

Energy-Efficient Environment Mapping via Evolutionary Algorithm Optimized Multi-Agent Localization

Ahmed Hallawa

Chair of Integrated Signal Processing Systems
Kopernikusstrae 16
Aachen, Germany
hallawa@ice.rwth-aachen.de

Giovanni Iacca

Chair of Integrated Signal Processing Systems
Kopernikusstrae 16
Aachen, Germany
iacca@ice.rwth-aachen.de

Stephan Schlupkothen

Chair of Integrated Signal Processing Systems
Kopernikusstrae 16
Aachen, Germany
schlupkothen@ice.rwth-aachen.de

Gerd Ascheid

Chair of Integrated Signal Processing Systems
Kopernikusstrae 16
Aachen, Germany
ascheid@ice.rwth-aachen.de

ABSTRACT

Miniature autonomous sensory agents (MASA) can play a profound role in the exploration of hardly accessible unknown environments, thus, impacting many applications such as monitoring of underground infrastructure or exploration for natural resources, e.g. oil and gas, or even human body diagnostic exploration. However, using MASA presents a wide range of challenges due to limitations of the available hardware resources caused by their scaled-down size. Consequently, these agents are kinetically passive, i.e. they cannot be guided through the environment. Furthermore, their communication range and rate is limited, which affects the quality of localization and, consequently, mapping. In addition, conducting real-time localization and mapping is not possible. As a result, Simultaneous Localization and Mapping (SLAM) techniques are not suitable and a new problem definition is needed. In this paper we introduce what we dub as the Centralized Offline Localization And Mapping (COLAM) problem, highlighting its key elements, then we present a model to solve it. In this model evolutionary algorithms (EAs) are used to optimize agents' resources off-line for an energy-efficient environment mapping. Furthermore, we illustrate a modified version of Vietoris-Rips Complex we dub as Trajectory Incorporated Vietoris-Rips (TIVR) complex as a tool to conduct mapping. Finally, we project the proposed model on real experiments and present results.

CCS CONCEPTS

•Computing methodologies → Evolvable hardware; •Computer systems organization → Sensor networks;

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '17 Companion, Berlin, Germany

© 2017 ACM. 978-1-4503-4939-0/17/07...\$15.00

DOI: <http://dx.doi.org/10.1145/3067695.3084201>

KEYWORDS

Evolutionary computation, localization, wireless sensors, swarm intelligence

ACM Reference format:

Ahmed Hallawa, Stephan Schlupkothen, Giovanni Iacca, and Gerd Ascheid. 2017. Energy-Efficient Environment Mapping via Evolutionary Algorithm Optimized Multi-Agent Localization. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017*, 6 pages.

DOI: <http://dx.doi.org/10.1145/3067695.3084201>

1 INTRODUCTION

One of the powerful attributes associated with Evolutionary Algorithms (EAs) is the fact they can be applied straight forwardly to many optimization and search problems without, or with limited, a priori knowledge of the problem under investigation. In the context of exploration of unknown environments, localization and mapping processes are challenging to design due to the lack of mathematical representation associating all variables with the defined objectives. Furthermore, adding constraints on the resources of the agents conducting exploration, which is the case when using MASA, emphasises the need of an optimization technique that can conduct the optimization process with limited a priori knowledge, which presents EAs as an adequate solution. Consequently, the presented solution in this work uses EAs at the core of its scheme, however, to motivate the presented work, in this section potential applications using MASA for exploration of unknown environment are highlighted. This is followed by a problem description and paper organization description.

1.1 Motivation

MASA offer a robust solution to many applications such as monitoring underground infrastructure. In this application, it is critical to monitor leaks and ruptures of the infrastructure, e.g. underground water supplies, oil and gas pipes. Furthermore, the system needs to capture blockage and overflow. Although literature is rich with solutions to this problem, these solutions rely, in many cases, on installation of static sensors [11] which is costly and not effective as the infrastructure network is usually huge and/or not accessible for such installations.

Another application for MASA is the exploration for resources such as oil in *cold heavy oil production with sand* (CHOPS) [9] [12]. In this application millimeter-sized sensor agents are injected via a pump and extracted to produce a profile of the potential oil distribution, in addition to temperature and pressure profiles. It is important to highlight that large sensors (> mm-sized) can not be used as they will be shattered by the injection pump.

Furthermore, many medical applications can rely on MASA. Recently, literature proposed the use of micro-meter sized robots in human body [6]. In these applications MASA would be suitable, specially for diagnostic purposes, as controlling propulsion of these mirco-robots still presents many challenges. As mentioned earlier, SLAM is not suitable to be used in such applications as in MASA are kinetically passive and have very limited available resources, thus a new problem definition is needed. Next the COLAM problem definition is discussed.

1.2 Problem Description

The problem can be defined as follows: Given a set of observations O_i from different sensors S_j embedded on a set of N sensory agents over discrete time t , with $F(i, S_j, t)$ as the function describing resource consumption to capture these observations, COLAM problem is estimating the positions of the N agents $\forall t$ and finding the map that describes the unknown environment under investigation G_m with the least possible resource consumption for each agent and with the least number of agents N . It is worth mentioning that in case the unknown environment is hostile, the problem is extended to include loosing M agents from the N injected ones in the environment which changes the localization objective of the problem to be the estimation of the positions of $N - M$ extracted agents instead of N .

1.3 Paper Organization

In the next section, a background on environment mapping is presented, section 3 describes the proposed scheme to solve the defined COLAM problem. Furthermore, section 4 is dedicated to define case studies used to test the proposed model, and finally in sections 5 and 6, results and conclusion are presented respectively.

2 BACKGROUND

2.1 Mapping via rips-complex

Mapping environments is one of the key objectives in COLAM (Section 1.2). In this context, mapping is referred to as the ability to reconstruct the environment topology G_m from collective sensor readings O_i of extracted N agents.

In this context, it is paramount in the abstraction of the environment to embed all available knowledge —such as the estimated agent positions and trajectories— and link it to physical environment features such as the presence of obstacles or the general layout of the environment. This will also allow the accumulation of knowledge granted from conducting simulations and real experiments. Furthermore, in order to facilitate the use of EA to *evolve* the environment model and converge to the “real” environment, the environment must be abstracted in such a way that its “genotype” can be easily representation and evaluated in comparison with other environments.

2.2 Vietoris Rips Complex

VR complex is an *abstract simplicial complex*, and its construction can be divided into two main steps: geometric and topological. Figure 2 illustrates these two steps starting from a cloud of points, which in COLAM could be the positions of all agents at all times. The geometric step is essentially the establishment of a VR neighbourhood graph $G_\epsilon(s)$. The topological step is called VR extension graph $V_\epsilon(s)$. To understand these two steps, it is needed first to understand what is an abstract simplicial complex, which is introduced in the next subsection.

2.2.1 Abstract Simplicial Complex. Given a set of vertices $S = \{s_1, \dots, s_n\}$, a simplicial complex Δ over S is a subset from S with the property that $s_i \in \Delta \forall i$ and if $\Lambda \in \Delta$ then all subsets of Λ are also in Δ . An element of Δ is called a face of Δ , and the dimension of a face Λ of Δ is defined as $|\Lambda| - 1$ where $|\Lambda|$ is the number of vertices of Λ [2].

Dimensions 0 and 1 of the faces are called vertices and edges, respectively, furthermore, $\dim \phi = -1$.

The maximal faces of Δ under inclusion are called facets of Δ . The dimension of the simplicial complex Δ is the maximal dimension of its facets; in other words:

$$\dim \Delta = \max\{\dim \Lambda \mid \Lambda \in \Delta\} \quad (1)$$

We denote the simplicial complex Δ with facets $\Lambda_1, \dots, \Lambda_q$ by

$$\Delta = \langle \Lambda_1 \dots \Lambda_q \rangle \quad (2)$$

and we call $\{\Lambda_1, \dots, \Lambda_n\}$ the *facet set* of Δ .

Finally, if we replace each simplex in a simplicial complex with the set of its vertices, we get a system of subsets of the vertex set. In doing so we neglect the geometry of the simplices, which allows us to focus on the combinatorial structure. This is what is called an abstract simplicial complex.

2.2.2 Vietoris Rips Expansion. As mentioned in the introduction of this section, a VR complex can be divided into two steps, the definition of the first step being as follows. Given $S \subseteq R^d$ and scale $\epsilon \in R$, the VR neighborhood graph is a neighborhood graph $(G_\epsilon(S), w)$, where [5]:

$$G_\epsilon(S) = (S, E_\epsilon(S)) \quad (3)$$

$$E_\epsilon(S) = \{\{u, v\} \mid d(u, v) \leq \epsilon, u \neq v \in S\} \quad (4)$$

$$G_\epsilon(S) = (S, E_\epsilon(S)) \quad (5)$$

Conducting this step computationally is not exhaustive, and there are many well-established algorithms capable of implementing this neighbourhood graph.

The second step, which is the establishment of the complex itself, can be summed up as follows. Given a neighbourhood graph (G, w) , $(V(G), w)$ can be established if all edges of a simplex Δ are in G , then Δ is in $V(G)$. For $G = (V, E)$ [15]:

$$V(G) = V \cup E \cup \{\Delta \mid (\Delta, 2) \subseteq E\} \quad (6)$$

and for $\Delta \in V(G)$

$$w(\Delta) = \begin{cases} 0, & \text{for } \Delta = \{v\}, v \in V \\ w\{(u, v)\}, & \text{for } \Delta = \{u, v\} \in E \\ \max w(\tau), & \text{otherwise.} \end{cases} \quad (7)$$

3 METHODOLOGY

3.1 System Layout

Detailed overview of the system is laid out Figure 1. The system has three phases, initialization phase, virtual loop phase and a real environment phase. The initialization phase starts with an initial *real* run (step i). This leads to the initialization of motion model estimation (details of this is in section 3.3). The output of this process combined with *knowledge base*, where previously solved problems using this system are stored, to create an initial simulation environment that can be then used in the virtual loop. The virtual loop represents a typical EA loop where minimization of the number of used agents and the consumed resources used to conduct localization is executed while ensuring adequate localization and consequently mapping.

3.2 Virtual loop

the virtual loop is the core of the development process of the agent by optimizing all tunable parameters available in the agent design by achieving localization and mapping with the least possible resources.

- (1) Agents EA optimization: A generation of agents is produced using a multi objective evolutionary algorithm. The objective is to find the least number of agents and tune the agent's design parameters which would consume the least resources possible while conducting an adequate localization and, consequently, mapping. The objective function is then minimization of consumed resources and maximization of mapping quality.
- (2) Simulation: Each generation produced from the EA module is the evaluated on the simulation environment. This environment is generated in the initialization phase.
- (3) Localization: Finding estimates to agent's position off-line.
- (4) Mapping: Using the localization output a map of the simulation environment is generated.
- (5) Compare: A comparison between the reconstructed environment and the simulation environment is conducted to quantify the quality of the mapping process and to feed the output to the EA module for further selection.
- (6) Stop criterion: Is the stopping criteria to the EA process, this can be an upper limit of the computational resources used or reaching a predefined mapping quality with an upper limit to resource consumption.

3.3 Localization

To enable the localization and hence also the environment mapping, the agents are equipped with ultra-sonic transceivers with enable e.g. two-way time of flight measurements. Thereby, the agents take measurements which will—after the extraction of the agents from the environment—be used by the centralized processing unit to obtain the agent positions. For the sake of simplicity, and because

the time of flight measurements can easily be converted into distance information, it is assumed that the distance measurements are subject to additive Gaussian noise:

$$\tilde{d}_{ij,t} = d_{ij,t} + n_{ij}, \quad (8)$$

where $d_{ij,t} = \|\mathbf{p}_{i,t} - \mathbf{p}_{j,t}\|_2$ denotes the true distance between the agents i and j at time step t , which have respectively the Cartesian positions $\mathbf{p}_{i,t}$ and $\mathbf{p}_{j,t}$. Moreover, $n_{ij} \sim \mathcal{N}(0, \sigma_{ij}^2)$ denotes the additive and independently distributed measurement noise. Therewith, the random variable describing the noisy distance measurement is denoted as $\tilde{d}_{ij,t}$. This measurement noise assumption is common and can be found e.g. in [1, 4, 13, 14].

Due to the COLAM approach, where the localization is being conducted jointly for all agents and considering all measurements, an *infinite horizon* approach is used, which is briefly summarized in the following. The principle thereof can be found e.g. in [8, 10].

3.3.1 Infinite Horizon Localization. Considered in the following are N agents in the two-dimensional space which have positions $\mathbf{p}_{i,t}, i = 1, \dots, N$ at time step t . The state vector for each agent i at time step t is denoted by $\mathbf{s}_{i,t}$ and is used to describe also the motion of the agent. The precise definition of the state vector components, such as velocity, is given in Section 3.3.2.

The motion and measurements of the passive agents can therewith be described by

$$\mathbf{s}_{i,t+1} = f(\mathbf{s}_{i,t}, \mathbf{v}_{i,t}) \quad (9)$$

$$\mathbf{y}_i = h(\mathbf{s}_{i,t}, \mathbf{w}_{i,t}), \quad (10)$$

where $f(\cdot)$ is called the state evolution or motion model function and $\mathbf{v}_{i,t}$ is the process noise that describes e.g. motion model inaccuracy. Function $h(\cdot)$ describes the measurements of the agents which are subject to measurement noise $\mathbf{w}_{i,t}$.

In the following, it is assumed that

$$\mathbf{s}_{i,t+1} = f(\mathbf{s}_{i,t}) + \mathbf{v}_{i,t}, \quad (11)$$

$$\mathbf{y}_i = h(\mathbf{s}_{i,t}) + \mathbf{w}_{i,t} \quad (12)$$

where $\mathbf{v}_{i,t} \sim \mathcal{N}(0, P)$ and $\mathbf{w}_{i,t} \sim \mathcal{N}(0, Q)$, with $Q = \text{diag}(\dots, \sigma_{ij}^2, \dots)$ in accordance with (8).

Therewith, the infinite horizon maximum a posteriori estimation problem can be given as, [10]:

$$\arg \max_{\{\mathbf{s}_{i,t}\}_{i=1, \dots, N}^{t=1, \dots, \infty}} p \left(\{\mathbf{s}_{i,t}\}_{i=1, \dots, N}^{t=1, \dots, \infty} \mid \{\mathbf{y}_{i,t}\}_{i=1, \dots, N}^{t=1, \dots, \infty} \right), \quad (13)$$

where $p(\cdot)$ denotes the probability density function. (13) can be further simplified to:

$$\arg \max_{\{\mathbf{s}_{i,t}\}_{i=1, \dots, N}^{t=1, \dots, \infty}} \sum_{t=1}^{\infty} \sum_{(i,j) \in \mathcal{E}} \log p_{d_{ij}}(d_{ij}(\mathbf{p}_i, \mathbf{p}_j) \mid \tilde{d}_{ij}) + \sum_{t=1}^{\infty} \sum_{i=1}^N \log p_{s_i}(\mathbf{s}_{i,t+1} \mid \mathbf{s}_{i,t}) + \sum_{i=1}^N \log p_{s_{i,0}}(\mathbf{s}_{i,0}), \quad (14)$$

where $\mathcal{E} = \{(i, j) \in \{1, \dots, N\}^2 \mid \exists \tilde{d}_{ij}, i \neq j\}$ denotes the set of measurements.

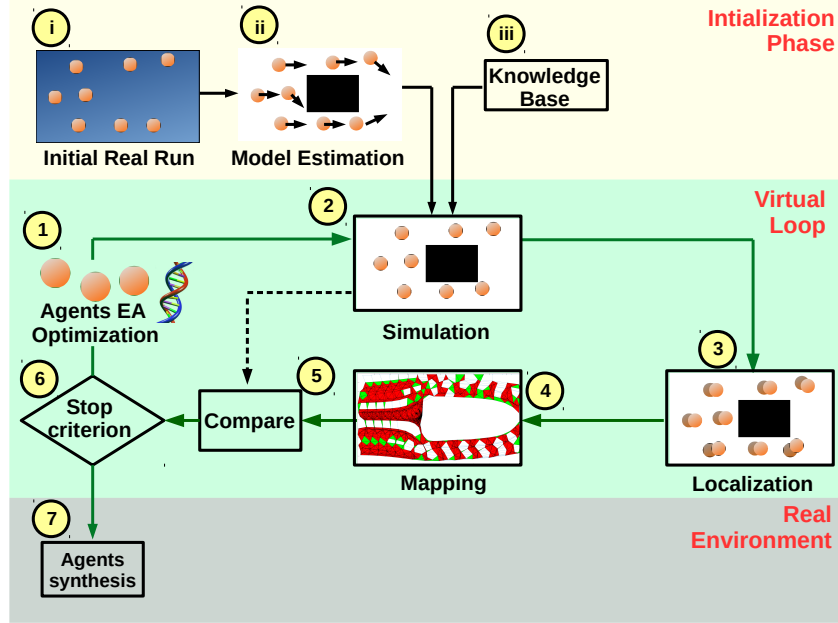


Figure 1: System Layout

Due to the Gaussianity assumption of $\mathbf{v}_{i,t}$ and $\mathbf{w}_{i,t}$ we obtain finally:

$$\arg \min_{\{s_{i,t}\}_{i=1,\dots,N}^{t=1,\dots,\infty}} \sum_{t=1}^{\infty} \sum_{(i,j) \in \mathcal{E}} \frac{(\tilde{d}_{ij,t} - \|\mathbf{p}_{i,t} - \mathbf{p}_{j,t}\|_2)^2}{2\sigma_{ij}^2} + \sum_{t=1}^{\infty} \sum_{i=1}^N \frac{1}{2} (\mathbf{s}_{i,t+1} - f(\mathbf{s}_{i,t}))^T \mathbf{P}^{-1} (\mathbf{s}_{i,t+1} - f(\mathbf{s}_{i,t})) + \sum_{i=1}^N \frac{1}{2} (\mathbf{s}_{i,0} - \hat{\mathbf{s}}_{i,0})^T \mathbf{P}_0^{-1} (\mathbf{s}_{i,0} - \hat{\mathbf{s}}_{i,0}), \quad (15)$$

where \mathbf{P}_0 denotes the covariance matrix of the initial state assumption for $t = 0$.

3.3.2 Agent Motion Model. To localize moving agents, assumptions regarding the motion model must be made. In this work, we assumed that the motion of the agents can be described by the following discrete-time model, [7]:

$$\mathbf{s}_{i,t+1} = \begin{bmatrix} x_{i,t+1} & y_{i,t+1} & v_{i,t+1} & \phi_{i,t+1} & \omega_{i,t+1} \end{bmatrix}^T \quad (16)$$

$$= \begin{bmatrix} x_{i,k} + \frac{2v_{i,k}}{\omega_{i,k}} \sin(\omega_{i,k}T/2) \cos(\phi_{i,k} + \omega_{i,k}T/2) \\ y_{i,k} + \frac{2v_{i,k}}{\omega_{i,k}} \sin(\omega_{i,k}T/2) \sin(\phi_{i,k} + \omega_{i,k}T/2) \\ v_{i,k} \\ \phi_{i,k} + \omega_{i,k}T \\ \omega_{i,k} \end{bmatrix} + \mathbf{v}_{i,t}, \quad (17)$$

where the process noise $\mathbf{v}_{i,t} \sim \mathcal{N}(\mathbf{0}, \mathbf{P})$ has the covariance matrix

$$\mathbf{P} = \text{blkdiag} \left[0 \quad 0 \quad T^2 \sigma_v^2 \quad \sigma_\omega^2 \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T^2 \end{bmatrix} \right], \quad (18)$$

and where T denotes the sampling period. In (17), $x_{i,t}$ and $y_{i,t}$ denote respectively the x and y position of agent i . The speed $v_{i,t}$,

heading angle $\phi_{i,t}$ and turn rate $\omega_{i,t}$ are defined in continuous time by $v = \sqrt{(dx/dt)^2 + (dy/dt)^2}$, $\phi = \text{atan}((dy/dt)/(dx/dt))$, $\omega = d\phi/dt$, [7].

3.4 Mapping

In this section, a proposed environment abstraction technique we dubbed as *Trajectory Incorporated Vietoris Rips (TIVR) complex* is introduced, this technique is presented to be used in the mapping process in step 4 of the virtual loop.

A Trajectory Incorporated Vietoris-Rips (TIVR) complex is simply a VR complex that in addition to the set edges $E_\epsilon(S)$ adds a new set T of edges. T consists of edges set between any agent in any position at time t_i and its position at time t_{i+1} for all i . This is done even if that edge doesn't comply with equation 4. To sum-up the benefits of using TIVR complex:

- It has the ability to accumulate simulation and real runs knowledge for both positions and trajectories.
- It has the ability to project from the given cloud of points and trajectories a quasi-probabilistic map with zones reflecting the probability of existence of an obstacle.
- Due to its structure, it can be straightforwardly integrated into the EA and establishing a genotype-phenotype mapping can be easily done.
- Computationally, it can be obtained with wide range of methods such as inductive or incremental.

3.5 Real environment

In this phase, agents are synthesized to perform a real run on the environment with the optimized agents. This may lead to new observations that can enhance the simulation environment used via the knowledge base (step iii in the initialization phase) and

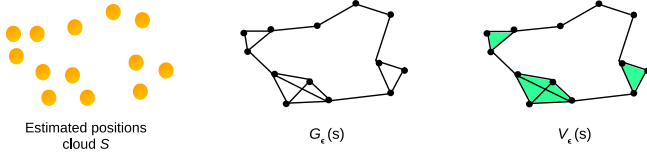


Figure 2: VR complex: an example [5]

consequently improve the optimization process of the agent design in later runs.

4 CASE STUDY

To test the proposed scheme, we project it on a case study that can be described as follows:

- (1) A group of agents are to be injected to four different unknown environments.
- (2) After their extraction, and from the set of observed sensed data, it is required to conduct localization and mapping as stated in the COLAM problem definition.
- (3) Each agent has two possible off-line tunable parameters: communication rate and communication power.
- (4) Agent power, which is directly associated with these two parameters, is the resource that needs to be minimized while achieving the needed localization and mapping.
- (5) As defined in the COLAM problem definition, the number of agents used in the mapping processes also needs to be minimized.

As stated in (1), the case study includes four unknown maps. Each is $200\text{ m} \times 100\text{ m}$, and can be described as follows: a map with a single obstacle located in the centre of the map, a map with two consecutive obstacles located in the centre of the map, a funnel shaped water channel with an obstacle in the middle going through the whole channel, and a map with an obstacle located in the upper right corner. The model described in Section 3 is used to solve defined COLAM problem on these unknown maps. In this case-study non-sorting genetic algorithm [3] NSGAI1 is used as the MOEA module due to limited number of parameters and objectives. It is assumed that the communication frequency is constant for all agents, furthermore, all communications are done with no interference or multi-path. In addition, the proposed TIVR is used for mapping. Next section is dedicated to highlight results.

5 RESULTS

To test mapping using the introduced trajectory incorporated TIVR complex, simulations were conducted on different environment typologies of size $200\text{ m} \times 100\text{ m}$. In Figure 3 (map 1) the environment had a single object in the middle. The red color (3 simplex in the TIVR graph) reflects that that area is more dense with edges and nodes and thus the probability of having an obstacle there is less than the green one (2 simplex in TIVR graph). Figures 4 (map 2), 5 (map 3), and 6 (map 4) illustrate respectively the TIVR complex for a map with two consecutive obstacles, a water channel with a pipe in the middle, and a map with an obstacle in the upper right corner, respectively.

Using a population of 50 and for 25 iterations (computational power limit stopping criterion), the cost functions are set as follows:

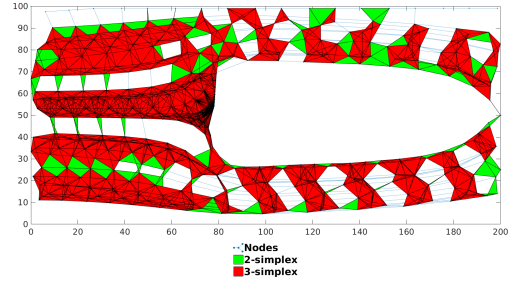


Figure 3: Reconstructed Single Obstacle Environment via TIVR complex

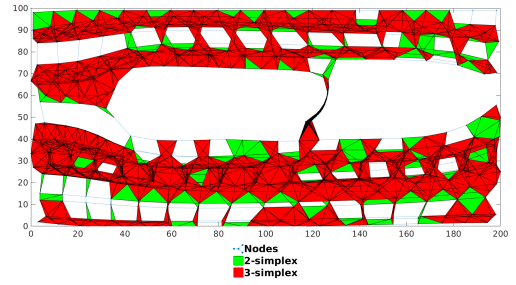


Figure 4: Reconstructed Two Obstacles via TIVR complex

$$M = \frac{|A \cap B|}{\kappa + |A \setminus B| + |B \setminus A|} \quad (19)$$

Where M is mapping quality, $|A \cap B|$ represents the area where the obstacle in the reconstructed map A and the real map B overlap on the map grid, $|A \setminus B|$ is the area of the obstacle which is in the reconstructed map A but not in the real one B , $|B \setminus A|$ is the area of the obstacle that is in the real map B but not in the reconstructed one A and κ is the $\max(|A|, |B|)$. Consequently, it holds $M \in [0, 1]$. On the other hand:

$$P = N(C_p + \phi C_f) \quad (20)$$

Where P is power coefficient, N is the number of agents, C_p is the communication power factor and C_f communication frequency and ϕ is a weighting coefficient. Table 1 highlights solutions from the Pareto optimal front when number of agents (N) are 30.

Map ID [#]	Comm. Frequency [Hz]	Comm. Range [m]	Power Coefficient [#]	Mapping Quality [%]
1	26.1	18.8	1290.2	95%
2	31.4	15.6	1363.2	86%
3	29.5	16	1317	93%
4	27.7	16.2	1268.4	90%

Table 1: Pareto optimal picked solutions at $N=30$

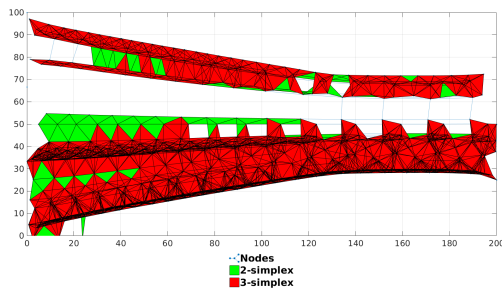


Figure 5: Reconstructed Tunnel Water Channel via TIVR complex

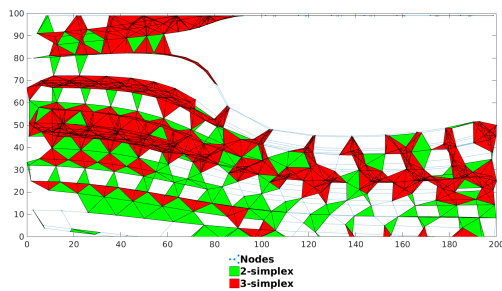


Figure 6: Reconstructed Side Blocked Channel via TIVR complex

6 CONCLUSION

In this work a modified version of SLAM problem dubbed as COLAM is defined. In this problem a centralized off-line localization and mapping is conducted to reconstruct an unknown environment topology with the least number of agents and with the least consumption of the available agent’s resources. A full description of a system that solves this problem is laid out. Moreover, a projection on a case study is presented. In this case study, agents are optimized in compliance with the COLAM problem to conduct mapping on four different environments. NSGAii is used with two objective functions: maximization of mapping accuracy and minimization of consumed power. Furthermore, an introduced technique dubbed as TIVR is used for mapping.

7 ACKNOWLEDGEMENTS



This project has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement No 665347.

We also gratefully acknowledge the computational resources provided by RWTH Compute Cluster from RWTH Aachen University under project RWTH0118.

REFERENCES

[1] G.C. Calafiore, L. Carlone, and Mingzhu Wei. 2010. Position estimation from relative distance measurements in multi-agents formations. In *18th Mediterranean Conference on Control Automation (MED)*, 2010. 148–153. DOI : <http://dx.doi.org/10.1109/MED.2010.5547601>

[2] Erin W Chambers, Vin De Silva, Jeff Erickson, and Robert Ghrist. 2010. Vietoris-rips complexes of planar point sets. *Discrete & Computational Geometry* 44, 1 (2010), 75–90.

[3] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.

[4] G. Destino and G. Abreu. 2011. On the Maximum Likelihood Approach for Source and Network Localization. 59, 10 (2011), 4954–4970. DOI : <http://dx.doi.org/10.1109/TSP.2011.2161302>

[5] Robert Ghrist and Abubakr Muhammad. 2005. Coverage and hole-detection in sensor networks via homology. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*. IEEE, 254–260.

[6] Islam SM Khalil, Herman C Dijkstra, Leon Abelmann, and Sarthak Misra. 2014. MagnetoSperm: A microrobot that navigates using weak magnetic fields. *Applied Physics Letters* 104, 22 (2014), 223701.

[7] X. Rong Li and V. P. Jilkov. 2003. Survey of maneuvering target tracking. Part I. Dynamic models. *IEEE Trans. Aerospace Electron. Systems* 39, 4 (Oct 2003), 1333–1364. DOI : <http://dx.doi.org/10.1109/TAES.2003.1261132>

[8] Christopher V Rao. 2000. *Moving horizon strategies for the constrained monitoring and control of nonlinear discrete-time systems*. Ph.D. Dissertation. UNIVERSITY OF WISCONSIN MADISON.

[9] Stephan Schlupkothen and Gerd Ascheid. 2016. Joint Localization and Transmit-Ambiguity Resolution for Ultra-Low Energy Wireless Sensors. In *2016 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE’16)*.

[10] A. Simonetto and G. Leus. 2014. A moving horizon convex relaxation for mobile sensor network localization. In *2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM)*. 25–28. DOI : <http://dx.doi.org/10.1109/SAM.2014.6882329>

[11] Ivan Stoianov, Lama Nachman, Sam Madden, Timur Tokmouline, and M Csail. 2007. PIPENET: A wireless sensor network for pipeline monitoring. In *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*. IEEE, 264–273.

[12] Elena Telnishnikh, J van Pol, and HJ Wörtche. 2015. Micro Motes: a highly penetrating probe for inaccessible environments. In *Intelligent Environmental Sensing*. Springer, 33–49.

[13] Paul Tseng. 2007. Second-Order Cone Programming Relaxation of Sensor Network Localization. *SIAM J. on Optimization* 18, 1 (Feb. 2007), 156–185. DOI : <http://dx.doi.org/10.1137/050640308>

[14] Zizhuo Wang, Song Zheng, Stephen Boyd, and Yinyu Ye. 2006. *Further relaxations of the SDP approach to sensor network localization*. Technical Report.

[15] Afra Zomorodian. 2010. Fast construction of the Vietoris-Rips complex. *Computers & Graphics* 34, 3 (2010), 263–271.