PURESAFE
Preventing hUman intervention for IncREased SAfety inFrastructures Emitting ionizing radiation

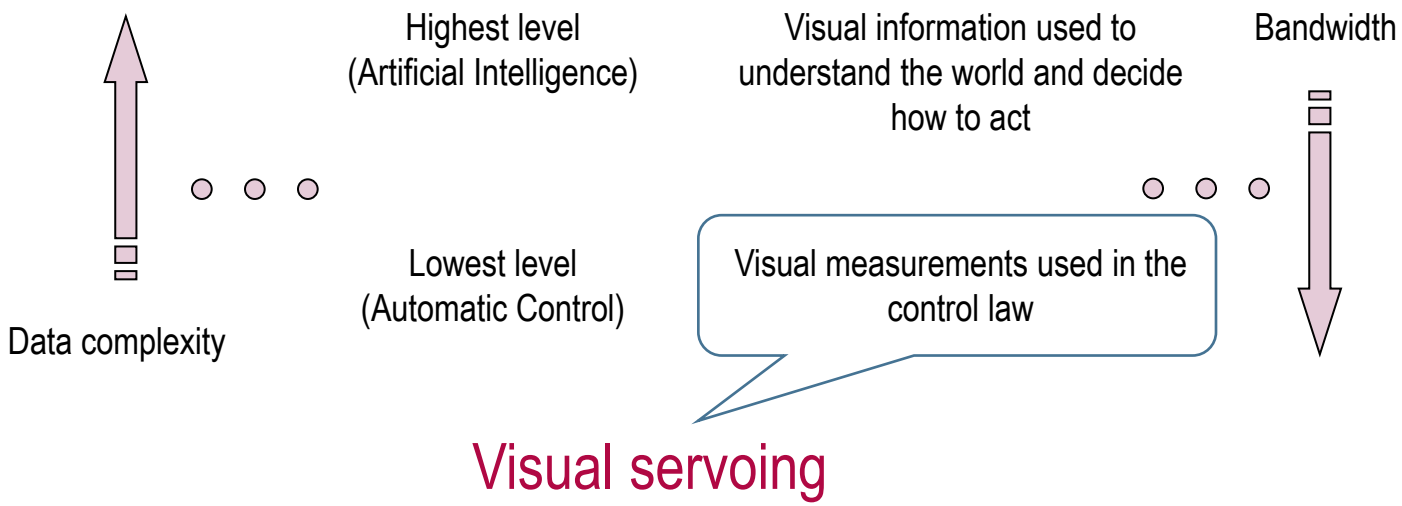
Vision for Robotic Manipulation

BRUNO SICILIANO

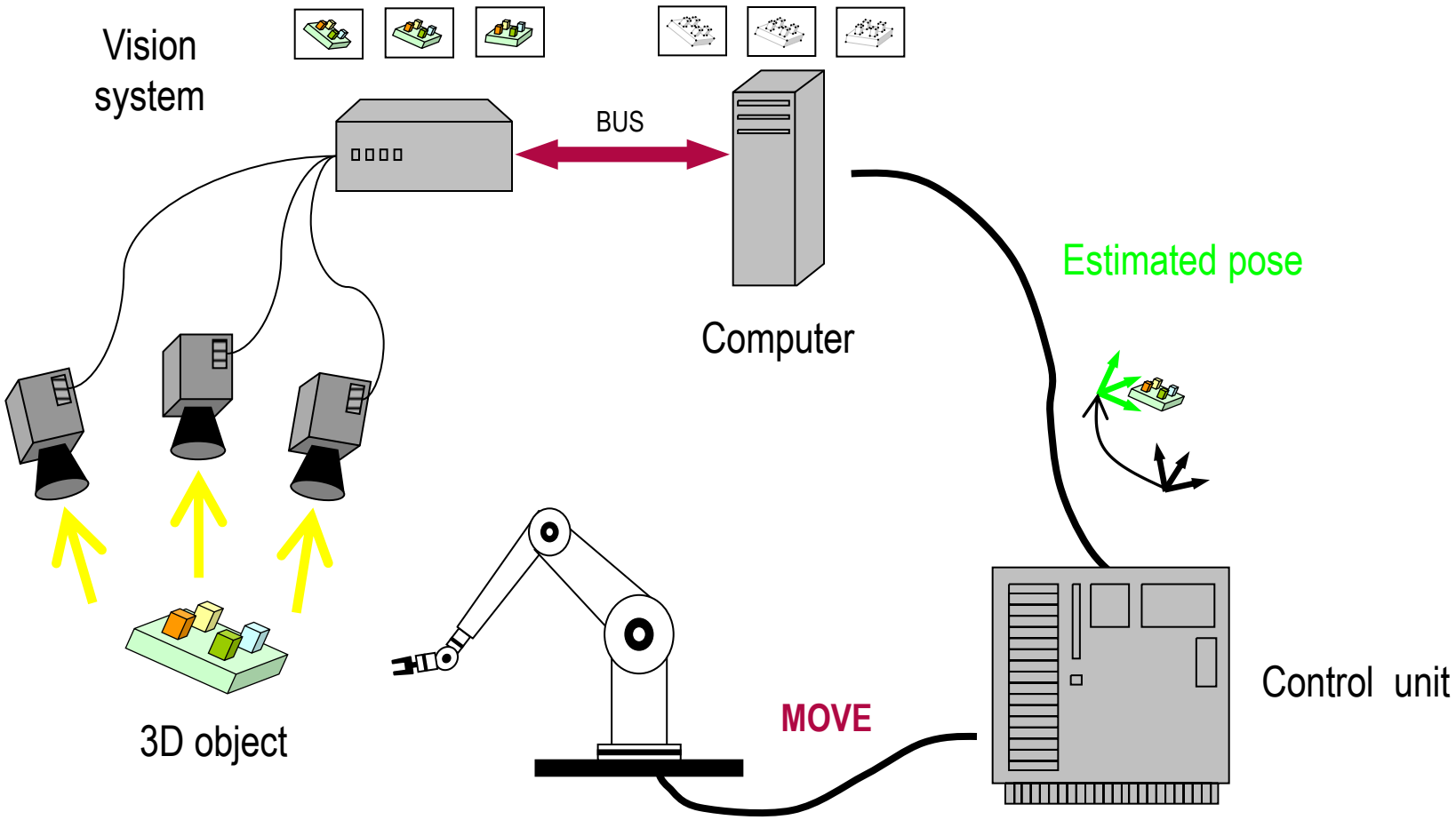
DIETI UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE

www.prisma.unina.it

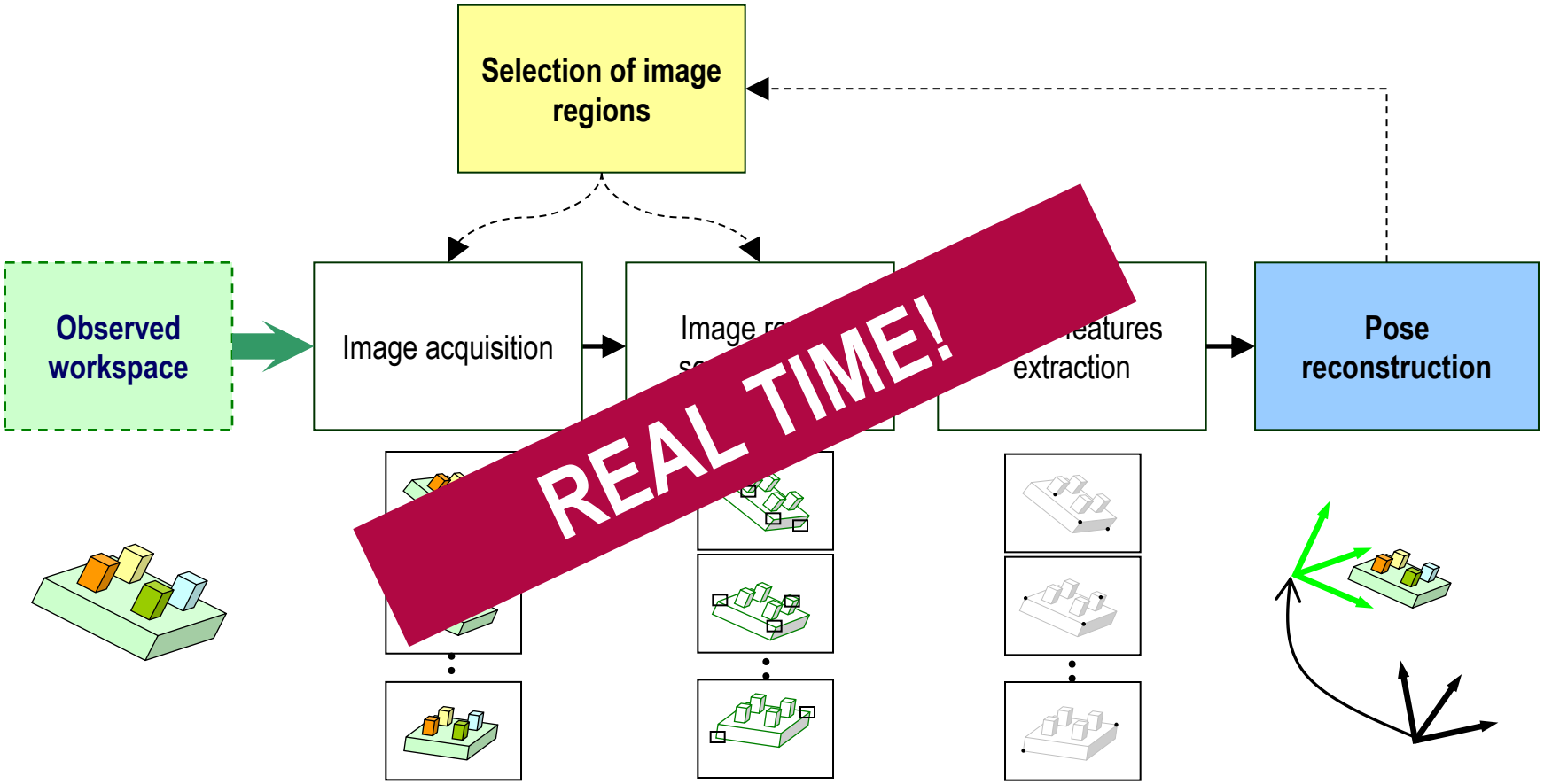
- Artificial intelligence vs. automatic control
  - Vision capabilities bring the robots nearer to human skills
  - The information that can be extracted form images is very rich and can be used at different levels of a hierarchical architecture



- Dynamic look-and-move



- Reconstruction of object pose from observed workspace



- 3D monocular robotic ball catching
- Visual grasp of unknown objects
- Real-time deformable object tracking
- Aerial visual manipulation

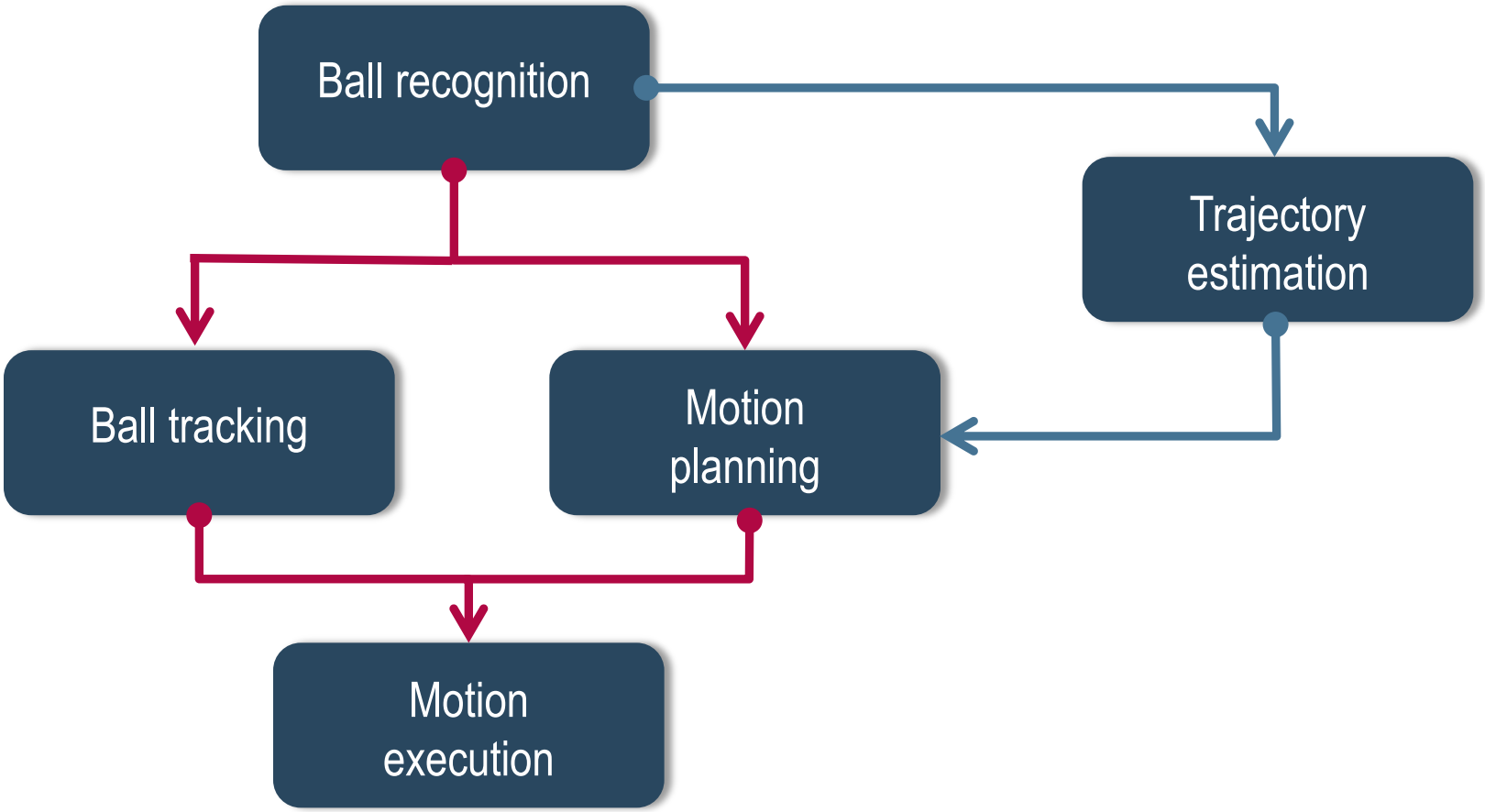
Motivation

- Why ball catching tasks …
  - Advanced robotic systems, which are required to perform quick reactions in response to visually perceived movements in a partially structured environment, are no doubt a good benchmark where testing new control algorithms and new estimation/prediction processes
  - They require
    - smart sensing
    - object tracking and motion prediction
    - on-line trajectory planning
    - motion coordination

Contribution

- In the literature …
  - Most of the presented approaches use either a stereo visual system to solve the 3D catching problem or a single camera for the 2D case
  - This scenario is reasonable because 3D tracking of the ball takes benefits from triangulation methods while, in the case of a single camera, only 2D information is directly available
  - However, a high frame rate and optics with a good accuracy are required to achieve an accurate and fast trajectory prediction, i.e. a successful catch
- By using only one camera, the cost of the equipment can be reduced. Moreover, the calibration procedure for one camera is easier than in the stereo case
- System
  - A robot manipulator with a standard CCD camera mounted in an eye-in-hand configuration which is driven by visual information in order to track a thrown ball

How It Works


Ball recognition
Trajectory estimation
Ball tracking
Motion planning
Motion execution

Ball Recognition

- The whole image is elaborated until the ball is detected, then a dynamic windowing technique, which is based on a first-order prediction algorithm of the ball motion in the image plane, is employed after the first detection
- An equalized color-based clustering is adopted in the image processing, and it makes use of the Hue, Lightness, and Saturation Color Space

Ball Tracking

- Without loss of generality
  - The camera frame is considered coincident with the hand (end-effector) frame, with the camera optical axis aligned with the hand approaching axis
  - The proposed visual control law belongs to the category named Resolved-Velocity Image-Based Visual Servoing, hence the manipulator dynamics is taken into account directly by the low-level robot controller
- Partitioned approach
  - The rotational components of the robot motion will be reserved to the ball tracking task
  - The positional components of the camera motion have to be generated in a way as to intercept the ball trajectory

- Notation
  - Normalized image coordinates of the ball centroid

$$\boldsymbol{s} = \begin{bmatrix} X & Y \end{bmatrix}^{\mathrm{T}}$$

  - Ball position w.r.t. the camera frame

$$\boldsymbol{p}_o^c = \begin{bmatrix} x^c & y^c & z^c \end{bmatrix}^{\mathrm{T}} = z^c \begin{bmatrix} X & Y & 1 \end{bmatrix}^{\mathrm{T}}$$

  - Absolute velocity of the camera frame expressed in the camera frame

$$\boldsymbol{v}_c^c = \begin{bmatrix} \dot{\boldsymbol{p}}_c^{c^{\mathrm{T}}} & \boldsymbol{\omega}_c^{c^{\mathrm{T}}} \end{bmatrix}^{\mathrm{T}}$$

  - Absolute velocity of the ball expressed in the camera frame

$$\boldsymbol{v}_o^c = \begin{bmatrix} \dot{\boldsymbol{p}}_o^{c^{\mathrm{T}}} & \boldsymbol{\omega}_o^{c^{\mathrm{T}}} \end{bmatrix}^{\mathrm{T}}$$

- Differential relationship in the image plane

$$\dot{\boldsymbol{s}} = \boldsymbol{L}_s \boldsymbol{v}_c^c + \boldsymbol{L}_s \boldsymbol{\Gamma}(-\boldsymbol{p}_o^c)\boldsymbol{v}_o^c$$

  - Interaction matrix

$$\boldsymbol{L}_s = \begin{bmatrix} \boldsymbol{L}_{sp} & \boldsymbol{L}_{so} \end{bmatrix} = \begin{bmatrix} -1/z^c & 0 & X/z^c & XY & 1-X^2 & Y \\ 0 & -1/z^c & Y/z^c & 1+Y^2 & -XY & -X \end{bmatrix}$$

$$\boldsymbol{\Gamma}(\cdot) = \begin{bmatrix} -\boldsymbol{I}_3 & \boldsymbol{S}(\cdot) \\ \boldsymbol{0} & \boldsymbol{I}_3 \end{bmatrix}$$

- Control law (rotational part)

$$\boldsymbol{\omega}_c^c = \boldsymbol{L}_{so}^{\dagger} \left[\boldsymbol{K}_{so,eb2}(\boldsymbol{e}_s)\boldsymbol{\tau}_{eb1}(\boldsymbol{e}_s) - \hat{\boldsymbol{L}}_{sp} \left(\dot{\boldsymbol{p}}_c^c - \hat{\dot{p}}_o^c + \boldsymbol{S}(-\hat{\boldsymbol{p}}_o^c)\hat{\boldsymbol{\omega}}_o^c \right) \right] + \hat{\boldsymbol{\omega}}_o^c$$

- Control gains

$$\boldsymbol{\tau}_{eb_1}(\boldsymbol{e}_s) = \begin{cases} \mathbf{0} & \text{if } \|\boldsymbol{e}_s\| \leq e_{b1} \\ \left(1 - \frac{e_{b1}}{\|\boldsymbol{e}_s\|}\right)\boldsymbol{e}_s & \text{if } \|\boldsymbol{e}_s\| > e_{b1} \end{cases}$$

$$\boldsymbol{K}_{so}(\boldsymbol{e}_s) = \begin{cases} k_o \boldsymbol{I}_2 & \text{if } \|\boldsymbol{e}_s\| \leq e_{b2} \\ k_o e^{\beta_o\left(\frac{\|\boldsymbol{e}_s\|}{e_{b2}} - 1\right)} \boldsymbol{I}_2 & \text{if } e_{b2} < \|\boldsymbol{e}_s\| \leq e_{b3} \\ k_o e^{\beta_o\left(\frac{e_{b3}}{e_{b2}} - 1\right)} \boldsymbol{I}_2 & \text{if } \|\boldsymbol{e}_s\| > e_{b3} \end{cases}$$

Motion Planning and Execution

- Fifth-order polynomial vector to compute the desired trajectory for the camera on-line in the 3D Cartesian space
- Control law (linear part)

$$\dot{\boldsymbol{p}}_c^c = \boldsymbol{R}_c^{\mathrm{T}} \left(\dot{\boldsymbol{p}}_{c,d} + \boldsymbol{K}_p \boldsymbol{e}_p \right)$$

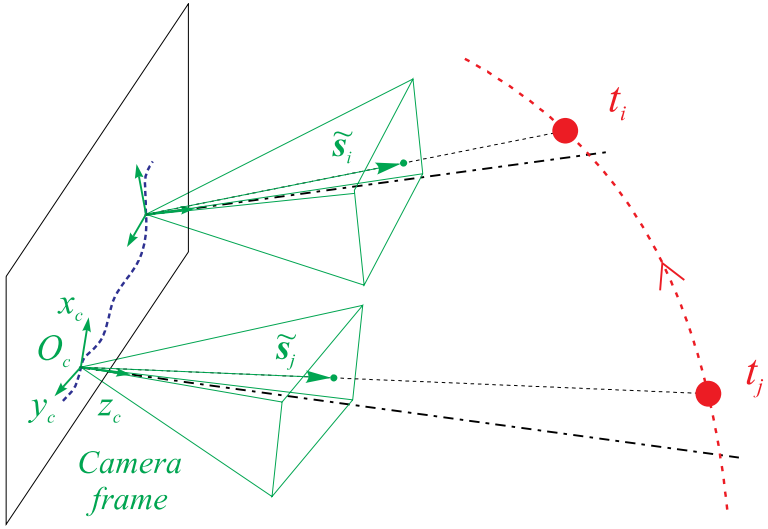
- Joint velocity control

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}(\boldsymbol{q}) \boldsymbol{T}_c \boldsymbol{\upsilon}_c^c + \boldsymbol{N}_J \boldsymbol{K}_r \dot{\boldsymbol{q}}_r$$

  - Redundancy resolution
    - avoiding joint limits
    - avoiding kinematic singularities
    - limiting motion of the track

Trajectory Estimation

- Interpolation of 2D measurements along time
- Enhanced robustness by collecting visual data moving camera along significant baseline
  - Orientation controlled to keep ball in field of view
- Optical ray passing through absolute origin $\boldsymbol{c}_k$ of camera and feature vector $\boldsymbol{r}_k$ at time t_k



$$\begin{cases} (r_{y,k} - c_{y,k})x + (c_{x,k} - r_{x,k})y + r_{x,k}c_{y,k} - r_{y,k}c_{x,k} = 0 \\ (r_{z,k} - c_{z,k})x + (c_{x,k} - r_{x,k})z + r_{x,k}c_{z,k} - r_{z,k}c_{x,k} = 0 \end{cases}$$

Trajectory Estimation

- Neglecting air drag factor

$$\boldsymbol{p} = \boldsymbol{p}_0 + \dot{\boldsymbol{p}}_0 t + 0.5 \boldsymbol{g} t^2$$

- Resulting linear system with n_l measurements

$$\boldsymbol{A} \begin{bmatrix} \boldsymbol{p}_0^{\mathrm{T}} & \dot{\boldsymbol{p}}_0^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} = \boldsymbol{b}$$

- Weighted pseudo-inverse of $\boldsymbol{A}$
  - Measurements less affected by air drag (ball velocity smaller)
  - Higher image resolution (ball closer to camera)

Trajectory Estimation

- The trajectory estimation provided by the linear algorithm is employed as a starting point for a non-linear estimation algorithm, that continuously refines the current estimation using new available ball observations and a more accurate ball trajectory model

$$\min_{\boldsymbol{p}_0, \dot{\boldsymbol{p}}_0} \sum_{k=1}^{n} \left\| \frac{1}{\hat{z}_k^c} \begin{bmatrix} \hat{x}_k^c \\ \hat{y}_k^c \end{bmatrix} - \boldsymbol{s}_k \right\|$$

  - estimated ball position $\hat{\boldsymbol{p}}_k^c = \begin{bmatrix} \hat{x}_k^c & \hat{y}_k^c & \hat{z}_k^c \end{bmatrix}^{\mathrm{T}} = \boldsymbol{R}_{c,k}^{\mathrm{T}} \left(\hat{\boldsymbol{p}}_k - \boldsymbol{c}_k \right)$
  - ballistic ball motion with air drag $\ddot{\boldsymbol{p}}(t) = \boldsymbol{g} - \dfrac{c_w \pi d_b^2 \rho_a}{2 m_b} \left\| \dot{\boldsymbol{p}}(t) \right\| \dot{\boldsymbol{p}}(t)$

Trajectory Estimation

- When the refinement process stops, the catching path can be generated
  - The hand orientation is controlled in order to have a direction of the camera equal to the tangent to the estimated ball trajectory at the predicted catching point
  - The hand starts to close its fingers and is moved following the same predicted path of the ball, while its velocity will be decreased in a fixed time (or displacement) until zero, in order to allow the dissipation of the impact energy in a sufficient time interval
- Estimated parameters
  - The position and linear velocity term are computed from the previous numeric integration
  - The angular velocity can be obtained as $\hat{\boldsymbol{\omega}}_o^c = \left(\hat{\boldsymbol{p}}_o^c \times \dot{\hat{\boldsymbol{p}}}_o^c\right) / \|\hat{\boldsymbol{p}}_o^c\|^2$

3D Monocular Robotic Ball Catching

V. Lippiello, F. Ruggiero, B. Siciliano

PRISMA Lab
www.prisma.unina.it

February 2013

- 3D monocular robotic ball catching
- Visual grasp of unknown objects
- Real-time deformable object tracking
- Aerial visual manipulation

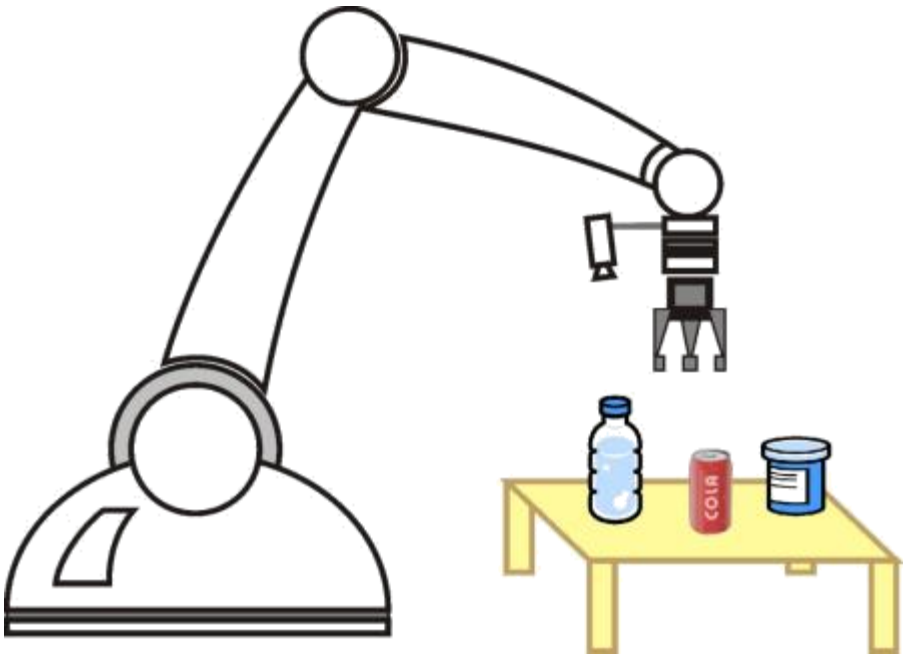
- Visual grasp of unknown objects
  - An object model reconstruction algorithm is required
  - Main required characteristics:
    - The algorithm must be fast
    - High accuracy is not required

    and ...

    - The reconstruction process should be suitable for grasping

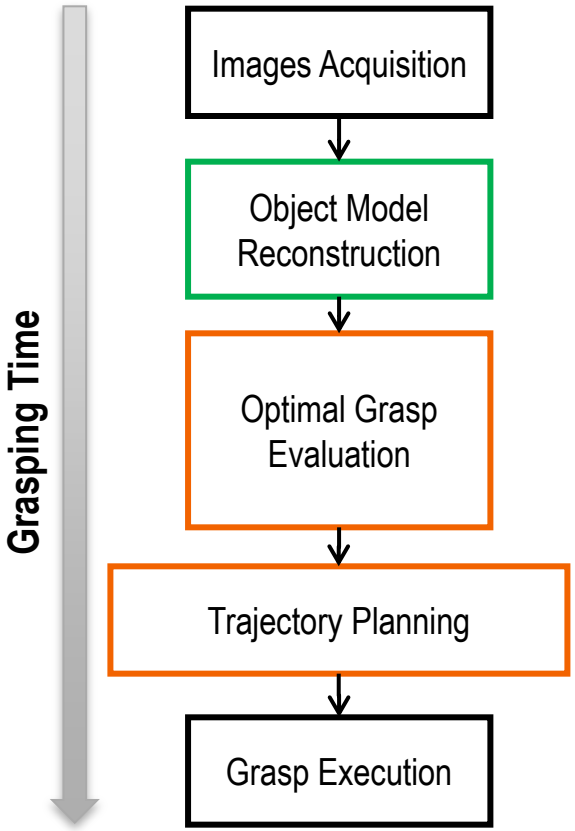
      ⇓

      it becomes an active component of the grasp process

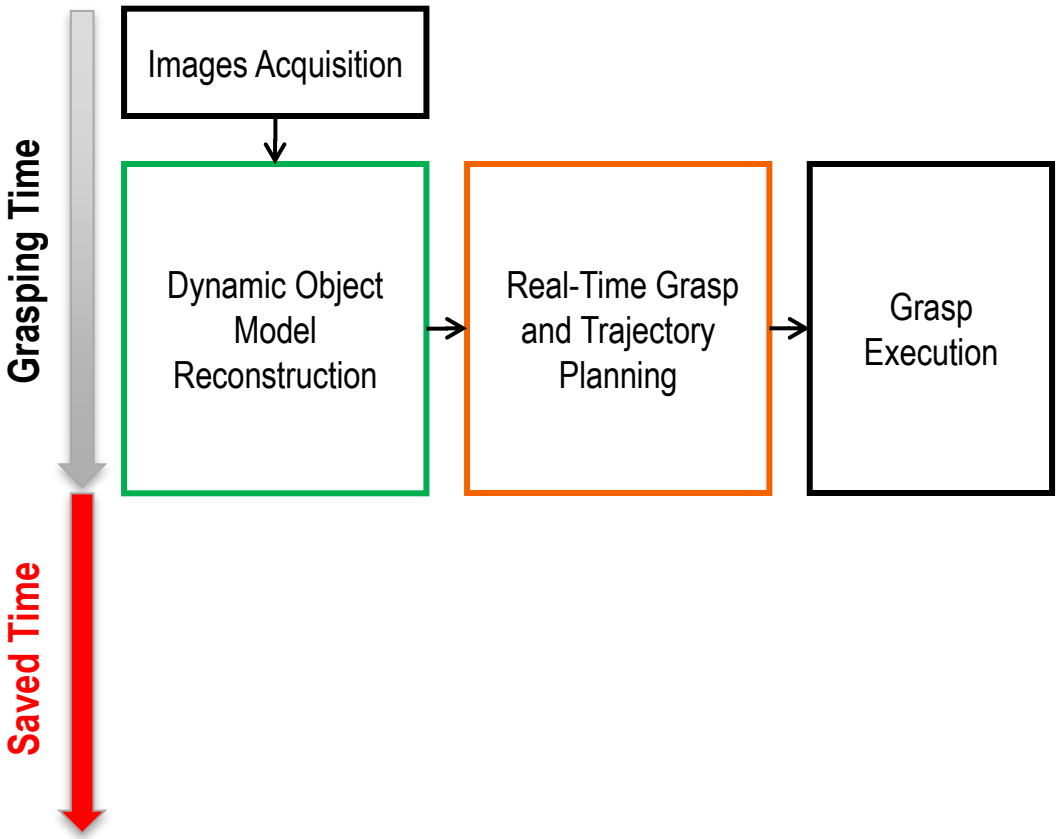


- Visual grasp of unknown objects

classical serial approach



proposed parallel approach

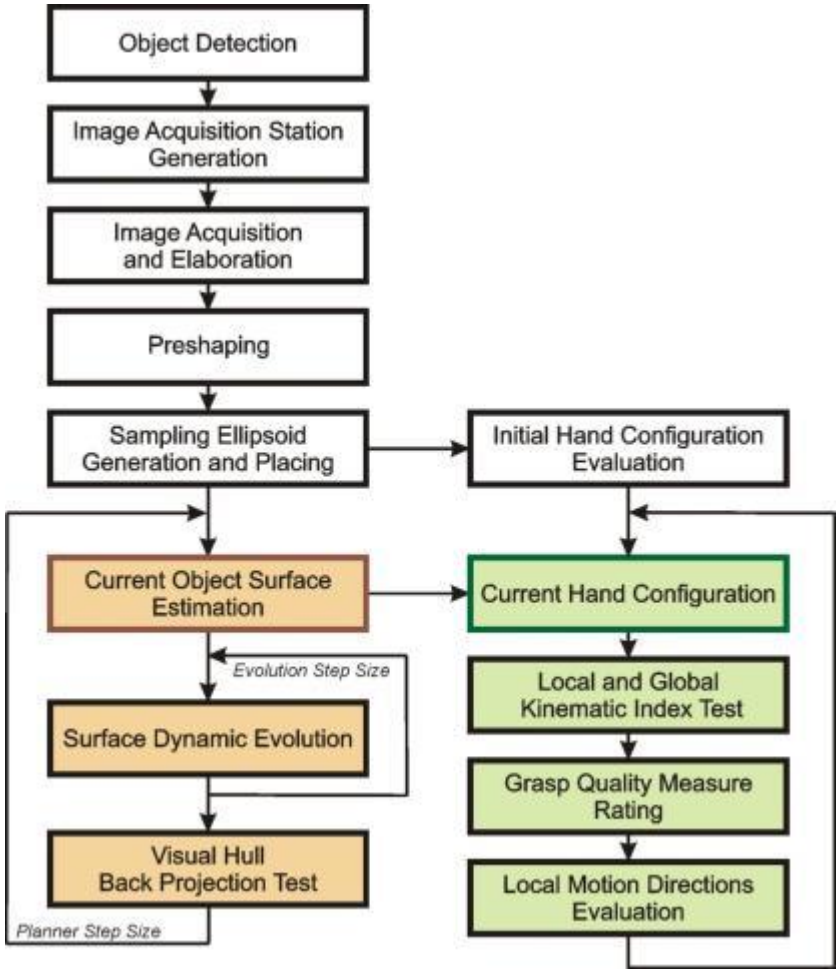


Assumptions and Goals

- Assumptions
  - An eye-in-hand camera configuration is considered
  - Multi-fingered hand
  - The observed object is
    - a rigid body fixed in the space
    - from a topology point of view, an orientable surface with genus 0
    - distinguishable with respect to the background and other objects
- Goals
  - Grasp a 3D unknown object while reconstructing at the same time its surface
    - The reconstruction of the object is a secondary outcome of the proposed method
    - The fingers move on the current (virtual) reconstructed surface towards local minima according to suitable grasp quality measures
    - A safety distance is held resulting in a floating effect around the object

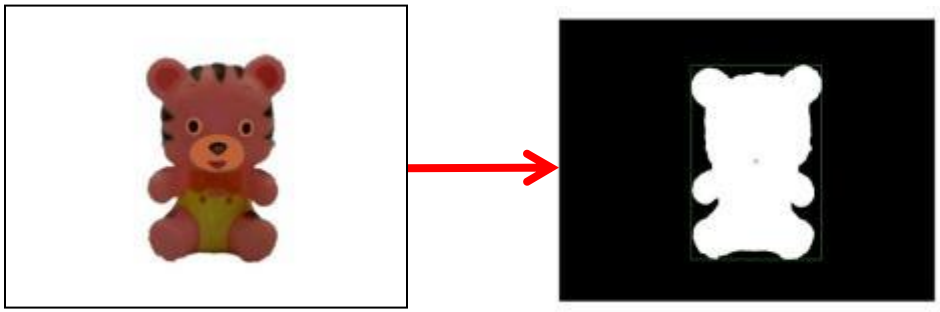
Floating Visual Grasp Algorithm

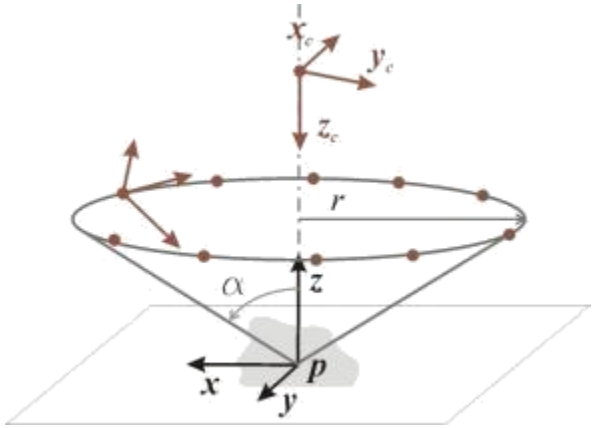
- Data flow
  - Some preparation steps
  - Object surface reconstruction
  - Local grasp planner
- The reconstruction algorithm updates in real-time the estimation of the current reconstructed object surface
- The local planner, on the basis of the current surface estimation, computes the fingers trajectories toward the current local optimal configuration for the grasp
- Reconstruction and local planning can be run in parallel resulting in a very fast grasp method

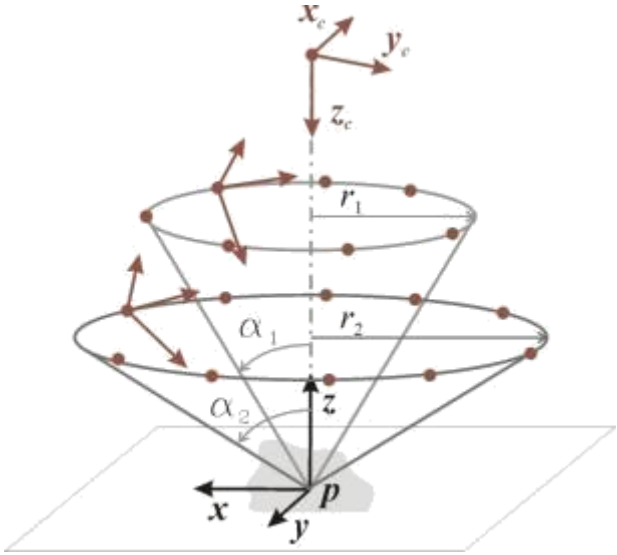


Preparation Steps

- Image acquisition stations
  - One, two or more circular trajectories at a constant distance from the object with different view angles
  - n acquired images
- Object silhouettes extraction
- Elaborations to enhance the quality of silhouettes

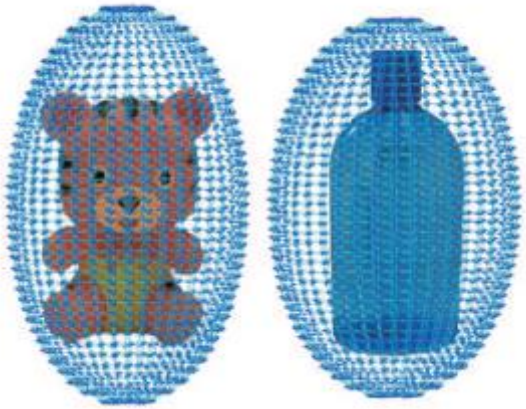






Preshaping

- Preshaping algorithm
  - The aim is to find the minimum ellipsoid which surrounds the object
  - For each image, the four planes of the Cartesian space containing the origin of the camera frame and two adjacent vertices of the corresponding silhouette bounding-box in the image plane are considered, resulting in 4n Cartesian planes
  - The intersections of these planes create a polyhedron which contains the object visual hull and whose vertices can be easily computed by solving a linear programming problem
  - Once the vertices have been computed, the central moments can be evaluated and thus the pseudo-inertia tensor; its eigenvectors and eigenvalues define the principal axes of inertia of the ellipsoid surrounding the object



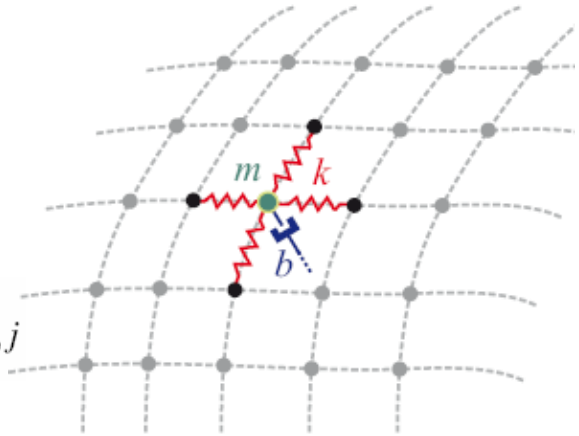
$$\mu_{i,j,k} = \sum_{x_v \in \mathcal{P}} (x_{v_x} - \bar{x}_{v_x})^i (x_{v_y} - \bar{x}_{v_y})^j (x_{v_z} - \bar{x}_{v_z})^k$$

$$I = \begin{bmatrix} \mu_{2,0,0} & \mu_{1,1,0} & \mu_{1,0,1} \\ \mu_{1,1,0} & \mu_{0,2,0} & \mu_{0,1,1} \\ \mu_{1,0,1} & \mu_{0,1,1} & \mu_{0,0,2} \end{bmatrix}$$

Object Surface Reconstruction

- Cross reticular topology with a virtual mass associated to each ellipsoid sample point, with springs linked to closest cross points and a spatial damper

$$m\ddot{\boldsymbol{x}}_{i,j} + b\dot{\boldsymbol{x}}_{i,j} + k(4\boldsymbol{x}_{i,j} - \boldsymbol{x}_{i-1,j} - \boldsymbol{x}_{i,j+1} - \boldsymbol{x}_{i+1,j} - \boldsymbol{x}_{i,j-1}) = \boldsymbol{f}_{i,j}$$



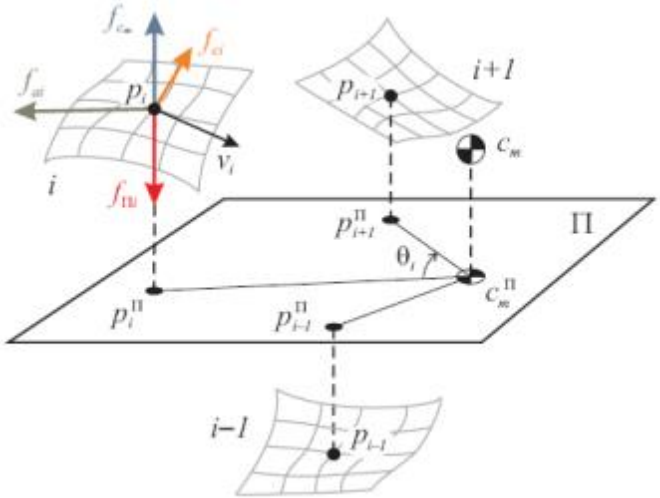
- The equilibrium is achieved when the elastic forces generated by the grid and the attractive forces generated by the visual hull become equal
- The accuracy of the reconstruction process depends
  - on the number of views
  - on the distribution of the acquisition stations
  - on the density of the reconstruction sphere

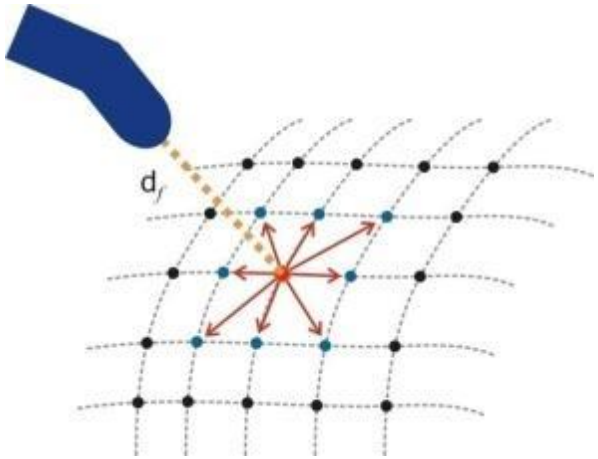
Local Grasp Planner

- The local grasp planner generates the fingertips trajectories on the current reconstructed surface (keeping a fixed floating safety distance)
  - Starting from the current grasp configuration, the planner generates the motion of the fingertips from the current position to a new set of points of the update surface
  - At each contact point of the current grasp configuration is associated a suitable field of forces, which is used to generate the motion of the fingertips
  - The process is repeated in a recursive manner, until improvements of the quality measure are obtained
  - The planner ends its job when the object reconstruction algorithm reaches an equilibrium; then, the planner computes the final grasp configuration and the floating distance is progressively reduced to achieve the desired grasp action
- For fine manipulation the initial grasp configuration is chosen as an equilateral grasp in a plane parallel to the two minor axes of the ellipsoid

Grasp Quality Measure

- Planar grasps in 3D space where the contact points lie in the same grasp plane
- At each current contact point, a field of forces is computed as a sum of suitable contributions:
  - Aligning the contact points in the same grasp plane
  - Attracting the grasp plane towards the center of mass of the current reconstruction surface
  - Attaining an equilateral grasp configuration
  - Enlarging the area of the grasp polygon
  - Avoiding joint limits overcoming kinematic singularities and finger collisions
- This field is projected onto the tangential plane to the current reconstruction surface at the current contact point



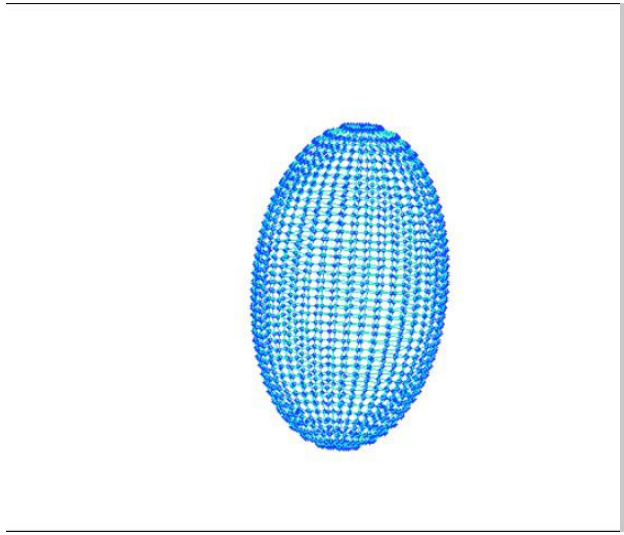


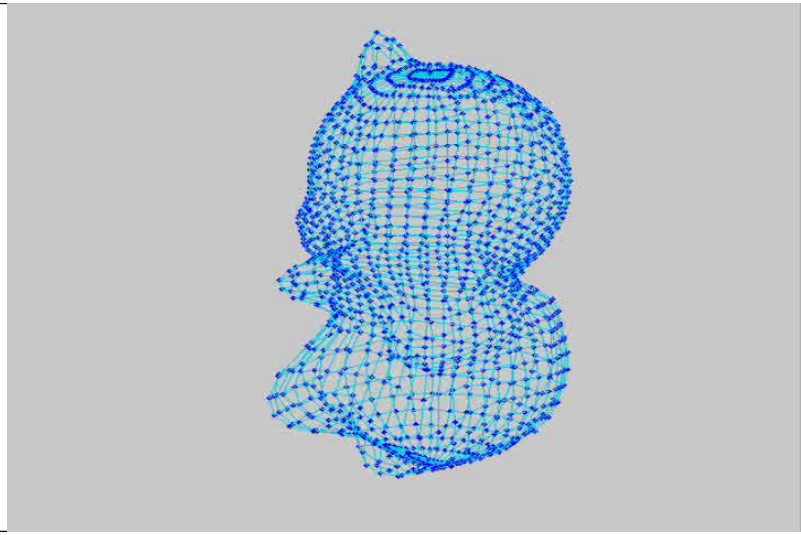
Finger Trajectory Planner

- The local grasp planner produces a sequence of intermediate target grasp configurations at each iteration of the object reconstruction algorithm
  - The intermediate configurations are used to generate the fingers paths
- The sequence of intermediate configurations can be suitably filtered by a spatial low-pass filter in order to achieve a smooth path for the fingers on the object surface
  - Only the final configuration has to be reached exactly, while the intermediate configurations can be considered as "via points"
- A fixed floating distance is held during the motion and is progressively reduced at the end of the reconstruction process to produce the final grasp action

Experiments

- Performance of the reconstruction algorithm
  - Object: teddy bear (12 images with α=80°, 6 images with α=50°, 1 image from the top)



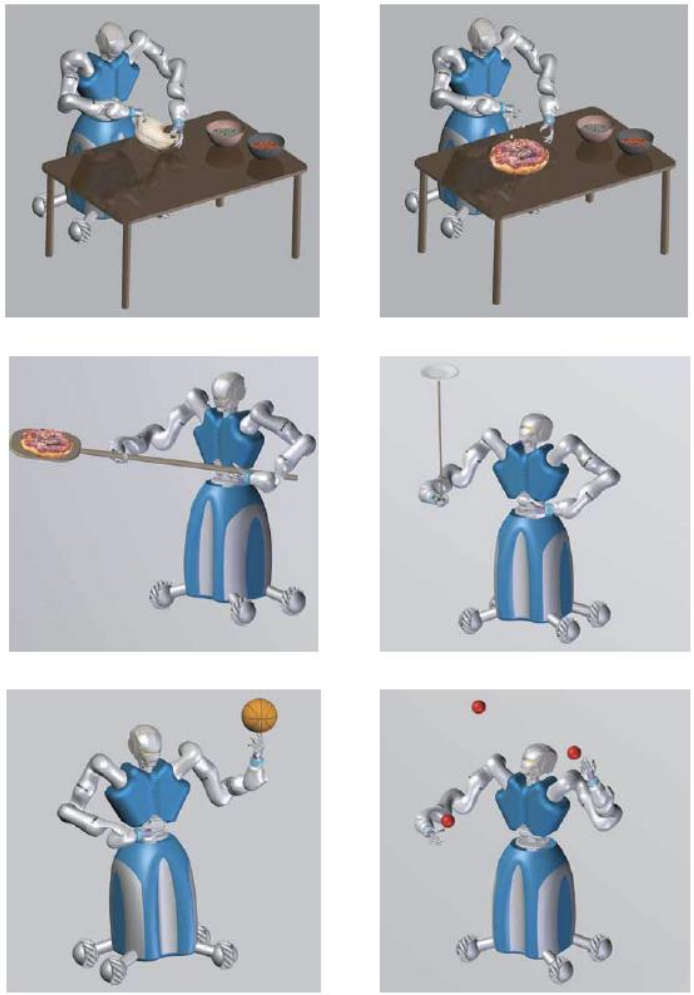


- Dynamic parameters
  - $M = 10^{-3}\ kg$, $k = 0.3 \cdot 10^{-3}\ N/m$, $b = 0.09 \cdot 10^{-3}\ Ns/m$, and $F_a = 5\ N$

- 3D monocular robotic ball catching
- Visual grasp of unknown objects
- Real-time deformable object tracking
- Aerial visual manipulation

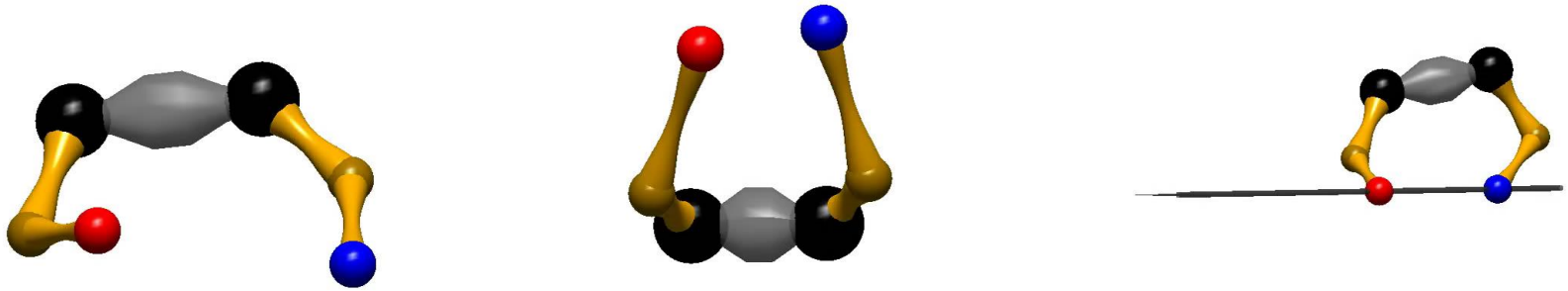
The RoDyMan Project

- Goal
  - Derivation of a unified framework for dynamic manipulation where the mobile nature of the robotic system and the manipulation of non-prehensile non-rigid or deformable objects are explicitly taken into account
- Achievements
  - Novel techniques for 3D object perception, dynamic manipulation control and reactive planning
  - Innovative mobile platform with a torso, two lightweight arms with multi-fingered hands, and a sensorized head for effective execution of complex manipulation tasks, also in the presence of humans

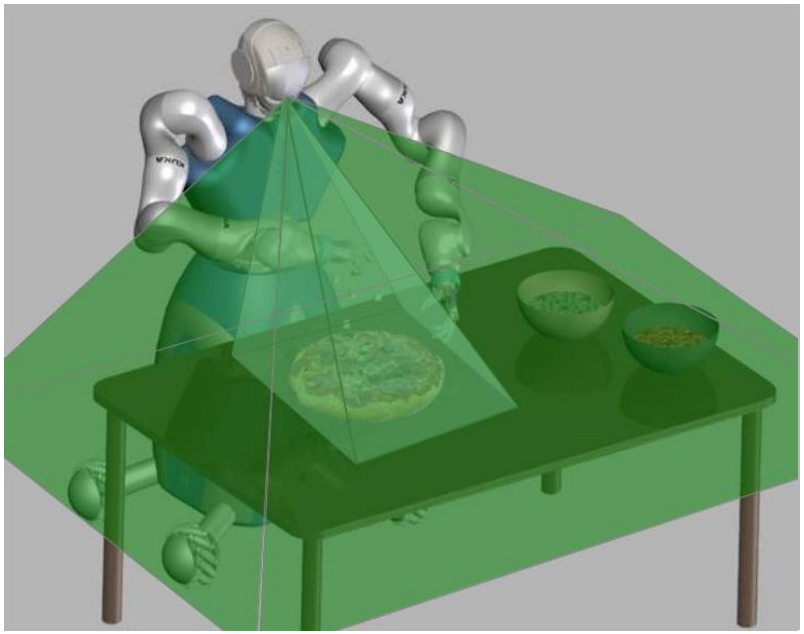


A Challenging Benchmark

- Validation
  - Dynamic manipulation will be tested on an advanced demonstrator, i.e. pizza making process, which is currently unfeasible with the prototypes available in the labs, where the application scenario is conceived to emulate the human ability to carry out a challenging robotic task



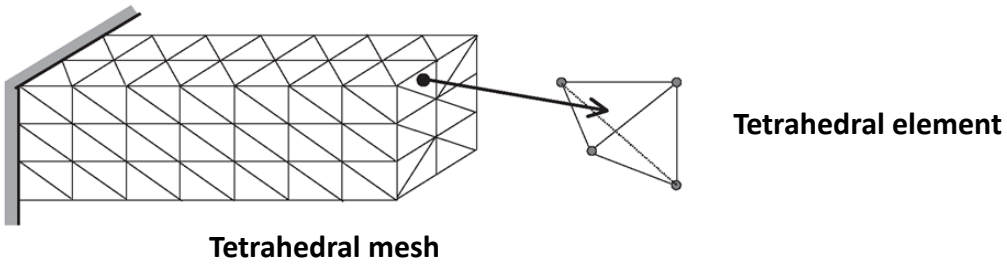
- Dynamic manipulation of deformable objects
  - Perception is a main challenge
  - Use of passive (monocular, stereo cameras) or active (laser scanners, ToF cameras, …) vision sensors
    - Real-time deformable object tracking (e.g. the pizza dough)
    - Environment awareness
      - Localize other objects, robots, people
      - Obstacle avoidance, path planning



Real-time Deformable Object Tracking

- Challenges
  - **Large deformations**, **plastic deformations**
  - Textureless object
  - Occlusions
  - **Real-time**
- RGB-D sensor (active depth sensor)
- Modelling
  - Physics-based model
    - Finite Element Method
- Tracking
  - Prior segmentation of the object
  - Rigid and non-rigid iterative closest points algorithms

- Finite Element Method (FEM)
  - Based on continuous mechanics
  - Approximation over elements (tetrahedrons) of a mesh representing the object:



    - Linear interpolation of the displacements $\hat{\mathbf{u}}$ of the vertices of the element $\mathbf{e}$

$$\mathbf{u}(\mathbf{x}) = \mathbf{N}_e(\mathbf{x})\hat{\mathbf{u}}$$

  - W.r.t. other models (parametric, mass-spring systems), able to model various sorts of deformations (highly elastic, viscous, plastic), better propagation of deformations (**volumetric effects**)
  - Computationally challenging

Linear Co-rotational Approach

- Computation of deformation internal forces $\mathbf{f}_e$ on the vertices of element $\mathbf{e}$
  - Linear elastic internal forces
    - Linear elasticity (Hooke's law) $\sigma_e = \mathbf{C}_e \varepsilon_e$ σ_e: stress ε_e: strain $\mathbf{C}_e$: stiffness tensor

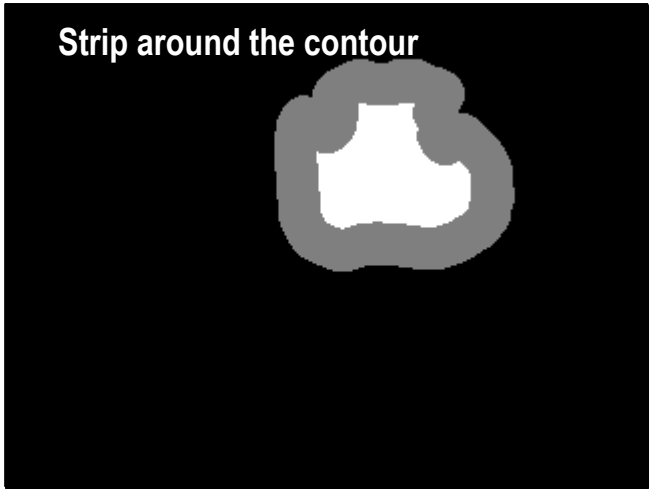
      $\mathbf{f}_e = \mathbf{K}_e \hat{\mathbf{u}}$ $\mathbf{K}_e$: stiffness matrix

      depends on Young modulus and Poisson ratio, can be pre-computed
  - Computationally efficient, but sensitive to rotation transformations of the element
- Co-rotational approach
  - Rotational invariance for the elements, so as to handle large deformations
  - Decomposition of the deformation gradient into a rotation $\mathbf{R}_e$ and a pure deformation giving

$$\mathbf{f}_e = \mathbf{R}_e \mathbf{K}_e (\mathbf{R}_e \hat{\mathbf{x}} - \hat{\mathbf{x}}_0) \qquad \hat{\mathbf{u}} = \hat{\mathbf{x}} - \hat{\mathbf{x}}_0$$

Segmentation

- Temporal coherence by adapting frame-by-frame the segmented area to a strip around the contour



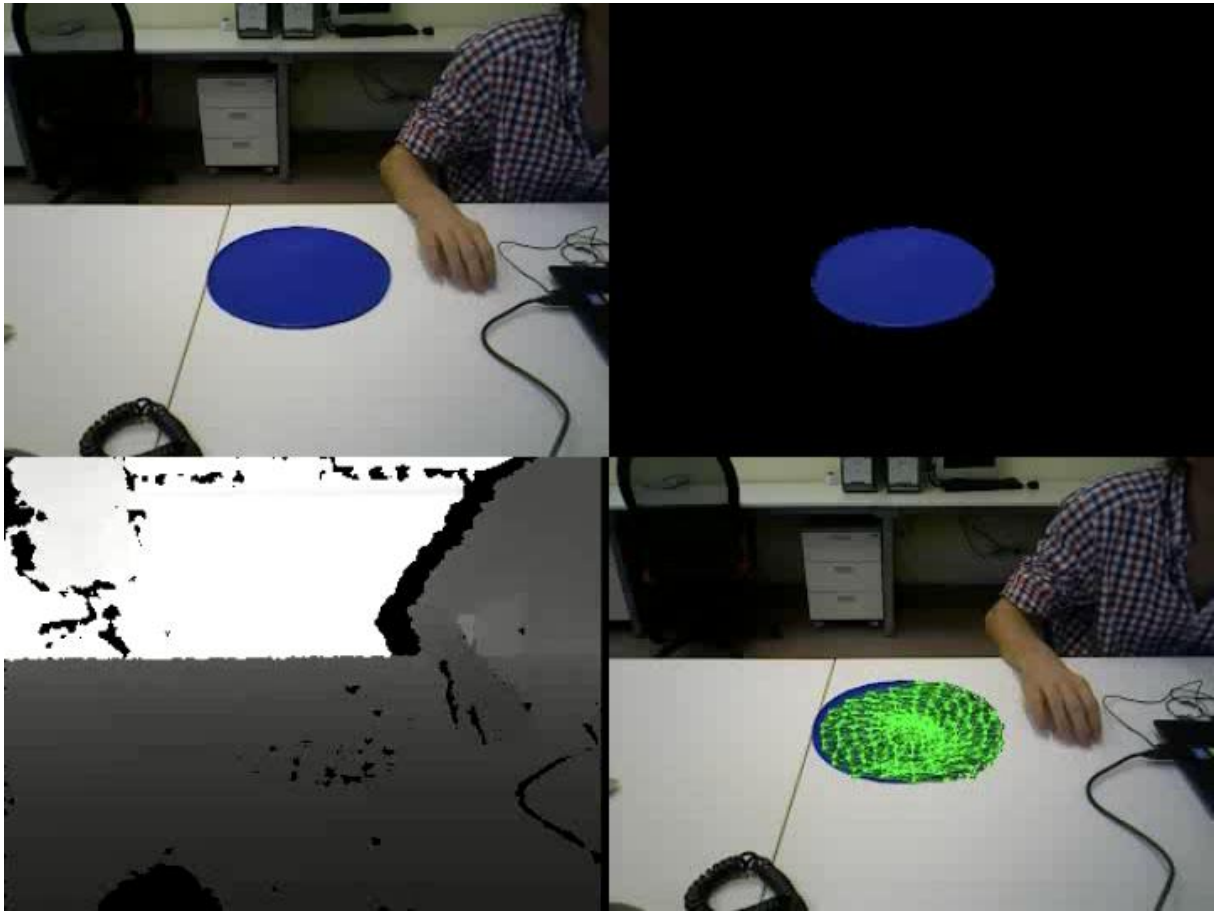
  - Energy minimization effective on the strip
  - Faster
- Real-time issues: CUDA implementation

- Rigid process
  - Iterative Closest Point algorithm, between mesh (rigid) and segmented point cloud, to track fast rigid motions
- Non-rigid process
  - Computation of external forces $\mathbf{f}_{ext}$ exerted on vertices based on segmented point cloud
    - Closest points (nearest neighbour search) visible surface of the mesh/segmented point cloud (compression forces)
    - Closest points (nearest neighbour search) segmented point cloud/segmented point cloud (stretching forces)
    - Elastic forces between correspondences
- Numerical resolution of the ODEs integrating the internal and external forces (Euler implicit integration + conjugate gradient), to update the mesh

$$\mathbf{M}\ddot{\hat{\mathbf{u}}} + \mathbf{D}\dot{\hat{\mathbf{u}}} + \mathbf{f} = \mathbf{f}_{ext}$$

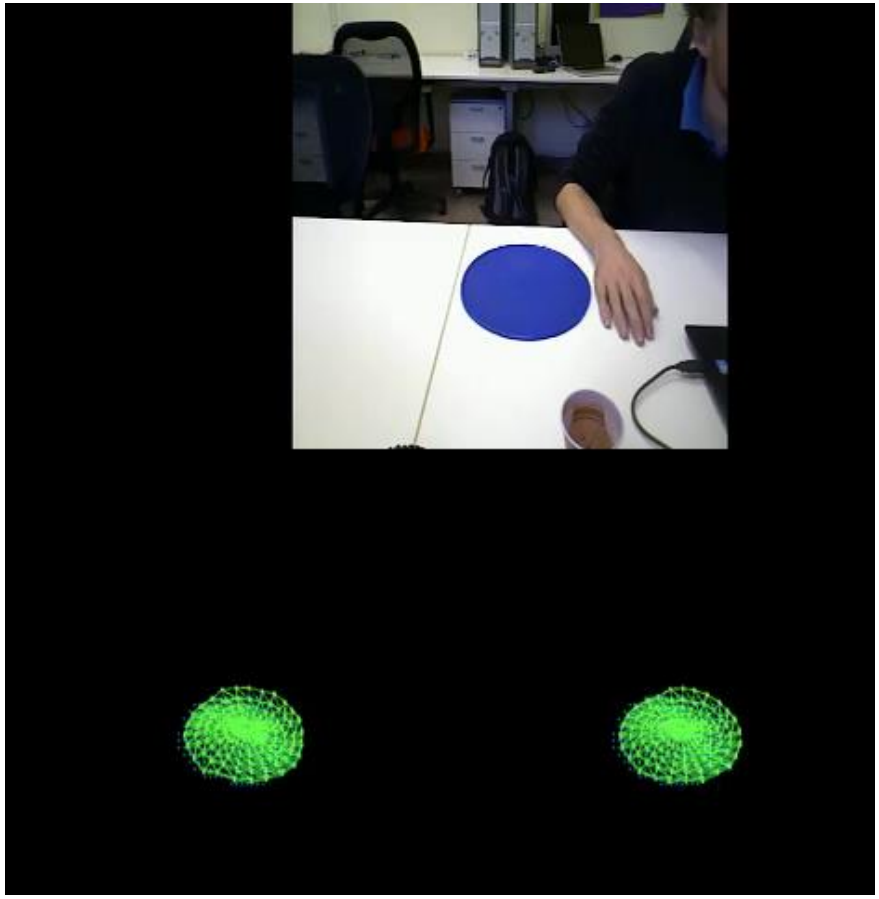
- Real-time implementation based on the SOFA library

Results



Execution: ~40 fps

On-going Work



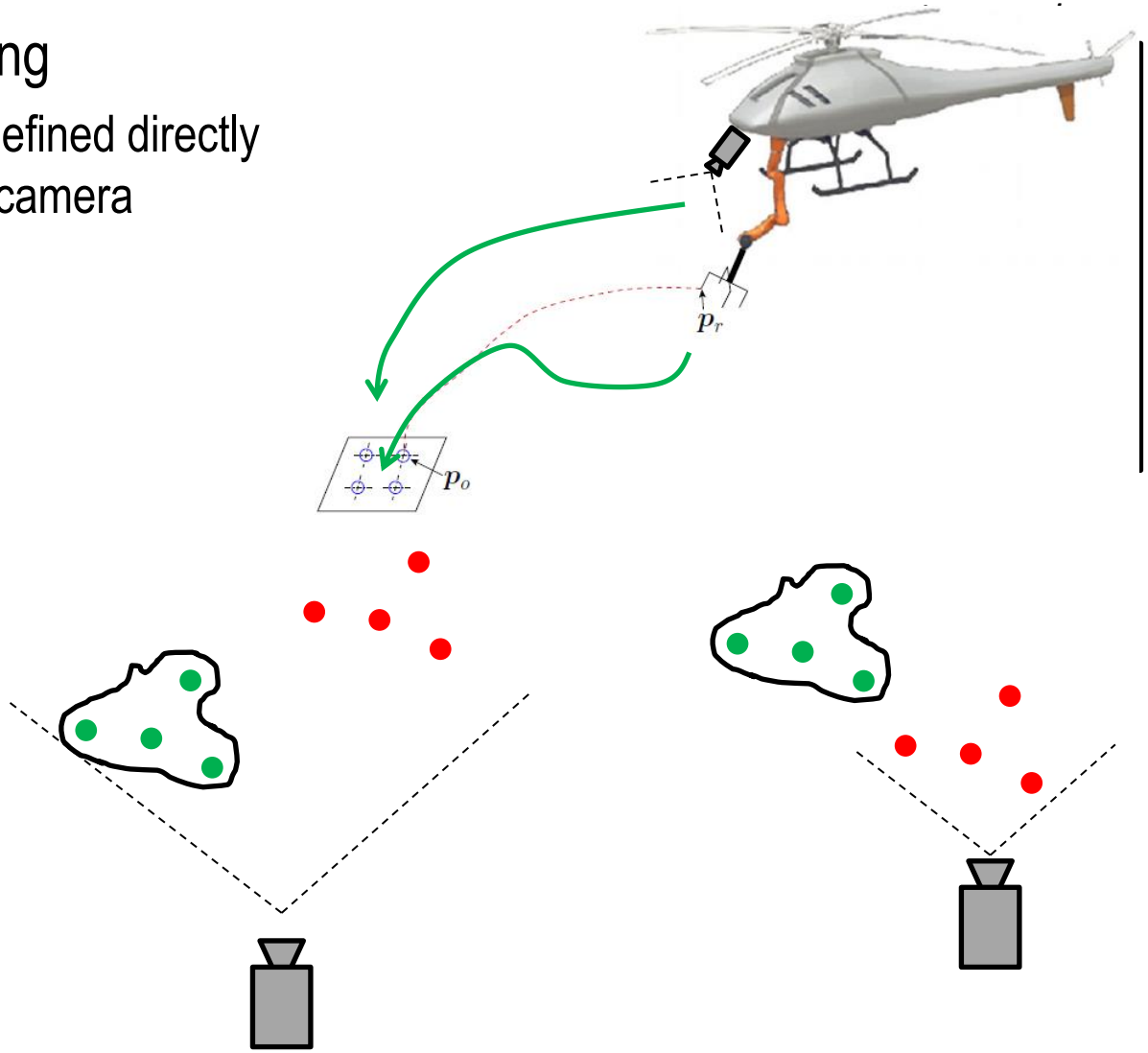
Effect of plastic deformation

- 3D monocular robotic ball catching
- Visual grasp of unknown objects
- Real-time deformable object tracking
- Aerial visual manipulation

- Goal
  - Development and experimental validation of the first cooperative free-flying robot system for assembly and structure construction

Use of Vision for Aerial Manipulation

- Image-based visual servoing
  - Control input (control law) defined directly into the image plane of the camera
- Camera configuration
  - Eye-to-hand
  - Eye-in-hand
  - Onboard-eye-to-hand



Kinematics of the Observed Points

- Image point coordinates

$$\boldsymbol{s}_o = (x_o \;\; y_o)^\top \in \mathbb{R}^2$$

- Target image kinematics

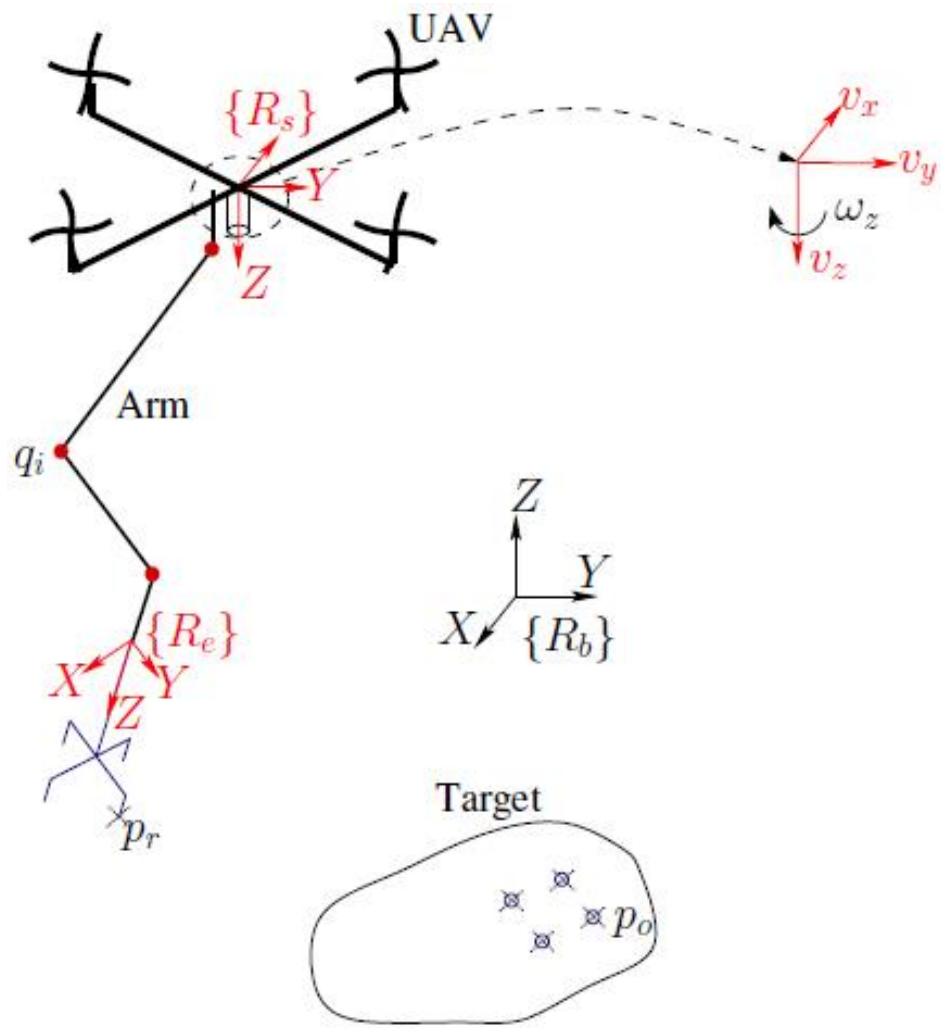
$$\dot{\boldsymbol{s}}_o = \boldsymbol{L}_o \mathbf{v}$$

  - L_O interaction matrix

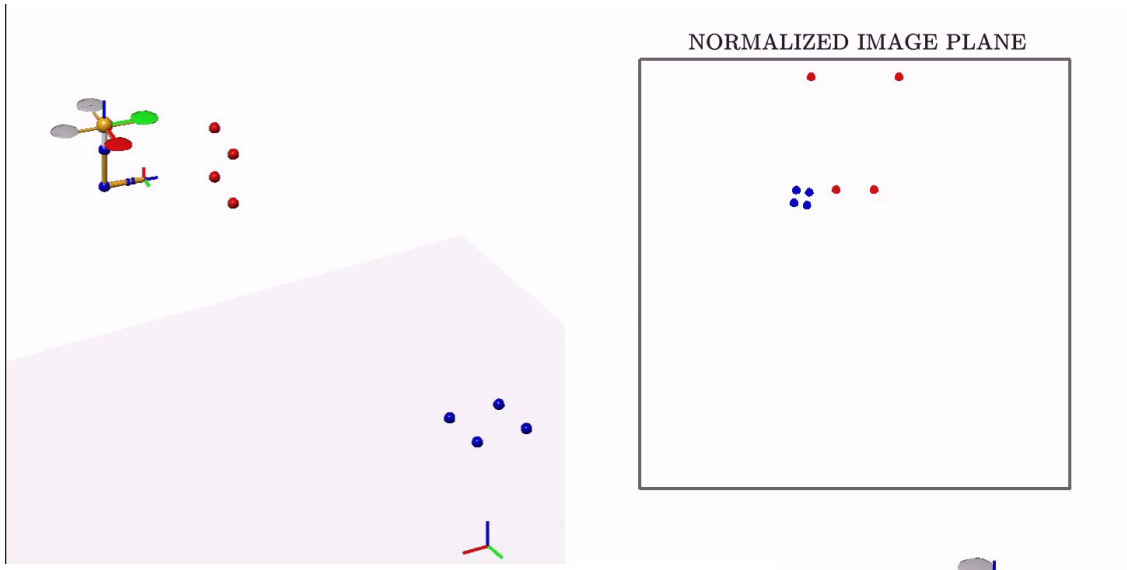
- Carried points kinematics

$$\dot{\boldsymbol{s}}_r = \boldsymbol{L}_r \, \dot{\boldsymbol{q}}$$

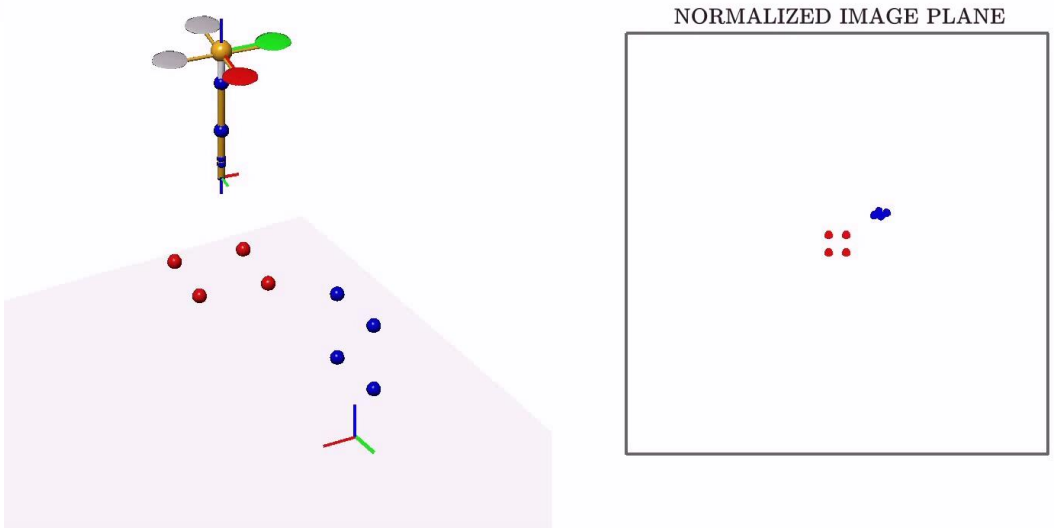
  - q robotic arm joint vector



Point-based Visual Servoing



- Motion of the image features in the image plane is smooth and quite straightforward



- Drawback: Cartesian motion in the 3D space can be undesirable and/or unpredictable, that could not be borne in practice

- Point-based image moments

$$m_{pq} = \sum_{i=1}^{m} (y_i - y_{i-1})\, \sigma_{i,pq}$$

$$\begin{cases} \sigma_{i,pq} = \sum_{k=0}^{p+1} \sum_{j=0}^{q} a_{kj}\, f_x\, f_y \\ f_x = x_i^k\, x_{i-1}^{p+1-k}\,, \quad f_y = y_i^j\, y_{i-1}^{q-j} \end{cases}$$

- Point-based image moments kinematics

$$\begin{aligned} \dot{m}_{pq} &= \sum_{i=1}^{m} \left[\sigma_{i,pq} (\boldsymbol{L}_{y_i} - \boldsymbol{L}_{y_{i-1}}) \dot{\boldsymbol{\zeta}} + (y_i - y_{i-1}) \boldsymbol{L}_{\sigma_{i,pq}} \dot{\boldsymbol{\zeta}} \right] \\ &= \left\{ \sum_{i=1}^{n} \left[\sigma_{i,pq} (\boldsymbol{L}_{y_i} - \boldsymbol{L}_{y_{i-1}}) + (y_i - y_{i-1}) \boldsymbol{L}_{\sigma_{i,pq}} \right] \right\} \dot{\boldsymbol{\zeta}} \\ &= \boldsymbol{L}_{m_{pq}}\, \dot{\boldsymbol{\zeta}}\,, \end{aligned}$$

where

$$\boldsymbol{L}_{m_{pq}} = \sum_{i=1}^{m} \left[\sigma_{i,pq} (\boldsymbol{L}_{y_i} - \boldsymbol{L}_{y_{i-1}}) + (y_i - y_{i-1}) \boldsymbol{L}_{\sigma_{i,pq}} \right] \in \mathbb{R}^{1 \times \acute{n}}$$

Moment-based Visual Control

- Feedback visual error $\boldsymbol{e}_m = \boldsymbol{s}_{mr} - \boldsymbol{s}_{mo} \in \mathbb{R}^6$
- Visual error kinematics

$$\dot{\boldsymbol{e}}_m = \begin{pmatrix} \boldsymbol{L}_{mr} & -\boldsymbol{L}_{mo_v} & -\boldsymbol{L}_{mo_{\omega_z}} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{v} \\ \omega_z \end{pmatrix} - \begin{pmatrix} \boldsymbol{L}_{mo_{\omega_x}} & \boldsymbol{L}_{mo_{\omega_y}} \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \end{pmatrix} = \boldsymbol{J}_m \dot{\boldsymbol{\xi}} - \overline{\boldsymbol{L}}_{m\omega} \bar{\boldsymbol{\omega}}$$

- Proposed control law that nullifies visual error (main task)

$$\dot{\boldsymbol{\xi}} := \boldsymbol{J}_m^+ \left(-\lambda_1 \boldsymbol{e}_m + \bar{\boldsymbol{L}}_{m\omega} \bar{\boldsymbol{\omega}}\right) + \left(\boldsymbol{I}_{n+4} - \boldsymbol{J}_m^+ \boldsymbol{J}_m\right) \dot{\boldsymbol{\xi}}_2$$

- Secondary task: nullify error on image points coordinates

$$\dot{\boldsymbol{\xi}}_2 := \boldsymbol{J}_c^+ \left(\boldsymbol{L}_{\bar{\omega}} \bar{\boldsymbol{\omega}} - \lambda_2 \boldsymbol{e}_c\right) + \left(\boldsymbol{I}_{n+4} - \boldsymbol{J}_c^+ \boldsymbol{J}_c\right) \dot{\boldsymbol{\xi}}_3$$

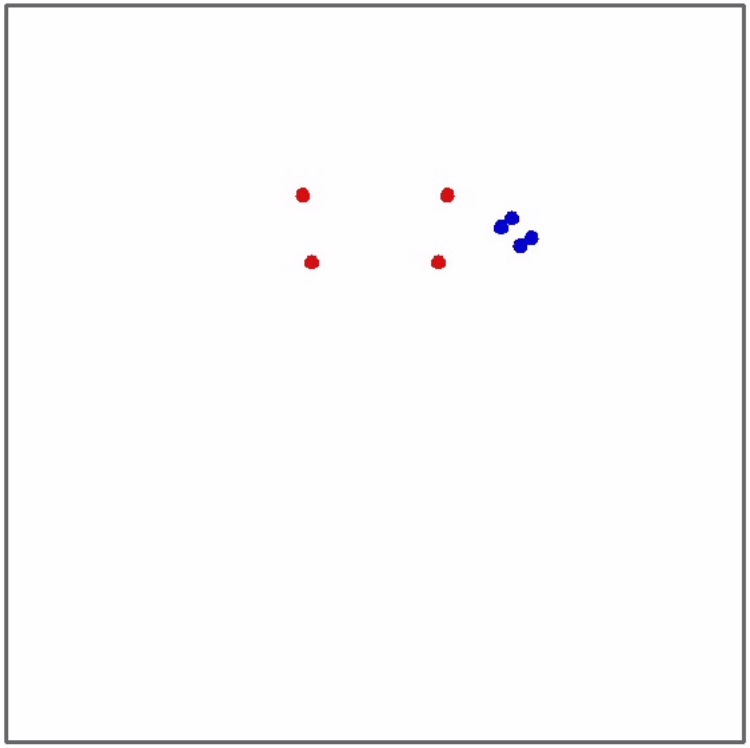
- Third task: aligning manipulator with UAV's center of gravity

$$\dot{\boldsymbol{\xi}}_3 = \left(\dot{\boldsymbol{q}}_g^\top \; \boldsymbol{0}_4^\top\right)^\top \qquad \dot{\boldsymbol{q}}_g := \boldsymbol{J}_t^+ \left(-\lambda_3 \bar{\boldsymbol{t}} - \bar{\boldsymbol{t}}_0\right) + \left(\boldsymbol{I}_n - \boldsymbol{J}_t^+ \boldsymbol{J}_t\right) \dot{\boldsymbol{q}}_l$$

- Fourth task: maximize dexterity

$$\dot{q}_{li} = -\lambda_4 \left[q_i - \frac{1}{2}\left(q_{mi} + q_{Mi}\right) \right]$$

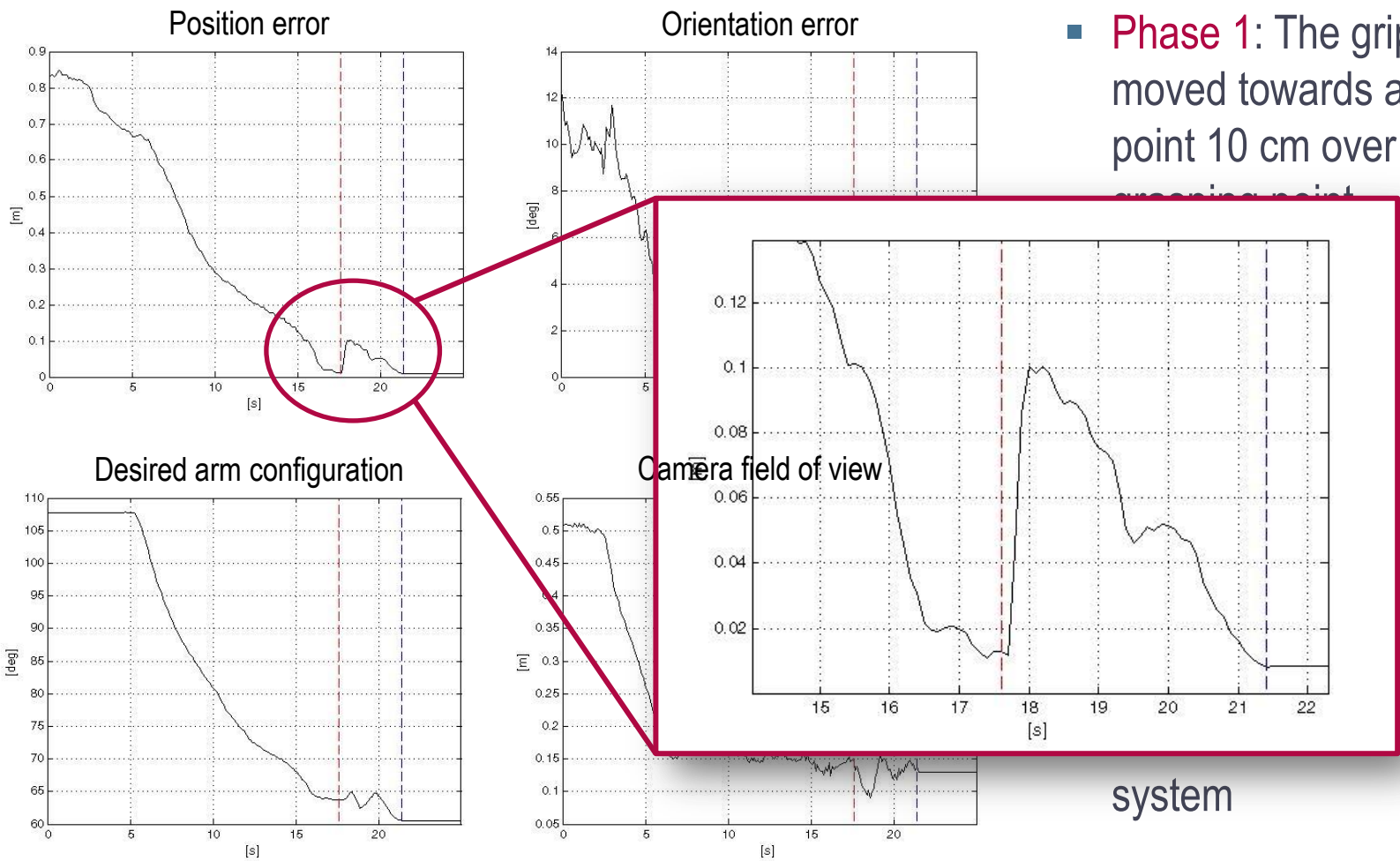
Orthogonal Assembly



Visual Grasping of a Bar

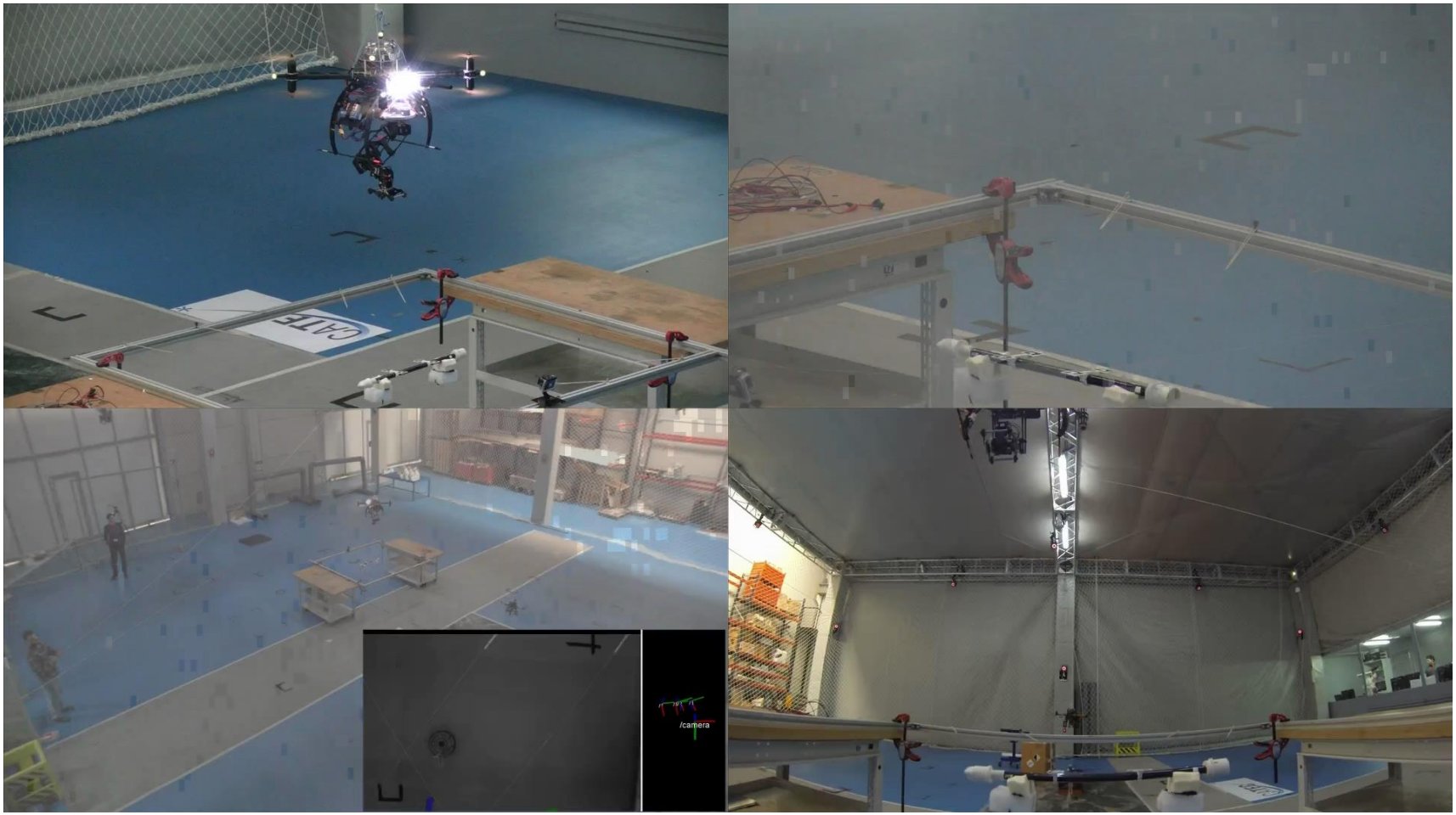
- The grasp of a bar has been experimentally performed with the use of visual information provided by a camera mounted on the vehicle base (position-based approach)
- The hierarchical task composition employed for the visual grasping experiment, in the adopted hierarchical order, is as follows:
  1. Camera field of view
  2. Gripper orientation
  3. Gripper position
  4. Arm-joint limits (i.e. desired arm configuration)
  5. Yaw approaching angle

Experiments



- Phase 1: The gripper is moved towards a way point 10 cm over the grasping point
- al
- is
- smoothly
- t position
- asp
- to the system

Results



Thanks Very Much Indeed ☺

THE PRISMA TEAM

Bruno Siciliano • Luigi Villani • Vincenzo Lippiello
Alberto Finzi • Silvia Rossi • Franco Cutugno
Fanny Ficuciello • Fabio Ruggiero • Agostino De Santis
Rafik Mebarki • Antoine Petit • Jun Nishiyama • Alejandro Donaire
Francesca Cordella • Mariacarla Staffa • Antonio Origlia
Daniela D'Auria • Luigi Pelliccia • Salvatore Iengo
Giuseppe Loianno • Luca Buonocore • Jonathan Cacace
Diana Serra • Mahdi Momeni
Valeria Federico • Jonathan van der Meer

FUNDING

8.3 M€