Contributions to LiDAR based SLAM and Computational Ethology

Marina Aguilar Moreno

PhD Advisor:
Prof. Manuel Graña Romay

Department of Computer Science and Artificial Intelligence
University of the Basque Country (UPV/EHU)

Donostia - San Sebastian, 18th July 2023

eman ta zabal zazu
Universidad del País Vasco | Euskal Herriko Unibertsitatea
INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Outline

Outline

Abstract Contribution 1

Contribution about LiDAR Based SLAM

- LiDAR sensors are used for scanning and reconstruction of indoor and outdoor environments.
- SLAM is one of the applications that takes advantage of this sensors, improving the speed and accuracy.
- There exists many brands LiDAR with different specifications. For this contribution we want to test quality of low-cost M8 Quanergy LiDAR.
- Due to the low resolution, unable to apply Deep Learning based approaches to these sparse data.
- Implementation of traditional methods based algorithm for SLAM to check result quality by using:
  - Iterative Closest Point
  - Coherent Point Drift
  - Normal Distribution Transform
- We propose a hybrid SLAM algorithm that achieves accurate results over in-house datasets captured with the low-cost LiDAR sensor.

Abstract Contribution 2

Contribution to Computational Ethology

- Computational Ethology
  - discipline that studies the animal behavior making use of the advances in Computer Vision and Artificial Intelligence.
  - quantitative approach to compare the effect of new medicines in different subjects.
- There exist a wide variety of sensors for experimentation. In this Thesis, data were collected with a piezoelectric platform and a top camera.
- Research question proposed: Is it possible to discriminate different phenotypes with data from piezoelectric signal by using Artificial Intelligence techniques?
- We propose a pipeline to obtain behavioral data for locomotion periods to train the models.
- An exhaustive exploration with Machine Learning based methods, Neural Networks and Convolutional Neural Networks to answer this research question.
- Concluding that it is feasible to discriminate phenotypes on the basis on pressure signal and Artificial Intelligence.

Outline

Outline

Background

- SLAM (Simultaneous localization and mapping) is widely used in robotics to create maps and estimate the path.
- Sensors used in SLAM:
  - Light detection and ranging (LiDAR) sensors:
    - based on a light beam that computes distances from the sensor to surrounding obstacles creating a point cloud that represents the enviornment.
    - Many brands: Velodyne, SICK, Hokuyo,...
    - Can be mounted in ground vehicles, unmanned aerial vehicles (UAV) or manually.
    - Other sensors can go with LiDAR: cameras, inertial measurement units (IMU), navigation systems (GNSS, GPS)...

State of the Art

There exist many LiDAR applications:

- SLAM + Robotics
- Remote sensing: Forestry data analisys, urban landscape creation, industrial measurements...
- Autonomous driving: real-time navigation, road inventories...
- 3D mapping: tree crop location, analysis of coastal barriers, surveying...

Computational approaches:

- Traditional methods:
  - The Iterative Closest Point method
  - The Normal Distribution Transform method
  - LOAM (LiDAR Odometry and Mapping)
- Artificial intelligent methods:
  - Machine learning algorithms
  - Neural Networks
  - Reinforcement learning

Data resources:

- Public data repositories
- Simulation environments

Outline

Motivation and Objectives

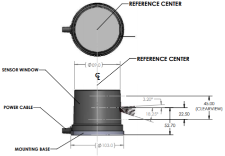
- Experimentation with LiDAR M8 Quanergy
- Data processing for SLAM applications
- Preliminary study to compare three traditional methods:
  - Iterative Closest Point
  - Coherent Point Drift
  - Normal Distribution Transform
- Proposal of a new hybrid method that improves the accuracy of the algorithm.

Outline

LiDAR M8 Quanergy

Low-cost LiDAR with Time-of-Flight (TOF) Technology

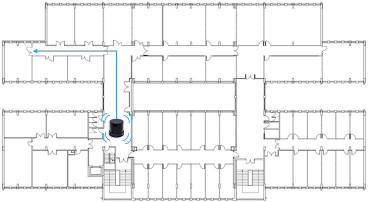
Parameter	M8 sensor specifications
Detection layers	8
Returns	3
Minimum range	0.5m (80% reflectivity)
Maximum range	>100m (80% reflectivity)
Spin rate	5Hz - 20Hz
Intensity	8 bits
Field of view	Horizontal 360° - Vertical 20° (+3°/-17°)
Data outputs	Angle, Distance, Intensity, Synchronized Time Stamps



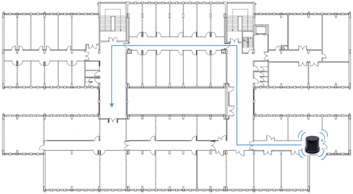
Location and Experimental Settings

- Location: 3rd floor of the Computer Science School of the UPV/EHU in San Sebastian.
- LiDAR on a manually-driven mobile platform.

In-house Dataset #1



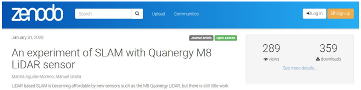
In-house Dataset #2



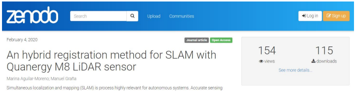
Location and Experimental Settings

- Location: 3rd floor of the Computer Science School of the UPV/EHU in San Sebastian.
- LiDAR on a manually-driven mobile platform.

In-house Dataset #1

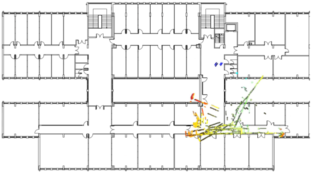


In-house Dataset #2



The Failure of Deep Learning

- M8 Quanergy sensor fails with novel state-of-the-art methods, such as Deep Global Registration[1].
- Due to the low resolution of this low-cost LiDAR compared to others brands:
  - M8 Quanergy LiDAR: 8 lasers and 1.3M points/s.
  - Kitti Dataset used in Deep Global Registration → Velodyne HDL-64E: 64 lasers and 2.2M points/s.
- Traditional methods to validate the quality of M8 Quanergy LiDAR measurements.

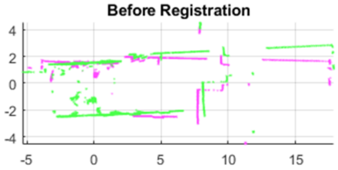


[1] Choy, C., Dong, W., Koltun, V., Deep global registration (2020) Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, art. no. 9157005, pp. 2511-2520.

Outline

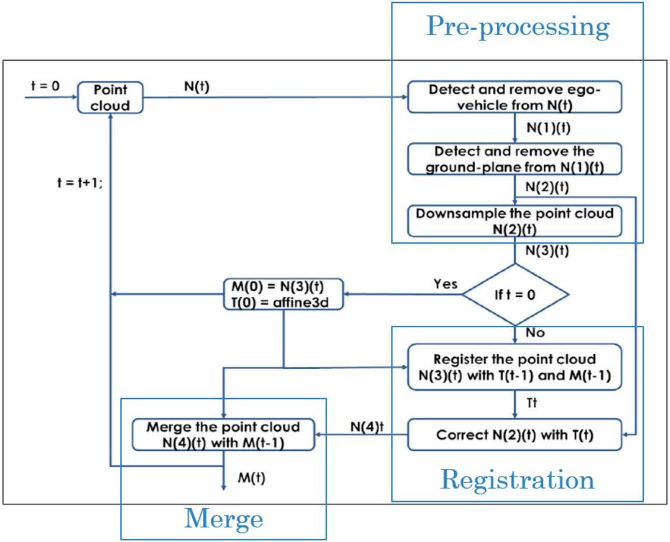
Point Cloud Registration

- Given 2 point clouds in time t_1(reference) and t_2 (moving), where $t_1 < t_2$:
  - Find the best transformation matrix to match both point clouds.
  - Apply this transformation to the moving point cloud.
- Merge the resulting point cloud with the reference point cloud.
- The result obtained is a new point cloud with more information.



Generic SLAM Framework

General diagram of point cloud registration algorithm.



Generic SLAM Framework

General template of point cloud registration algorithm.

Input: sequence of point clouds $\{N(t)\}_{t=0}^{T}$ captured by the LiDAR
Output: overall point cloud $M(T)$, sequence of registered transformations $\{\mathcal{T}_t\}_{t=1}^{T}$
For $t = 0, \ldots, T$

1. $N^{(1)}(t) \leftarrow$ remove ground plane from $N(t)$
2. $N^{(2)}(t) \leftarrow$ remove ego-vehicle from $N^{(1)}(t)$
3. $N^{(3)}(t) \leftarrow$ down-sample $N^{(2)}(t)$
4. If $t = 0$ then $M(0) = N^{(3)}(t)$; GOTO For
5. $(\mathcal{T}_t, e_t) \leftarrow$ register $\mathcal{T}_{t-1}\left(N^{(3)}(t)\right)$ to $M(t-1)$
6. $N^{(4)}(t) \leftarrow \mathcal{T}_t\left(N^{(2)}(t)\right)$
7. $M(t) \leftarrow merge\left(M(t-1), N^{(4)}(t)\right)$

The Iterative Closest Point (ICP) method[2]

- Given 2 data sets: P with N_p points and X with N_x points:
- Initialize the vector $P_0 = P$, the iteration $k = 0$ and the quaternions $\bar{q_0} = [\bar{q_R}|\bar{q_T}]^t = [q_0 q_1 q_2 q_3|q_4 q_5 q_6]^t = [1, 0, 0, 0|0, 0, 0]^t$
- Repeat this loop until convergence with threshold τ:
  - Compute the closest points between P_k and X: $Y_k = \mathcal{C}\left(P_k, X\right)$.
  - Compute the registration: $(\bar{q_k}, d_k) = \mathcal{Q}_k\left(P_0, Y_k\right)$:
    - Calculate cross-covariance matrix $\mathcal{Q}_k\left(P_0, Y_k\right)$, and compute the eigenvalues and unit eigenvectors that are $q_{\bar{R}k}$.
    - Calculate $q_{\bar{T}k} = \bar{\mu_x} - R(q_{\bar{R}k})\bar{\mu_p}$, where μ is the center of mass of each point set and R the rotation matrix, both known.
    - Obtain the transformation matrix $\bar{q_k} = [q_{\bar{R}k}|q_{\bar{T}k}]^t$ and the mean square error d_k.
  - Apply the matrix to the point P_0 , $P_{k+1} = \bar{q_k}\left(P_0\right)$.
  - Compute the mean square error d: If $d_k - d_{k+1} < \tau$, finish the loop.

---

[2] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes", IEEE Transactions on Pattern Analysis and Machine Intelligence 14, 2 (1992), pp. 239-256.

The Coherent Point Drift (CPD) method[3]

- Formulates the alignment of 2 point clouds as a probability density estimation problem:
  - point cloud $Y = \{\mathbf{y}_i\}_{i=1}^{M}$ represents the Gaussian Mixture Model (GMM) centroids,
  - point cloud $X = \{\mathbf{x}_i\}_{i=1}^{N}$ represents the data points generated by these centroids.
- The GMM probability density function is $p(\mathbf{x}) = \omega \frac{1}{N} + (1-\omega)\sum_{m=1}^{M} \frac{1}{M} p(\mathbf{x}\,|m)$.
- We obtain the GMM centroids locations minimizing the log-likelihood function:

$$E\left(\theta, \sigma^2\right) = -\sum_{n=1}^{N} \log \sum_{m=1}^{M} P(m)\, p(\mathbf{x}\,|m),$$

where θ is the rotation, translation and scale parameters and σ^2 is the covariance.

- We apply Expectation-Maximization (EM) algorithm to find θ and σ^2, minimizing the objective function:

$$Q = -\sum_{n=1}^{N} \sum_{m=1}^{M} P^{old}(m\,|\mathbf{x}_n) \log\left(P^{new}(m)\, p^{new}(\mathbf{x}\,|m)\right).$$

[3] A. Myronenko and X. Song, "Point Set Registration: Coherent Point Drift", IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 12 (2010), pp. 2262-2275.

The Normal Distribution Transform (NDT) method[4]

- The approach is similar to the occupancy grids.
- The space is divided in cells, where the probability of a sample falls in a cell is: $p(\mathbf{x}) \sim N(q, \Sigma)$.
- Let be the spatial mapping T: $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} cos\phi & -sin\phi \\ sin\phi & cos\phi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$
  - Build the NDT of the first scan.
  - Initialize the estimate for the parameters of the mapping T.
- Repeat this loop until convergence:
  - For the second scan, apply mapping T.
  - Determine the corresponding normal distribution for each mapped point, q and Σ.
  - Compute score function: $score(\bar{p}) = \sum_i exp\left(\frac{-(\bar{x_i}' - \bar{q_i})^t \sum_i^{-1} (\bar{x'_i} - \bar{q_i})}{2}\right)$, where $\bar{p}$ is the parameter vector to estimate.
  - Calculate the parameters vector $\bar{p}$ by optimizing the score function with Newton's Algorithm.

[4] Peter Biber and Wolfgang Straßer, "The Normal Distributions Transform: A New Approach to Laser Scan Matching", in vol. 3, (2003), pp. 2743 - 2748 vol.3.

Hybrid Point Cloud Registration Algorithm

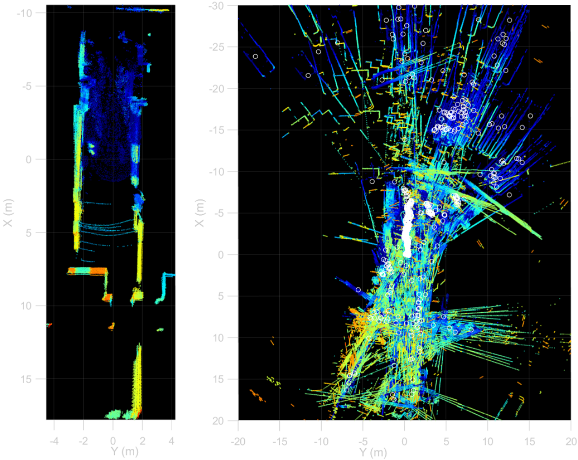
Input: sequence of point clouds $\{N(t)\}_{t=0}^{T}$ captured by the LiDAR
Output: overall point cloud $M(T)$, sequence of registered transformations $\{\mathcal{T}_t\}_{t=1}^{T}$
Method = "ICP"
For $t = 0, \ldots, T$

1. $N^{(1)}(t) \leftarrow$remove ground plane from $N(t)$
2. $N^{(2)}(t) \leftarrow$ remove ego-vehicle from $N^{(1)}(t)$
3. $N^{(3)}(t) \leftarrow$ downsample $N^{(2)}(t)$
4. If $t = 0$ then $M(0) = N^{(3)}(t)$; GOTO step 1
5. $(\mathcal{T}_t, e_t) \leftarrow$ register $\mathcal{T}_{t-1}\left(N^{(3)}(t)\right)$ to $M(t-1)$ using Method
6. If $e_t > \theta_e$ then Method = "NDT"; GOTO step 5
7. $N^{(4)}(t) \leftarrow \mathcal{T}_t\left(N^{(2)}(t)\right)$
8. $M(t) \leftarrow merge\left(M(t-1), N^{(4)}(t)\right)$

Outline

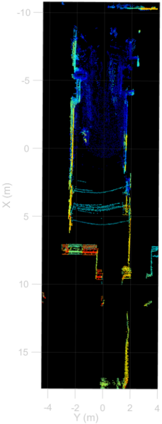
Results over the In-house Dataset #1

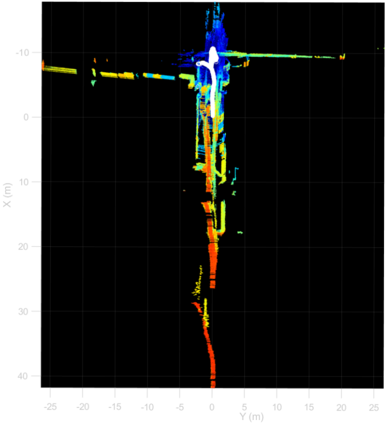
Registration of the cloud points before reaching the turning point (left) and the estimated trajectory (white dots) and registered cloud of points using ICP (right).



Results over the In-house Dataset #1

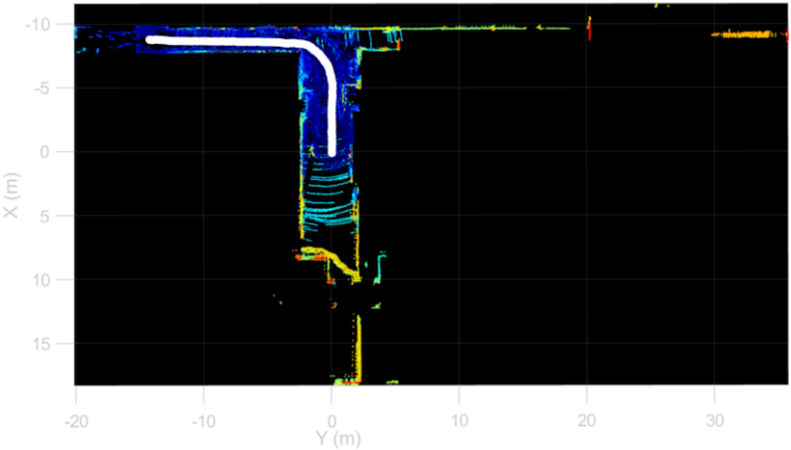
Registration of the cloud points before reaching the turning point (left) and the estimated trajectory (white dots) and registered cloud of points using CPD (right).





Results over the In-house Dataset #1

Estimated trajectory and registered cloud of points using NDT.

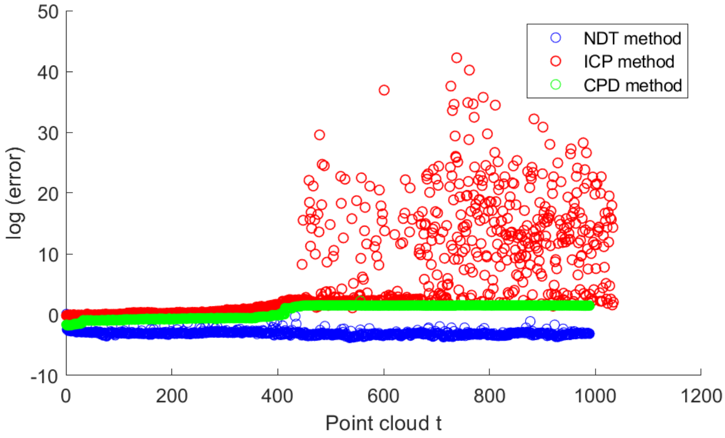


Results over the In-house Dataset #1

Projection of the NDT registered point cloud on the plan of the building.

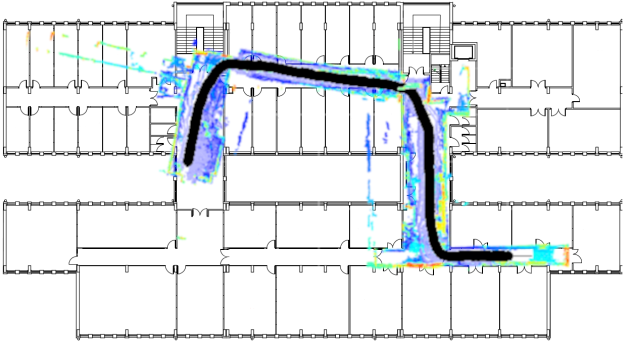
Results over the In-house Dataset #1

Evolution of the logarithmic registration error for NDT, CPD, and ICP methods.



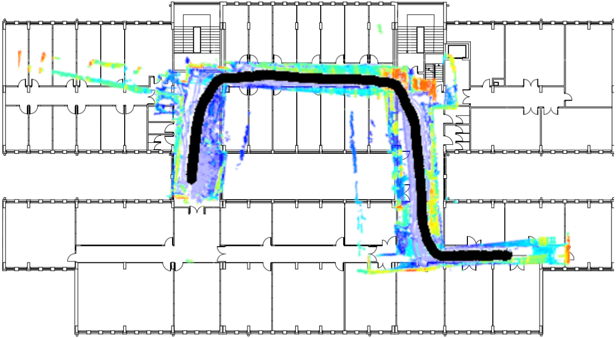
Results over the In-house Dataset #2

Projection of the ICP registered point cloud on the plan of the building with the estimated trajectory.



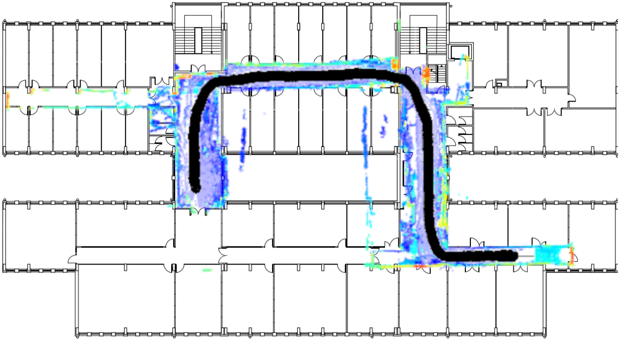
Results over the In-house Dataset #2

Projection of the NDT registered point cloud on the plan of the building with the estimated trajectory.



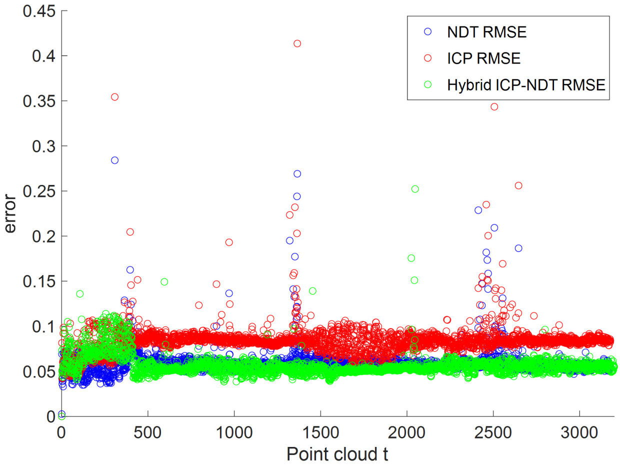
Results over the In-house Dataset #2

Projection of the HRA registered point cloud on the plan of the building with the estimated trajectory.



Results over the In-house Dataset #2

Time evolution of the registration RMSE for NDT, ICP, and HRA methods.



Results over the In-house Dataset #2

Performance of ICP, NDT, and HRA methods along the experimental path.

	ICP method	NDT method	HRA method
Maximum RMSE	0.4136	0.2841	0.2522
Median RMSE	0.0835	0.0589	0.0554
Cumulative RMSE	265.29	187.21	176.20

Outline

Conclusion

- Evaluation of M8 Quanergy LiDAR in SLAM application in two in-house datasets.
  - The SLAM algorithm includes point cloud pre-processing, registration, transformation and merger.
- Report a comparison between three registration methods for point cloud registration:
  - Iterative Closest Point (ICP).
  - Coherent Point Drift (CPD).
  - Normal Distribution Transform (NDT).
- Proposition of a novel Hybrid Point Cloud Registration Algorithm (HRA):
  - ICP + NDT.
- Results:
  - ICP and CPD obtain larger error than NDT for the dataset #1.
  - NDT is better than ICP and CPD in turning points for the dataset #1.
  - HRA improves both ICP and NDT RMSE for dataset #2.
  - HRA obtains better reconstruction than ICP and NDT for dataset #2.

Publications Produced about this Topic

1. Aguilar-Moreno, M., Graña, M. (2021). A Comparison of Registration Methods for SLAM with the M8 Quanergy LiDAR. In: Herrero, Á., Cambra, C., Urda, D., Sedano, J., Quintián, H., Corchado, E. (eds) 15th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2020). Advances in Intelligent Systems and Computing, vol 1268. Springer, Cham.
2. Aguilar-Moreno, M., Graña, M. (2020). An Hybrid Registration Method for SLAM with the M8 Quanergy LiDAR. In: de la Cal, E.A., Villar Flecha, J.R., Quintián, H., Corchado, E. (eds) Hybrid Artificial Intelligent Systems. HAIS 2020. Lecture Notes in Computer Science(), vol 12344. Springer, Cham.
3. Aguilar-Moreno, M., Graña, M. (2022), On registration methods for SLAM with low resolution LiDAR sensor, Logic Journal of the IGPL; jzac037, https://doi.org/10.1093/jigpal/jzac037.

Outline

Outline

Background

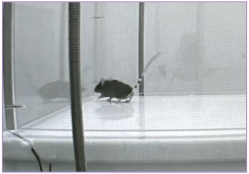
Behavior:

- The set of muscular responses of a living being because of an external stimulus and internal motivation.

Computational Ethology (CE):

- Discipline that studies the animal behavior.
- Using the advances in Computer Vision and Artificial Intelligence.
- Focused on the natural behavior to perform real-world tasks
- In unrestricted environments.
- Quantitative behavior characterization.

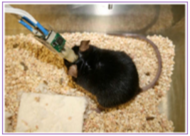
Pharmacological point of view: CE is useful to test new medicines comparing the effect in different subjects, obtained by genetic modification.

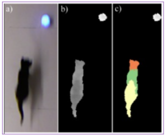


State of the Art

Sensors:

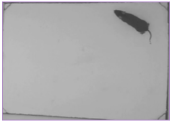
- RGB / depth / infrared cameras
- Pressure sensors
- Inertial sensors
- Microphones

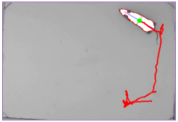




Applications based on Artificial Intelligence:

- Tracking applications: DeepLabCut, Bonsai, SLEAP, ...
- Behavior classification: JAABA, DeepEthogram, VAME, ...
- Strain classification: SVM, k-NN



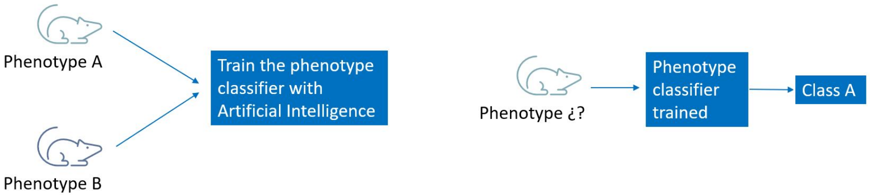


Outline

Motivation and Objectives

Research question proposed:

- Is it possible to implement a strain classifier from pressure signal and Artificial Intelligence?
- We focus on:
  - Spectrogram from piezoelectric signal
    - Features
    - Images
  - Locomotion periods.
- Application of Artificial Intelligence techniques.



Outline

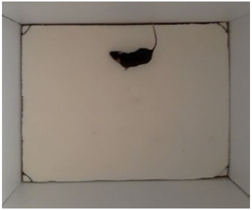
Animal and Experimentation

12 mice with 2 different strains:

- 7 wild-type (WT): non-mutated gene.
- 5 transgenic Fmr1-knockout (Fmr1-KO): animal model to study Fragile X Syndrome.

Recording system:

- Opaque-walled cage
- Base: piezoelectric platform with 3 sensors (20 kHz)
- Top video camera (25 fps)
- Computer with Spike software to record piezoelectric signal.

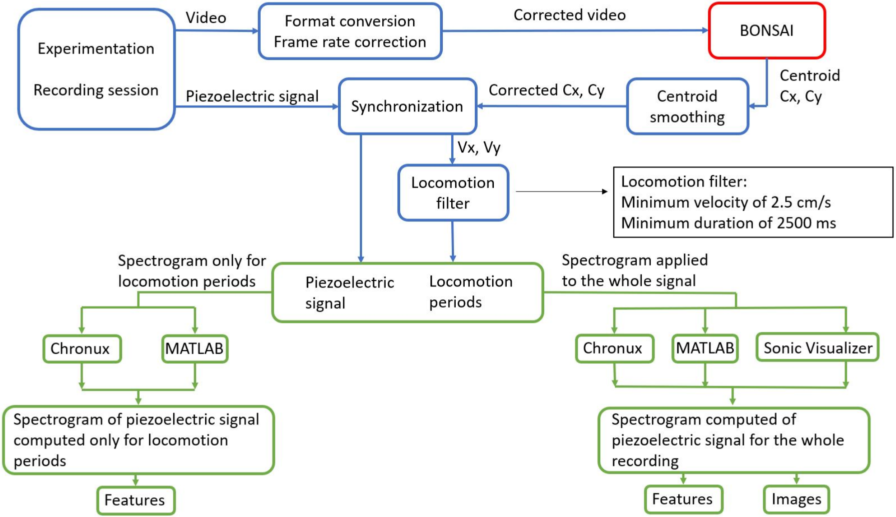


Animals introduced individually.
Procedure in accordance with EU directives for animal protection.

Outline

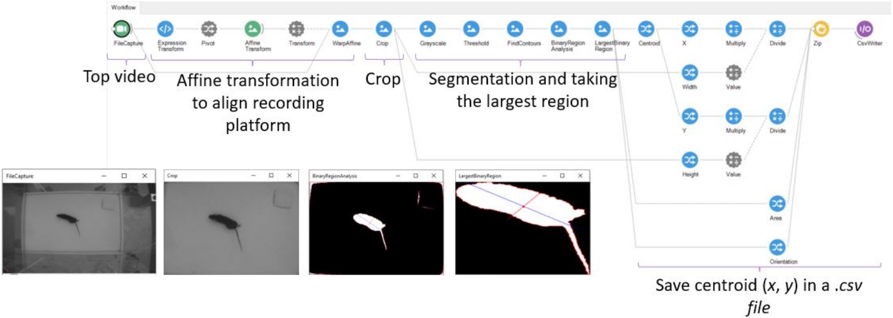
Behavioral Data Processing

Data processing pipeline



Behavioral Data Processing

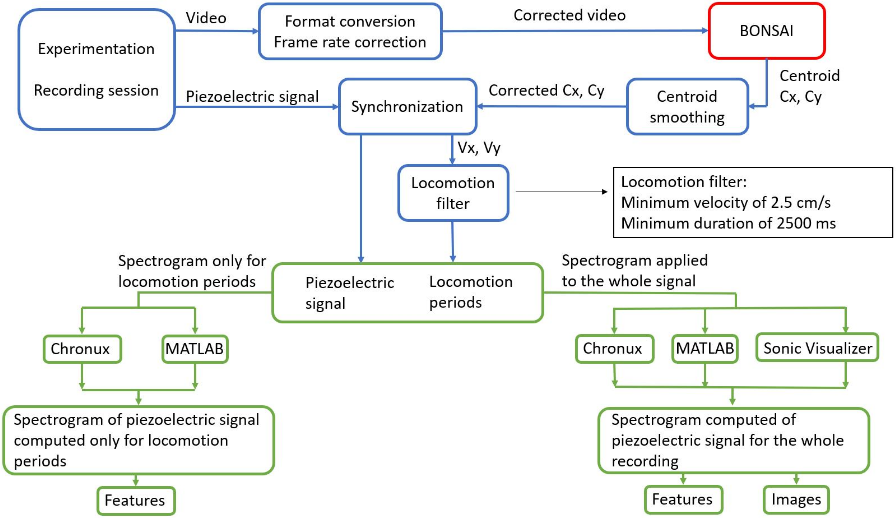
Bonsai[5] Pipeline for Video Processing



[5] https://bonsai-rx.org/

Behavioral Data Processing

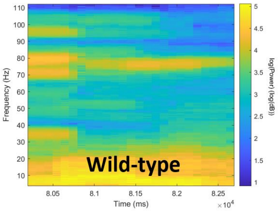
Data processing pipeline

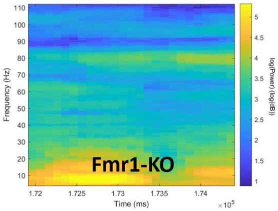


Parameters for Spectrogram Computation

Parameters for Spectrogram Computation with Chronux Library and Images

Parameters	Value 1	Value 2	Default values
Window size (s)	1	2	-
Windows step (s)	0.1	0.2	-
Tapers	[4, 2]	[3, 5]	[3, 5]: A numeric vector [TW K] where TW is the time-bandwidth product and K is the number of tapers, less than or equal to 2TW-1
Frequency of interest (Hz)	[1.5 - 40]	[4 - 112]	[0 - Fs/2] (Fs: sampling frequency)

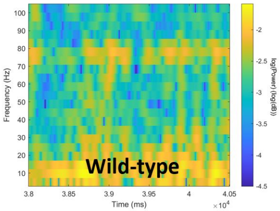


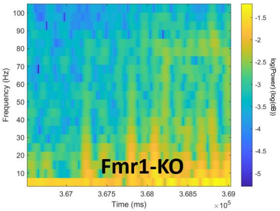


Parameters for Spectrogram Computation

Parameters for Spectrogram Computation with Sonic Visualizer and Images

Parameter	Value	Range of values
Colour	Green	[Green, Sunset, ... , Wasp, Ice, ...]
Scale	dB	[Linear, Meter, dB^2, dB, Phase]
Window size	256	[32, 64, 128, 256, 512, ... , 16384, 32768]
Overlap	93.75%	[none, 25%, 50%, 75%, 87.5%, 93.75%]
Show	All bins	[All Bins, Peak Bins, Frequencies]
Scale	Linear	[Linear, Log]

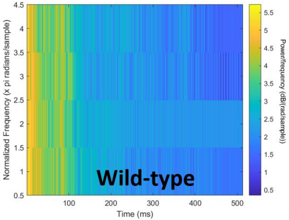


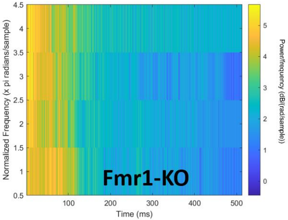


Parameters for Spectrogram Computation

Parameters for Spectrogram Computation with MATLAB and Images

Parameters	Value 1	Value 2	Value 3	Range of values
Number of sections	8	4	-	Integer
Overlap	0.5	0.1	-	[0 - less than window]
Window	Hamming	Chebyshev	Tukey	[Bartlett-Hann, Bartlett, Gaussian ,..., triangular]





Formulation Problem and AI based Models

- Binary classification problem to discriminate two phenotypes:
  - WT (class 0)
  - Fmr1-KO (class 1)
- Inputs for classification:
  - Spectrogram features
  - Spectrogram images
- Models:
  - Machine learning models
  - Neural Network
  - Convolutional Neural Networks + Transfer learning

Models for Classification

Machine Learning Models

- Decision trees
- Linear discriminant analysis
- Logistic regression
- Gaussian Naive Bayes
- SVM
- k-NN
- Boosted trees
- Bagged trees
- Subspace discriminant
- Subspace k-NN
- RUSBoosted trees

Experiments	Chronux	MATLAB	Sonic Visualizer	Total
Segmented chunks	8	12	-	20
Whole spectrogram	8	12	1	21
Total	16	24	1	41

Models for Classification

Neural Network

- Multi-layer Perceptron (MLP)
- Parameters selected with a grid search:
  - 255 batch size
  - 700 epochs
  - Adam optimizer

MLP layers	Neurons	Activation function	Dropout normalization
Input layer	variable	-	-
First hidden layer	400	relu	0.2
Second hidden layer	200	relu	-
Third hidden layer	60	relu	-
Fourth hidden layer	35	relu	-
Output layer	1	sigmoid	-

Models for Classification

Convolutional Neural Networks with Transfer Learning

- AlexNet
- GoogLeNet
- ResNet50

Parameter	Value
Solver	Adam
Learning rate	0.0001
Mini batch size	52
L2 Regularization	0.0001
Folds for Cross-validation	5

Algorithm	Layers	Total learnables
AlexNet	25 (depth 8)	56 876 418
ResNet50	177 (depth 50)	23 538 690
GoogLeNet	144 (depth 22)	5 975 602

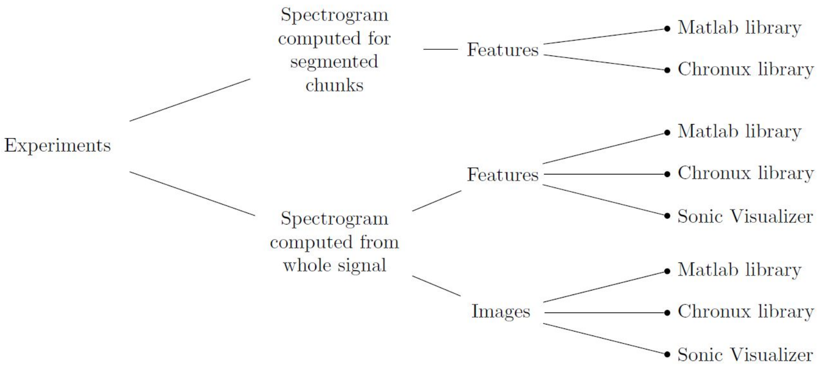
Model Training and Evaluation

- Dataset divided into two parts:
  - 80% train set
  - 20% test set
- 5-fold cross-validation
- Metrics: Accuracy, AUC, Recall, Precision, F1 score

Algorithm	Execution time
Machine Learning algorithms	$\approx$ 1s
MLP	$\approx$ 2min
AlexNet	$\approx$ 35min
ResNet50	$\approx$ 4h
GoogLeNet	$\approx$ 2h

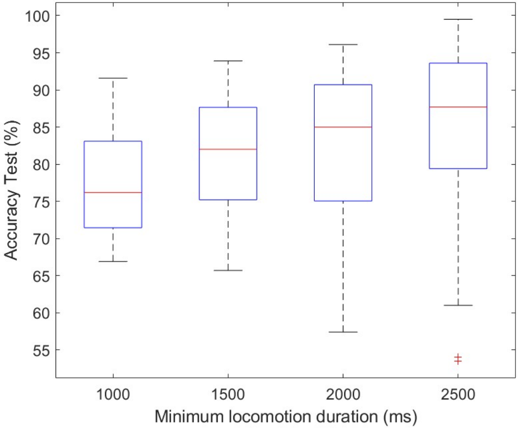
Outline

Results


Experiments
Spectrogram computed for segmented chunks
Features
Matlab library
Chronux library
Spectrogram computed from whole signal
Features
Matlab library
Chronux library
Sonic Visualizer
Images
Matlab library
Chronux library
Sonic Visualizer

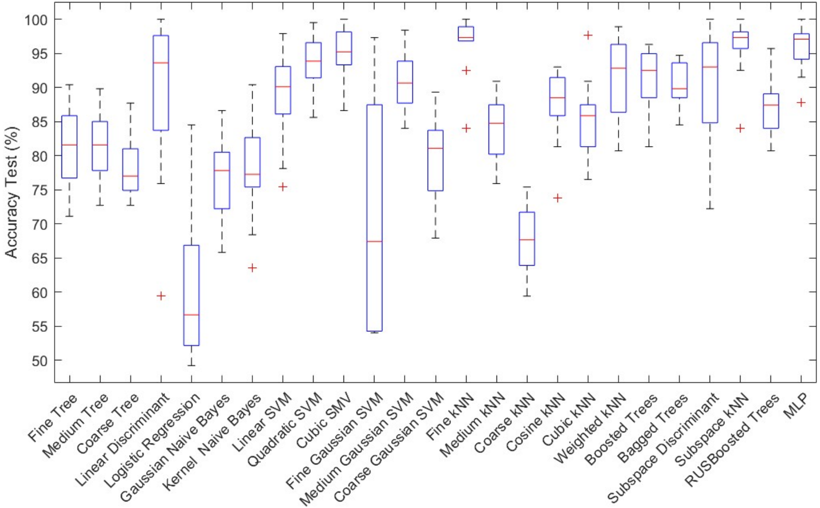
Results with Machine Learning Algorithms

Results for Different Minimum Locomotion Duration



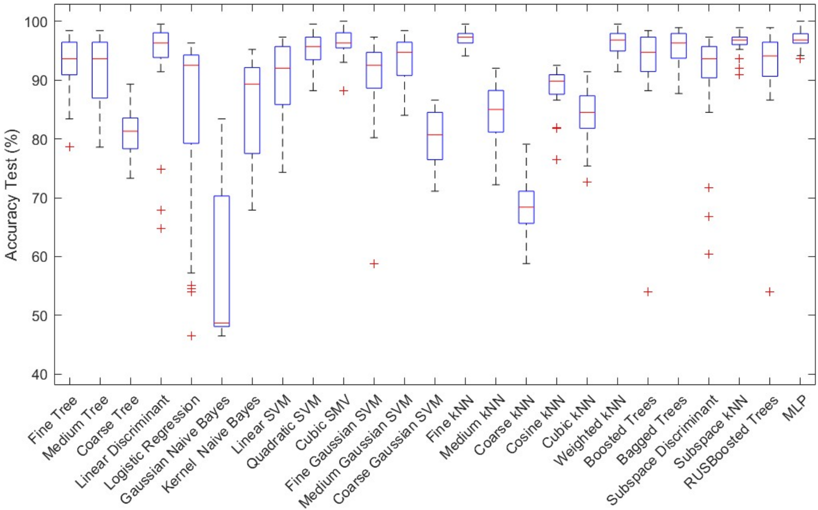
Results with Machine Learning Algorithms

Results for Spectrogram Features Computed only for Segmented Chunks



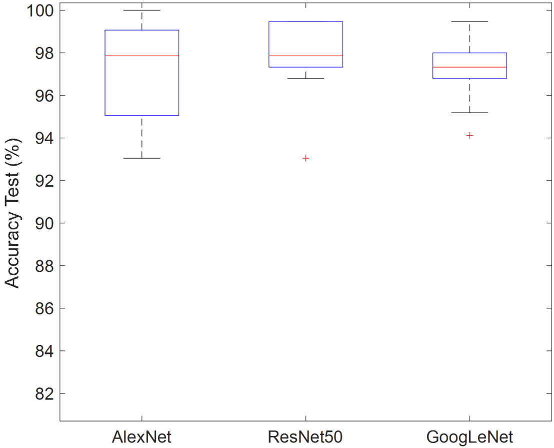
Results with Machine Learning Algorithms

Results for Spectrogram Features Computed from the Whole Signal



Results with Transfer Learning

Results for Image Classification with Chronux Library, MATLAB and Sonic Visualizer



Outline

Conclusion

- This work proposed the research question:
  - Is it possible to discriminate phenotypes with pressure signals and Artificial Intelligence?
- Binary classification problem with 2 different animal models:
  - Wild-type
  - Fmr1-KO
- Spectrogram from the pressure signal during locomotion periods:
  - Chronux Library
  - MATLAB
  - Sonic Visualizer
- Different Machine Learning based methods, NN and CNN have been tested with:
  - Spectrogram features
  - Spectrogram images
- Yes, we can differentiate phenotypes with high accuracy, precision, recall and F1 score.

Publications Produced about this Topic

1. Aguilar-Moreno, M., Graña, M. (2023). Computational Ethology: Short Review of Current Sensors and Artificial Intelligence Based Methods. In: Iliadis, L., Maglogiannis, I., Alonso, S., Jayne, C., Pimenidis, E. (eds) Engineering Applications of Neural Networks. EANN 2023. Communications in Computer and Information Science, vol 1826. Springer, Cham.
2. Aguilar-Moreno, M., Graña, M. (2023), Phenotype Discrimination based on pressure signals by transfer learning approaches, International Work-Conference on Artificial Neural Networks (IWANN 2023), accepted and presented in Congress.

Outline

Conclusion and Findings in LiDAR Based SLAM

- Installation and configuration of M8 Quanergy LiDAR sensor.
- Recording several datasets to implement SLAM algorithms.
- Comparison of the results obtained with three traditional methods:
  - Iterative Closest Point
  - Coherent Point Drift
  - Normal Distribution Transform
- Proposal of he Hybrid Registration Algorithm (HRA) with the joint of the ICP and NDT methods.
- Obtaining a better reconstruction of the surface with the HRA proposed in this Thesis.

Conclusion and Findings in Computational Ethology

- Development of a pipeline to process data from a recording system composed of a piezoelectric platform and a video camera:
  - Spectrogram features
  - Spectrogram images
- Application of AI techniques for animal model classification.
- Answer the research question proposed in this Thesis.

Other Results and Awards

1. Graña M, Aguilar-Moreno M, De Lope Asiain J, Araquistain IB, Garmendia X. (2020). Improved Activity Recognition Combining Inertial Motion Sensors and Electroencephalogram Signals. Int J Neural Syst. 2020;30(10): 2050053. https://doi.org/10.1142/S0129065720500537.
2. Second prize in the INIZIA 2023 call for proposals in the category of New Innovative Initiatives organised by BIC ARABA.
3. Finalist in the Manuel Laborde Werlindel 2022 call in the category of New Innovative Initiatives organised by BIC Gipuzkoa.

Future Work in Computational Ethology

- Apply the approach tested in the second contribution in an experimental study on healthy ageing in the elderly.
- The study is about fragility, which is a syndrome that affects to elderly.
- Predict this syndrome in early stages to mitigate its effects, improving the quality of life of the society.

Acknowledgements

- Thanks to my advisor Professor Manuel Graña.
- Thanks to Professor Xavier Leinekugel for allowing me to spend my international stay in his research group at the Mediterranean Institute of Neurobiology in Marseille.
- Thanks to the members of the Computational Intelligence Group.
- Thanks to the members of the Computer Science and Artificial Intelligence Department.
- Grant PRE2018-085294 funded by MCIN/AEI 10.13039/501100011033 and by "ESF Investing in your future".

Contributions to LiDAR based SLAM and Computational Ethology

Marina Aguilar Moreno

PhD Advisor:
Prof. Manuel Graña Romay

Department of Computer Science and Artificial Intelligence
University of the Basque Country (UPV/EHU)

Donostia - San Sebastian, 18th July 2023

eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

The Iterative Closest Point (ICP) method

- Given 2 data sets: P with N_p points and X with N_x points:
- Initialize the vector $P_0 = P$, the quaternions $\bar{q_0} = [\bar{q_R}|\bar{q_T}]^t = [q_0 q_1 q_2 q_3 q_4 q_5 q_6]^t = [1,0,0,0,0,0,0]^t$ and the iteration index $k = 0$
- Repeat this loop until convergence with threshold τ:
  - Compute the closest points between P_k and X: $Y_k = \mathcal{C}(P_k, X)$.
  - Compute the registration: $(\bar{q_k}, d_k) = \mathcal{Q}_k(P_0, Y_k)$:
    - Calculate cross-covariance matrix $\mathcal{Q}_k(P_0, Y_k)$, and compute the eigenvalues, whose unit eigenvectors are $\bar{q_{Rk}}$, corresponding to the maximum eigenvalue is the optimal rotation.
    - Calculate $\bar{q_{Tk}} = \bar{\mu_x} - R(\bar{q_{Rk}})\bar{\mu_p}$, where μ is the center of mass of each pointset and R the rotation matrix, both known.
    - Obtain the transformation matrix $\bar{q_k} = [\bar{q_{Rk}}|\bar{q_{Tk}}]^t$ and the mean square error d_k.
  - Apply the matrix to the point P_0 , $P_{k+1} = \bar{q_k}(P_0)$.
  - Compute the mean square error d: If $d_k - d_{k+1} < \tau$, finish the loop.

The Coherent Point Drift (CPD) method I

- Formulates the alignment of 2 point clouds as a probability density estimation problem:
  - point cloud $Y = \{\mathbf{y}_i\}_{i=1}^{M}$ represents the Gaussian Mixture Model (GMM) centroids,
  - point cloud $X = \{\mathbf{x}_i\}_{i=1}^{N}$ represents the data points.
- Registration tries to maximize the likelihood X as a sample of the probability distribution modeled by Y after the application of the transformation $T\left(Y, \theta\right)$, where θ are the transformation parameters.
  - The GMM model is formulated as $p\left(\mathbf{x}\right) = \omega \frac{1}{N} + \left(1 - \omega\right) \sum_{m=1}^{M} \frac{1}{M} p\left(\mathbf{x} \left| m\right.\right)$
  - All Gaussian conditional distributions are isotropic with the same variance σ^2, i.e. $p\left(\mathbf{x} \left| m\right.\right) = \left(2\pi\sigma^2\right)^{-D/2} \exp\left(\frac{\|\mathbf{x} - \mathbf{y}_m\|^2}{2\sigma^2}\right)$.
  - We parametrize the GMM centroids locations by a set of parameters θ and estimate them maximizing the likelihood or minimizing the log-likelihood function: $E\left(\theta, \sigma^2\right) = -\sum_{n=1}^{N} \log \sum_{m=1}^{M} P\left(m\right) p\left(\mathbf{x} \left| m\right.\right)$.

The Coherent Point Drift (CPD) method II

- We apply Expectation-Maximization (EM) algorithm to find θ and σ, minimizing the objective function:
  $Q = -\sum_{n=1}^{N} \sum_{m=1}^{M} P^{old}(m \,|\mathbf{x}_n) \log \left(P^{new}(m)\, p^{new}(\mathbf{x} \,|m) \right).$
- For rigid transformations, the objective function takes the shape:

$$Q\left(\mathbf{R}, \mathbf{t}, s, \sigma^2\right) = \frac{1}{2\sigma^2} \sum_{n,m=1}^{N,M} P^{old}(m \,|\mathbf{x}_n) \left\|\mathbf{x}_n - s\mathbf{R}\mathbf{y}_m - \mathbf{t}\right\|^2 + \frac{N_p D}{2} \log \sigma^2,$$

such that $\mathbf{R}^T\mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1.$

The Normal Distribution Transform (NDT) method

- Given 2 point clouds, build the NDT model of the first scan:
  - The space is divided in cells that has at least 3 points to:
    - Collect all points $x_{i=1..n}$.
    - Calculate the mean $\bar{q}= \frac{\mathbf{1}}{\mathbf{n}} \sum_i \bar{x_i}$,
    - Calculate the covariance matrix $\Sigma = \frac{1}{n} \sum_i (\bar{x_i} - \bar{q}) (\bar{x_i} - \bar{q})^t$.
  - The spatial mapping T: $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} cos\phi & -sin\phi \\ sin\phi & cos\phi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$
    - Initialize the estimate for the parameters of the mapping T.
  - Repeat this loop until convergence:
    - For the second scan, map the points into the coordinate frame of the first scan according to the parameters.
    - Determine the corresponding normal distribution for each mapped point.
    - Compute score function: $score(\bar{p}) = \sum_i exp\left(\frac{-(\bar{x_i}' - \bar{q_i})^t \sum_i^{-1} (\bar{x_i'} - \bar{q_i})}{2}\right)$.
    - Calculate the parameters vector $\bar{p}$ by optimizing the score function with Newton's Algorithm.