Environmental Codes for Autonomous Position Determination

Kamen Kanev, Hirotaka Kato
Research Institute of Electronics
Shizuoka University
Hamamatsu, Japan
kanev@rie.shizuoka.ac.jp, gs10011@s.inf.shizuoka.ac.jp

Reneta Barneva, Zhuojun Fu
Department of Computer and Information Sciences
State University of New York
Fredonia, USA
barneva@cs.fredonia.edu, fu0317@fredonia.edu

Shigeo Kimura
Institute of Nature and Environmental Technology
Kanazawa University
Kanazawa, Japan
skimura@t.kanazawa-u.ac.jp

Abstract— **This work focuses on pervasive environmental codes that serve as an interface to augmented vision devices and provide support for localization and automated navigation. We begin with a concise overview of methods for automating the localization of humans and autonomous agents including mobile robots. Automated localization is based on mapping where positions, orientations, and other localization parameters are determined either on a plane or in a three-dimensional space. While various devices such as sonar and ultrasound locators, laser scanners, visible light and infrared cameras, etc. are considered for gathering of the necessary mapping information the focus of our work is on the innovative system for environmental semantic encoding that we have developed. In this system, we have successfully implemented semantic surfaces with embedded marking which provide additional information alone, separately and independently from all the visual features and properties of the surrounding physical surfaces.**

Keywords - position determination; mapping; semantic surfaces; navigation; SLAM; surface encoding; CLUSPI.

I. INTRODUCTION

Human activities as well as activities of autonomous agents and mobile robots are essentially connected to the surrounding environment. An interface layer that is responsible for the interactions taking place within such activities could therefore be established and maintained. In this work we attempt to establish such a layer by introducing a special type of environmental semantic encoding that is implemented with *ubiquitous semantic surfaces* [7]. An introduction to semantic surfaces and details about our environmental semantic encoding approach are given in Section II. In this section, we continue with an introduction to a more general localization and mapping. For illustration of autonomous position determination we will refer to robotic mapping, e.g., the building of a map of a local environment surrounding a robot.

The simultaneous robot orientation and map building is an estimation problem known as SLAM (Simultaneous Localization and Mapping) [12]. SLAM is an essential capability for any autonomous agent or a mobile robot traveling in unknown environments where globally accurate position data is not available [9, 10]. High uncertainty often exists in such environments so the capability to map them is essential in order to allow a robot to be deployed with a minimal infrastructure.

A variety of sensors such as sonar and ultrasound, laser, visible light and infrared as well as digital cameras are commonly used to gather information and to “capture” the local environment. For now, we assume that maps are static, that is, no relative movement of environment features exist and no intermittent changes in such features are allowed. However, despite this assumption, the uncertainties of the robot state can become arbitrarily large [12]. This stresses upon the necessity of more reliable tracking of the exact positions of landmarks and other environmental features that can reduce the uncertainty of the robot state. The environmental codes that we consider in this work are specifically designed for environment enhancements facilitating such tracking and potential use in SLAM and FastSLAM [10, 11].

FastSLAM is an efficient SLAM algorithm, which decomposes the SLAM problem into two, namely a robot localization problem and a collection of landmark estimation problems. It uses a modified particle filter for assessing the posterior over robot paths instead of the extended Kalman filter (EKF), which reduces the running time from linear to logarithmical in respect to the number of landmarks [10, 11]. Another advantage of FastSLAM over the EFK is its multi-hypothesis data association. Since each FastSLAM particle represents a specific robot path, data association decision can be performed on a per-particle basis. High weights will be assigned to correct data association in terms of high chances for future resampling. If the data association is incorrect, the weight will be low and then the association will be removed later.

In Section II, we describe the semantic encoding method that we have developed and provide details about the implemented software system that employs it. In Section III, we present experimental measurements and validation of the developed system. In Section IV, we elaborate on the approach of 3D mapping with semantics and SLAM. Finally, we conclude in Section V with some plans for further work.

II. SEMANTIC SURFACES

In linguistics, *semantics* refers to the meaning carried by a language. In the case of mapping, we define *semantic surface* as a surface with embedded marking which provides additional information alone, separately and independently from all the visual features and properties of the surface [7]. Such an embedded surface marking can be implemented, for example, as a (dense) grid of landmarks with links to nodes with specific meaning. We believe that the semantic surfaces as defined here, although applicable outdoors, are best employed in indoor environment and for small scale navigation. In the following sections we discuss in more detail various methods and approaches for surface marking and creation of semantic surfaces that are suitable for position determination of autonomous agents and mobile robots.

A. Environmental semantic encoding

In our daily life, we often encounter different codes embedded or attached to various products and equipment. Such codes usually carry digital information, specific to the artifact they are associated with, and are used for its tagging and consequent identification. Typical examples are barcodes employed in shops for merchandise management and control. Although most of the barcodes currently in use are still one-dimensional, more advanced two-dimensional barcodes such as QR and Datamatrix codes, Color codes, and others are being adopted. In addition to their business use, both old-type and newer codes are becoming more accessible for ordinary people through various gadgets, such as camera enabled mobile phones and smart phones, etc. Typical applications involve scanning of a code by a mobile device camera, decoding and extracting information embedded in it, and providing related feedback to the user. This is a very powerful application model that allows on-time delivery of up-to-date context-dependent information, dynamically adjustable to meet the specific need of the current user. It takes advantage of the continuous 3G/WiFi connectivity of the current mobile devices, of their ability to take snapshots of the surrounding environment by a simple press of a button, and of the user profiles containing usage history and preferences information stored on the device.

In a similar way, autonomous agents such as mobile robots can also take advantage of these codes. Context dependent information provided in this way could be further extended and even better targeted if localization information is available. Standard GPS-based tracking and position determination, however, is generally not sufficient for reliable precise localization indoors. To address this issue, we introduce here our research on ubiquitous environmental digital codes for global positioning and navigation [4]. These codes are specifically designed to seamlessly blend in the surrounding environment by ether being practically undetectable by naked human eye due to the size and shape of the employed marks or by becoming part of the environmental patterns naturally covering walls and other surrounding surfaces. Methods for generation of unobtrusive surface codes that blend well with existing printed content have been reported in [1,8] and our work on direct embedding of such codes in surfaces and into the bulk of physical objects have been reported in [3].

In this paper, we focus on larger-scale codes that are integrated with various patterns on surrounding surfaces [4]. The codes do not interfere with the look and feel of the surrounding environment and thus do not disturb the humans. Autonomous agents and mobile robots, however, can extract the codes from their surroundings and employ them for localization (position and orientation) determination.

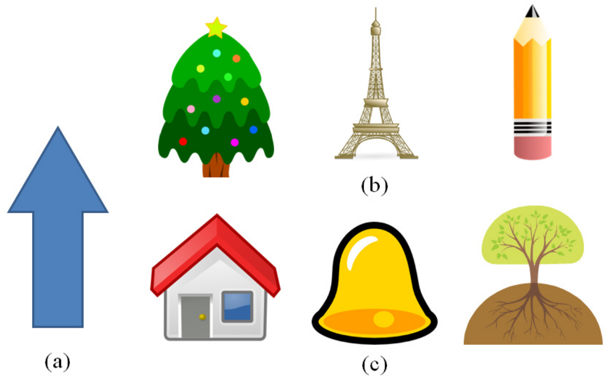

Figure 1. A sample human recognizable arrow (a) and algorithmically recognizable (b) and non-recognizable (c) "A-shaped" objects

There are known approaches addressing interior design patterns [2], where figures with different distinguishable shapes are used for the encoding. As discussed in [2], experimental interior design patterns based on 4-figure and 6-figure digital encoding have been created and consequent figure recognition and decoding test have been conducted. Reported experimental results suggest that better system performance and higher recognition rates would need to be secured before its final adoption.

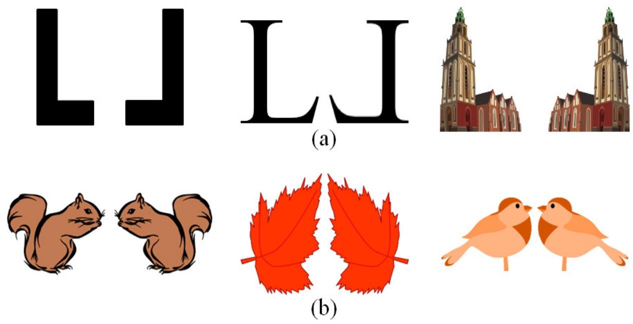

Figure 2. A sample human recognizable (a) and algorithmically recognizable (b) "L-shaped" objects

In contrast to the above discussed interior design patterns, the encoding scheme that we employ in our work uses non-shape based figure discrimination. Its computational complexity is significantly lower than the F-descriptor based method employed in [2]. Our method can work with single figure type patterns where no figure type discrimination is required and digitally encoded data is simply related to the figure rotations. In this way various, differently shaped and sized graphical objects can serve as figures in our patterns where only their orientation matter. For reliable determination of the figure rotational angles, however, graphical objects with easily distinguishable main or up direction should be used.

The arrow shaped figure shown in Fig. 1(a) is a good example of direction recognition by humans, since its pointing direction is unmistakable. But it is also a good example of a graphical object with algorithmically-easy determinable direction. Simple methods to determine the arrow direction would be to find its mass center and to connect it to the most distant arrow point, to calculate and use moments of higher order, etc. Since the figure orientation will be determined algorithmically, it does not necessarily have to match the human judgment. We can, therefore, say that any figure with easily determinable main direction by the employed algorithms could be considered an arrow or an A-shaped object. To humans it may not look like or even remotely resemble an arrow but it can be treated as an A-shaped object as long as its main direction is well determined by the employed software. Some examples of objects that can and cannot be considered as A-shaped are given in Fig. 1(b) and Fig. 1(c), respectively.

A simple extension of this idea would be considering L-shaped objects. Same as for A-shaped objects, the main direction of the L-shaped objects should be well defined and easily determined algorithmically. In addition to this, L-shaped objects and their mirror images should be easily distinguishable. For illustration, examples of human recognizable L-shapes and their mirror images are shown in Fig. 2(a). Further examples of algorithmically recognizable L-shaped objects and their respective mirror images are shown in Fig. 2(b).

B. Software system

We have developed an experimental software system (Fig. 3) for basic figure management, for design, generation, and printing of environmental semantic codes, and for code extraction, analysis, and consequent localization in semantically encoded environments.

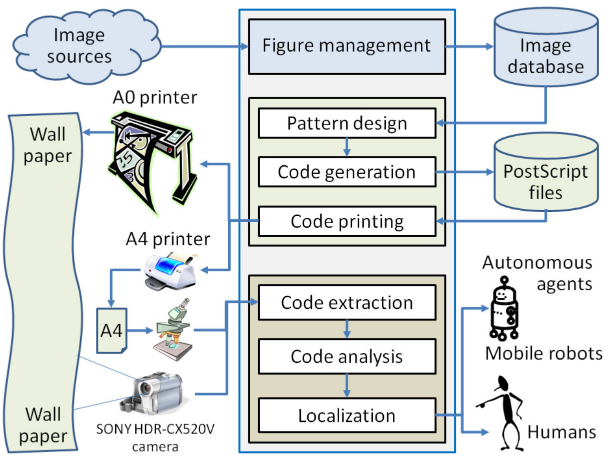

Figure 3. A schematic representation of the developed experiment software system

The system is quite flexible and allows creation of a vast variety of environmental encoding patterns. Depending on the choice of code components or figures, impressive wallpaper patterns close to real artworks could be created.

The elementary figures used by our system are initially organized in a figure database. Employed figures are essentially images in different formats which fall into one of the following two categories:

- raster images: these images are stored in files with extensions bmp, tiff, gif, jpeg, etc. that store arrays of image pixels, and
- vector images: these images are stored in files with extension svg, wmf, etc. and contain drawing information.

Raster images do not scale well, i.e., they get jagged under large magnification, which makes it difficult to use the same designed pattern for differently sized wallpaper codes. Vector images on the other hand scale very well although some standard image processing techniques cannot be directly applied to them.

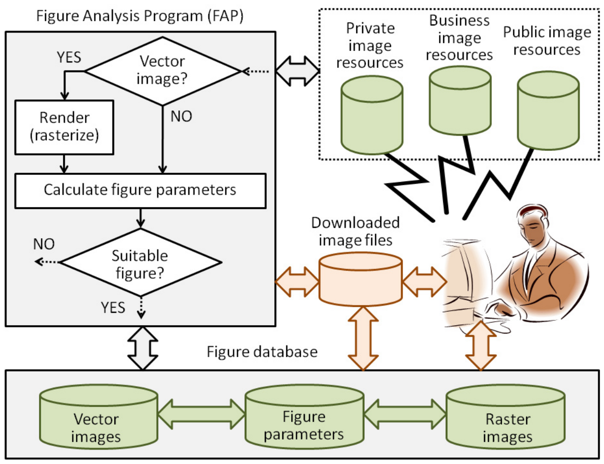

Figure 4. Figure management component of the system

In our system, we support both types of images and we always attempt to employ the most suitable type for the task in question. For example, required figure characteristics such as center of gravity, mass distribution, moments, etc. are calculated based on a raster image file of the figure. If such file is not available, it is automatically generated by scan-conversion (rendering) of the corresponding vector file. In the designed patterns, on the other hand, we use vector representations of the employed figures whenever available.

1) Figure management

The process of building the figure database is schematically represented in Fig. 4. First copyright free publicly available, commercial, and private image sources are searched and images that are judged as potentially suitable for environmental semantic encoding are fed into the Figure Analysis Program (FAP). The program accepts all most popular image formats for both raster and vector images. Raster images supplied to the program are immediately evaluated [4] and those found unsuitable are rejected. For positively assessed raster images an attempt is made to locate and download a vector version of the same figure if available. If a vector image is supplied to the program FAP first renders it and then proceeds with its evaluation. For all positively evaluated figures FAP calculates and stores the necessary figure parameters in the database. Once the database is populated with sufficient number of suitable figures the code generation process can be initiated.

2) *Pattern design*

At this stage (Fig. 5), the visual appearance of the environmental semantic code is chosen. As earlier discussed, figures of various shapes, sizes and colors can be employed in the environmental code. Depending on the fundamental code parameters the pattern designer may or may not be limited to certain types of figures, e.g., A-shaped, L-shaped, etc. This, however, is completely transparent to the pattern designer since the software will allow him to use only suitable figures in his design [4]. Figures are arranged on a predetermined grid where the selection of figures for the grid positions is controlled by the pattern designer. Note that other placement parameters of the figures such as figure rotations and small displacements from the grid positions are controlled by the encoding engine. The pattern designer can arrange figures interactively, programmatically, and by combining the previous two methods. This way a fine artistic arrangement can be done interactively and then programmatically replicated to cover large areas.

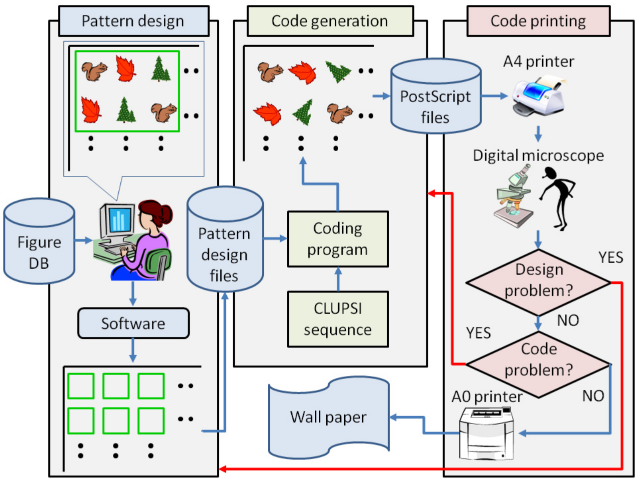

Figure 5. Pattern design, code generation, and code printing components of the system

3) *Code generation*

At this stage (Fig. 5) the Pattern Encoding Engine (PEE) is invoked to calculate the figure rotations that will carry the localization and other semantic information. Some encoding options may also introduce small relative displacement of the figures in respect to their predefined grid positions. Our PEE is based on the Cluster Pattern Interface (CLUSPI) class of codes [5,6]. These codes have no physical margins, blocks, or any other features that may fragment their appearance. CLUSPI encoded semantic surfaces and environmental encoding based on them is a powerful mechanism for creating ubiquitous environments where humans (through specialized devices), autonomous agents, and mobile robots can reliably determine their positions, orientations, and also obtain environment related semantic information.

4) *Code printing*

The environmental semantic codes created by our software are stored in PostScript files that can be directly send to a printing device (Fig. 5). If all images (figures) embedded in the file are of vector type, the code from the file looks well when printed both on small A4 sized and on large A0 sized sheets. This feature of the code is extremely convenient for experimental work. Proofs for visual inspection of the developed codes, for example, can be conveniently printed on A4 sheets. Same proofs can then be used for code consistency checking and structural verification. While the real code is to be used as wall paper (A0 printouts) by using the A4 proof and a digital microscope we are able to verify all essential properties of the printed code.

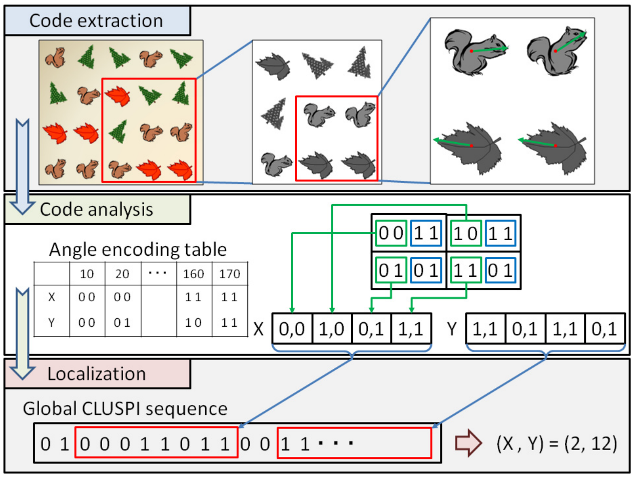

Figure 6. Code extraction, analysis, and localization components of the system

5) *Code extraction*

Typical applications of CLUSPI codes are for digital enhancement of printed materials. In such cases the code reader can be easily tuned to retain a predetermined distance from the encoded surface, e.g., by installing a spacer, which greatly simplifies the code extraction process. However, for large-scale codes embedded in wall paper, for example, positions and orientations of the camera in respect to the encoded surface may vary significantly. With the change of distance, optical parameters of the camera may need adjustment (depending on the focus depth). To tackle this issue we employ an autofocus camera (SONY HDR-CX520V). Another problem is the changing size of the viewing area and thus the total number of figures that can be analyzed which will be discussed in following works. In all cases each frame of the camera video stream is first converted to grayscale with its brightness and contrast normalized (Fig. 6). Then zones containing figures are identified and the figures are separated from the background through standard image processing techniques [15]. Then the barycentre, the orientation (rotational angle), and other parameters (e.g., L-shape mirroring parameters, etc.) are computed for each extracted figure. The obtained data is finally organized in a 2D array corresponding to the figure arrangement on the surface.

6) Code analysis

The data contained in the 2D array with extracted figure parameters obtained at the previous step is first converted to short sequences of bits. As shown in Fig. 6, an angle derived from the figure orientation within the pattern carries four bits of information (see the angle encoding table). The first two bits are allocated for encoding the X-coordinate and the remaining two ones are used for encoding the Y-coordinate. The bits for each coordinate from the figures forming a virtual block [4] are arranged in two 8-bit sequences, one for each of the coordinates.

7) Localization

Localization information is calculated on the basis of the bit-sequences derived at the previous step (Fig. 6). The CLUSPI scheme encodes coordinates with a global binary sequence having special properties, namely such that any subsequence with a predetermined length (which is a parameter of the code) is unique [1]. Based on that, we can determine the positional coordinates of a set of figures by matching the subsequences that they encode with the global CLUSPI encoding sequence [5,6]. The implementation of this decoding is, of course, more complex and includes redundancy management and error recovery procedures which will be discussed in future works.

III. EXPERIMENTS AND EVALUATION

As shown in Fig. 3 and discussed in Section II, generated environmental semantic codes are first printed on A4-sized paper for visual inspection, code consistency checking, and structural verification.

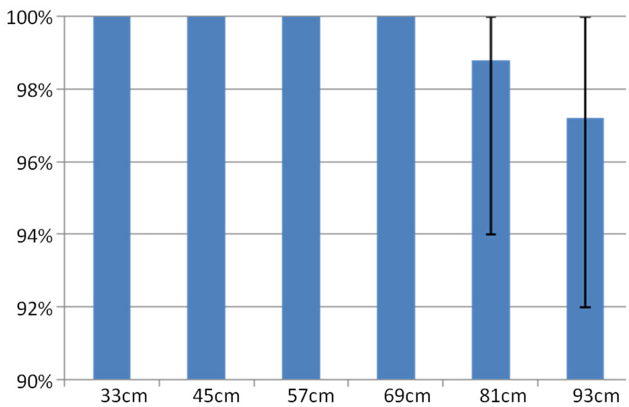

Figure 7. Recognition rate vs. distance (Bars show average; overlaid segments show minimum and maximum values.)

Using such proof pages, we have conducted a range of experiments and measurements of the recognition rates for specific codes at various distances and viewing angles. Obtained results are directly applicable to full-sized environmental encoding patterns by appropriate scaling. The detailed measurement results that we report here are based on a sample coded pattern for position determination with minimal sequences of four figures as shown in Fig. 6. A compact group of four figures (e.g., 2x2) can be viewed with the employed digital microscope from a minimum distance of about 33 cm.

Results for six sets of measurements starting with the minimum distance and incrementing by 12 cm are shown in Fig. 7. At each distance, five spots on the patterned sheet have been randomly chosen and 10 measurements have been done at each spot (50 measurements altogether for each distance). As shown in Fig. 7, no position determination errors have been detected for the distances within the 33-69 cm range and the average success rate remains over 95% for the 69-93 cm range. Above that distance, the recognition rate becomes too low for practical use.

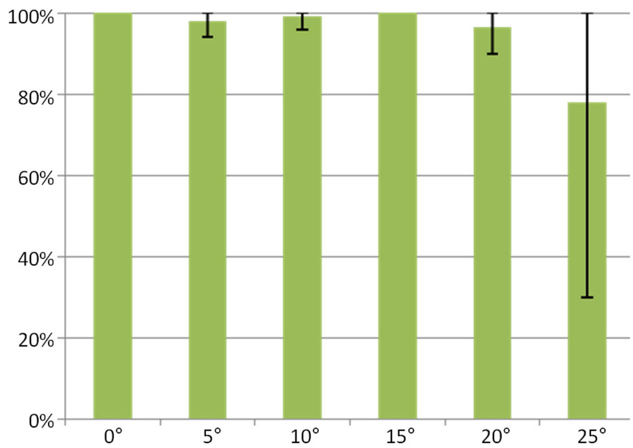

Figure 8. Recognition rate vs. angle (Bars show average; overlaid segments show minimum and maximum values.)

Similarly, results for six sets of measurements at angles from 0° to 25° (in 5° increments) from the vertical are shown in Fig. 8. Again, 10 measurements at 5 random spots have been taken for each of the angles. As shown in Fig. 8 average success rate stays over 95% for all angles up to 20°. Above that angle, recognition rate becomes too low for practical use.

IV. 3D MAPPING WITH SEMANTICS

We consider the pervasive environmental codes in the framework of autonomous position determination as a tool for localization support. The main purpose of a semantic surface in respect to a typical geometric map is to provide “some type of reasoning based on individual entities in the map and/or their classes” [13]. For instance, robots with rescue systems are often designed to assist rescuers to locate victims in various disaster environments such as an earthquake scene. Those systems require reliable maps with specific and detailed object information. In order to identify and properly report the precise localization of injured victims, all available degrees of freedom (e.g., 6DOF) need to be used on top of the available 3D maps. This is an issue that can be addressed by the 6D SLAM method. For real-time applications, however, the time that simple SLAM takes for pose (position and orientation) estimates often becomes a problem. In such situations semantic knowledge could be extracted from the surrounding environment and used to improve the efficiency and to speed up the 6D SLAM method. "A semantic 3D map for mobile robots is a metric map that contains in addition geometric information of 3D data points and assignments of these points to known structures or object classes." [12,13] Semantic knowledge could, therefore, be provided, through semantic maps implemented as semantic surfaces. The process of 3D mapping with semantics could be divided into four major steps:

- 3D scanning by 6D SLAM,
- 3D scene interpretation,
- objects detection and localization, and
- semantic map presentation.

However, "Prior to the mapping, the object database needs to be initialized and filled with object descriptions both for the 2D and 3D representations" [12,13]. Therefore, due to the 3D nature of such semantic maps, all the data need to be rendered before it can be employed in 6D SLAM computations.

Symbolic level robot planning often relies on such rendered semantic maps and takes advantage of the specific background knowledge embedded in them. Extracted semantic knowledge is then used for reasoning about objects or object classes present in the map. As reported in [14] using semantically labeled points results in a speedup with no loss of quality in computing time for matching of two 3D scans.

V. CONCLUSION

In this work, we discussed various approaches for autonomous agent localization identifying their advantages and drawbacks. After a concise review of the existing methods, we drew reader's attention to the benefit of using semantic surfaces consisting of embedded marking with predefined links to specific semantic nodes. Semantic surfaces are especially useful for indoor and small scale navigation where the other methods have some deficiencies.

An experimental software system for environmental coding generation, analysis and processing has been developed providing flexibility and allowing creation of a vast variety of environmental encoding patterns some of which are close to artworks and applicable as decorative wallpaper patterns.

Our intention is using the system for generating semantic surface environments which will be employed in real applications with autonomous agents. In particular, our plans for further work include carrying out thorough experiments to investigate how the recognition will be influenced in an environment where the semantic surfaces will be subjected to wear and tear.

ACKNOWLEDGMENT

This work was funded in part by Cooperative Research Projects at Research Institute of Electronics, Shizuoka University and Grant-in-Aid for Scientific Research (B) 20300269.

REFERENCES

[1] Barneva, R.P., Brimkov, V.E., and Kanev, K., Theoretical issues of cluster pattern interfaces, Combinatorial Image Analysis, Lecture Notes in Computer Science, No 5852, 2009, Springer-Verlag Berlin Heidelberg, pp. 302-315.

[2] Hiyama, A., Saito, S., Tanikawa, T., and Hirose, M., Design flexibility in seamless coded pattern for localization, In Proc. of the 2007 ACM Symp. on Virtual Reality Software and Technology VRST 2007, Newport Beach, California, USA, November 5-7, 2007, pp. 219-220.

[3] Kanev, K., Gnatyuk, P., and Gnatyuk, V., Laser marking in digital encoding of surfaces, Advanced Materials Research, Vol. 222, 2011, pp.78-81.

[4] Kanev, K., Kato, H., and Koroutchev, K., Encoding of surfaces for global positioning and navigation, The Journal of Three Dimensional Images, Vol. 24, No. 3, 2010, pp. 51-57.

[5] Kanev, K. and Kimura, S., Clustering-scheme-encoded interfaces providing orientation feedback, US Patent No 7991191, 2011.

[6] Kanev, K. and Kimura, S., Digital information carrier, JP Patent No 4368373, 2009.

[7] Kanev, K., Mirenkov, N., Brimkov, B., and Dimitrov, K., Semantic surfaces for business applications, Int. Conf. on Software, Services and Semantic Technologies, Sofia, Bulgaria, 2009, pp. 36-43.

[8] Kanev, K. and Orr, T., Enhancing paper documents with direct access to multimedia for more intelligent support of reading, In Proc. of the IEEE Conf. on the Convergence of Technology and Professional Communication, Saratoga Springs, New York, USA, 23-25 Oct. 2006, pp. 84-91.

[9] Milford, M.J., Robot navigation from nature, STAR 41, pp. 1–7.

[10] Montemerlo, M. and Thrun, S., FastSLAM: a scalable method for the simultaneous localization and mapping problem in robotics, Springer, Berlin, Germany, 2007.

[11] Montermerlo, M., Thrun, S., Koller, D., and Wegbreit, B., FastSLAM: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges, In Proc. of IJCA 2003.

[12] Nüchter, A., 3D robotic map: The simultaneous localization and mapping problem, STAR 52, pp.9-27, 2009, pp. 109-172.

[13] Nüchter, A. and Hertzberg, J., Towards semantic maps for mobile robots, Robotics and Autonomous Systems, 56, 2009, pp. 915-926.

[14] Nüchter, A., Wulf, O., Lingemann, K., Hertzberg, J. Wagner, B., and Surmann., H. RoboCup 2005: Robot Soccer World Cup IX 3D Mapping with Semantic Knowledge. Lecture notes in computer science (0302-9743), 4020, p. 335.

[15] O'Gorman, L., Sammon, M.J., Seul, M., Practical algorithms for image analysis: descriptions, examples, programs, and projects, 2nd ed., Cambridge University Press, New York, USA, 2008.