Neuro-Evolutive System for Ego-Motion Estimation with a 3D Camera

Ivan Villaverde, Zelmar Echegoyen and Manuel Graña
Computational Intelligence Group
University of the Basque Country
http://www.ehu.es/ccwintco

ICONIP 2008
Neural Information Processing in Cooperative Multi-Robot Systems

Contents

- Introduction
- ToF 3D Camera
  - Data Preprocessing
- Neuro-Evolutive System
- Experimental Settings and Results
- Conclusions

Introduction

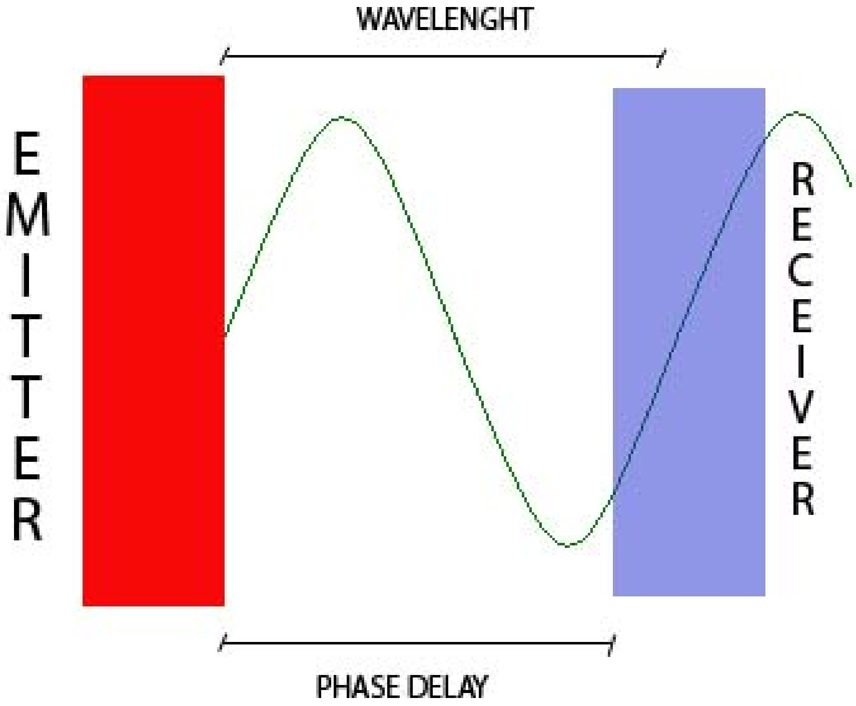
eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea

- Use of new ToF 3D cameras.
- Final objective: full SLAM capabilities on multirobot systems.
- First step: Learn data processing and feature extraction from the 3D data provided by the camera.
- Simple task: ego-motion estimation.

ToF 3D Camera

- SwissRanger SR-3000
- Phase measuring Time of Flight principle.

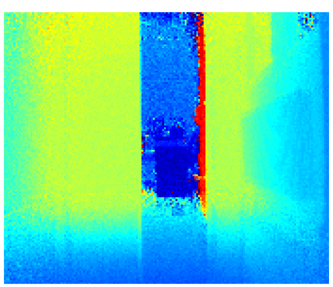
  - Led array illuminates the scene.
  - Known wavelength amplitude.
  - Phase delay used to measure traveled distance.

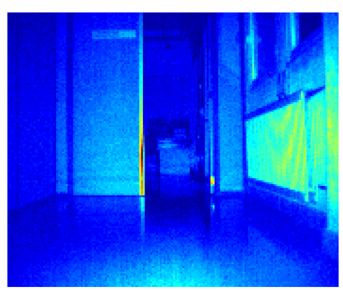


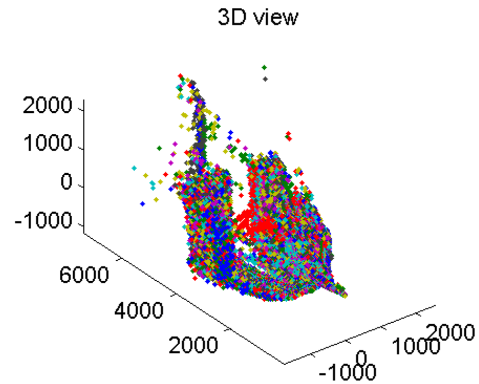
eman ta zabal zazu

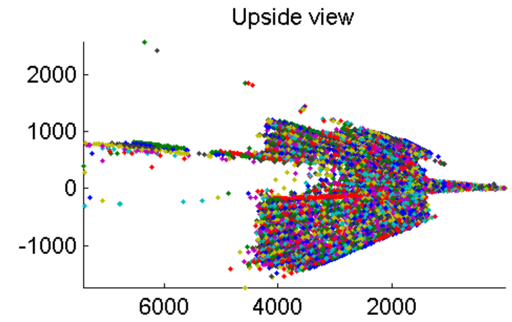
Universidad del País Vasco
Euskal Herriko Unibertsitatea

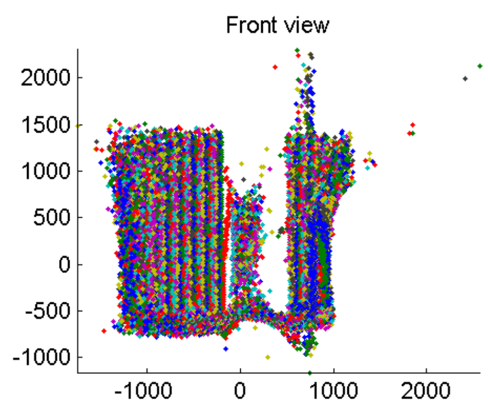
ToF 3D camera

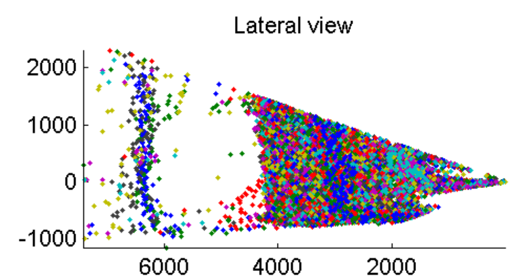










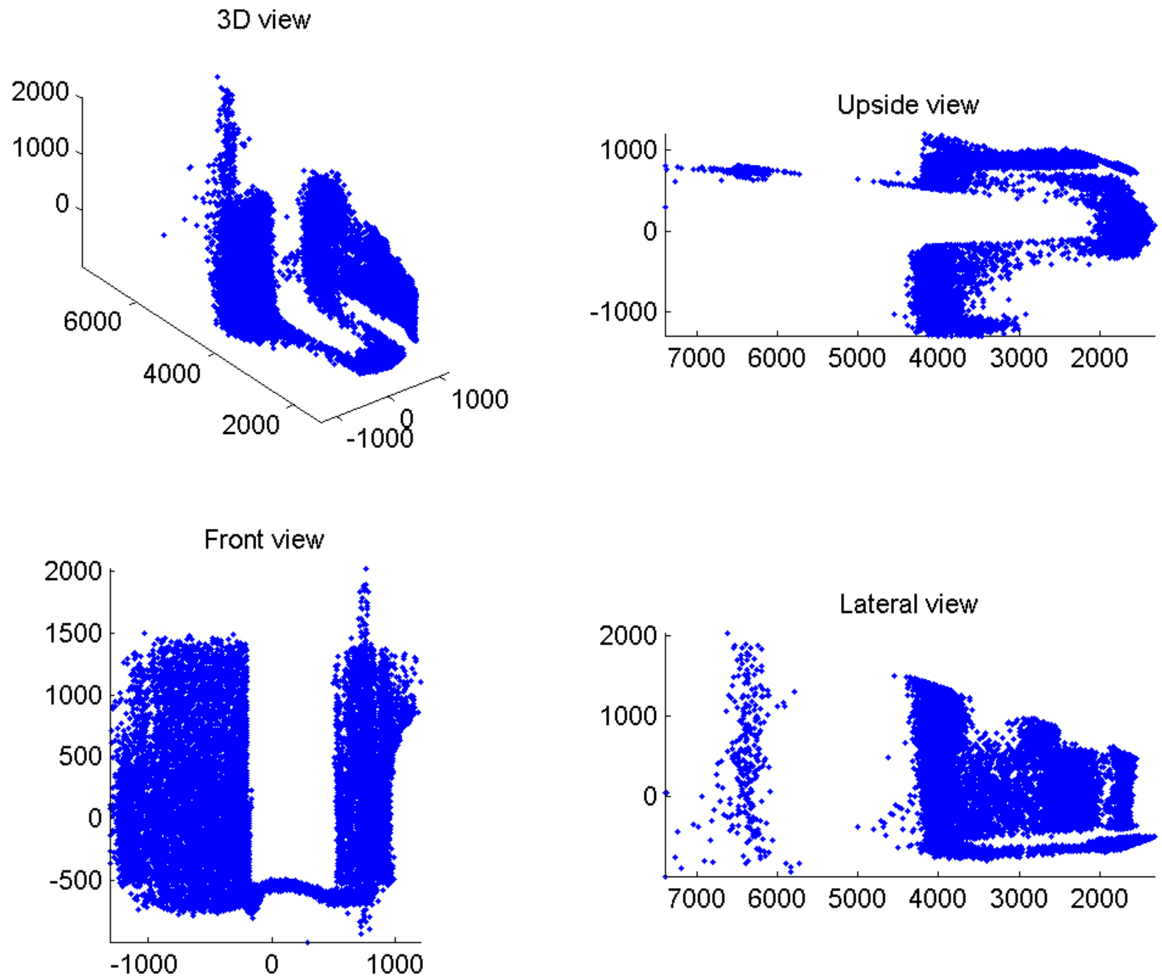


Data Preprocessing

- Pros:
  - Full 3D scene information.
  - On-line operation.
  - On-board operation.
- Cons:
  - Big data size.
  - Ambiguity range.
  - Specular reflections.
  - Measurement uncertainty.

Data Preprocessing

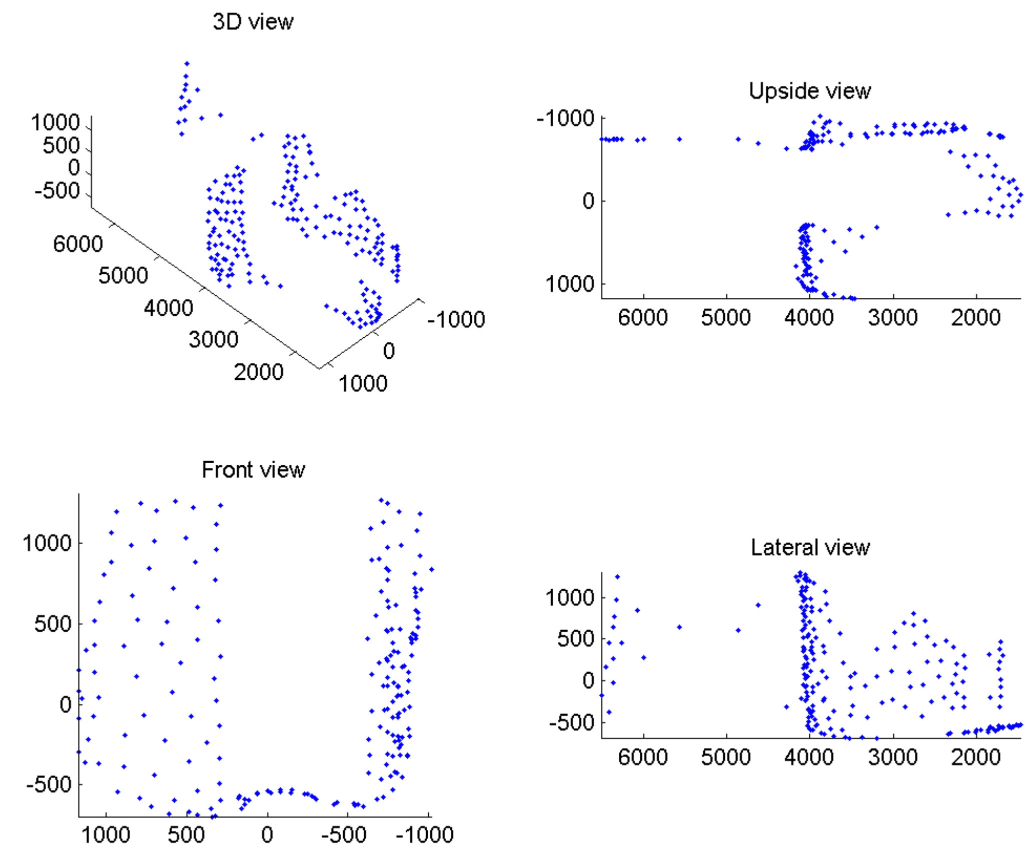
- Thresholding confidence value $C_i = I_i \times D_i$



Neuro-Evolutive System
Neural Gas

- First step: Neural Gas used to generate a codevector set that fits the data.
  - Codevector set *S*.
  - Keeps the spatial shape of the 3D data.
  - Reduces data amount to a fixed, small size.
  - Obtained using the SOM Toolbox for Matlab

http://www.cis.hut.fi/projects/somtoolbox/



Neuro-Evolutive System
Ego-motion Estimation

- Problem Statement:
  - Robot at time *t* defined by:
    - Position: $P_t = (x_t, y_t, \theta_t)$
    - Observed data: codevector set S_t
  - Time *t+1*:
    - S_{t+1} obtained from the camera.
    - P_{t+1} has to be estimated.

Neuro-Evolutive System
Ego-motion Estimation

- P_t and P_{t+1} are nearby points in 2D space.
- Same environment, but observed from different PoV.
  - Most objects visible from P_t should be also visible from P_{t+1}.
  - S_{t+1} should be similar to S_t, after a slight transformation.
  - This transformation gives the spatial relation between P_t and P_{t+1}.
- Objective: estimate the transformation *T* between S_t and S_{t+1}.

Neuro-Evolutive System
Evolution Strategy

- An ES is used to search for the transformation *T*.
- Individuals h_i are hypothesis about position P_{t+1}, and their traits the parameters of the transformation T_i between P_t and hypothesized P_{t+1}.

$$h_i = (x_i, y_i, \theta_i)$$

$$T_i = \begin{bmatrix} \cos(\theta_i - \theta_t) & -\sin(\theta_i - \theta_t) & x_i - x_t \\ \sin(\theta_i - \theta_t) & \cos(\theta_i - \theta_t) & y_i - y_t \\ 0 & 0 & 1 \end{bmatrix}$$

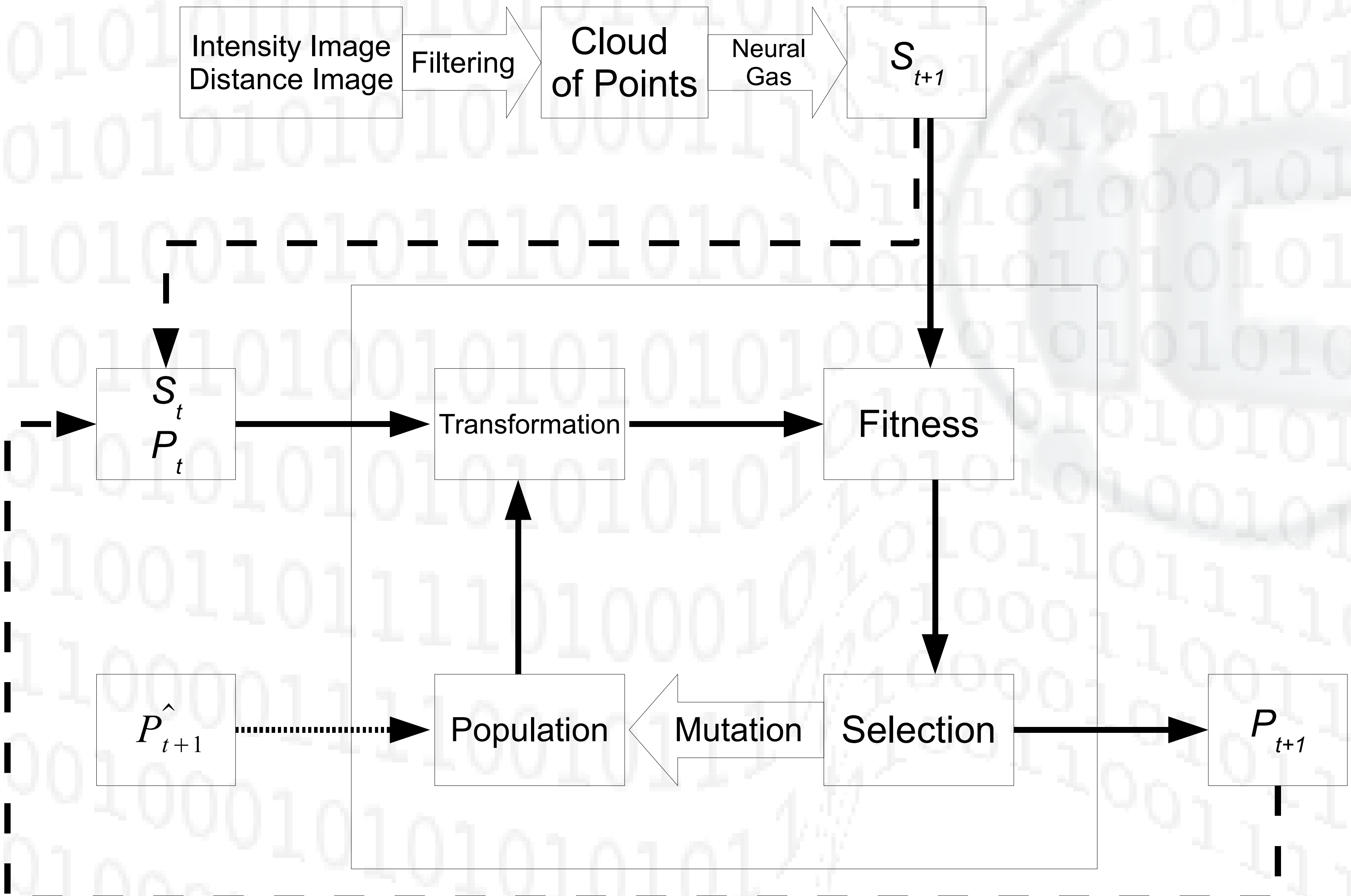
Neuro-Evolutive System
Evolution Strategy

- For each hypothesis h_i we have a prediction of the observed grid:

$$(\hat{S_{t+1}})_i = T_i \times S_t$$

- Fitness function as a matching distance between codevector sets, computed as the sum of the euclidean distances from each codevector in $(\hat{S_{t+1}})_i$ to its closest codevector in S_{t+1}.
- Initial population built randomly from h_0 = (0, 0, 0) (i.e. No transformation: the robot has not moved).
  - Optionally, an *a priori* estimation can be used.

Neuro-Evolutive System
Algorithm Flow Diagram



Experimental Settings

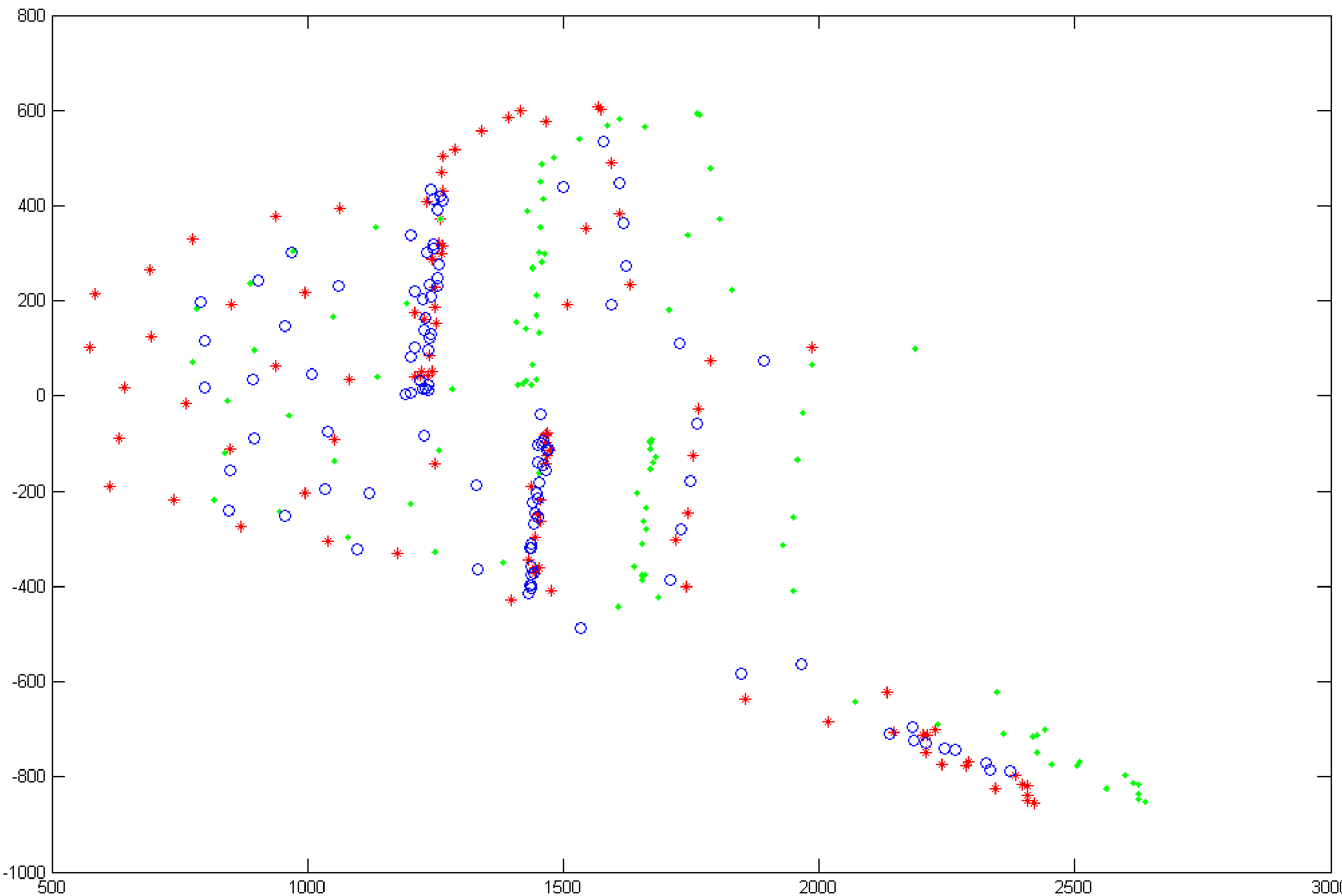
- Pre-recorded walks.
  - Odometry and optical views as reference.
  - Very noisy 3D images due non-optimal configuration.
- Experiment result: Sequence of transformations $T = (T_1, \ldots, T_t)$.
  - Robot position at time t: $P_t = T_t \times \ldots \times T_1 \times P_0$
- 100 node codevector sets.

Experimental Settings

- Evolution strategy implementation:
  - Population of 20 individuals.
  - New generation:
    - The 1/3 best fitted directly.
    - Remaining 2/3 generated from them, crossing pairs and mutating traits with 50% probability.
  - Fitness function as a matching distance between codevector sets, computed as the sum of the euclidean distances from each codevector in $(\hat{S}_{t+1})_i$ to its closest codevector in S_{t+1}.
  - Stopping condition: Best fitted have the same orientation and are within a threshold euclidean distance.

Experimental Results

- Matching results:



S_t = ·

$(\hat{S}_{t+1})_i$ = *

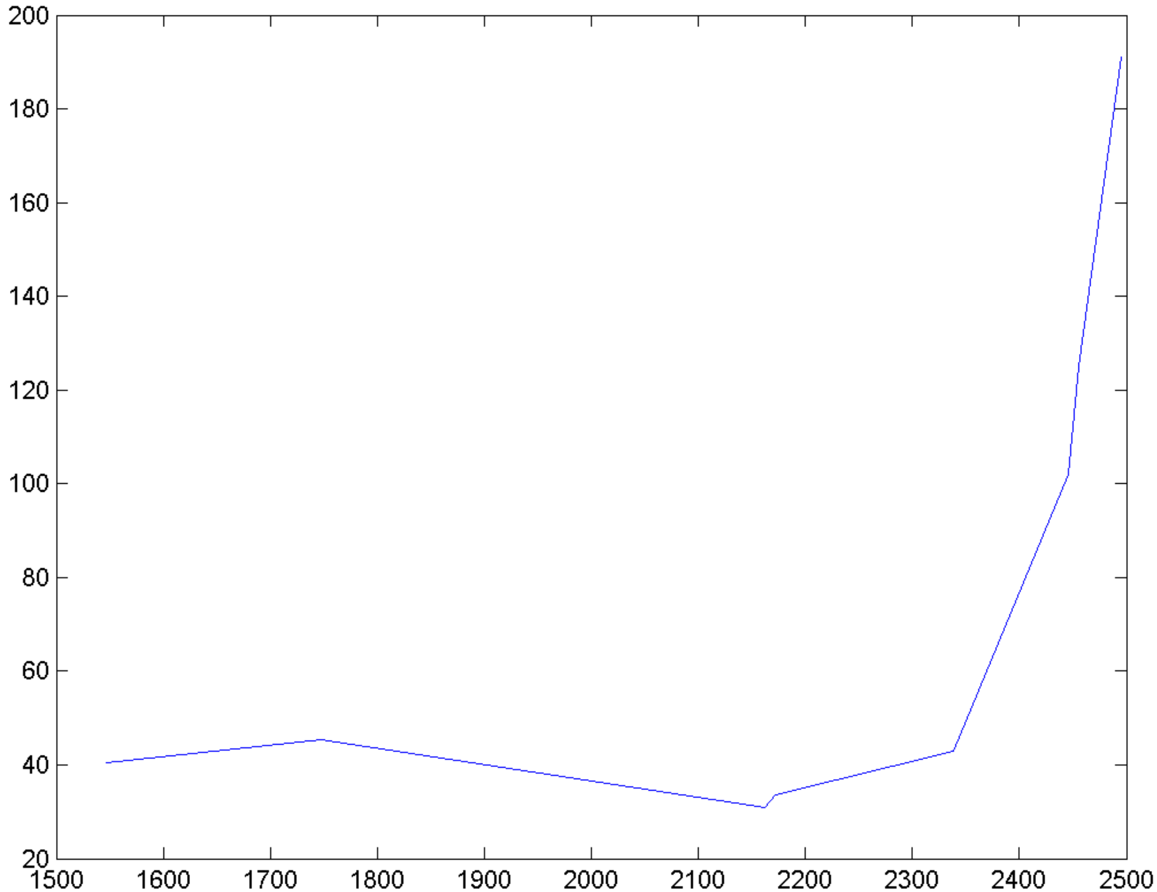
S_{t+1} = o

eman ta zabal zazu

Universidad del País Vasco Euskal Herriko Unibertsitatea

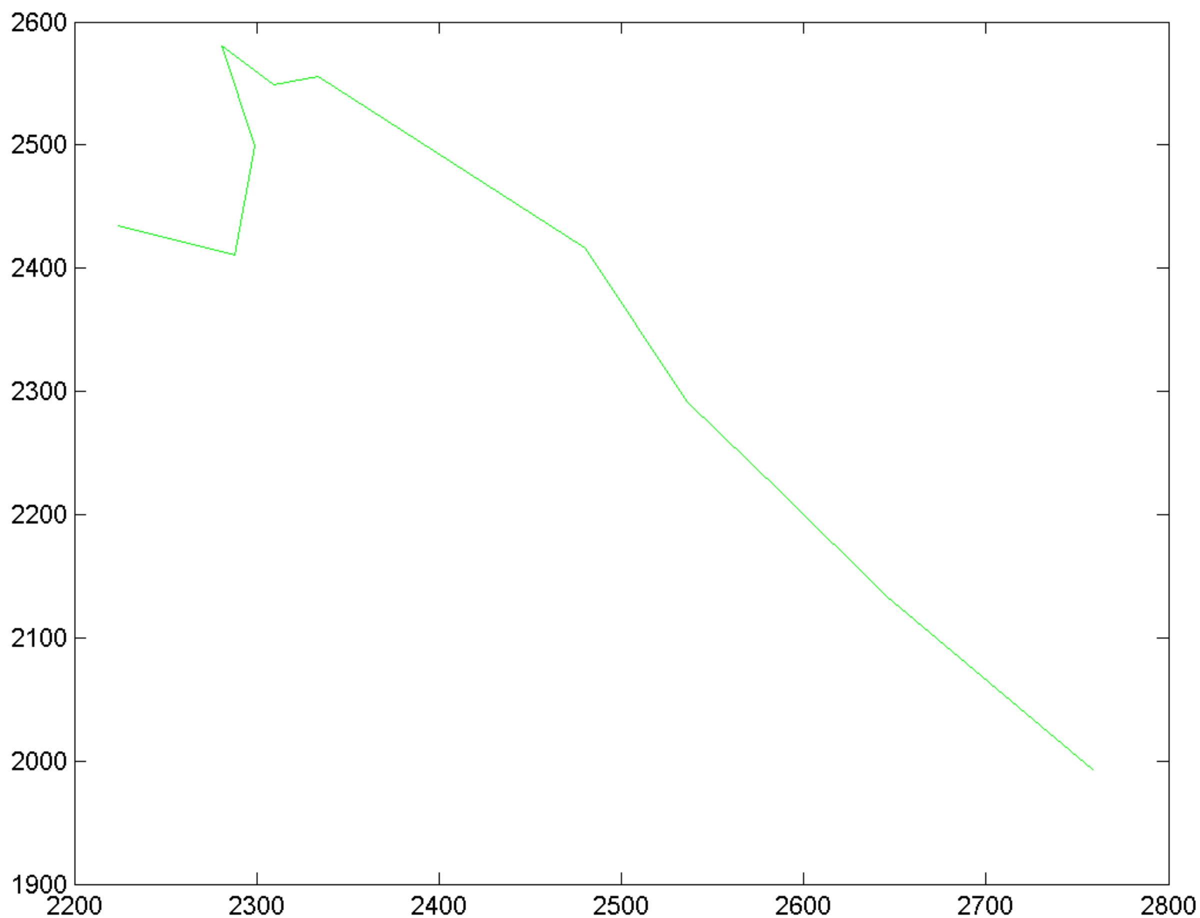
Experimental Results

- Correct ego-motion estimation:



Experimental Results

- Erroneous ego-motion estimation:



Conclusions

- Mobile robot ego-motion estimation algorithm.
  - 3D camera measurements.
  - Neuro-Evolutive system.
    - Neural Gas.
    - Evolution Strategy.
  - Correct estimation with good matching features.
- Future work:
  - Integration in a Kalman or particle filter SLAM architecture.
  - 3D environment reconstruction.
  - Use of point cloud registration techniques.

emán ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Questions, by Oberazzi
http://www.flickr.com/photos/oberazzi/318947873/