



Project Management and System Development

# Activity Classification with Persistent Homology Barcodes

Intermediate Presentation 27/06/2017

Project Supervisor: Benjamin Busam

Presented by: Ester Molero Hidalgo

[ester.molero@tum.de](mailto:ester.molero@tum.de)

# Project Reminder

Classify repetitive movements according to distance measurement between them



**Topological data analysis**  
Insensitive to orientation changes and  
does not suffer from extensive  
augmentation needs.

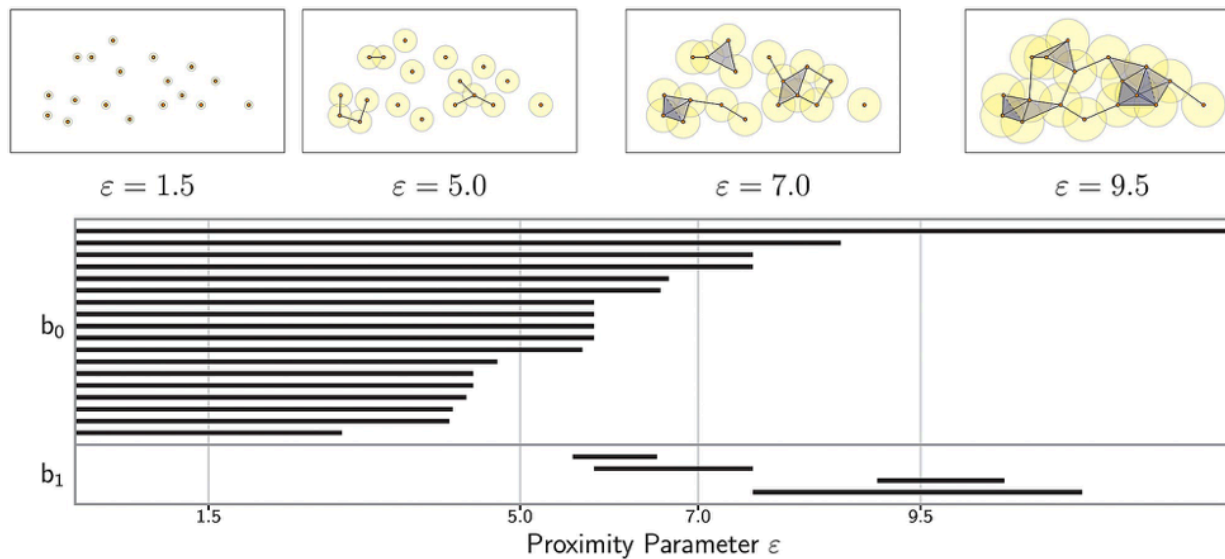


First trial of persistence homology  
for medical application.



# Project Reminder

Classify repetitive movements according to distance measurement between them



Persistence Homology

Source: <https://dsweb.siam.org/The-Magazine/Article/topological-data-analysis>  
Last Accessed: 26/6/17



# Tasks

1- Collect the data: 3D pose series

- Acquisition interface -> OpenIGTLink Library
- Obtain data generated by the FRAMOS Optical Tracking System of different movements repeatedly.

2- Distance function algorithm.

3- Calculate the homology barcodes for each data window. -> Ripser Library

[Ripster library home page](#)

4- Separate de barcodes via SVM (Support Vector Machine)

5- Evaluate de classification.



# Tasks

1- Collect the data: 3D pose series

- Acquisition interface -> [OpenIGTLink Library](#)
- Obtain data generated by the FRAMOS Optical Tracking System of different movements repeatedly.

2- Distance function algorithm.

3- Calculate the homology barcodes for each data window. -> Ripser Library

[Ripster library home page](#)

4- Separate de barcodes via SVM (Support Vector Machine)

5- Evaluate de classification.

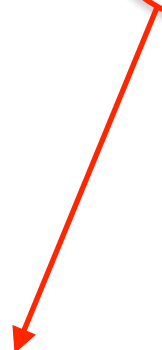
- Finished
- In progress
- To do



# Distance Function Algorithm Prototype

Output:  
Distance Matrix

$$\begin{bmatrix} 0 & d_{w_1, w_2} & d_{w_1, w_j} \\ \vdots & \ddots & \circled{d_{w_i, w_j}} \\ 0 & \dots & 0 \end{bmatrix}$$

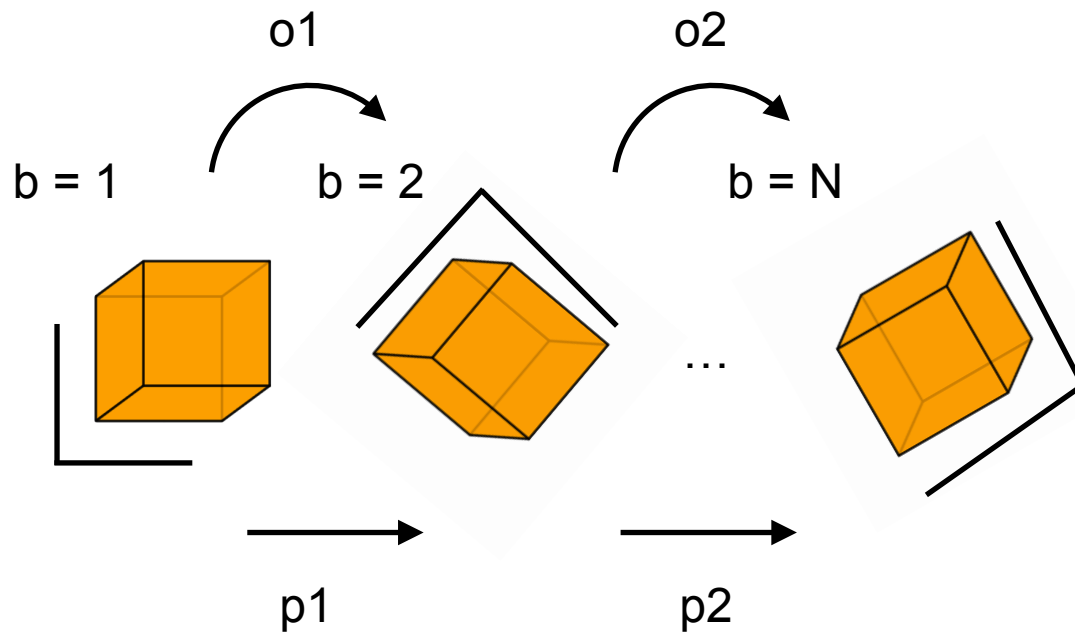


$$d(w_i, w_j) = \sum_{k=1}^{N-1} d(b_{i-k}, b_{j-k})$$

N = window size  
i, j = [1, ..., T] (T: number of frames)  
b = box  
w = window



# Distance Function Algorithm Prototype

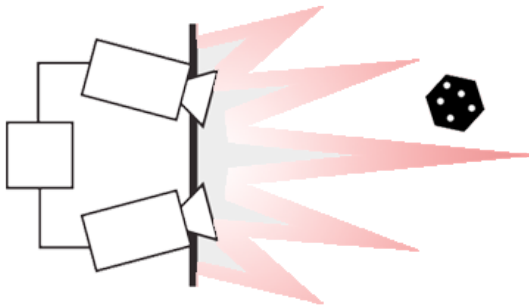


N = window size  
i, j = [1, ..., T] (T: number of frames)  
b = box  
p = translation  
o = orientation

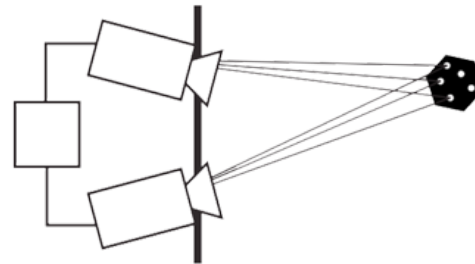
$$d(b_{i-k}, b_{j-k}) = d(p_{i-k}, p_{j-k}) + \lambda d(o_{i-k}, o_{j-k})$$

# Workflow Prototype

## 1. Data retrieval from Optical Tracking System

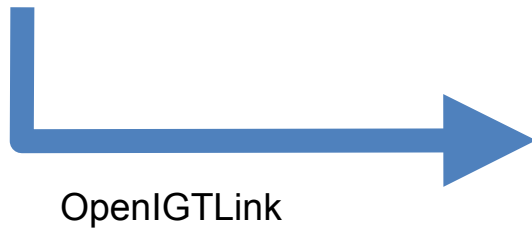


The object is lit using near IR light



Retro-reflective markers reflect back

Source: <http://www.ps-tech.com/3d-technology/optical-tracking>  
Last accessed: 26/6/17



## 2. Text File with pose matrices

```
Trac
-1 0 0 50
0 0.142857 0.989743 0
0 0.989743 -0.142857 50
0 0 0 1

-0.98861 -0.0988095 0.113525 49.0033
0.0988095 0.142857 0.984799 9.93347
-0.113525 0.984799 -0.131467 49.0033
0 0 0 1

-0.954892 -0.196632 0.222525 46.0531
0.196632 0.142857 0.970014 19.4709
-0.222525 0.970014 -0.0977491 46.0531
0 0 0 1
```





# Data Acquisition

**Server / Tracker**



Waits for client at a given host



Connection Successful

**Client / Receiver**



Request for a connection with the server



Network



Data Transfer

```
Trac
-1 0 0 50
0 0.142857 0.989743 0
0 0.989743 -0.142857 50
0 0 0 1

-0.98861 -0.0988095 0.113525 49.0033
0.0988095 0.142857 0.984799 9.93347
-0.113525 0.984799 -0.131467 49.0033
0 0 0 1

-0.954892 -0.196632 0.222525 46.0531
0.196632 0.142857 0.970014 19.4709
-0.222525 0.970014 -0.0977491 46.0531
0 0 0 1
```

Source: [inspect magazine](http://inspectmagazine.com)  
Last accessed: 26/6/17



# Workflow Prototype

## 2. Text File with pose matrices

```
Trac
-1 0 0 50
0 0.142857 0.989743 0
0 0.989743 -0.142857 50
0 0 0 1

-0.98861 -0.0988095 0.113525 49.0033
0.0988095 0.142857 0.984799 9.93347
-0.113525 0.984799 -0.131467 49.0033
0 0 0 1

-0.954892 -0.196632 0.222525 46.0531
0.196632 0.142857 0.970014 19.4709
-0.222525 0.970014 -0.0977491 46.0531
0 0 0 1
```

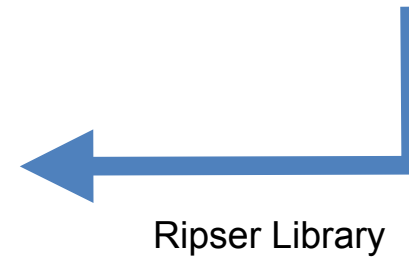
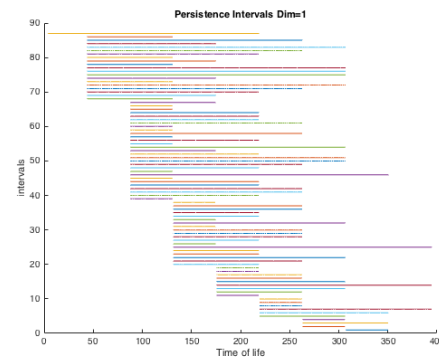
Distance Algorithm



## 3. Square Distance Matrix

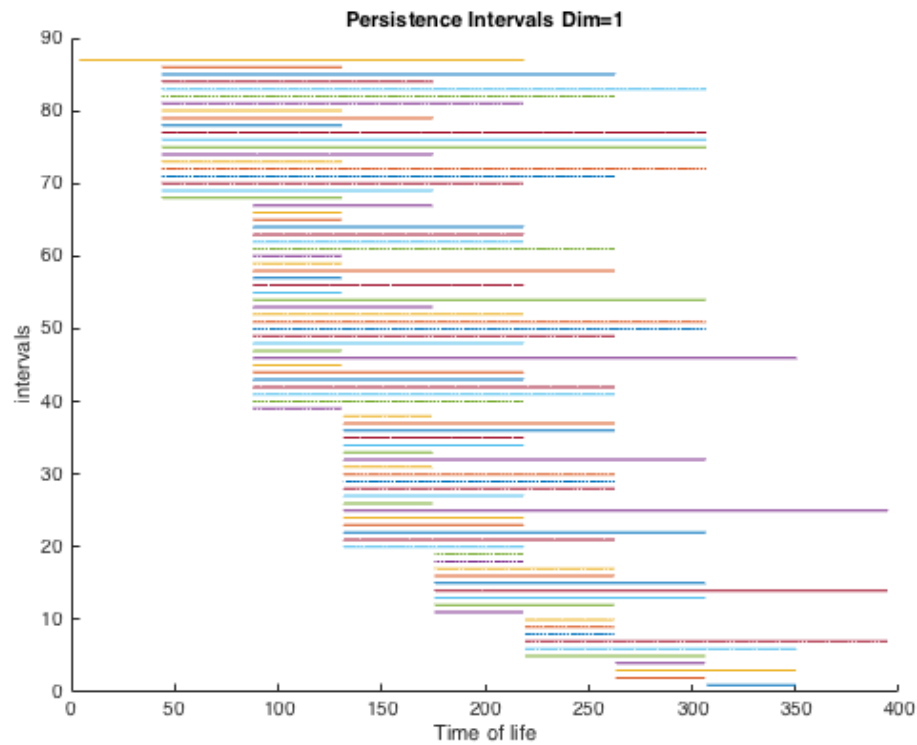
$$\begin{bmatrix} 0 & d_{w_1, w_2} & d_{w_1, w_j} \\ \vdots & \ddots & d_{w_i, w_j} \\ 0 & \dots & 0 \end{bmatrix}$$

## 4. Homology Barcodes

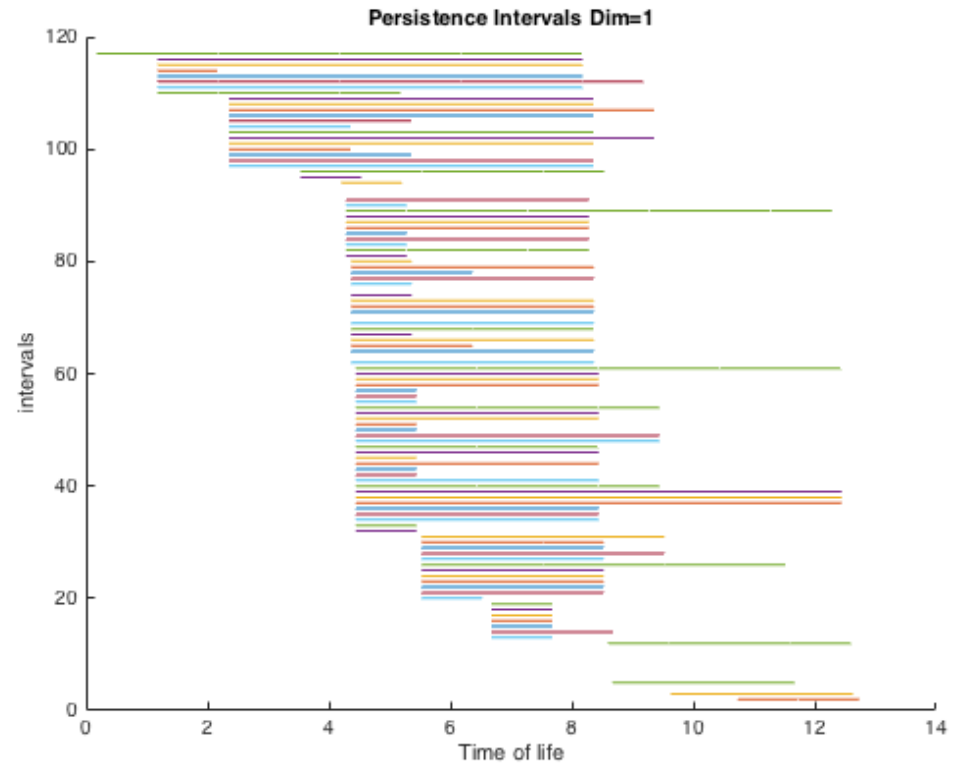


# Barcode Example (Artificial Data)

Linear Movement. Constant velocity

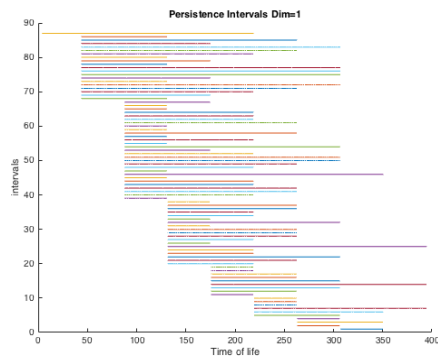


Circular movement. Constant angular velocity



# Workflow Prototype

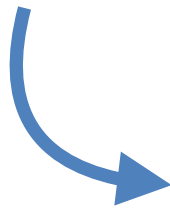
## 4. Homology Barcodes



SVM



**6. Barcode Classifier**  
(To implement)



**5. Persistent Images**  
(To implement)



For further information: <https://arxiv.org/abs/1507.06217>



# Problems Encountered

- Build OpenIGTLink Library in C++ -> Delay in the timeline
- Integration of Ripser Library in Matlab -> File based communication.

# Risks

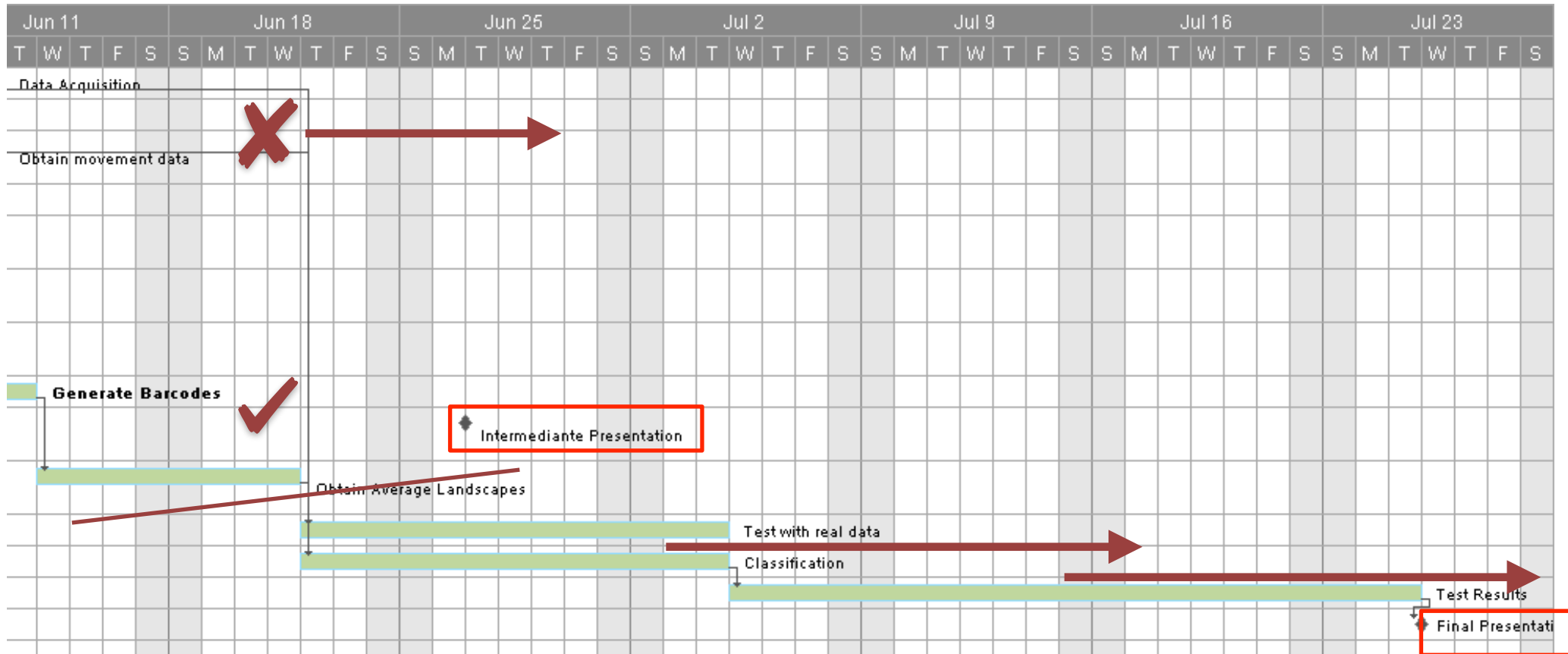
- Lack of robustness in the final classifier
- Long computation times

# Variations from initial scheme

- Persistent barcodes/images as input for classifier instead of average landscapes.



# Timeline



# Questions?





Project Management and System Development

# Activity Classification with Persistent Homology Barcodes

Intermediate Presentation 27/06/2017

Project Supervisor: Benjamin Busam

Presented by: Ester Molero Hidalgo

[ester.molero@tum.de](mailto:ester.molero@tum.de)