information can be used not only for real-time control purposes. Together with the location information (provided by localizers like tracking systems or robot encoders) the sensors can be used to build maps of the region they have already passed through, vaguely similar to a scanning process. The *map building function* $b_M(C, t, P) = M$ thus serves to insert local sensor information C into a possibly sparsely filled map M based on location information P. The *query function* $q_M(P, t) = C(P, t)$ extracts environment features from this map (if they are present) for P and time t. This type of map can be called a *map from local sensors* to differentiate it from the global maps defined earlier, which are static and/or complete at the time of using them. Map positions can be labeled with either of three basically different environment types: $C = C_0$ (*empty space*), C_+ (*occupied space*), and $C_?$ (*unknown*), where C_+ can comprise several application-dependent sub-types (e.g. admissible/safe and non-admissible/unsafe regions). A map initially contains only $C = C_?$ entries. New entries are appended to *feature lists* in their respective map positions, thus storing a complete history of the known states of features for that position. Voxel grids are a useful environment representation for many applications with irregularly fragmented environments; each voxel then stores its respective feature list (cf. [11] for more information on the access functions).

Resolution and Optimization

The theoretically achievable resolution of maps from local sensors is determined by four factors. *Sensor reliability* and *precision of localisation* are system parameters. *Feature shift* includes motion of the environment features – surgically speaking, insufficient fixation of the patient (*patient shift*) and/or moving soft tissue (*tissue shift*) – and is an unfortunate event which should be avoided. Finally, when defining the data entry function $b_M(.)$, it becomes immediately clear that for tool-based local sensors, the tool shape plays an important role in the localization of sensor information as well. More precisely, the spatial resolution of the local sensor – and by extension, of the local sensor map – is limited by the *sensor point spread function*: Sensor information C is implicitly convolved with the *tool shape* S during sampling. Since tools can rarely be approximated as points (or Dirac impulses) when e.g. surgical miller heads have diameters around 5 mm, this places a severe limit on the map resolution and its usefulness. When entering the environment features at position P, the information is thus smeared out over the tool volume.

This convolution can be expressed at two points in time: *immediately*, i.e. already when writing the new sensor data C into the map, or *delayed*, i.e. when reading from the map later. Delayed convolution implies that C is entered *into M exactly at position P* only, usually resulting in a very sparse map. This is more efficient when writing ($b_M(.)$ may be time-critical, [Stolka05]), but the information in the map is tool shape- and orientation-dependent – the later convolution upon reading becomes more complex since information about those factors needs to be retrieved or reconstructed. Immediate convolution, on the other hand, spreads out the sensor information during data entry. Most obviously, this makes any later interpretation of M independent of the tool orientation and shape S. Furthermore, immediate convolution has another advantage that is difficult to capture in the delayed variant: Additional information already present can be integrated in a straightforward way to improve data localization, most notably motion direction v and application knowledge.

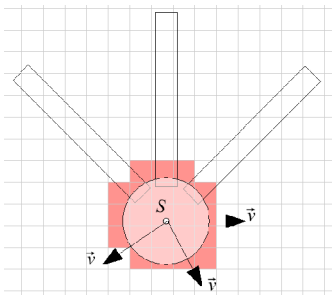

Figure 4: Schematic illustration of naive active shape S for a miller head (red; v: motion vector)

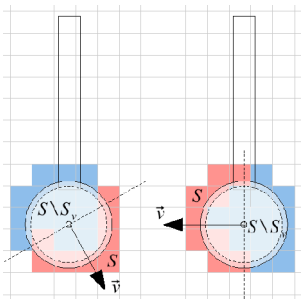

Figure 3: Schematic illustration of active shapes S_v (red) and passive shapes $S\backslash S_v$ (blue) for different motions v of a miller head

Morphological operators (e.g. shrinking) cannot be used to undo this convolution for several reasons. First, the map usually does not represent complete coverage of the environment, often resulting in an asymmetrical extension of regions along the explored boundary; second, any hierarchical classifier behavior (consistently preferring certain states over others when encountering them over parts of the sensor interaction volume) makes overlapping unpredictable; and third, destructive mapping may cause regions to physically disappear before they can be sampled from each side, resulting in further asymmetry in extension.

Instead, two optimizations can thus be used to enter better localized data into maps from local sensors. Any physical tool serving as a data generator for the local sensor will have a non-zero spatial extent, which translates into *active shape* components S_v interacting with the surrounding environment "at the front of the tool", and complementary *passive shape* components $S\backslash S_v$ (the set of points comprising the whole tool shape minus the set of points comprising the active shape) which are not in active contact with the environment but take up space in the region of interest, being "the rest" of the tool. Shape S and direction v together allow for exact determination those subshapes (Figure 4, Figure 3 show an example with a miller head as the tool, but this is easily extendable to e.g. laser beams. The same approach applies to the local sensors themselves, e.g. for a US probe).

Since the $S\backslash S_v$ does not contribute to the sensor information, but can be assumed to be empty after the tool leaves the current position, it is labeled with $C = C_0$ at each data entry, thus integrating knowledge about the tool geometry. Note that S_v extends beyond the physical tool volume in order to allow some potential information $C \neq C_0$ to remain in the map after being swept by $S\backslash S_v$. For volume-removing tools, this approach corresponds to *destructive mapping*, since previously occupied environment locations are mar-

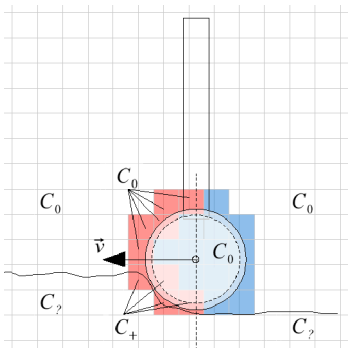

Figure 5: Entering new local sensor information C_+ into a partly known map by destructive mapping (new C_+ data only in regions swept by active shape, but not filled with C_0; passive shape enters C_0)

ked as removed from the current map this way. This first optimization integrates knowledge that could otherwise only be sampled if the sensor reached the same position again (a variant of *look-ahead*). This is especially useful in conjunction with the following second optimization, e.g. in layered milling.

Second, a static environment can be assumed when the workpiece is non-elastic (e.g. bone). When entering a new $C \neq C_0$ into the map, the second optimization dictates that this contact information may not be appended to locations where the last appended state was $C = C_0$ (as there can no longer be an obstacle; cf. Figure 5).

Access Functions

In summary, data entry for map building with tool-based local sensors can be expressed as

$$b_M(C, t, P, v) := M',$$
$$M' = M + \left(C \cdot P * S_v(v)\right) + \left(C_0 \cdot P * S \backslash S_v(v)\right)$$

returning a new map M', where C is the sensor information updating the old map M at position P with timestamp t. It is only appended at the positions resulting from convolution of P with the active shape S_v, which depends on the unit direction of the motion v. Furthermore, this is subject to the second modification, i.e. not entering sensor information $C \neq C_0$ where the last was $C = C_0$. The passive shape is labeled C_0. Therefore, the typical state development of a single map entry's last known entry over time can be described as $C_? \rightarrow C_+ \rightarrow C_0$, with both transitions being optional.

Querying the map can be expressed as

$$q_M(t, P, v) := C_{class} = \sum \left(\tau\left(M \cdot P * S_v(v), t\right)\right)$$

for relative positioning, i.e. moving the tool from the current position to a new one in its neighborhood. The positioned active shape $P*S_v(.)$ determines the region to select from M for querying – we are only interested in the features of the area to be touched by the active shape. The function $\tau(., t)$ projects out the environment information relevant for time t from the selected feature lists, choosing the features nearest to t in time. Finally, some sort of application-specific sensor data fusion $\Sigma(.)$ takes place to distill one overall environment feature from the selected spatio-temporal map region. Queries with absolute positioning, i.e. not based on a motion vector from the current position, have the form

$$q_M(t, P) := C_{class} = \sum \left(\tau\left(M \cdot P * S, t\right)\right)$$

to determine the environment state based on data all across the tool shape.

The temporal projection function $\tau(.,.)$ implements feature selection from each map entry's feature list. Since maps from local sensors may be generated by destructive mapping, for each map position the initial feature ($C_?$), the first known feature (C_+), and the possibly last feature (C_0) may differ, and what constitutes "interesting" information from the map depends on the goal of the query – i.e., whether to determine initial environment features, plan the next motion, or reconstruct the temporal development of the encountered environment (it is necessary to extract the head, tail, or an intermediate element of the features list in the respective case). In fact, the most current view of the map is basically empty except for narrow "bands" of information extending slightly beyond the explored region when using volume-removing tools – the regions already explored have been emptied and marked C_0, while other regions are still unexplored and therefore marked $C_?$.

Together, these methods provide optimal sensor information localization within a map from tool-based local sensors, independent of the original tool shape, especially when there is overlap between successive convolved tool shapes.

IV. Application and System

The presented methods were tested on the RONAF system („Robot-based Navigation for Milling at the lateral Skull Base") for automated milling of cavities in the skull bone for subdermal implantation of hearing aids.

The robot was an industrial model (Stäubli RX90) retrofitted with regard to speed and safety and suitable for medical use in hip and knee endoprosthesis milling applications (CASPAR, by Orto-Maquet). Sensors include – amongst others – a 6D force/torque sensor (JR3 90M31A with strain gage bridges, max. sensing range 63 N/5 Nm, resolution 1:4000) and a standard room microphone. The main tool is a surgical miller (electrically driven Aesculap microtron EC/GD622, up to 30.000 rpm) mounted perpendicularly to the robot tool flange to minimize deformation (Figure 1, cf. [12]).

V. Experiments and Results

Registration of the robot with the intervention region is performed by optical tracking (three-point-based [12] or by delineating a region to scan with an A-Scan ultrasound probe to reconstruct the 3D intervention volume [13]). The precision of this procedure (better than 2.5 mm) is unrelated to the navigation precision achievable by using the proposed map from local sensors.

The precision gained by an automatic tool calibration procedure added up to an improvement of as much as 2.1 mm in our experiments compared to the initial transfor-

mation determined from tool blueprints. Further repeated calibration runs yielded only improvements <0.02 mm (i.e. near-perfect calibration was achieved).

Deformation estimation experiments for the robot and tool setup were performed by approaching a rigid obstacle with varying orientations, measuring the forces and positions starting from the point of contact to crossing a predefined absolute force threshold F_{max} = 30 N. The gathered data was used to compute structure compliances along the tool axes. The actual deformation (or position error between actual and desired tool tip position) reached up to 2.0 mm, whereas after subtraction of the estimated deformation the remaining error dropped to ~0.2 mm, due to a small non-linearity of the deformation. Further experiments showed sufficient linearity and independence of the single axis deformations.

Exact sensor classification rates and procedures are described in detail in [11].

Finally, the usefulness and validity of the proposed mapping method was evaluated in a series of simulation experiments, aimed at investigating the actual influence of the mapping with and without optimizations.

To this end, we tested the proposed mapping scheme including four different optimization levels (optimization 1/directional active shape modification, 2/no entry into C_0, none, and both). In a 3D environment with a layered obstacle (top-to-bottom: air/C_0; bone/C_B (6 mm thickness); dura/C_D; 40x40x40 mm^3; 1 mm discretization; Figure 6), a simulated spherical miller tool (tool diameter 4.5 mm; active shape diameter 5.5 mm) interacted with the environment with both a Brownian motion and a systematic horizontal-parallel path (path offset 75% of tool radius; similar to the milling paths of RONAF), engaging in destructive mapping. Contacts were determined according to a single simulated sensor. The resulting maps were compared with the initial environment (reading their feature lists' last entries $C \neq C_0$, *Pre-Terminal Last-Known Map/PTM*) and with the final environment (their last entries, *Terminal Fate Map/TFM*). Using a simulation is a valid approach here, as our focus lies on evaluating the relative performance of mapping algorithms, not actual sensor data classification (described in [11]).

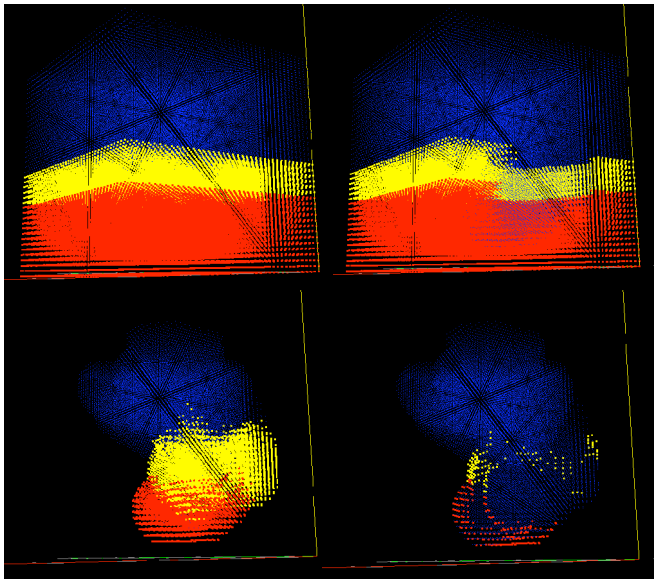

Figure 6: Simulation results after partial Brownian path exploration with destructive mapping and optimizations 1+2: initial environment and final situation (top left/right); PTM and TFM (bottom left/right) (all images from same perspective; blue: C_0, yellow: C_B, red: C_D, not shown: $C_?$)

In the following, classification denotes the result of mapping single map entries. As the relative cost of misclassification is highly application-specific, we discuss only the ratio of correctly/incorrectly classified map entries. For Brownian paths, the simulation was run repeatedly since the maps are dependent on path history (although results did not differ significantly). Furthermore, we assumed an underlying hierarchy $C_B > C_D > C_0 > C_?$ of sensor data classifier results (i.e. the dominance of partial contact states within the active shape over others; a reasonable approximation of the actual local sensor behavior).

Table 3: Map Building with Brownian Motion (correct/incorrect classifications; data of one example run for partial exploration; grey: improved cases)

	Opt –	***Opt 1***	***Opt 2***	***Opt 1+2***
PTM (partial)	12261 / 3362	15791 / 3836	13847 / 1776	**17351 / 2276**
TFM (partial)	13358 / 2265	17669 / 1958	15149 / 474	**18399 / 1228**
PTM (compl.)	50707 / 13281	50246 / 13754	56402 / 7586	**54978 / 9022**
TFM (compl.)	63986 / 2	64000 / 0	63988 / 0	**64000 / 0**

Table 4: Map Building with Horizontal-Parallel Motion (correct/incorrect classifications; grey: improved cases)

	Opt –	***Opt 1***	***Opt 2***	***Opt 1+2***
PTM (partial)	28160 / 7520	28182 / 9138	35680 / 0	**35666 / 1654**
TFM (partial)	24480 / 11200	32622 / 4698	32000 / 3680	**36959 / 361**
PTM (compl.)	44800 / 19200	43200 / 20800	54400 / 9600	**61979 / 2021**
TFM (compl.)	24236 / 39764	44894 / 19106	33836 / 30164	**64000 / 0**

For TFMs (the most current environment representation, useful for planning purposes) and PTMs (the most current environment representation before destruction, useful e.g. for *ex-post* registration with pre-operative data), the following results could be established.

With Brownian motion (the least restrictive assumption on tool behavior; Table 3) and partial exploration (200 non-air contact steps), activation of either Optimization 1 and Optimization 2 generally yielded better results than respective deactivation, while activating both was generally best

w.r.t. correct classifications. As for misclassifications, Optimization 1+2 was always 2nd-best (2nd only to Optimization 2).

With horizontal-parallel motions (200 contacts; Table 4), there was a huge positive influence by Optimization 2, while activation of Optimization 1 yielded vastly better results for TFMs (not PTMs) w.r.t. both correct classifications and misclassifications; activation of both Optimizations 1+2 was best overall.

For complete environment explorations (~2000 contacts), the results were basically the same.

From these results, it can be concluded that the proposed optimized mapping methods improve the localization of classified map entries (i.e. putting the right information in the right place) for reasonable approximations to real sensor motions. Depending on the purpose of mapping, one might argue for the use of different optimization combinations, however, using both presented methods simultaneously is generally superior to any other approach. It should be noted, however, that dominant classification results „bleed over“ into neighboring regions. This is an inherent property of the local sensing approach; its significance needs to be evaluated based on the intended application.

VI. Conclusions

We have presented a prototypical implementation of the “local sensors and mapping” concept together with the prerequisites for choosing or developing appropriate local sensors and their impact on the resulting map from local sensors. The concept has been partially experimentally verified. We could show that the additional local sensor information from the region of intervention can be sampled, classified into (surgically) relevant states, and served to build precise maps, with useful resolution optimizations in place.

In particular, two local sensors were presented and their validity for the demonstrated application shown. Local sensor mapping relies on precise localization (as discussed here w.r.t. tool calibration and deformation estimation), optimally localized data entry into the map, and finally reading environment features back from the map for navigation purposes. One usage scenario is local navigation (region skipping) based on the map from local sensors. An aspect described only implicitly in the present paper is the impact of situations where the tool sweeps regions of different subtypes $C \in C_+$, while the local sensor returns only a single one, thus blurring boundaries between regions in a non-trivial way that cannot be undone with simple morphological operators.

While the presented system is based on a surgical robot, it can easily be used with hand-held tools tracked by a localizer as well. This strategy promises safety improvements, especially when used in conjunction with map-based tool power output control systems like Navigated Control [7]. Other fields of application may include automated archaeology and other preparation methods.

Current work includes tool vibration data classification to close the gap between the low-pass force data which includes vectorial information and high-pass audio data which does not, and data fusion procedures for multiple local sensor setups during map reading.

Acknowledgements

This work is a result of the project „Robot-based navigation for milling at the lateral skull base (RONAF)“ of the special research cluster „Medical navigation and robotics“ (SPP 1124) funded by the Deutsche Forschungsgemeinschaft (DFG), performed in cooperation with the Universitäts-HNO-Klinik (Abt. HNO-Heilkunde) in Heidelberg/Germany. Further information can be found at *http://ai3.inf.uni-bayreuth.de/projects/ronaf/*.

References

[1] I. Boesnach, M. Hahn, J. Moldenhauer, Th. Beth, U. Spetzger, “Analysis of Drill Sound in Spine Surgery”, MRNV (Medical Robotics, Navigation and Visualisation), Remagen/Germany, 2004.

[2] D. Engel, J. Raczkowsky, H. Wörn, “Sensor-aided Milling with a Surgical Robot System”, CARS (Computer Assisted Radiology and Surgery), H.U. Lemke et al. (eds.), Paris/France, 2002.

[3] Ph. A. Federspil, U. W. Geisthoff, D. Henrich, P. K. Plinkert, “Development of the First Force-Controlled Robot for Otoneurosurgery”, The Laryngoscope, Lippincott Williams & Wilkins, Inc., Philadelphia, 2003.

[4] F. J. Harris, “On the Use of Windows for the Harmonic Analysis with the Discrete Fourier Transform”, Proceedings of the IEEE, Vol. 66, No. 1, Jan 1978.

[5] D. Henrich, “Robotergestütztes Fräsen an der lateralen Schädelbasis: Kraft-basierte lokale Navigation bei der Implantatbettanlage”, Robotik, Ludwigsburg/Germany, 2002.

[6] D. Henrich, Ph. Stolka, “Principles of Navigation in Surgical Robotics”, MRNV (Medical Robotics, Navigation and Visualisation), Remagen/Germany, 2004.

[7] V. Braun et al, „Navigated Control – Ein neuer Ansatz für das exakte Fräsen“, CURAC (Computer- und Roboterassistierte Chirurgie), Leipzig/Germany, 2002.

[8] D. Malthan et al., “Improvement of Computer- and Robot-Assisted Surgery at the Lateral Skull Base by Sensory Feedback”, MRNV (Medical Robotics, Navigation and Visualisation), Remagen/Germany, 2004.

[9] R. Rojas, “Theorie der neuronalen Netze”, Springer Verlag Berlin Heidelberg, ISBN 3-540-56353-9, 1993.

[10] J. Rosen, B. Hannaford, C. G. Richards, M. N. Sinnan, “Markov Modeling of Minimally Invasive Surgery Based on Tool/Tissue Interaction and Force/Torque Signatures for Evaluating Surgical Skills”, IEEE Transactions on Biomedical Engineering, Vol. 48, No. 5, May 2001.

[11] Ph. Stolka, D. Henrich, „Building Local Maps in Surgical Robotics“, IROS (IEEE/RSJ International Conference on Intelligent Robots and Systems), Edmonton/Canada, 2005.

[12] Ph. Stolka, D. Henrich, „Improving Navigation Precision of Milling Operations in Surgical Robotics“, IROS (IEEE/RSJ International Conference on Intelligent Robots and Systems), Beijing/China, 2006.

[13] Ph. Stolka, D. Henrich, „Robot-Based 3D Ultrasound Scanning and Registration with Infrared Navigation Support“, ICRA (IEEE International Conference on Robotics and Automation), Rome/Italy, 2007.

[14] S. H. Tretbar, Ph.A. Federspil, P.K. Plinkert, “Improved ultrasound based navigation for robotic drilling at the lateral skull base”, CARS (Computer Assisted Radiology and Surgery), Chicago/USA, 2004.