UNIVERSITY OF PADOVA

DEPARTMENT OF INFORMATION ENGINEERING

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

# *Distributed approach to dense and sparse camera networks*

Angelo Cenedese

*joint work with Ruggero Carli, Luca Schenato*

FeedNetback co-design for networked control systems

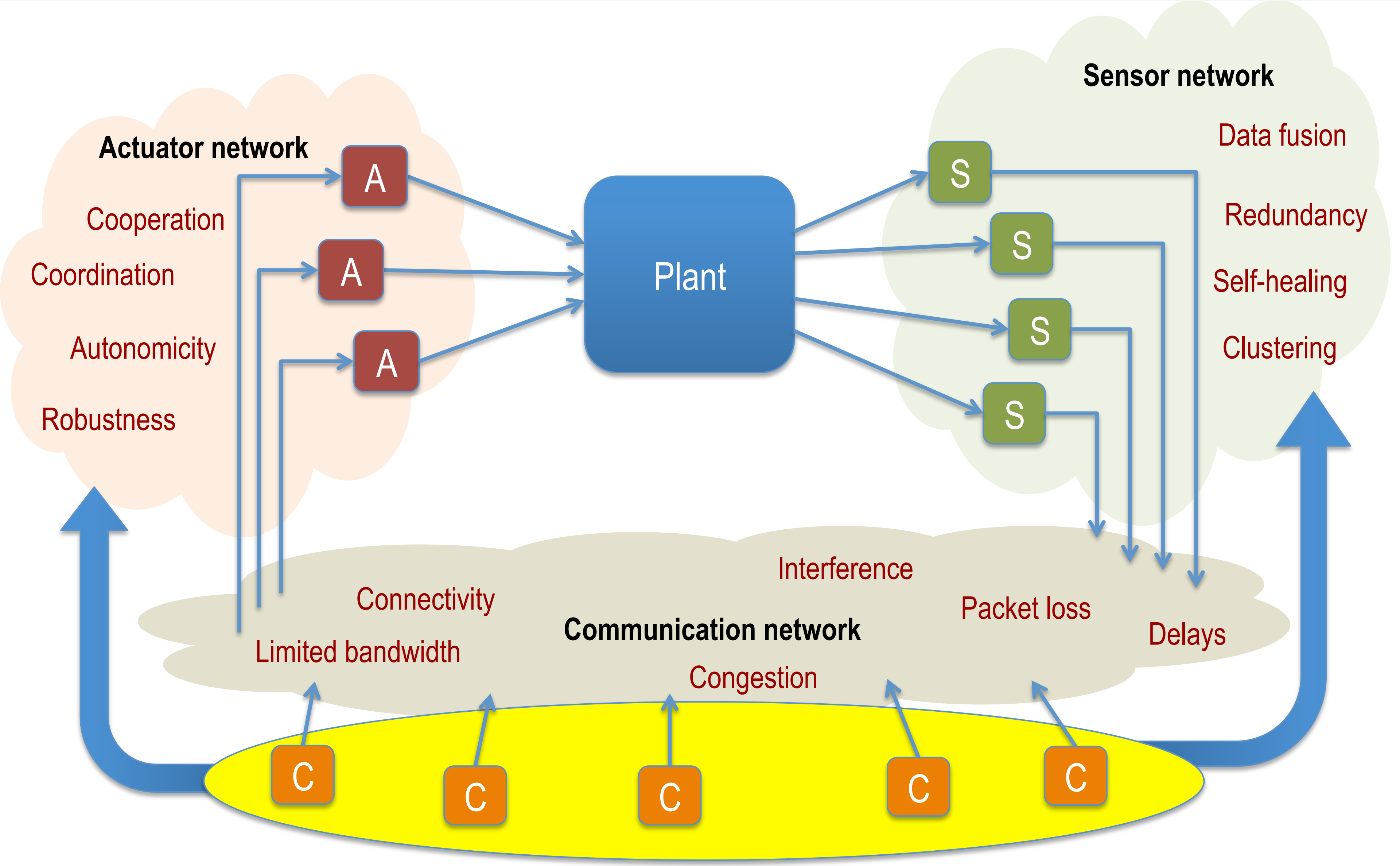
HYCON2 TAMING HETEROGENEITY AND COMPLEXITY

Max Planck Gesellschaft – Tübingen, 5th June 2013

# Outline

- Introduction on distributed systems
- Camera networks
- Sparse camera networks
    - Spatial topology learning
    - Distributed calibration
    - Tracking and patrolling
- Dense camera networks
    - Camera affinity
    - Distributed reconstruction
- Conclusions

# Large scale distributed systems



# Networked Controlled Systems (NCSs)

Sensor Network + Network Controlled Systems

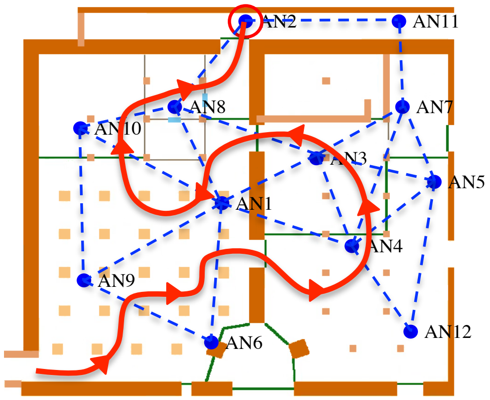
**Sensor and Actor Network (SAN):**

physically distributed dynamical systems interconnected by a communication network.

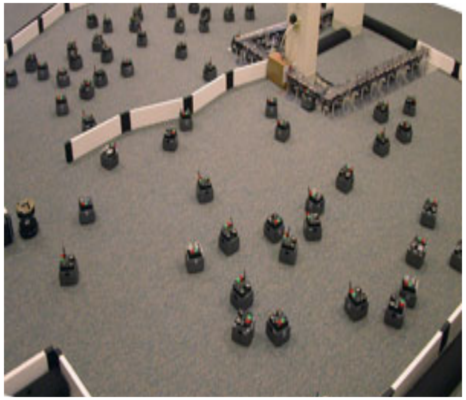
Nodes are sensors (*monitoring*) and/or actuators (*control*).

# Typical applications

## Localization & Tracking

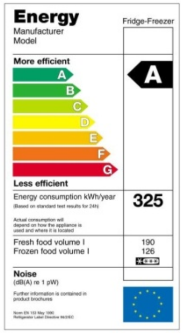



## Swarm robotics

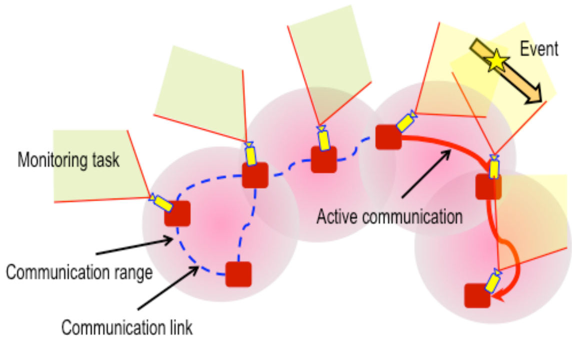


## Indoor monitoring

## Building Energy Management

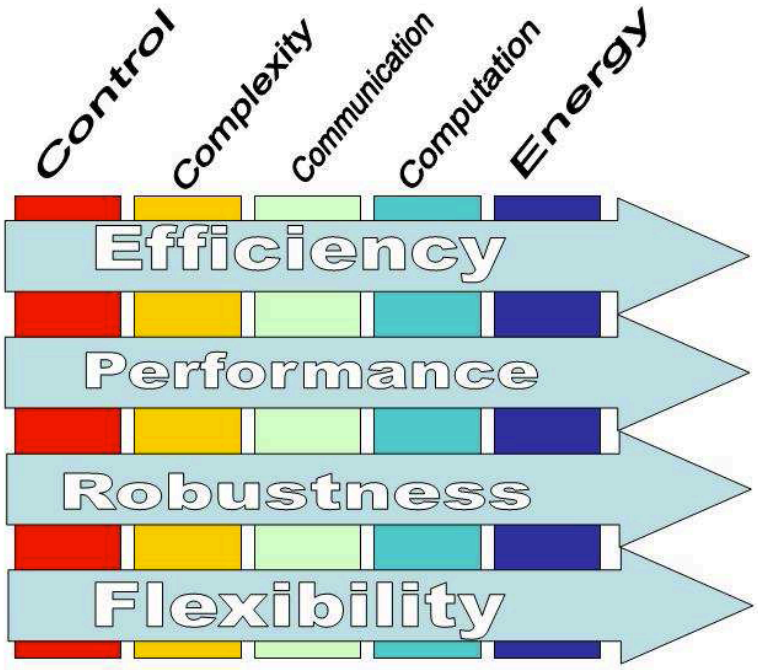



## Camera networks



# The challenge of distributed systems

- To generate a **co-design** framework, to integrate architectural constraints and performance trade-offs.




- This approach:
  - will enable the development of more efficient, robust and affordable networked control systems that **scale** and **adapt** with changing application demands
  - will contribute in mastering **complexity**

# Outline

- Introduction on distributed systems
- Camera networks
- Sparse camera networks
  - Spatial topology learning
  - Distributed calibration
  - Tracking and patrolling
- Dense camera networks
  - Camera affinity
  - Distributed reconstruction
- Conclusions

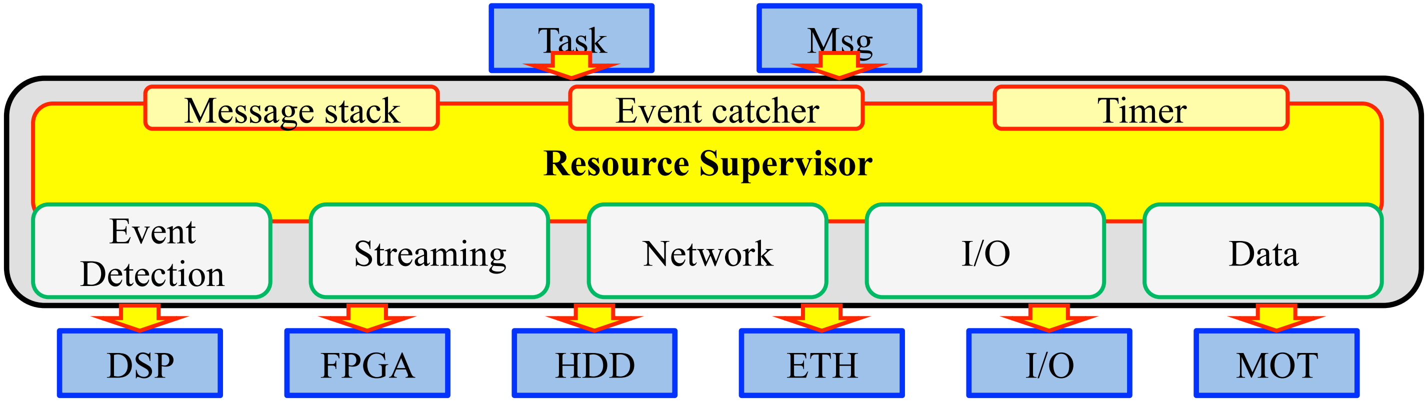
# Camera networks as distributed systems

- Cameras and computing power are both becoming more affordable.
- Companies are building ever larger networks of cameras and there is a drive to distribute the computing locally to the camera. It remains an open problem **how to control** such smart camera networks:
  - **Architecture:** scale and complexity of the system. Need to communicate large amounts of data efficiently and coordinate several cameras.
  - **Control & Complexity:** the number of network nodes in this application will be very large and therefore the issue of scalability will come to the forefront.
  - **Control & Communication:** enormous quantities of data collected by the cameras that, if communicated naively, are likely to swamp any network with traffic.
  - **Control & Computation:** of primary importance will be the issue of Control aware computing, especially since one of the camera platforms used is based around FPGA and DSP on board processors that have slow development cycles and limited computing flexibility.

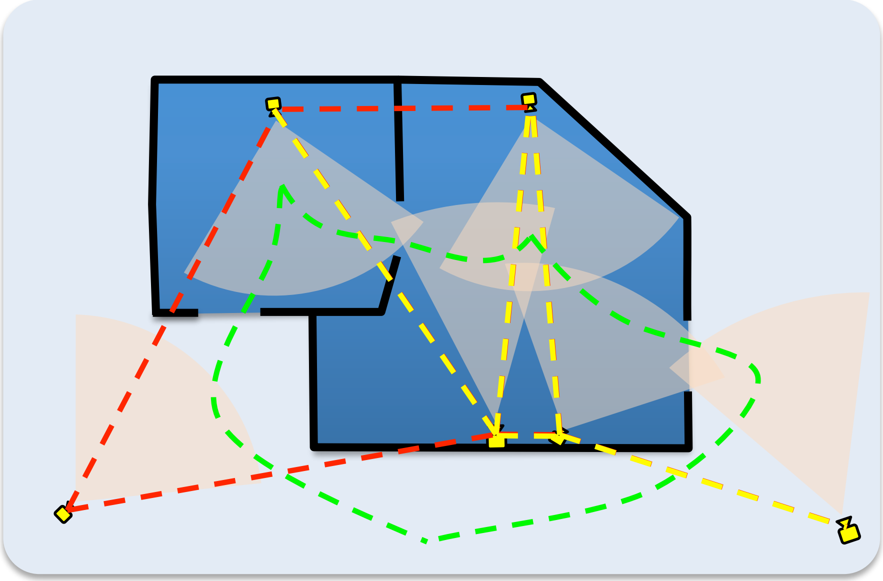
# Camera networks as multiagent systems

- Cameras are **actuated visual sensors** characterized by:
  - **Locality in sensing:** wedge shaped f.o.v. (optical cameras) or range sensor (depth cameras)
  - **Local information exchange** through a communication network
  - **Global** tasks and global performance index
  - **Graph** based interaction models

Camera networks are **smart** camera networks!



# Graph based interaction models



Communication links
Visibility links
Neighboring relationships

$$\mathcal{G} = (\mathcal{V}, \mathcal{E})$$

Agents (cameras)
Areas

- Protocols of interest are:
  - **Consensus:** the agents need to agree to a common decision on a task or a value
  - **Formation:** agents need to coordinate their “motion” for sensing
  - **Assignment:** the agents are resource limited and there is need for a fair assignment of tasks
  - **Coverage:** achieve best area coverage as a network global task
  - **Distributed estimation:** cooperative and collective gathering of data to produce better information

# Camera networks and robotic networks

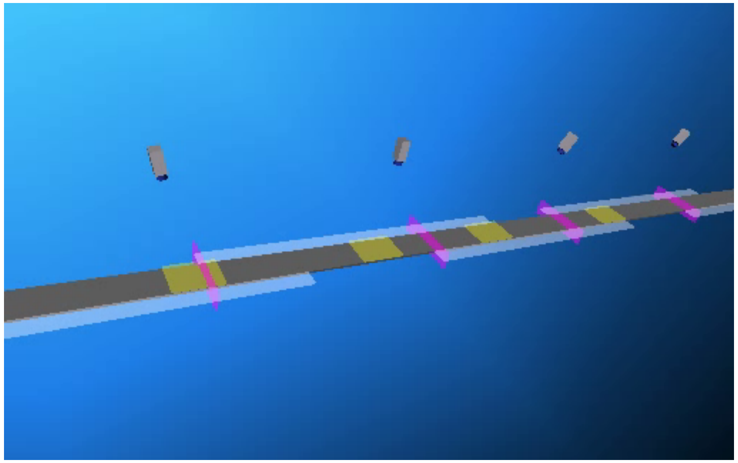
Camera networks and robotic networks share common features and common problems

**Boids models**

1. **Separation:** avoid collisions with your neighbors
2. **Alignment:** steer towards the local heading of your neighbors
3. **Cohesion:** keep contact with your neighbors and avoid being isolated

**Camera network patrolling**

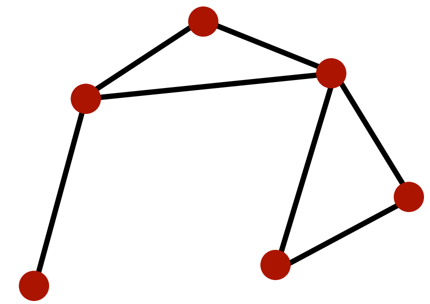
1. **Separation:** minimize the f.o.v. overlap between neighbors
2. **Alignment:** synchronize the speed of patrolling to attain optimal lag time
3. **Cohesion:** agree on the phase of the periodic movement



# Consensus algorithm

- Given a graph and some variable measured by the nodes

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}) \Rightarrow \{x_1, \ldots, x_n\}$$



- Distributed recursive algorithm adapted to the graph:

$$x_i\,(t+1) = f\left(x_1(t), \underbrace{x_{j1}(t), \ldots, x_{jN_i}(t)}_{\in \mathcal{N}_i}, t\right)$$

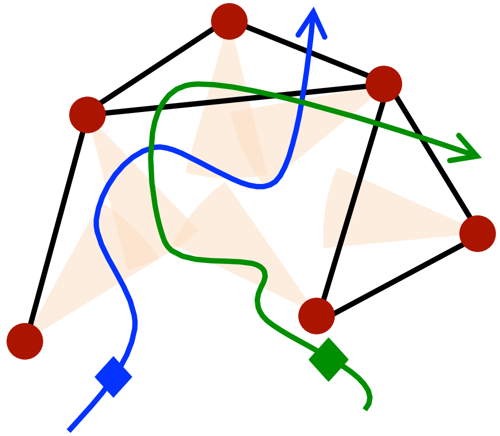
$$x_i\,(t+1) = p_{ii}(t)x_1(t) + \sum_{j \in \mathcal{N}_i} p_{ij}(t)x_j(t) \Rightarrow \boxed{\mathbf{x}(t+1) = P(t)\mathbf{x}(t)}$$

- Solution to the consensus problem:

$P(t)$ solves the
- consensus problem: $\mathbf{x}(t) = P^t \mathbf{x}_0 \longrightarrow \left(\mathbf{1} w_0^\top\right) \mathbf{x}_0$
- average consensus problem: $\mathbf{x}(t) = P^t \mathbf{x}_0 \longrightarrow \dfrac{\mathbf{1}\mathbf{1}^\top}{n} \mathbf{x}_0$

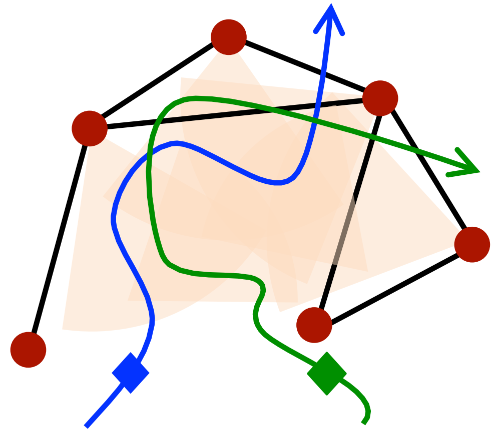
# Sparse and dense camera networks

**Sparse camera networks:**



- Cameras do not necessarily have overlapping fields of view.
- **Issues:** data association, network relationship learning, scene understanding.

**Dense camera networks:**



- Large overlapping fields of view between cameras: geometrical information can be used to calibrate the cameras and reconstruct the shapes and trajectories of objects in the 3D space.
- **Issues:** data and tasks distribution.

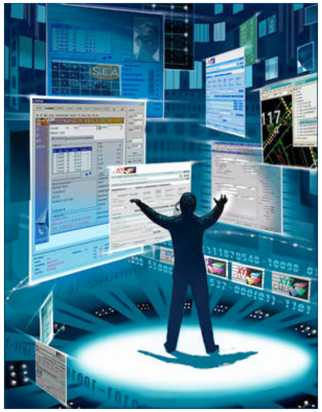
# Outline

- Introduction on distributed systems
- Camera networks
- Sparse camera networks
  - Spatial topology learning
  - Distributed calibration
  - Tracking and patrolling
- Dense camera networks
  - Camera affinity
  - Distributed reconstruction
- Conclusions

# Motivating application: smart video-surveillance systems

Today's reality

Tomorrow's vision



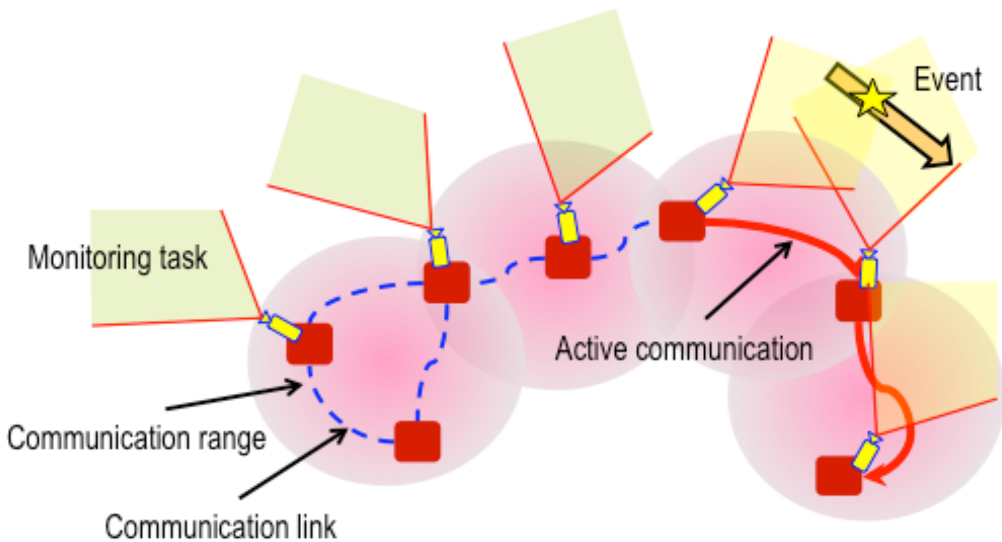
**Co-design approach: C³ communication-control-complexity**

- **"Dynamic" sensing**: sensors are actuated and their parameters tuned according to the dynamics of the scene
- Area coverage and flexibility according to operational needs
- **Scalability** in terms of adding nodes

# Smart video-surveillance system

The smart video-surveillance system realizes the **SAN paradigm**!

- **Actuated**: able to perform a set of tasks/actions such related to the video registration/coding and the interpretation of the scene
- **No central coordinator**: both the information and the surveillance tasks are **shared** among the nodes
- Performance of the system shows slow degradation in presence of faults



# (Some) open problems

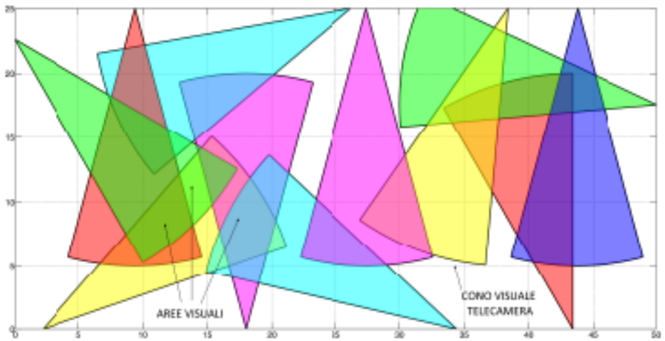
- **Setup phase:**
  - Topology learning and graph building problem: build the graph structure of the network from observations
  - Distributed calibration: compute the relative relationship among camera pairs
- **Operation phase:**
  - Manage patrolling and tracking tasks in a coordinated way
  - Provide autonomic behavior in presence of faults or hacking

# Topology learning: problem statement

- **Assumptions:**
  - States of the system are the **distinguishable** visible areas (overlapping and non overlapping fov's)
  - Observations are given as **binary** strings:

$$O_t \in \{0,1\}^K$$

| | | | | | |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 | 0 |




- **Problem:**
  - Given an observation sequence, infer the set of states and the underlying graph that constraints their transitions:

$$\mathcal{O} \Rightarrow \{\mathcal{S}, \mathcal{G} = (\mathcal{S}, \mathcal{E})\}$$

- **Solution:**

Two-step approach:

  - The correspondence between states and **static** observations provides a first-guess model (states and transition probabilities).
  - The discovery of hidden states is obtained through a splitting procedure from **dynamic** observations and the model is tuned accordingly.

# Topology learning: Hidden Markov Model (HMM)

- HMM is the model that better describes the problem: the state is not known a priori but need to be inferred from observations
  - state transition probability distribution $A \in \mathbb{R}^{N \times N}$:

$$a_{ij} = \mathbf{P}[q_{t+1} = S_j | q_t = S_i] \qquad 1 \leq i, j \leq N$$

  - observation symbol probability distribution $B \in \mathbb{R}^{M \times N}$:

$$b_j(v_i) = \mathbf{P}[O_t = v_i | q_t = S_j] \qquad 1 \leq i \leq M\,, \quad 1 \leq j \leq N\,,$$

  - the initial state distribution $\pi \in \mathbb{R}^N$:

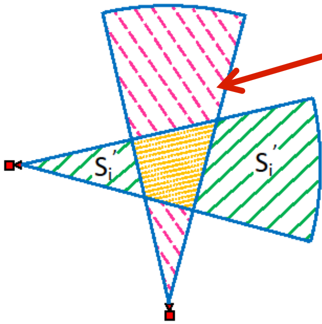
$$\pi_i = \mathbf{P}[q_1 = S_i] \qquad 1 \leq i \leq N\,,$$

**Baum-Welch algorithm:** expectation-maximization method to compute posterior estimates for the model parameters, based on a maximum likelihood approach.

The new model $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ is more likely than the original $\lambda$ to produce the observation sequence (or at least equally likely):

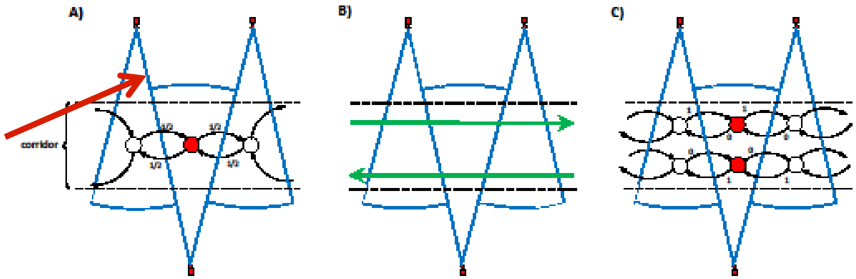
$$\mathbf{P}[\mathcal{O}|\bar{\lambda}] \geq \mathbf{P}[\mathcal{O}|\lambda)]$$

# Topology learning: the state splitting

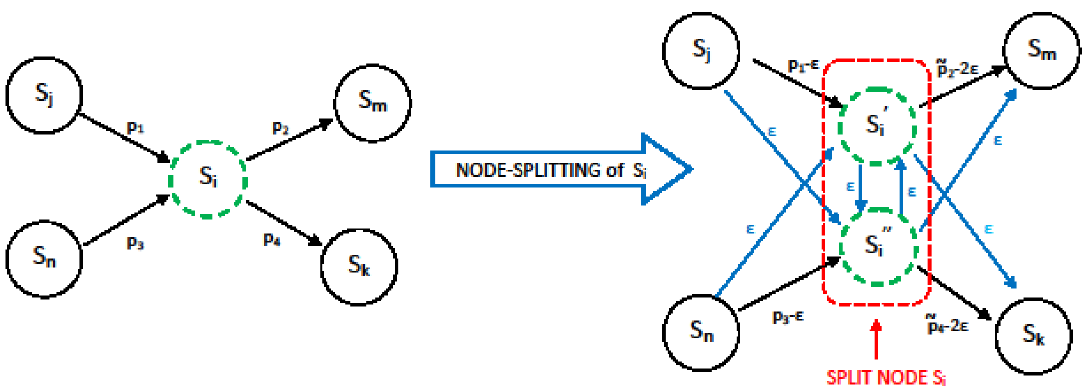


Topological Splitting

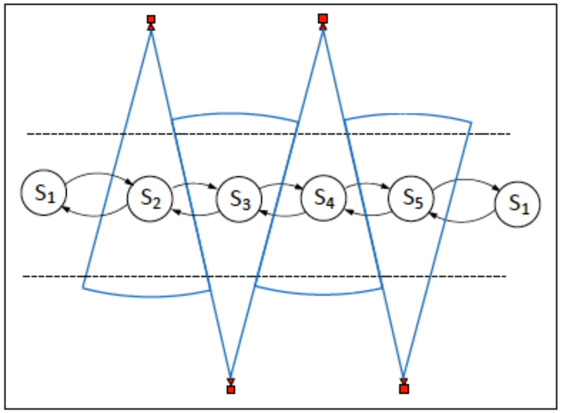
Logical Splitting

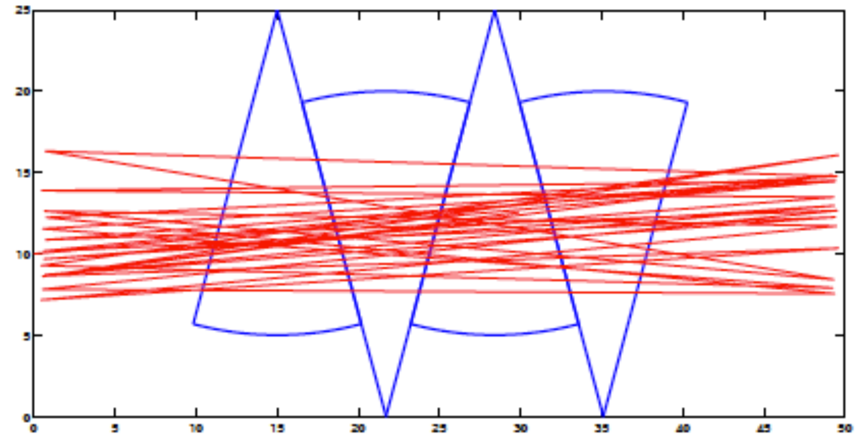


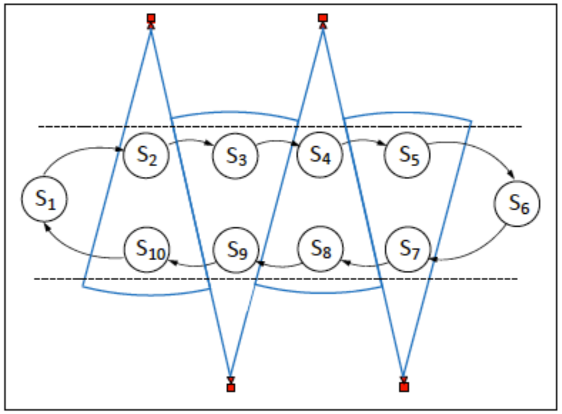

- The 2-step transitions are considered: $q_{t-1} \rightarrow q_t \rightarrow q_{t+1}$
- The orthogonality among the probability sequence is evaluated to split the states as a sort of "innovation measure".

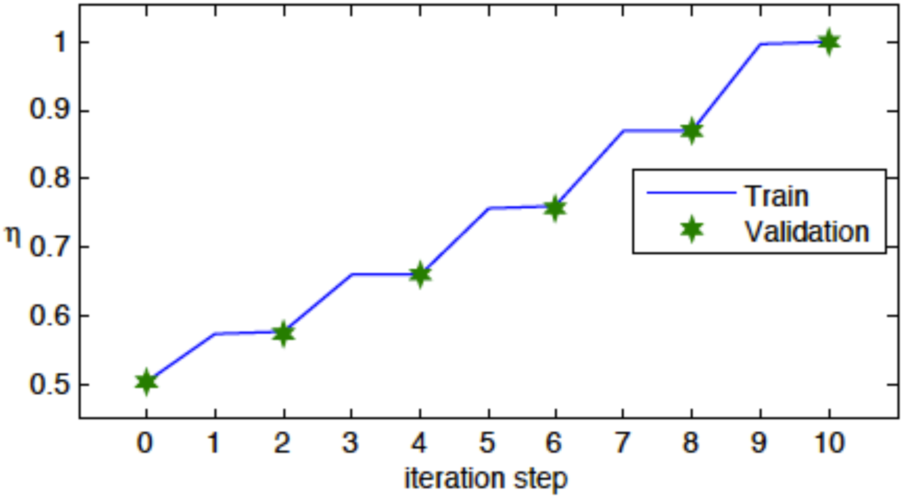


# Topology learning: Scenario I - corridor

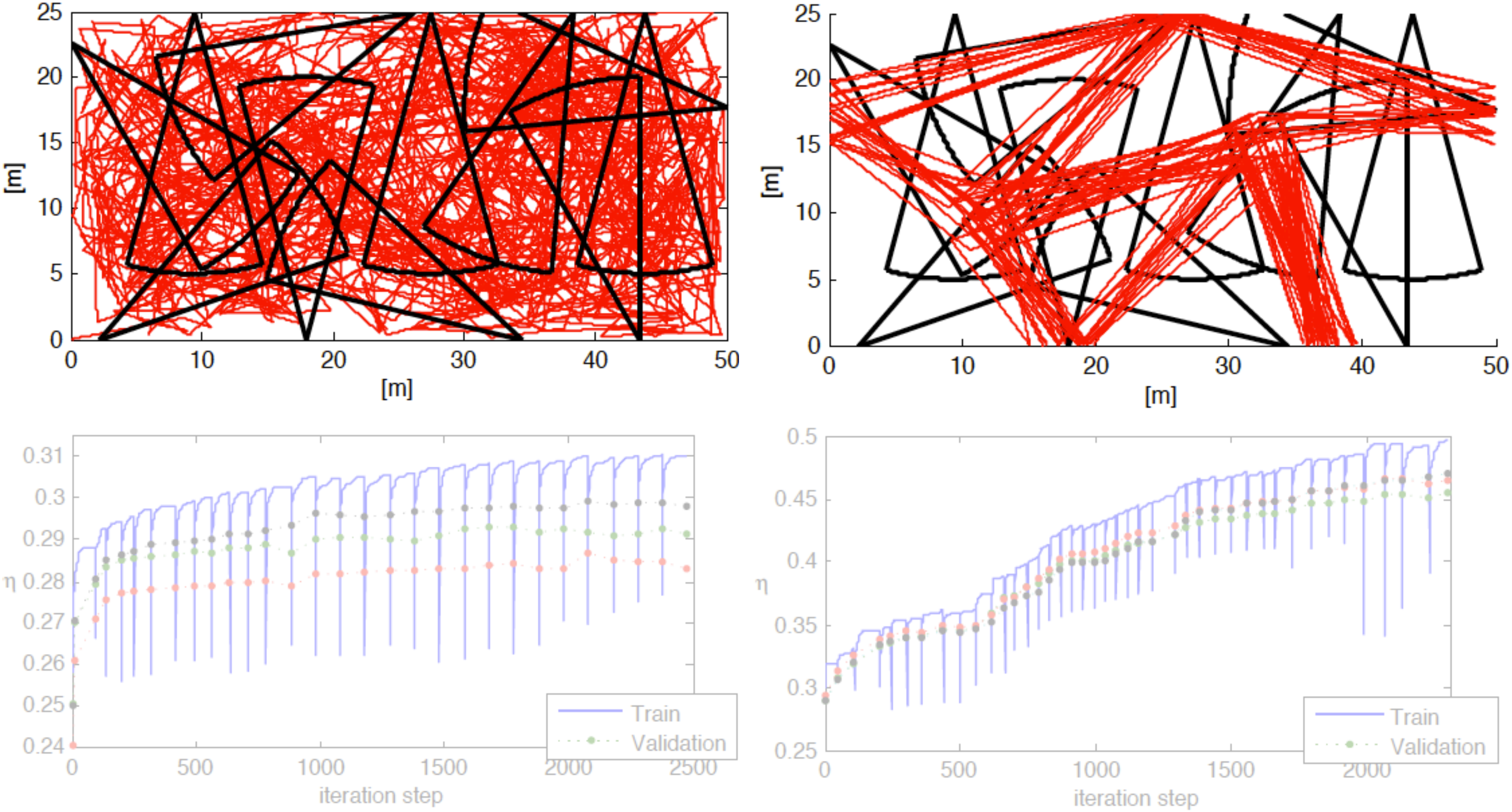


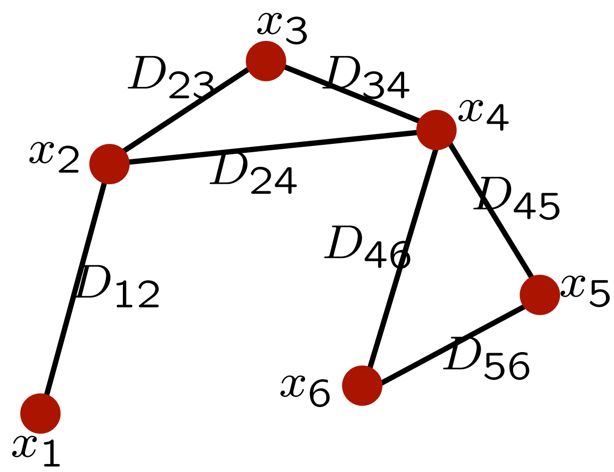



# Topology learning: Scenario II – park




*[Cenedese-Ghirardello-Guiotto-Paggiaro-Schenato, On the graph building problem in camera networks, NECSYS2010]*

# Distributed calibration: global registration




- Given a camera network and relative noisy distance measurements:

$$\overline{D_{ij}} = D_{ij} + \Delta D_{ij} \qquad \Delta D_{ij} \in \mathcal{N}(0, C_{ij})$$

- Recover the cameras' pose: $x_i \in \mathbb{R}^6$

$$D_{ij}^* = \arg\max_{D_{ij}} f(\overline{D_{ij}} | D_{ij})$$

$$\{D_{ij}^*\} = \arg\min_{\{D_{ij}\}} \sum (\overline{D_{ij}} - D_{ij})^\top C_{ij}^{-1} (\overline{D_{ij}} - D_{ij})$$

Mahalanobis distance

- In matricial form:

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \underbrace{(\bar{\mathbf{D}} - H\mathbf{X})^\top C^{-1} (\bar{\mathbf{D}} - H\mathbf{X})}_{W}$$

*$H$: visibility/communication matrix*

$$\frac{\partial W}{\partial \mathbf{X}} = 0 \quad \Rightarrow \quad \mathbf{X}^* = \left(H^\top C^{-1} H\right)^{-1} H^\top C^{-1} \bar{\mathbf{D}}$$

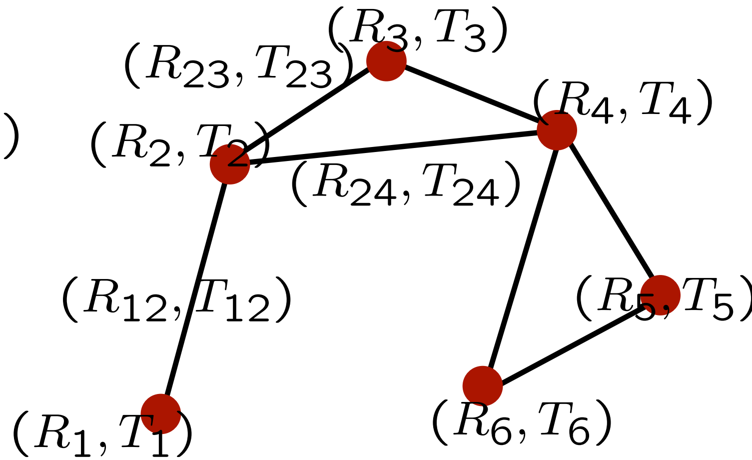
*[Lu-Milios, Globally Consistent Range Scan Alignment for Environment Mapping, 1997]*

# Distributed calibration: global registration

- Given a camera network and the pose representation:

$$R_i \in SO(3) + T_i \in \mathbb{R}^3 \Rightarrow g_i = (R_i, T_i) \in SE(3)$$

retrieve each camera pose from relative measurements $(\overline{R_{ij}}, \overline{T_{ij}})$




- Introduce the geodesic distance in *SO(3)*: $d_{SO(3)}(R_{ij}, \overline{R_{ij}})$

- Define the functional:

$$\varphi = \frac{1}{2} \sum_{\mathcal{E}} d^2_{SO(3)}(R_{ij}, \overline{R_{ij}}) + \|T_{ij} - \overline{T_{ij}}\|^2$$

"Pose reconstruction error"

- Solve the constrained minimization problem in a distributed fashion by separation:

$$\min_{\{R_i\},\{T_i\},\{\lambda_{ij}\}} \varphi(\{R_i\}, \{T_i\}, \{\lambda_{ij}\}) = \frac{1}{2} \sum_{\mathcal{E}} \underbrace{d^2_{SO(3)}(R_i^\top R_j, \overline{R_{ij}})}_{\varphi_R(\{R_i\})} + \underbrace{\|R_i^\top (T_j - T_i) - \lambda_{ij} \overline{t_{ij}}\|^2}_{\varphi_T(\{R_i\},\{T_i\},\{\lambda_{ij}\})}$$

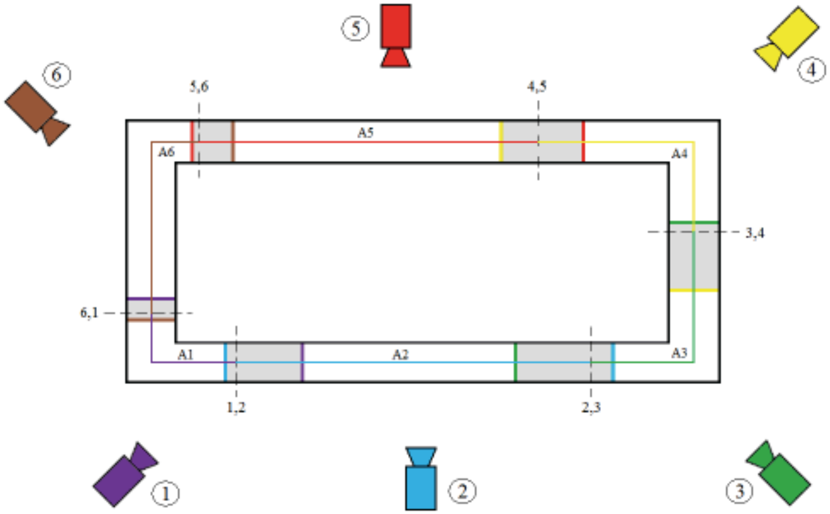
*[Tron-Vidal, Distributed Image-Based 3D Localization of Camera Sensor Networks, CDC2009]*

# Perimeter patrolling: introduction

- **Patrolling:** act of an agent that senses a different portion of the environment at a time, in order to detect events or anomalies

- **Problem:** Real-time PTZ multi-camera optimal perimeter patrolling
  - **Fixed position** cameras with **different** speeds and visible portions of the perimeter
  - Each camera is an **autonomous agent** capable of communication and independent decision making

- **Goal:** to provide an algorithm to achieve optimal patrolling policy
  - Distributed
  - Asynchronous
  - Parallelizable
  - Adaptive



# Perimeter patrolling: introduction


- **Robotic networks:** autonomous vehicles move in the environment to attain the optimal area coverage and to act coordinately

- **Smart camera networks:** particular case of the multiagent network to design an optimal patrolling strategy for the patroller, in terms of Pan-Tilt-Zoom (PTZ) commands

- Systems **constrained** by their motion dynamics, by their sensing capabilities, and by the communication protocols

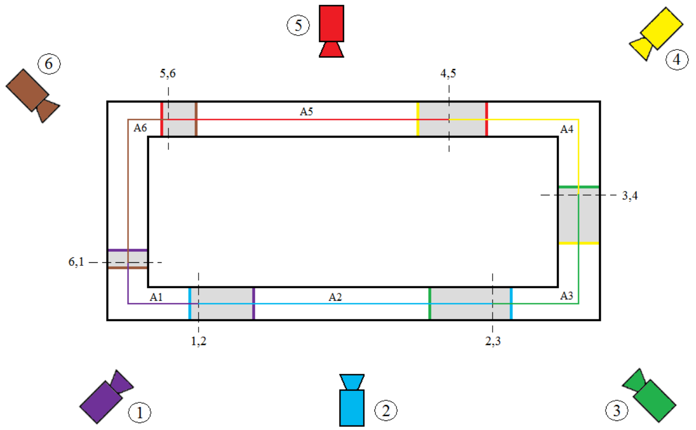
*[Pavone-Arsie-Frazzoli-Bullo, Equitable partitioning policies for robotic networks, ICRA09]*

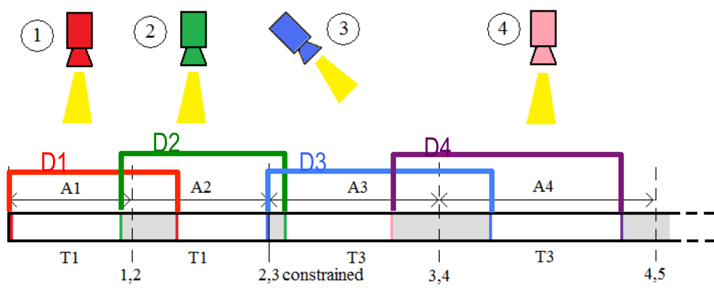
# Perimeter patrolling: rationale

- The focus is on the **coordination** and **communication** among cameras, studied in the context of distributed algorithms.

1. The problem of optimal perimeter patrolling is abstracted into a **distributed optimization problem** subject to **constraints** and with **asynchronous** communication and updates.

2. Strategy features:
   - Cameras **communicate** and **coordinate** only with the preceding and the following
   - Cameras **optimally** cooperate to patrol/partition the perimeter
   - Partitions require the same **minimal time** to patrol



# Perimeter patrolling: conjecture

- **Conjecture: patrolling as partitioning**

In a multiagent framework, the optimal strategy is **obtained by partitioning** the perimeter into subdomains and allowing the agents to patrol their domains to and fro at maximum speed

- Centralized optimal partitioning with

$$T_i^* = T_{lag}^*(A_i) = \frac{2|A_i|}{\overline{v}_i} = \frac{2(r_i - \ell_i)}{\overline{v}_i}$$

- Optimization problem:

$$\mathcal{P}_1 : \quad \begin{array}{lll} & \min_{A_1,\dots,A_N} & \max_i \{T_{lag}^*(A_i)\} \\ & s.t. & A_i \subseteq D_i \quad i = 1, \dots, N \\ & & \cup_{i=1}^N A_i = \mathcal{L} \end{array}$$

*[Czyzowicz-Gasieniec-Kosowski-Kranakis, Boundary Patrolling by Mobile Agents with Distinct Maximal Speeds, 2011]*

# Perimeter patrolling: optimal partitioning problem

- LP problem:

$$\mathcal{P}_2: \quad \begin{array}{lll} & \min_{\tau,\{r_i\}_{i=1}^{N-1},\{\ell_i\}_{i=2}^{N}} 2\tau & \\ s.t. & \frac{r_i-\ell_i}{\overline{v}_i} \le \tau & i=1,\dots,N \\ & \underline{d}_i \le \ell_i \le \overline{d}_i, \quad \underline{d}_i \le r_i \le \overline{d}_i & i=1,\dots,N \\ & r_i \ge \ell_{i+1} & i=1,\dots,N-1 \end{array}$$

$where\ \underline{d}_1 = \ell_1 = 0\ and\ \overline{d}_N = r_N = L$

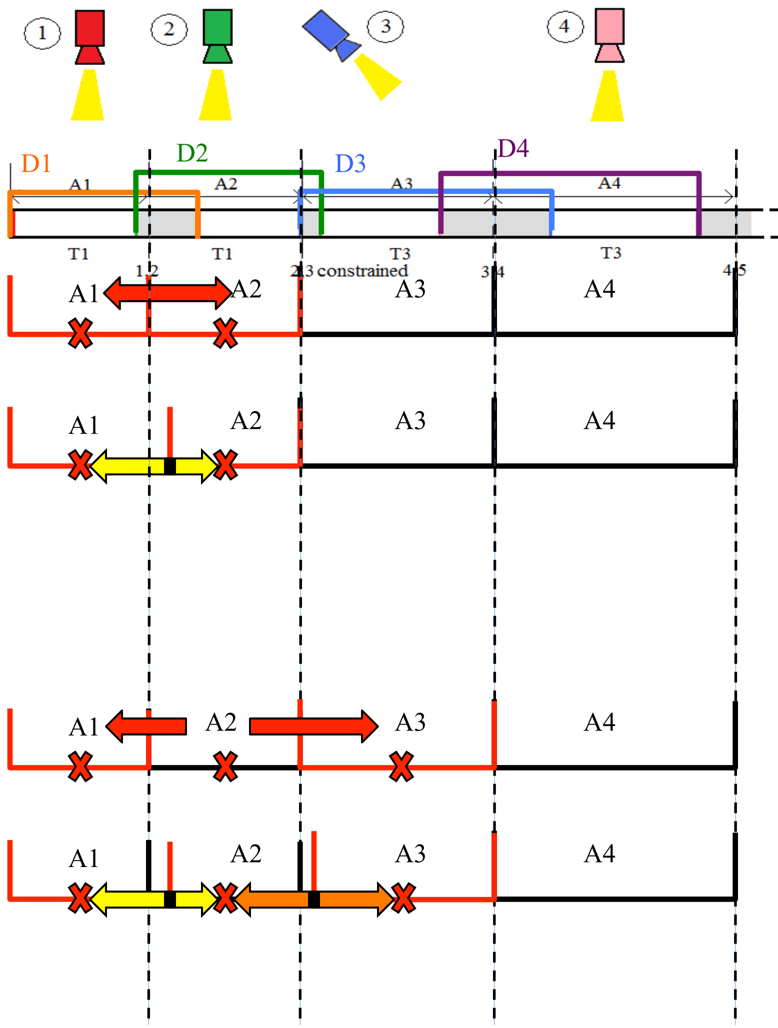
→ centralized solution that can be computed in a distributed fashion

- New optimization problem: unique solution as one of the original problem solutions

$$(\mathcal{P}_3) \quad \begin{array}{ll} \min_{\{r_i\}_{i=1}^{N-1},\{\ell_i\}_{i=2}^{N}} \sum_{i=1}^{N} \frac{1}{\overline{v}_i}(r_i-\ell_i)^2 & \\ \underline{d}_i \le \ell_i \le \overline{d}_i, \quad \underline{d}_i \le r_i \le \overline{d}_i & i=1,\dots,N \\ r_i \ge \ell_{i+1} & i=1,\dots,N-1 \end{array}$$

  - under gossip communication: scalable and parallelizable distributed algorithms
  - uniqueness of the minimizer yields convergence of iterative numerical algorithms

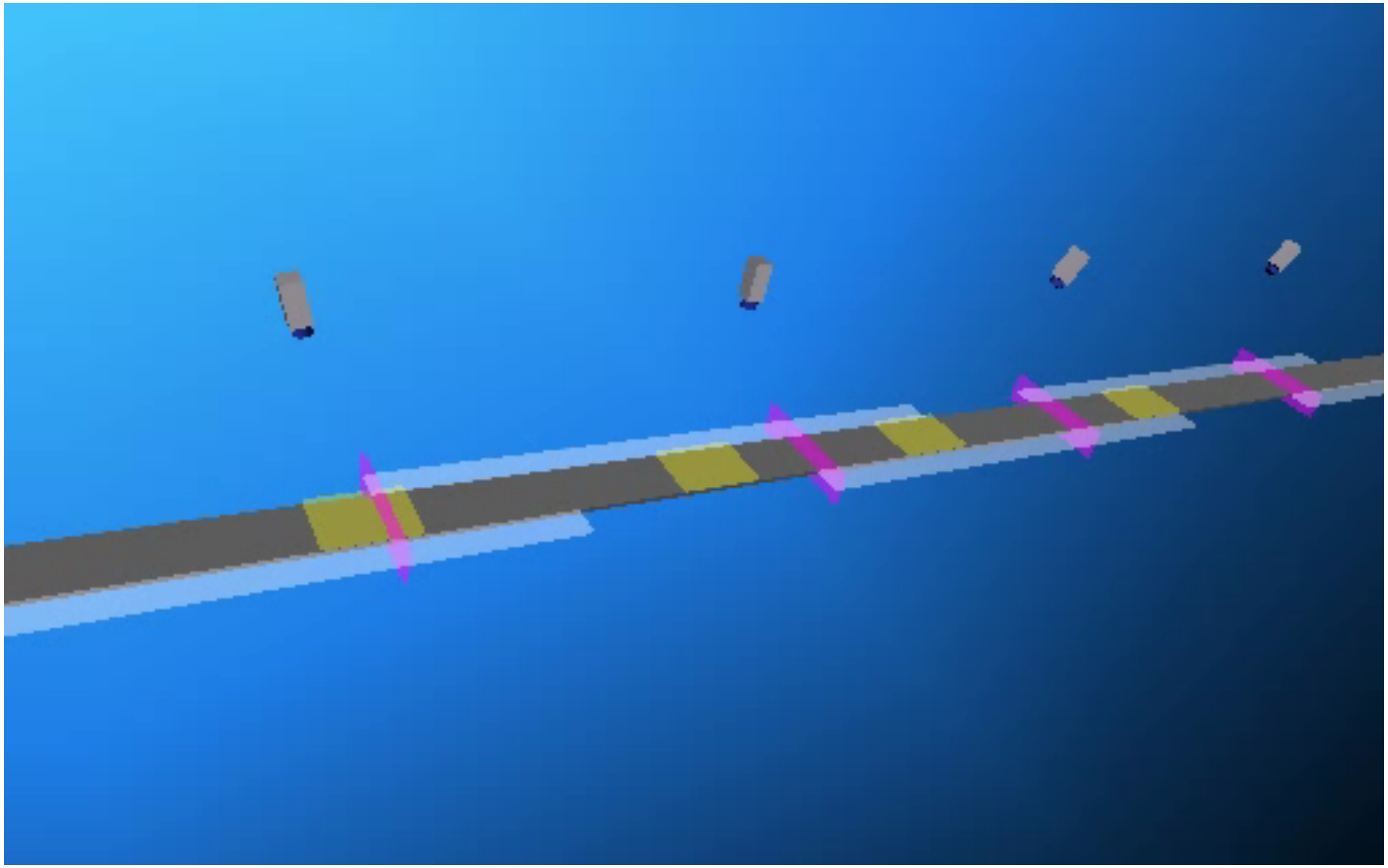
# Perimeter patrolling: communication schemes



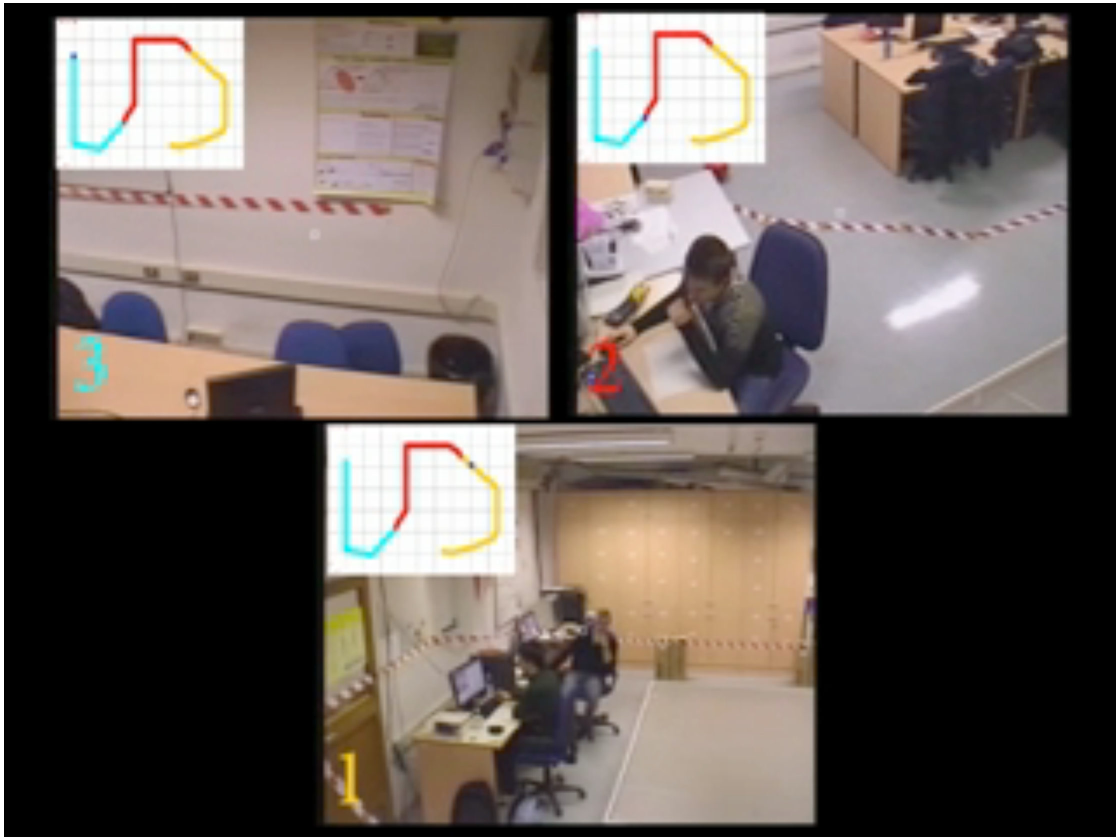

- **Symmetric Gossip algorithm**
  - Transmission iteration:
  - Extremes' iteration:
- **Asymmetric Broadcast algorithm**
  - Transmission iteration:
  - Extremes' iteration:
- **Deterministic/Probabilistic convergence**

*[Alberton-Carli-Cenedese-Schenato, Multi-agent perimeter patrolling subject to mobility constraints, ACC12]*

# Perimeter patrolling: Symmetric Gossip strategy



# Perimeter patrolling: Asymmetric Broadcast strategy

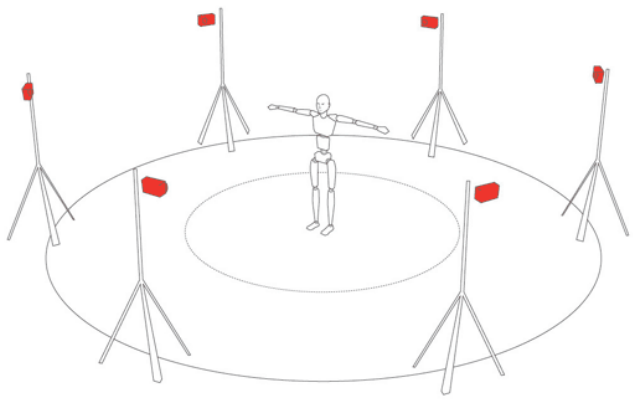


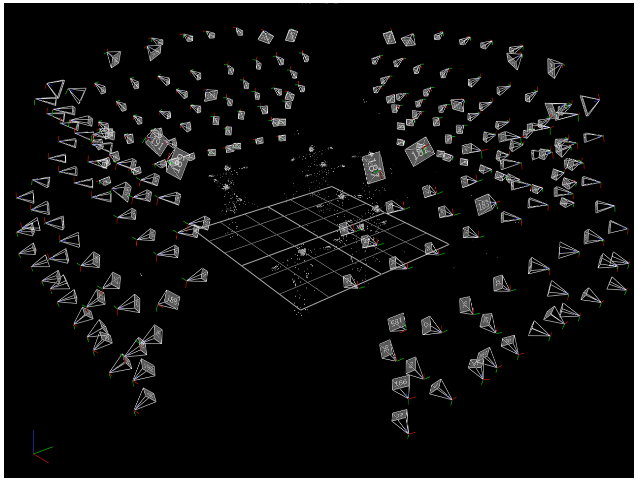
# Outline

- Introduction on distributed systems
- Camera networks
- Sparse camera networks
  - Spatial topology learning
  - Distributed calibration
  - Tracking and patrolling
- Dense camera networks
  - Camera affinity
  - Distributed reconstruction
- Conclusions

# Large scale MoCaps: distributed approach?

- The request of a **more and more accurate** estimation of the target positions is leading to the use of large camera network systems.
- The **real time use** of the system imposes stringent computational requirements.





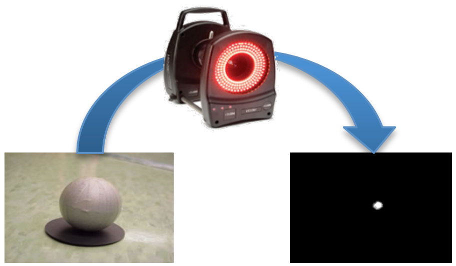
**Use the network as a computational grid!**

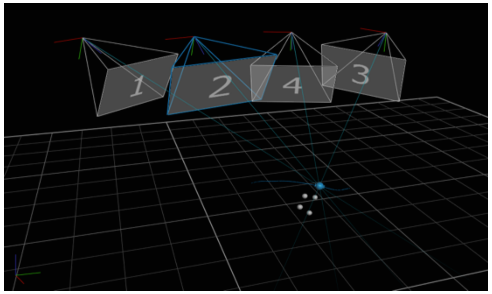
1. A binary tree structure: spreading the calculation on different nodes/cameras to tackle the issue of computational complexity when scaling up
2. A node pairing strategy: define and compute a suitable affinity measure between cameras
3. An algorithm: the correctness and the termination of the algorithm should be guaranteed

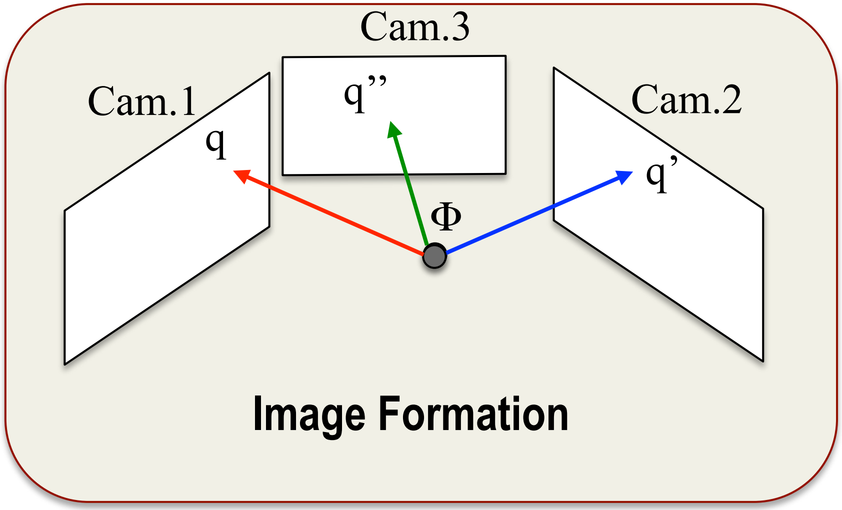
# 3D reconstruction by triangulation

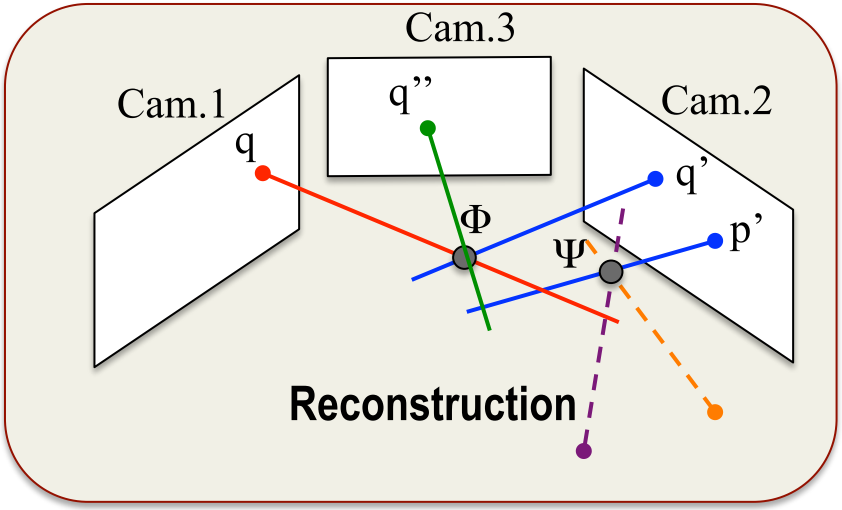
**How it works:**

- Markers appear as unlabeled dots on each camera image plane
- Matching → Back-projection → Reconstruction






*[Hartley-Sturm, Triangulation,CVIU97]*

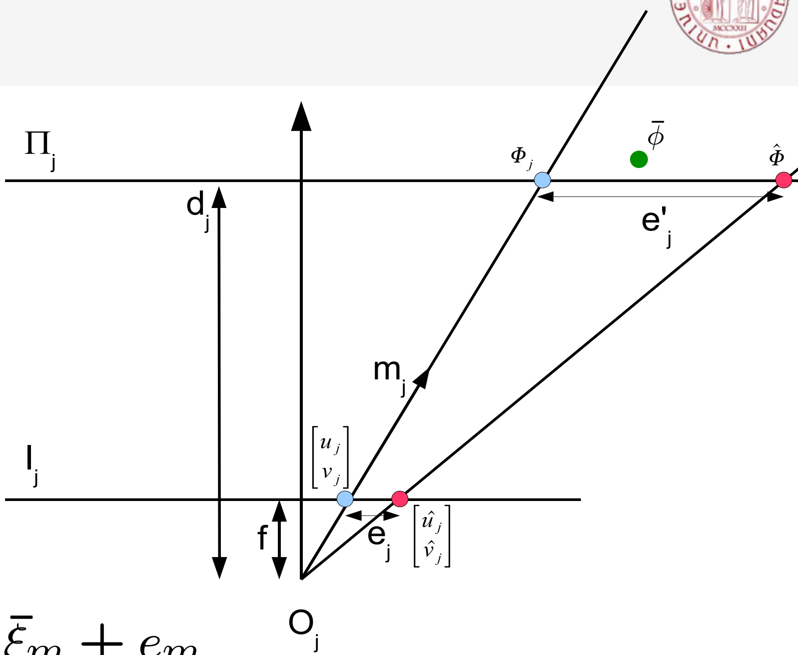
# The reconstruction problem

$$\bar{\phi} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + noise$$

$(P_1, d_1)$ $\qquad$ $(P_m, d_m)$

$$\xi_1 = \begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \bar{\xi}_1 + e_1 \quad \dots \quad \xi_m = \begin{bmatrix} u_m \\ v_m \end{bmatrix} = \bar{\xi}_m + e_m$$

$$\phi_1 = \bar{\phi} + e'_1 \quad \dots \quad \phi_m = \bar{\phi} + e'_m$$



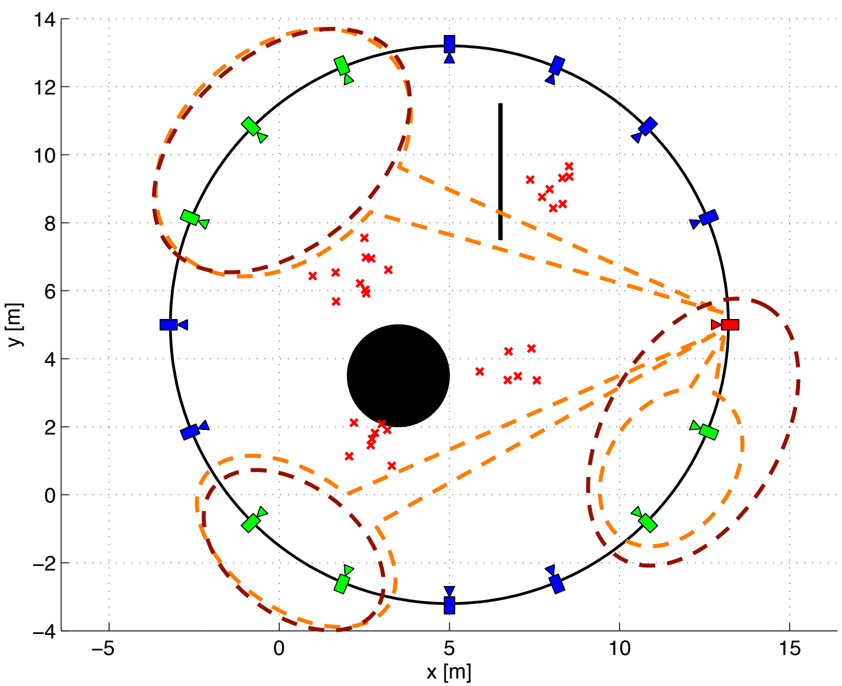

$$\hat{\phi} = \mathcal{F}(\phi_1, \phi_2, \cdots, \phi_m) \approx \bar{\phi}$$

**Reconstruction problem as an optimal estimation problem**

# The camera affinity: synthetic experiments

- *m* camera setup: the error about the target position provided by one camera *j* can be modeled as:

$$\Sigma_j = \underbrace{M\psi_j\psi_j^\top}_{\text{Error on } \textit{depth}} + \underbrace{\sigma_{e_j}^2 d_j^2 \Psi_j \Psi_j^\top}_{\text{Error on } \textit{target plane}} \quad \Rightarrow \quad \mathrm{trace}\left(\Sigma_j\right) \leq \underbrace{\sigma_e^2 \bar{d}^2}_{\text{Cameras parameters}} \frac{3}{\underbrace{m}_{\text{Network population}}\underbrace{\lambda_3[M(\phi)]}_{\text{Cameras positions}}}$$




***k*-affinity set problem:**

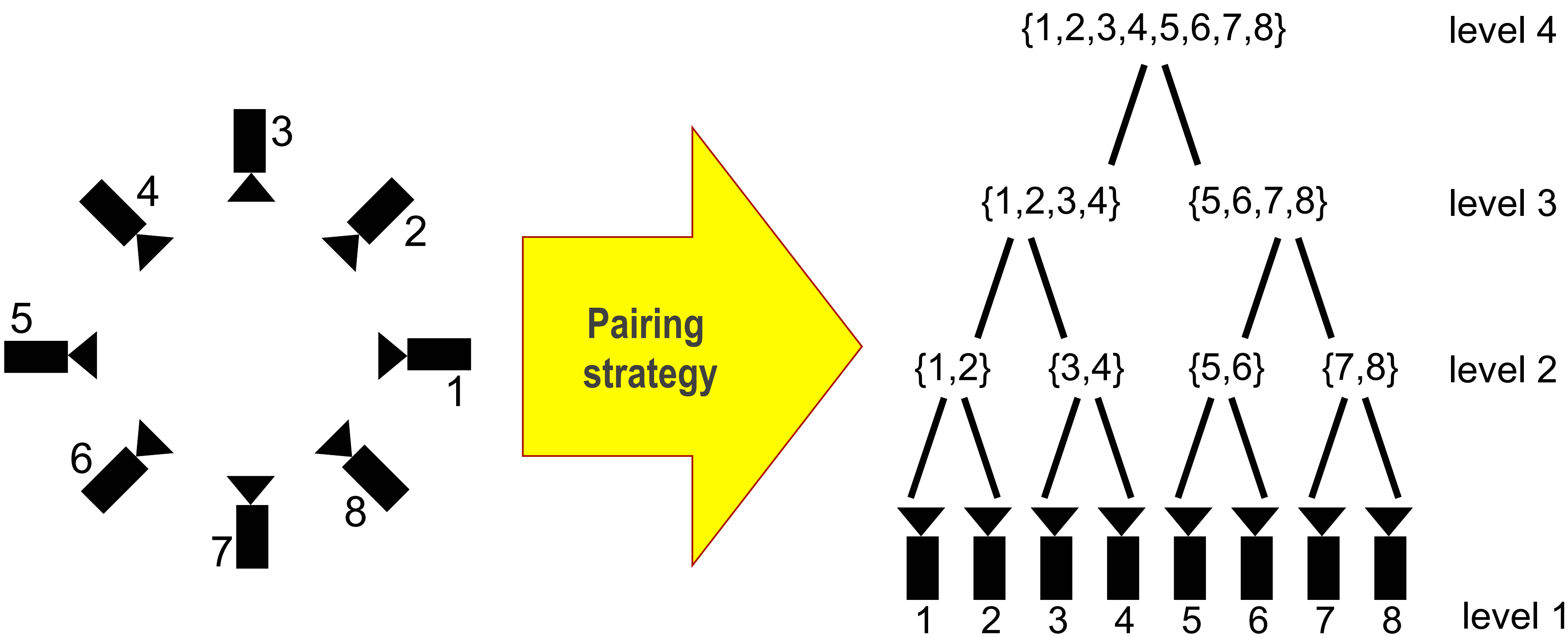
compute the *k*−1 cameras that allows the best reconstruction when associated to a specific camera

***k*-reconstruction set problem:**

compute the *k* cameras that allow the best reconstruction

*[Masiero-Cenedese, Reconstruction error in a motion capture system, arXiv:1203.3230v1 , 2012]*

# Binary tree structure

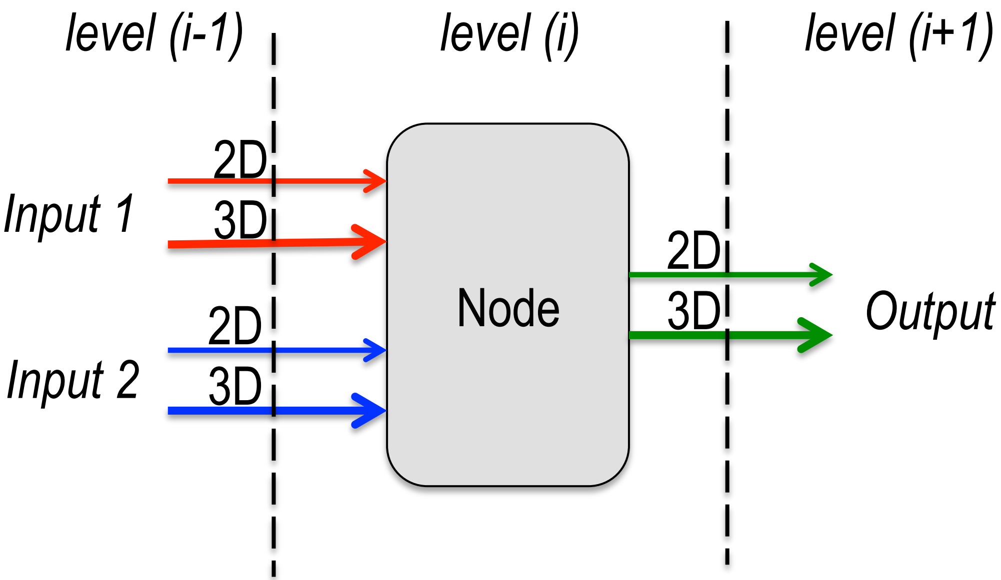



Topological structure

Binary tree structure

$$N \text{ nodes } \Rightarrow \log N \text{ levels } \Rightarrow (N-1) \text{ computational units}$$

# The algorithm



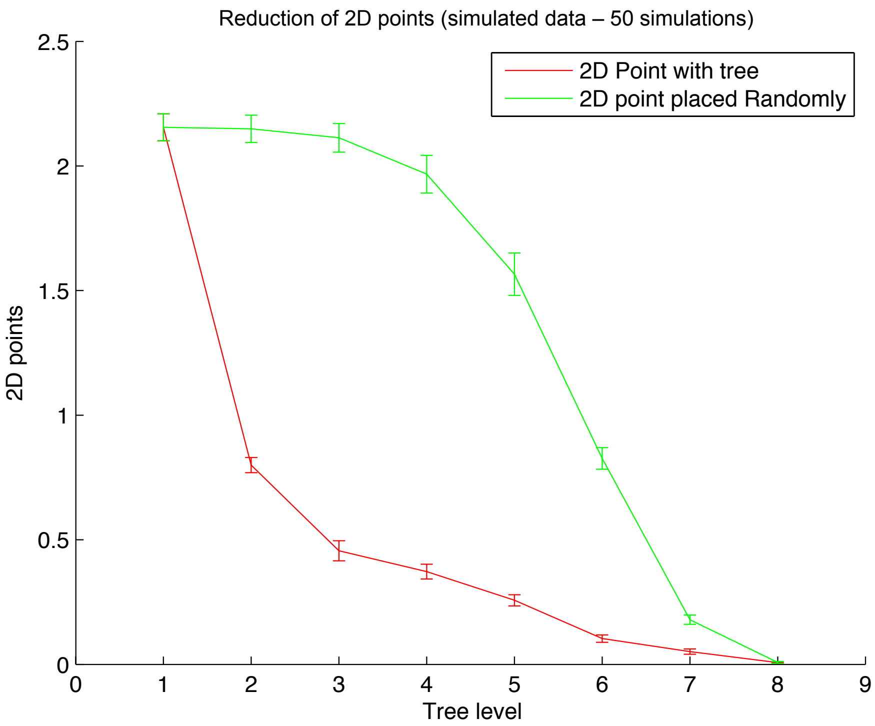

**3-Phase algorithm:** each node combines

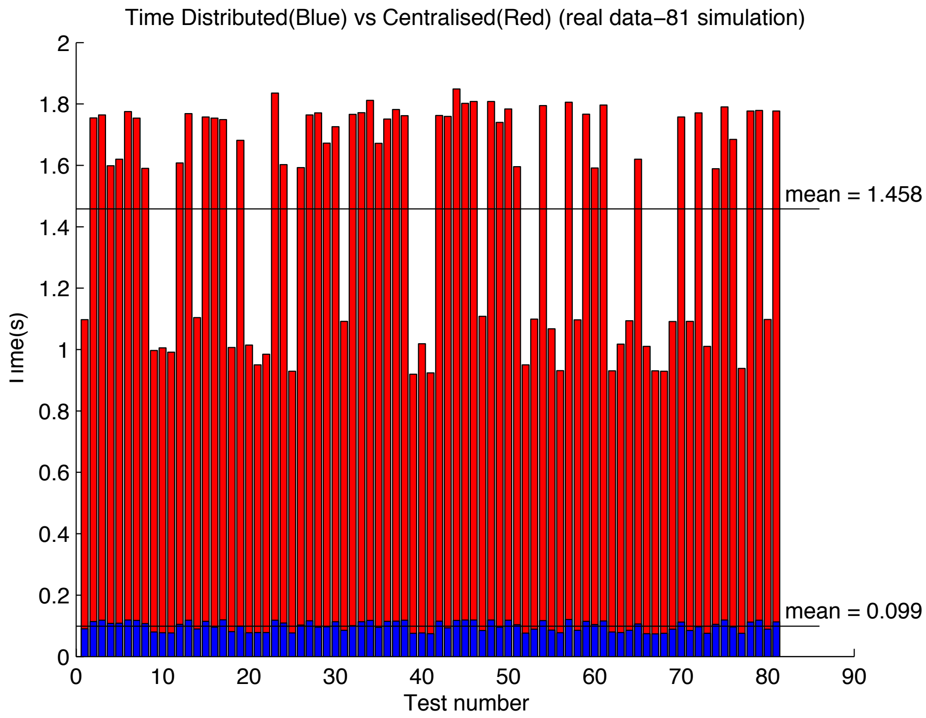
- **3D info** (Input 1) **+ 3D info** (Input 2)
- 2D info (Input 1) + **3D info** (Input 2)
- **3D info** (Input 1) + 2D info (Input 2)
- 2D info (Input 1) + 2D info (Input 2)

**To create new 3D points**
**To associate 2D points with existing 3D points**
**To reduce 2D points**

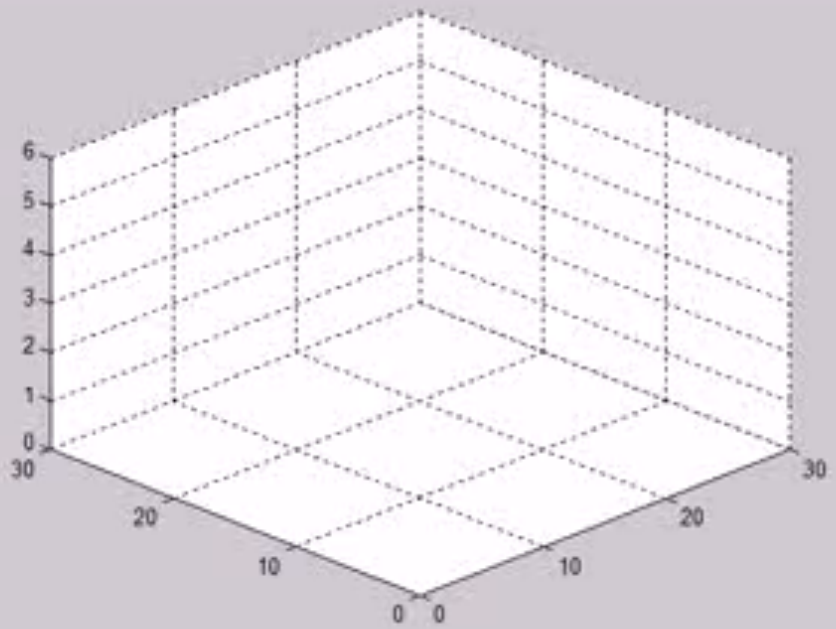
*[Cenedese-Felicini-Czyz-Smyth, A Distributed approach to 3D reconstruction in Marker Motion Capture systems, 2013]*

# Affinity based distributed algorithm – performance comparison

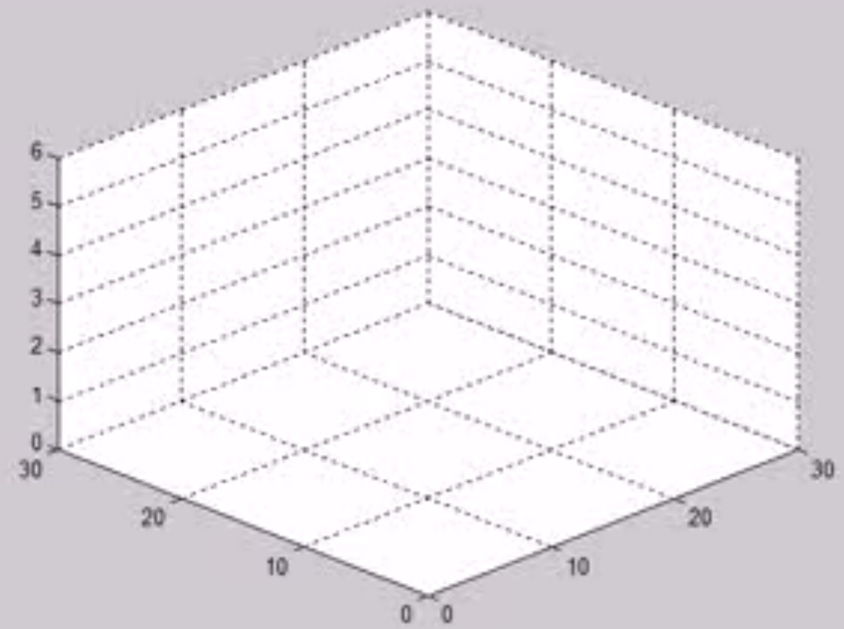


# Distributed vs centralized – video



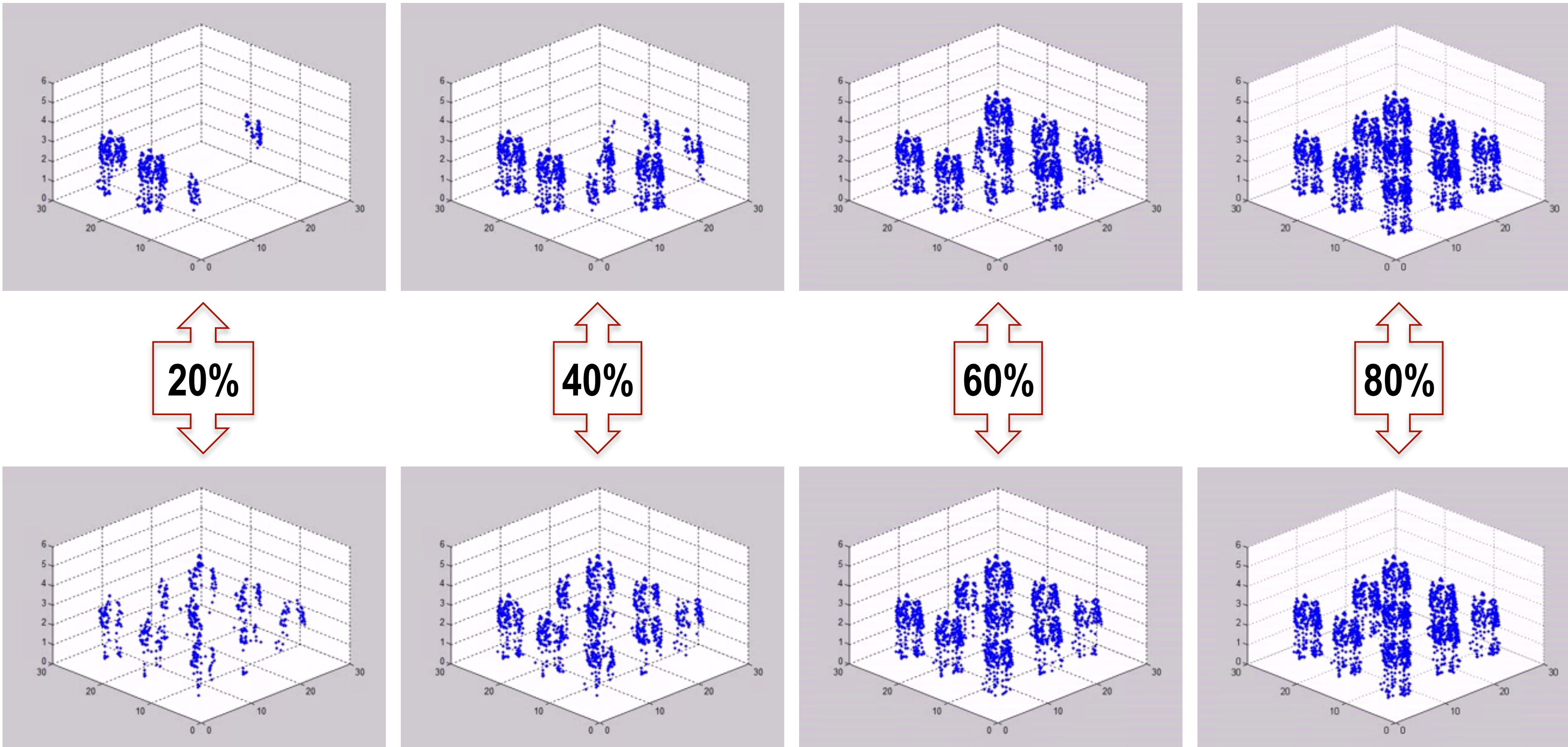

**Distributed (slowed down)**




**Centralized**

# Distributed vs centralized – snapshots

**Centralized**



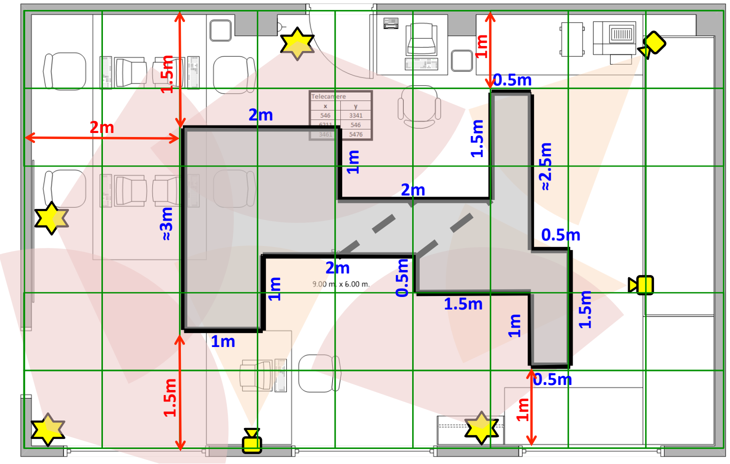

**Distributed**

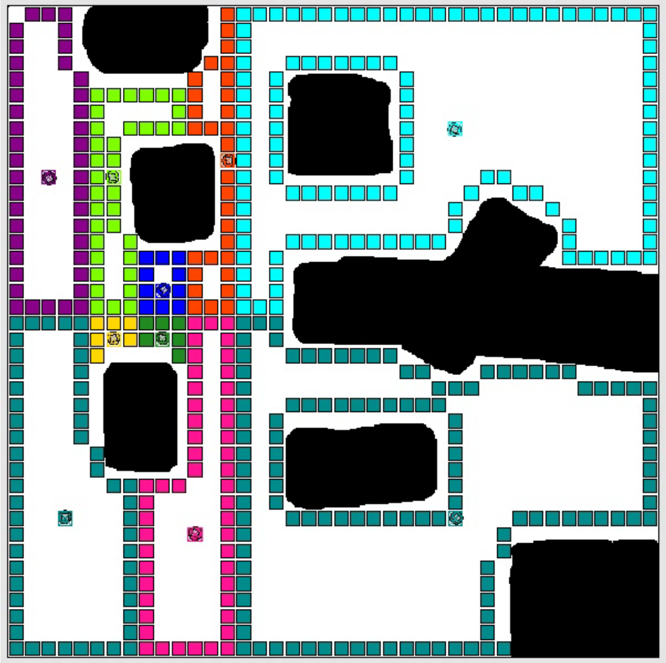
# Outline

- Introduction on distributed systems
- Camera networks
- Sparse camera networks
  - Spatial topology learning
  - Distributed calibration
  - Tracking and patrolling
- Dense camera networks
  - Camera affinity
  - Distributed reconstruction
- Conclusions

# Open research issues

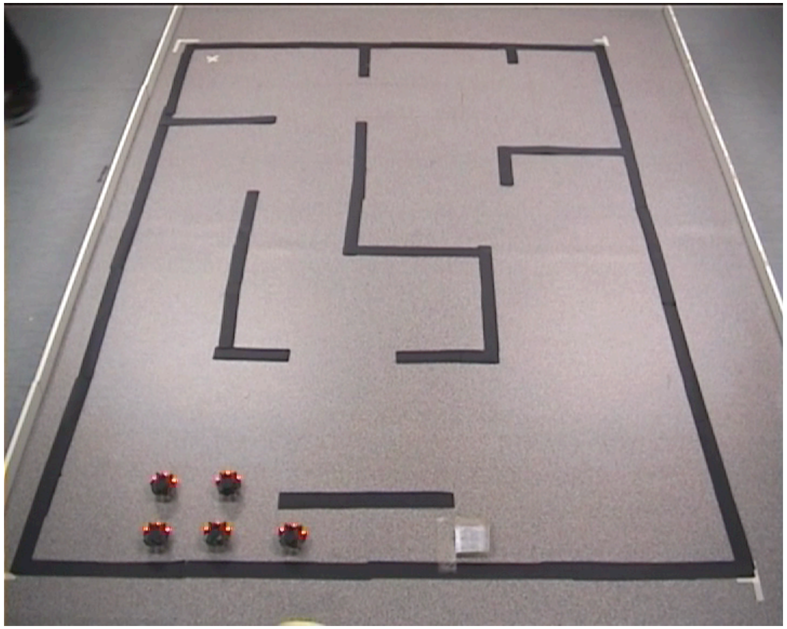
- Topology learning
- Distributed calibration
- Patrolling with non ideal domains
  - non-connected
  - with bifurcations
  - with occlusions
- Two-dimensional patrolling

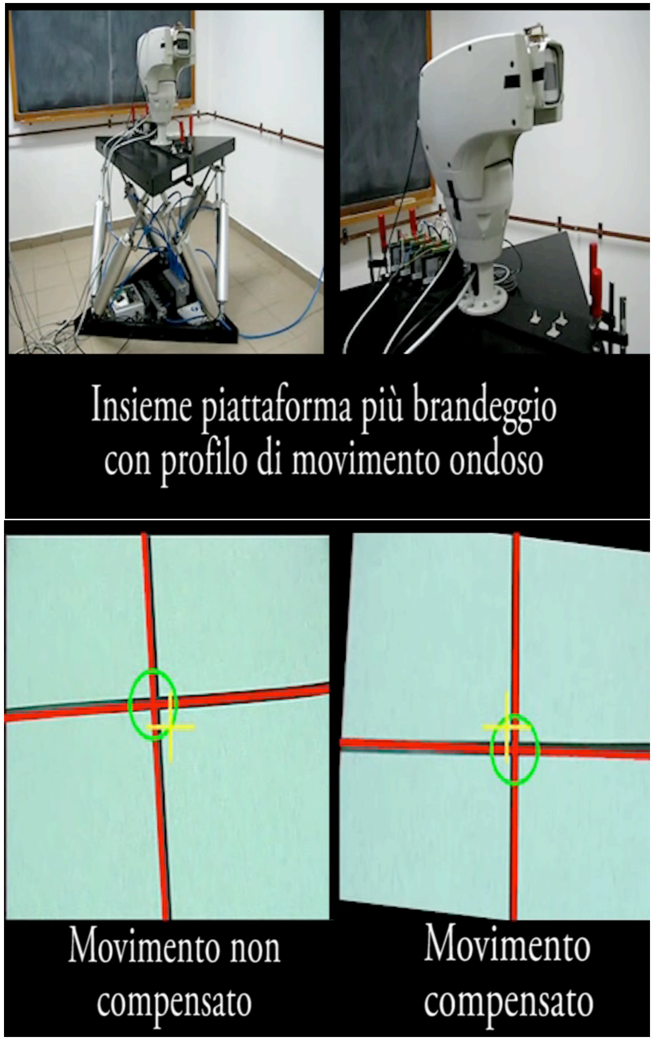


# Open research issues

- Mobile camera networks



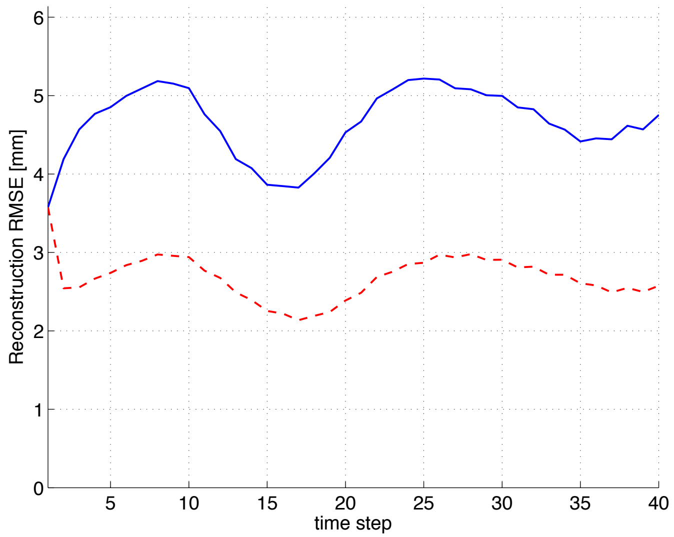


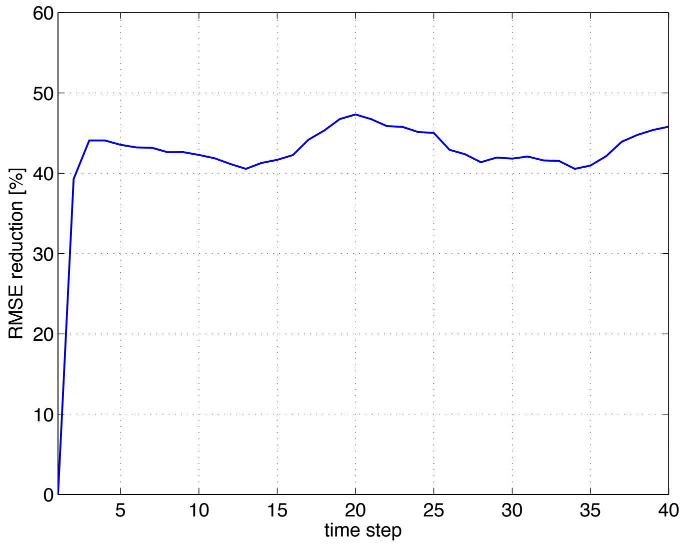
# Open research issues

- Consensus-based 3D reconstruction

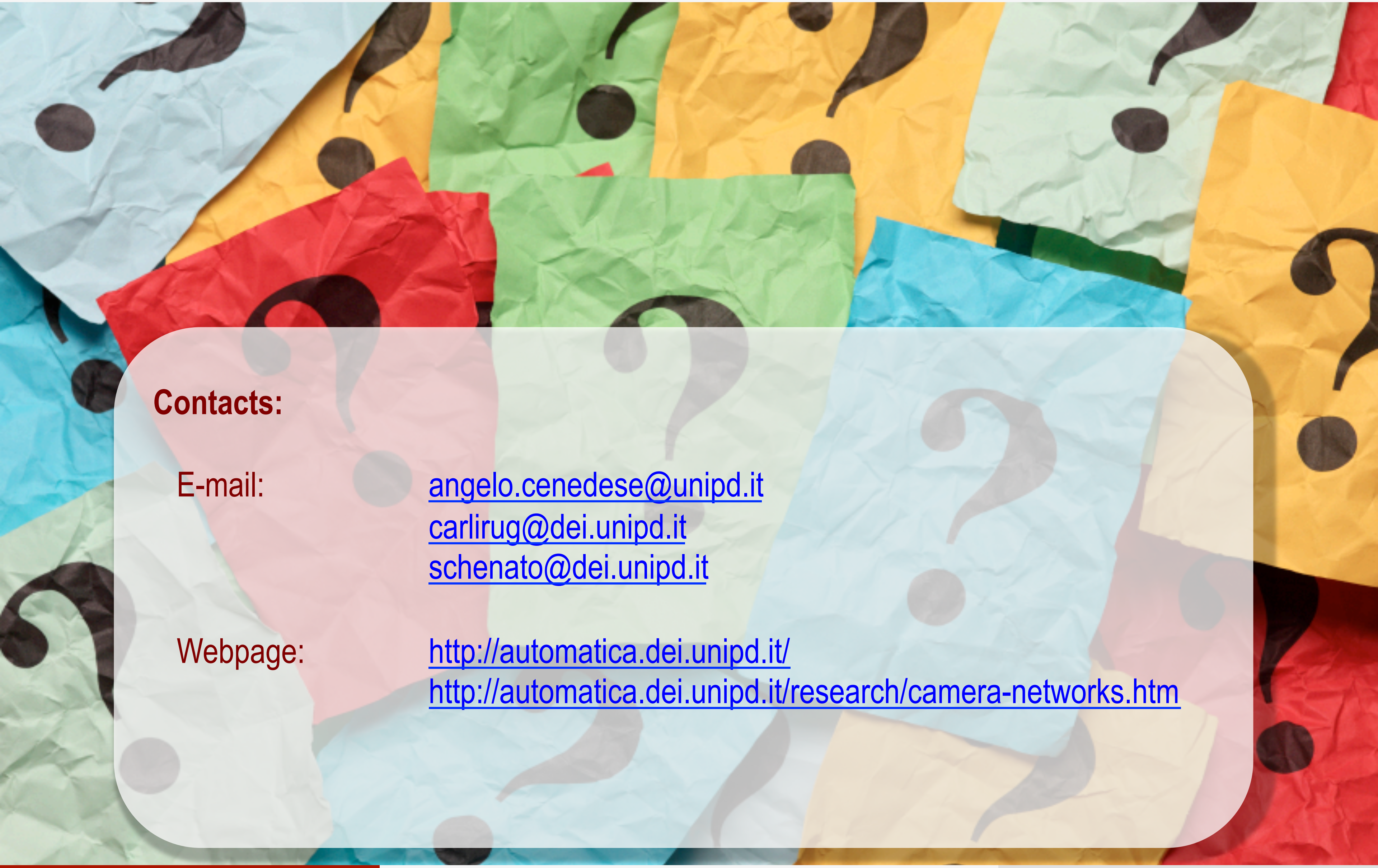
- Estimation and compensation of the camera synchronization delay in a network

RMSE in the reconstruction of 3D targets positions ($n$ = 100)



# Q&A



**Contacts:**

E-mail: angelo.cenedese@unipd.it
carlirug@dei.unipd.it
schenato@dei.unipd.it

Webpage: http://automatica.dei.unipd.it/
http://automatica.dei.unipd.it/research/camera-networks.htm