Università di Genova

UniGe | DIBRIS

Real-time Grasping Using Event-based Camera

In partial fulfillment of the requirements for the degree of
Laurea Magistrale in Robotics Engineering

Presented by:
Hocine DELALA

SUPERVISOR:
PROF. CARMINE RECCHIUTO

October 14, 2025

Introduction

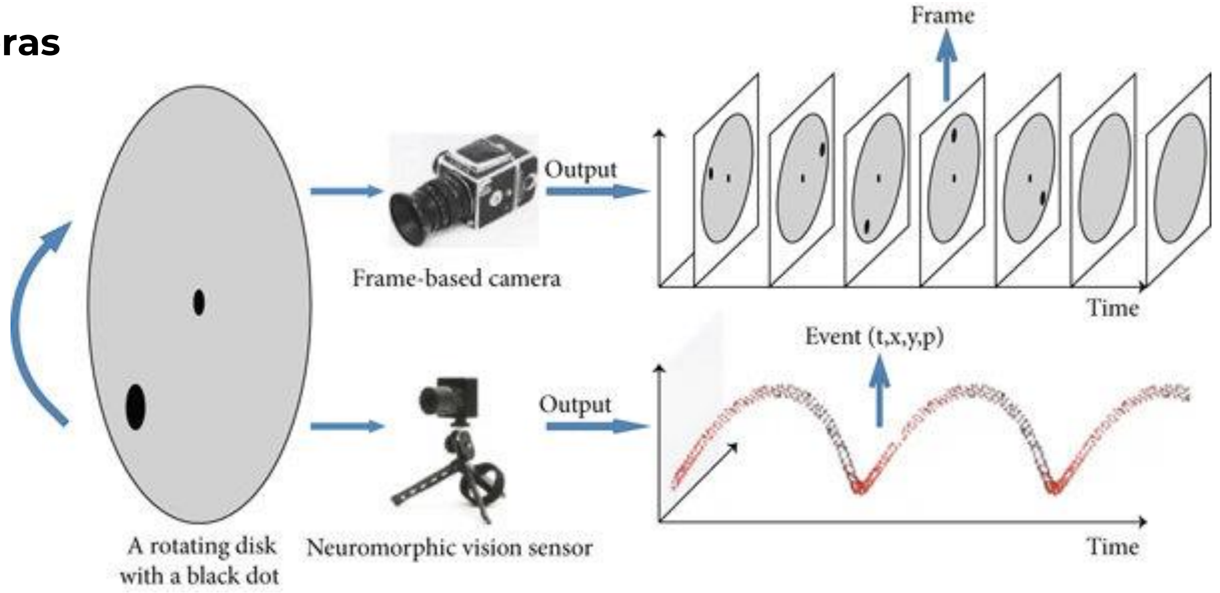
Traditional Frame-Based Cameras

- ❌ High Latency
- ❌ Motion Blur
- ❌ Poor Dynamic Range

⬇️ VS ⬇️

Event-Based Cameras (DVS)

- ✅ Microsecond Latency
- ✅ No Motion Blur
- ✅ 120dB Dynamic Range
- ✅ Power Efficiency



How can we utilize the advantages of Event-Based Cameras to achieve real-time grasping for industrial manipulation?

State of the Art - Existing Approaches

Model-Based [1]	Model-Free [1]
Requirements: ● Known CAD model. ● 3D registration. ● EMVS [2] for reconstruction. **Output:** ● 6-DoF pose.	**Requirements:** ● No CAD needed ● MEMS clustering ● EMVS for depth(one time) **Output:** ● Centroid + orientation

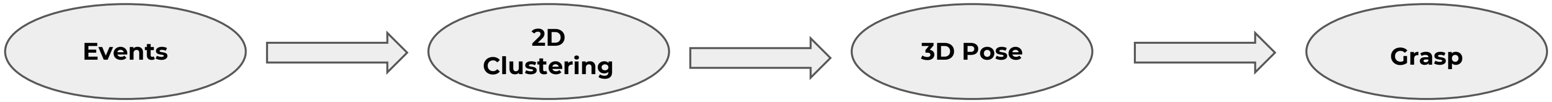
Where does our work fit?

Middle Ground
- Lightweight geometric priors (object dimensions)

- No 3D reconstruction (Simpler & faster)

- Suitable for industrial scenarios (catalogued objects)

Can we grasp objects reliably using only 2D event clustering and geometric priors?



The Answer is **YES!**

- ✓ Reconstruction-free perception pipeline.
- ✓ Static + dynamic grasping modes.
- ✓ Experimental validation (mm-level accuracy).

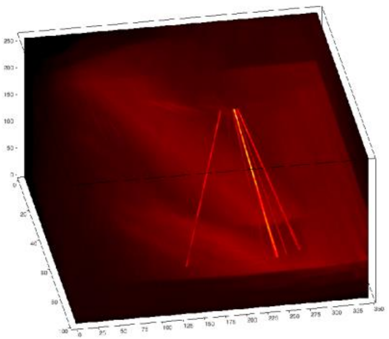
Methodology

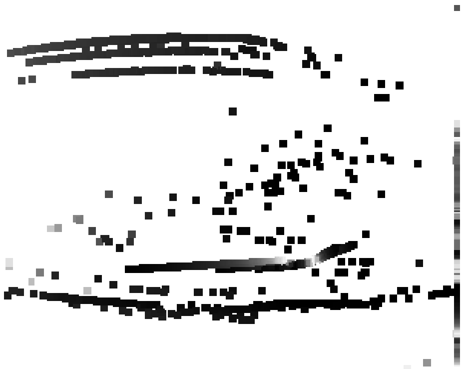
Initial Approach - EMVS (3D Reconstruction)

Offline Tests Results:

✗ Sparse, noisy reconstruction

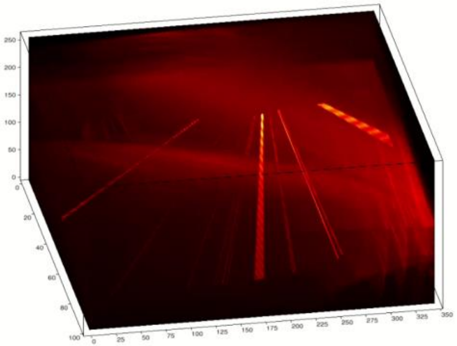
✗ Insufficient for object segmentation

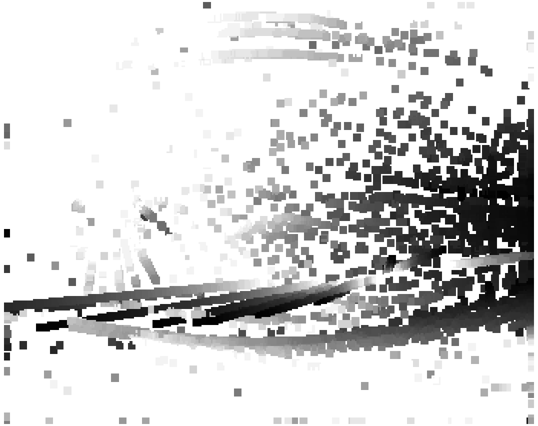




Limitations:

Complex and computationally heavy w.r.t 2D approaches.





Pivoting to 2D Clustering approach

Why 2D Clustering is Enough ?

- Grasp always occurs at the same Z-level.
- Known object height (geometric prior).

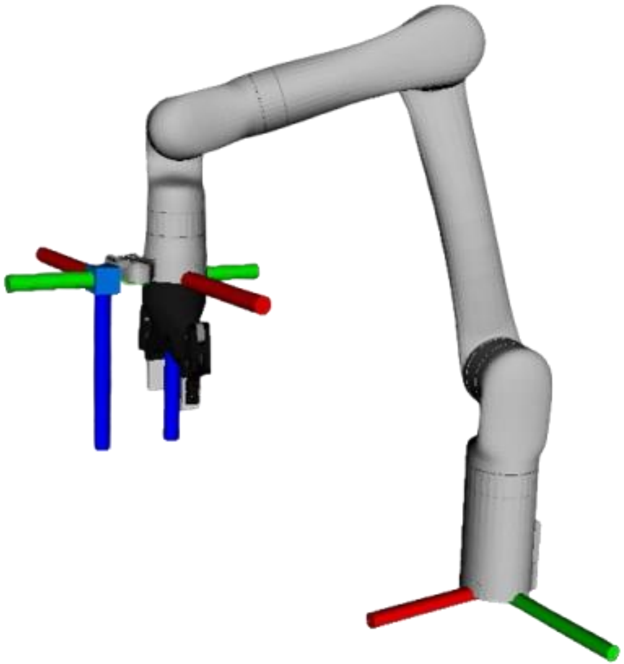
Perception Approach

Stage 1: 2D Clustering & Validation + 3D back-projection

Stage 2: Temporal 3D Filtering

Eye-in-Hand Camera Frame (URDF/TF)

- Eye-in-hand.
- Fixed joint.
- Transform measurement.
- Known height of the object.



Stage 1: 2D Clustering & Validation

Step 1: DBSCAN Clustering
- Accumulate recent events: $E_{\Delta t}$
- Cluster in pixel space (cluster eps, min samples)

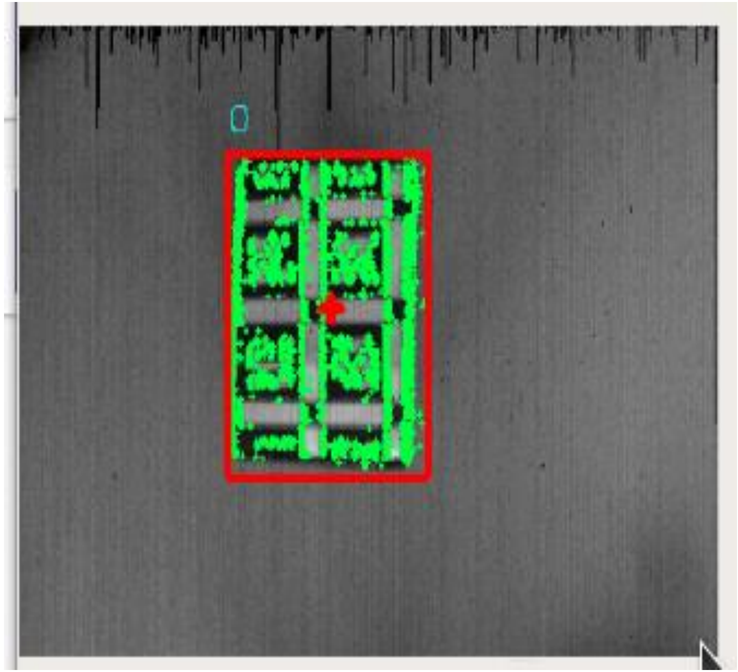
Step 2: Rectangle Fitting
- We fit the cluster to the object dim.
- Extract: center (u,v), dimensions, angle

Step 3: Size-Consistency Check (cluster Validation)

Rejects the cluster if the geometric error exceeds a threshold

Step 4: 3D Back-projection

Back-project the 3D position into the robot's base frame.



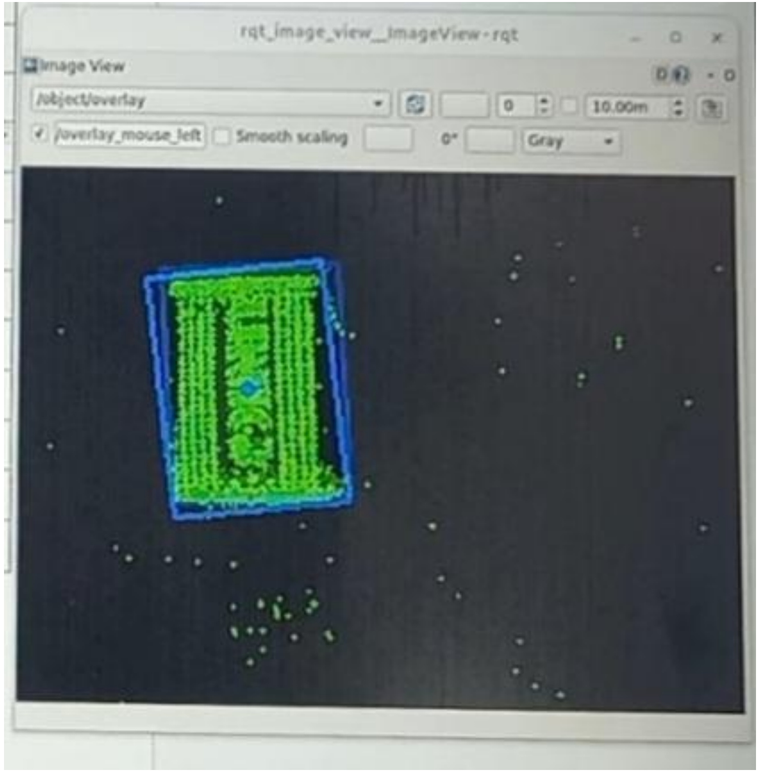
Stage 2: Temporal 3D Filtering

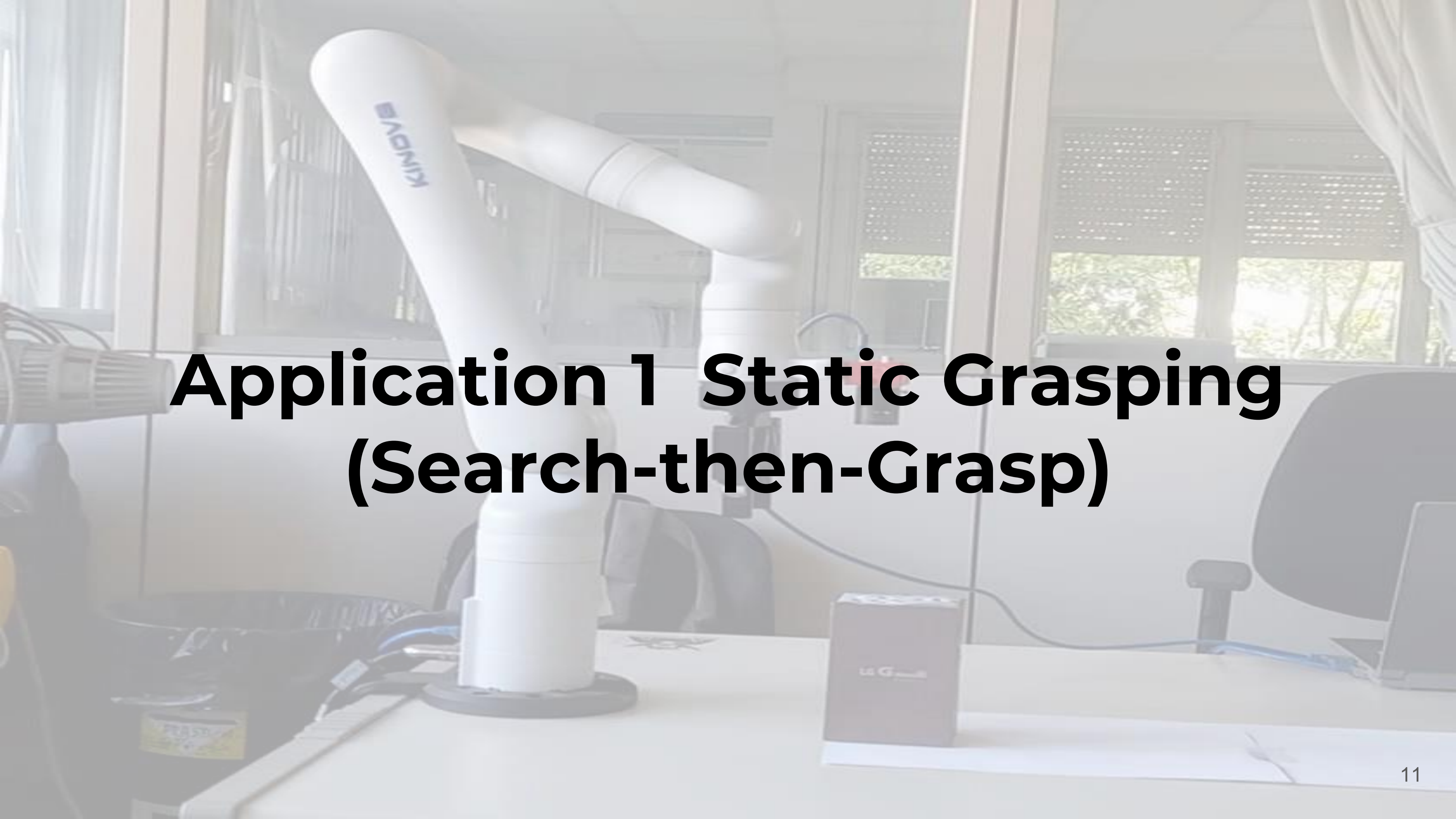
1. Pose Accumulation:
Collecting validated 3D poses from stage 1 while the robot is moving.

2. 3D DBSCAN Clustering
Apply a second layer of DBSCAN, this time in the 3D Base Frame.

3. Compute stable pose:
Final centroid averaging.

Output: Stable, filtered pose.





Application 1 Static Grasping
(Search-then-Grasp)

Scenario: Object stationary on stopped conveyor.

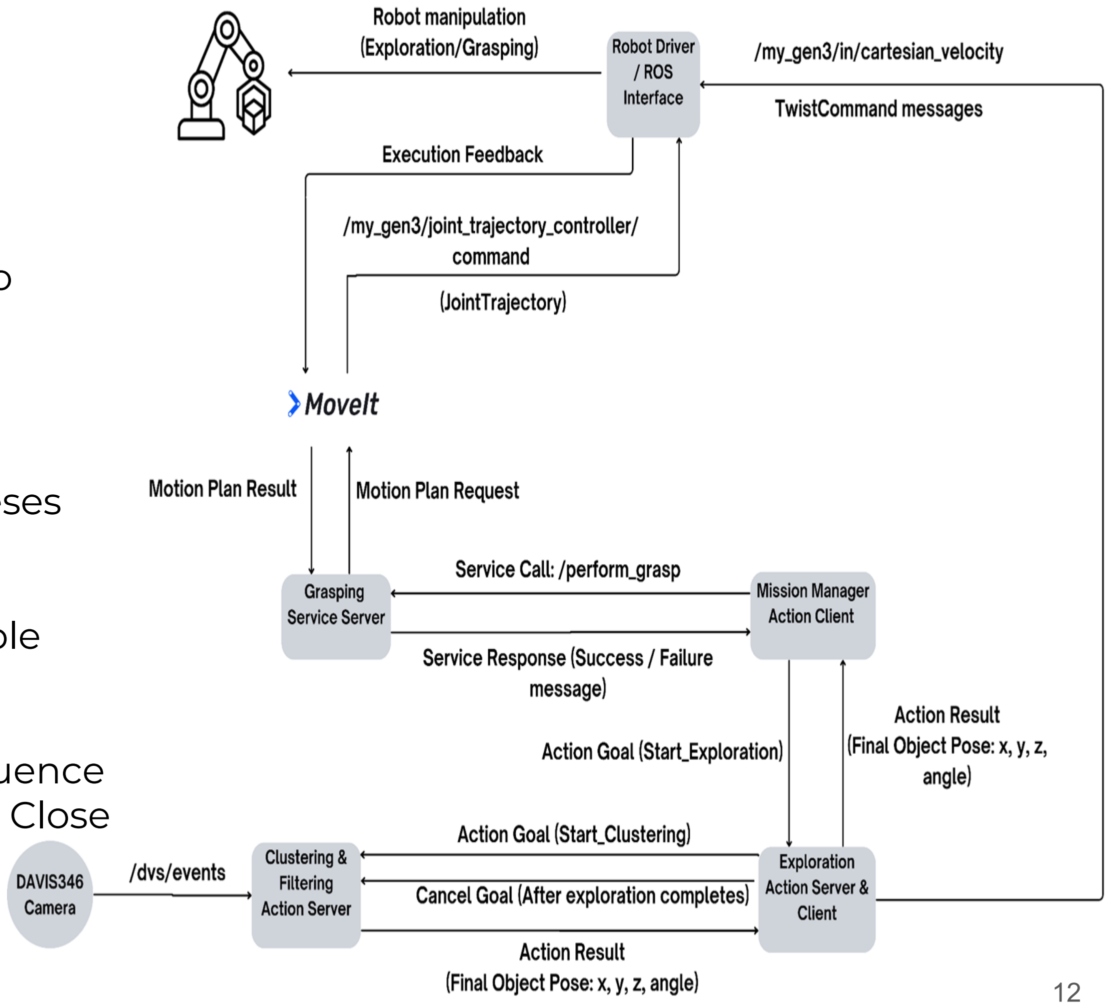
Workflow:

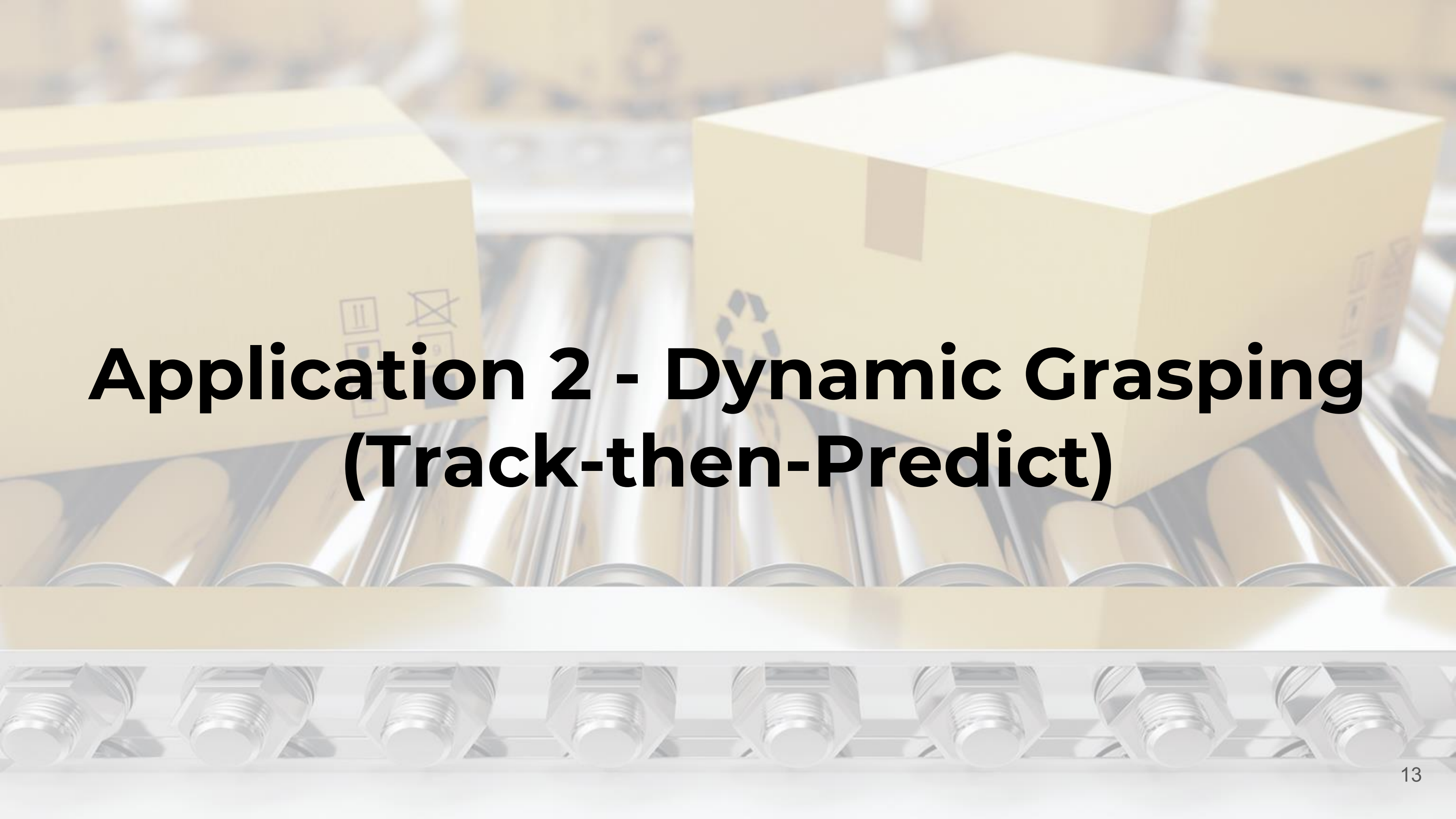
1. **Exploration:** Robot performs motion to generate events
• Square trajectory
• Yaw sweep

2. **Clustering:** Accumulate pose hypotheses during motion.

3. **Filtering:** Temporal 3D DBSCAN → stable pose.

4. **Grasping: MoveIt** executes grasp sequence
• Approach → Orient gripper → Descend → Close → Lift



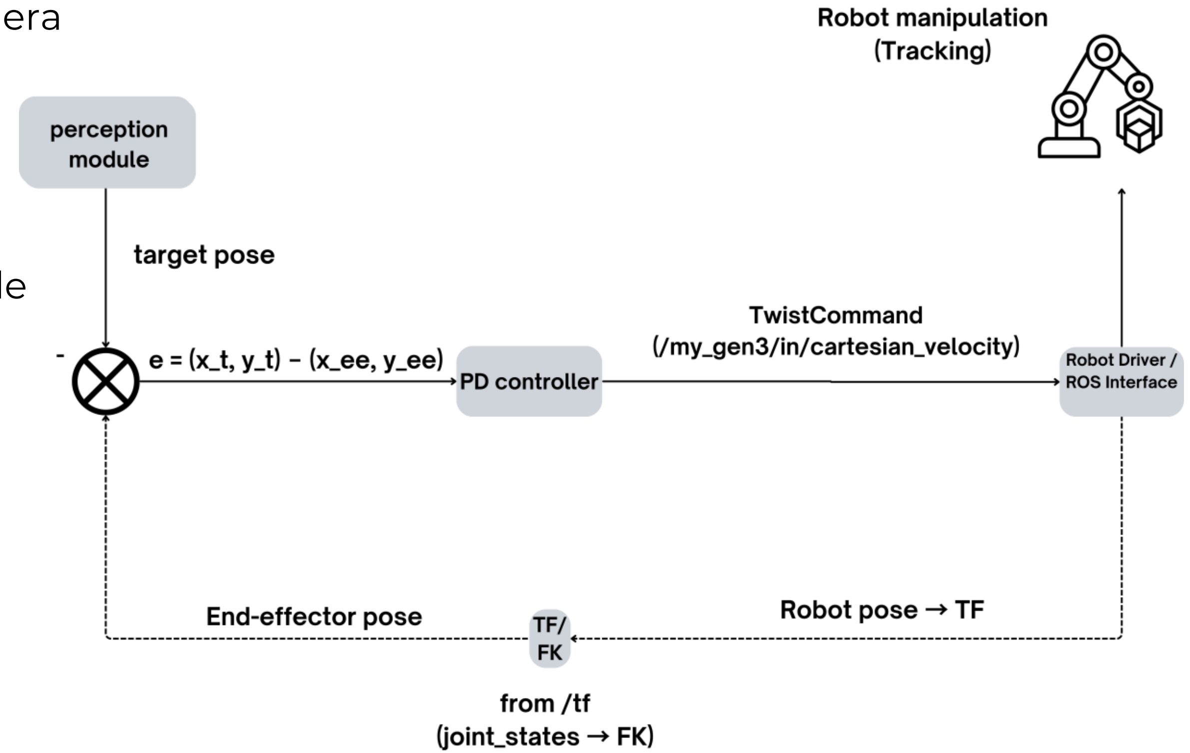


Application 2 - Dynamic Grasping (Track-then-Predict)

Position-Based Visual Servoing (PBVS)

Goal: Keep object centered in camera view while tracking

- Clustering node publishes stable poses continuously (~50 Hz).



Velocity Estimation & Predictive Interception

Velocity Estimation (Kalman Filter):
State vector (4D): $X = \left[\mathrm{x}, \mathrm{y}, v_x, v_y\right]^{\mathrm{T}}$
Input: Position measurements $[x, y]$ from clustering. **Output:** Velocity estimate $v_{est} = \left[v_x, v_y\right]$.

1. **PREDICT:** Assume constant velocity $x_{new} = x_{old} + v_{old} \times \Delta t$

2. **UPDATE:** When new measurement arrives
→ Refine velocity estimate by comparing prediction to measurement

3. **REPEAT:** More measurements → Better understanding of true velocity → Less influenced by noise

✓ Gave smoother better results than simple averaging (finite differencing v = Δp/Δt).

Linear Motion Prediction:

$$p_{pred} = p_{start} + v_{est} \cdot \Delta t$$

Feed-Forward Control:

$$v_{control} = \underset{\substack{\uparrow \\ \text{correction}}}{k_P \cdot e_{pred}} + \underset{\substack{\uparrow \\ \text{compensation}}}{v_{est}}$$

Scenario: Object moving on a conveyor

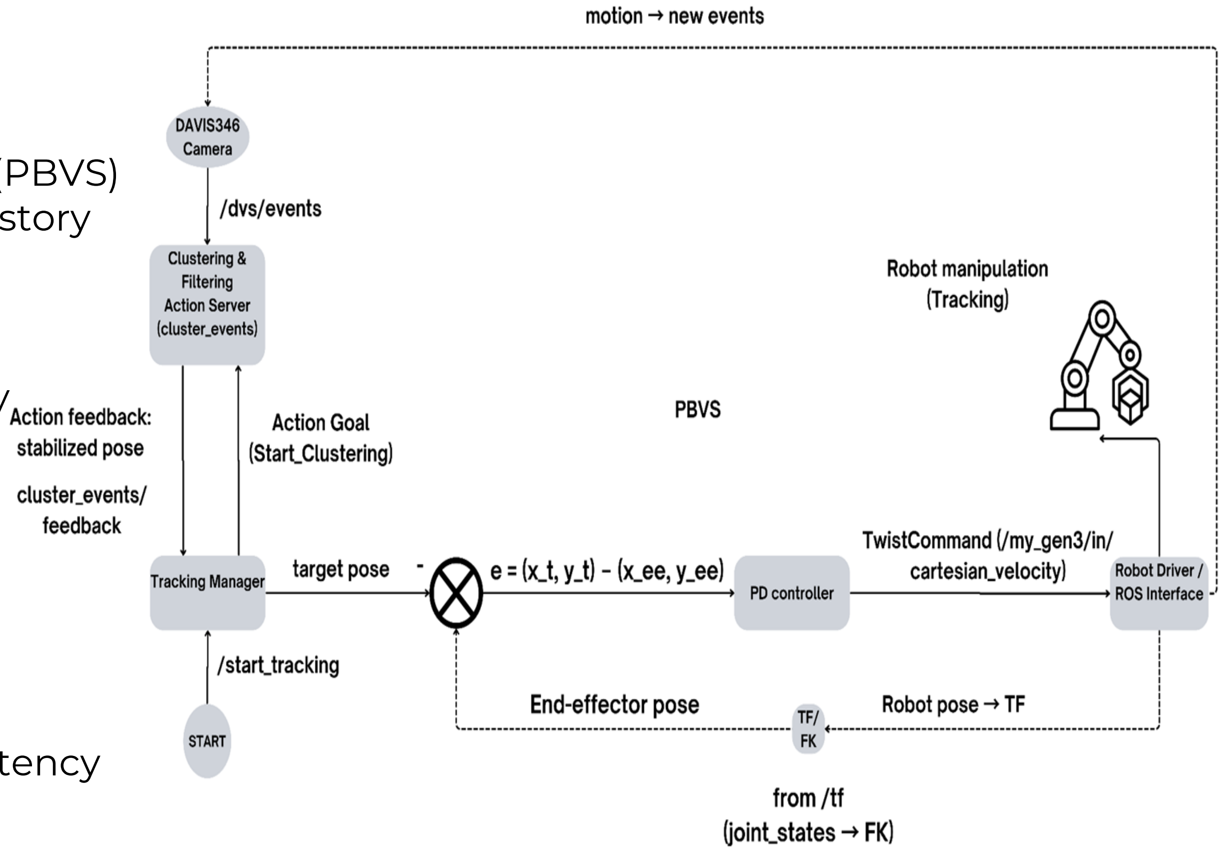
1. **TRACKING**

- Position-Based Visual Servoing (PBVS)
- Follow object, collect position history
- Controller: $v = k_P \cdot \mathrm{e} + k_d \cdot \dot{\mathrm{e}}$

2. **ESTIMATING**
- Kalman Filter on position history
- Estimate planar velocity: (v_x, v_y)

3. **BLIND DESCENT**
- Predict intercept position.
- Feed-forward control loop.
- Descend and close gripper.

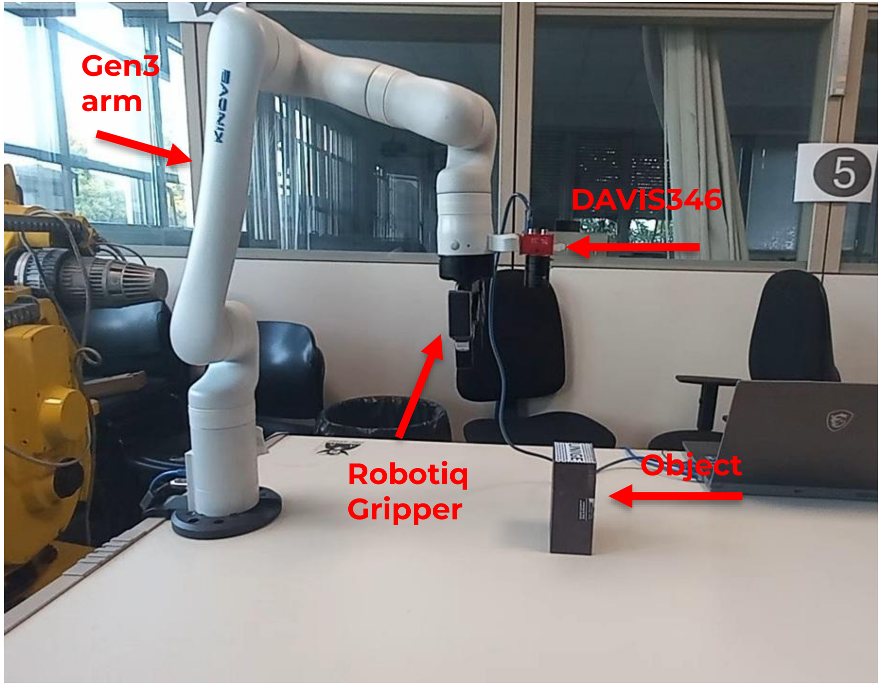
Predictive grasp compensates for latency



Experimental Setup

Hardware:

- **Robot:** Kinova Gen3 (6-DOF, 891mm reach)
- **Gripper:** Robotiq 2F-85 (85mm stroke)
- **Camera:** DAVIS346 (346×260 pixels, eye-in-hand)
- **Object:** Rectangular box (8.5×6.0×14.5 cm)

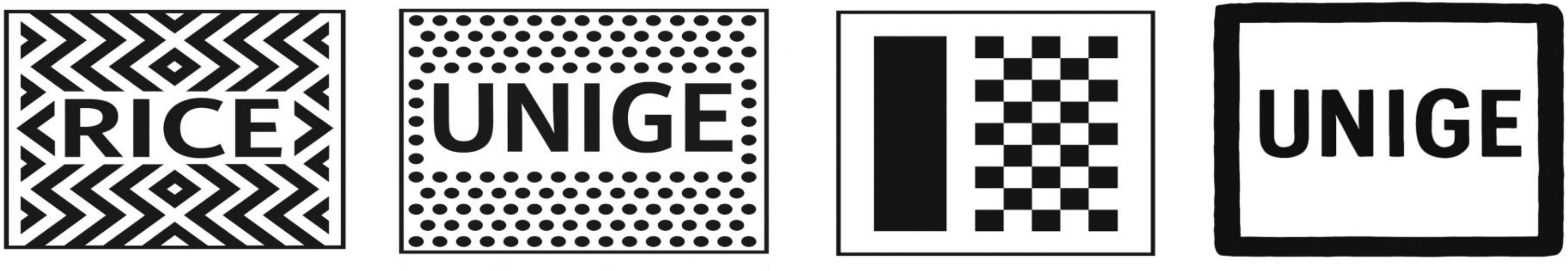


Software:

- ROS Kinetic (Docker container)
- MoveIt for motion planning
- Python scripts.

Experimental Variations in case of Static Grasping

Object Surface Patterns Tested:



Exploration Trajectories:

- Square (20×20cm XY)
- Yaw sweep (±30°)

Why test variations?

→ Evaluate robustness to texture and motion type
→ To find the pattern that is motion independent, we look for the pattern that excites the most events, regardless of the motions type.

Evaluation Metrics

1. **Centroid Accuracy (mm):**

$$e = \sqrt{(x_{\text{det}} - x_{\text{ref}})^2 + (y_{\text{det}} - y_{\text{ref}})^2}$$

The object was positioned in a known position and used as a reference point to calculate the error.

2. **Events Count:**

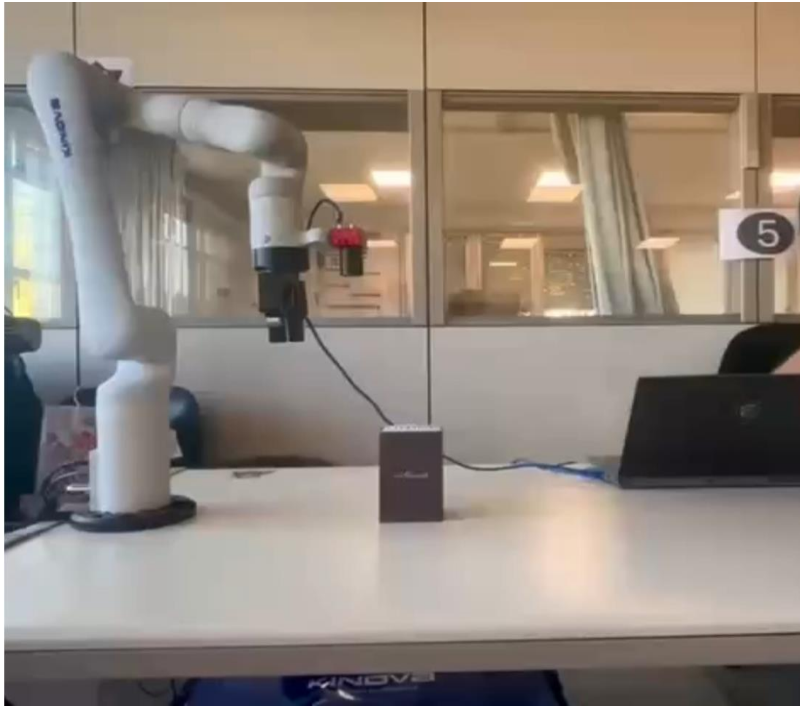
Number of events within detected bounding box

→ Indicates feature richness

3. **Grasp Success Rate :**

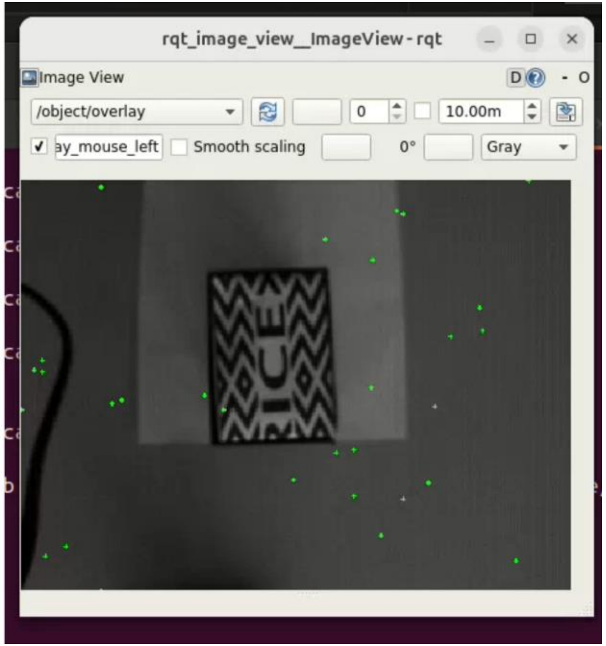
$$\text{Success Rate} = \frac{\text{Successful Grasps}}{\text{Total Attempts}} \times 100\%$$

Static Grasping (Square Trajectory): Visual Results



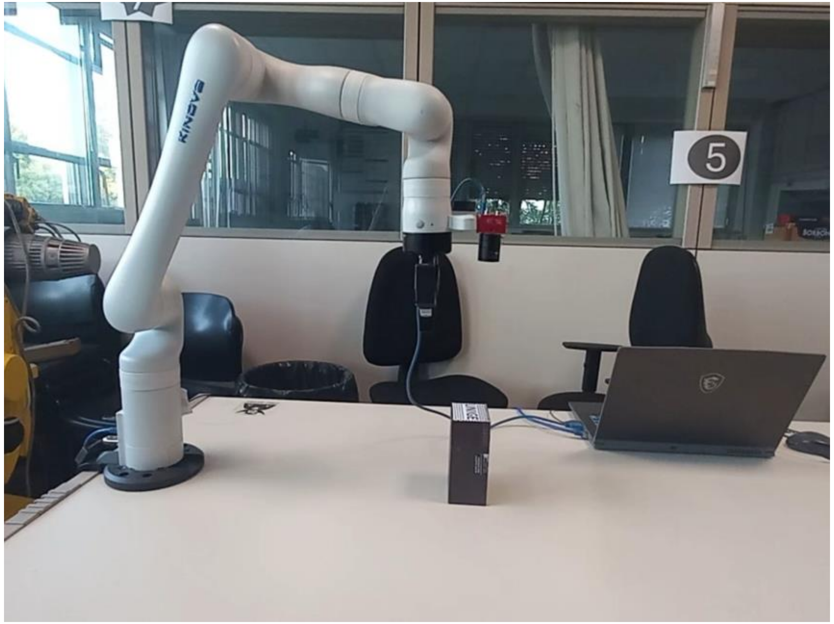
Static Grasping(Yaw sweep): Visual Results

Overlay image_view



yaw sweep exploration trajectory (no patterns)

Static Grasping (Orientation): Visual Results



Static Grasping Results Analysis

Square trajectory Accuracy

Trial	No pattern	RICE zigzag	UNIGE borders	UNIGE dots	Checkerboard
1	12.50	5.10	11.70	11.70	3.60
2	1.40	5.00	8.40	10.70	2.80
3	3.60	2.20	8.20	11.70	1.60
4	0.00	5.10	6.40	5.10	2.80
5	2.20	8.50	9.20	10.40	2.40
6	10.00	2.20	7.00	9.40	3.60
7	5.10	3.60	7.80	6.70	3.60
8	5.10	3.60	9.40	6.40	5.80
9	7.10	2.20	8.60	7.60	3.20
10	6.10	3.20	7.80	5.80	4.50
Average(mm)	**5.31**	**4.07**	**8.45**	**8.55**	**3.39**
Std. dev.	**3.25**	**2.11**	**1.90**	**2.35**	**1.42**
Success (%)	**90**	**100**	**80**	**100**	**100**

Yaw sweep trajectory Accuracy

Trial	No pattern	RICE zigzag	UNIGE dots
1	5.10	12.50	6.40
2	3.60	5.10	7.60
3	8.00	7.60	7.10
4	4.10	11.20	9.10
5	1.00	10.00	6.40
6	6.20	10.60	6.40
7	2.50	10.00	7.60
8	7.50	7.30	6.10
9	4.80	7.10	5.40
10	2.90	7.10	8.20
Average (mm)	**4.57**	**8.85**	**7.03**
Std. dev.	**2.22**	**2.33**	**1.11**
Success (%)	**100**	**100**	**100**

Static Grasping Results Summary

Explore Motion	Pattern	Avg. Centroid Err (mm)	Success (%)
Square	RICE zigzag	4.07	100
Square	Checkerboard	3.39	100
Yaw Sweep	No pattern	4.57	100
Yaw Sweep	UNIGE dots	7.03	100

✓ Square + patterns: Best accuracy (3-4mm)
✓ Yaw + edges: Prefers clean boundaries (No Pattern)
✓ No pattern: Still works! (5.31mm) → Critical for real industrial items

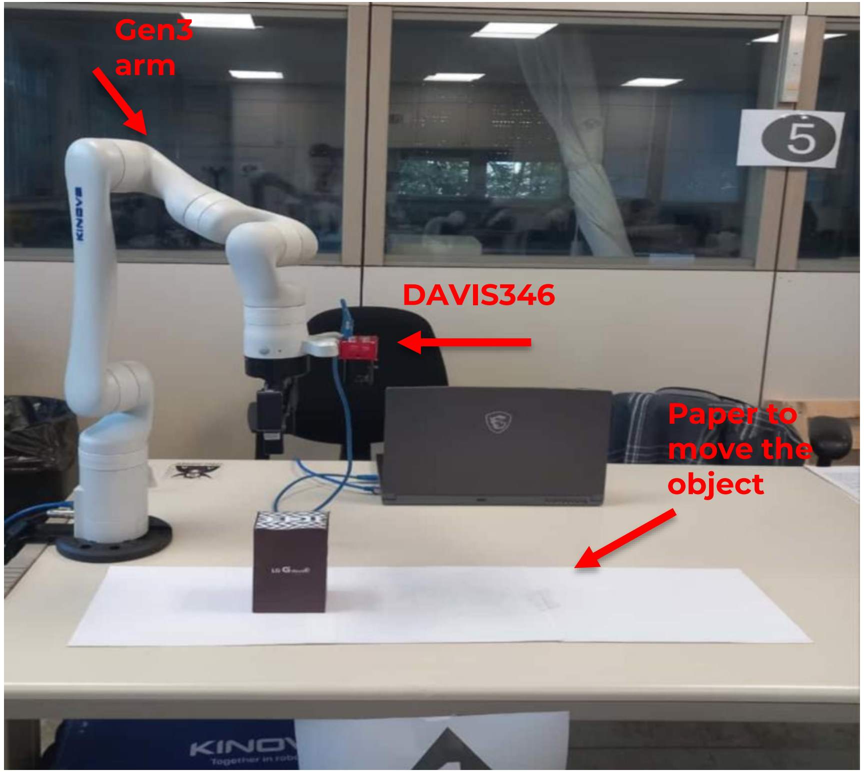
Dynamic Experimental Setup

Simulation Approach:
- Object placed on long paper sheet - Manually pulled across workspace

Tested Trajectories:
✓ Straight line (X direction)
✓ Diagonal (XY motion)

Challenges: Irregular, imprecise speed

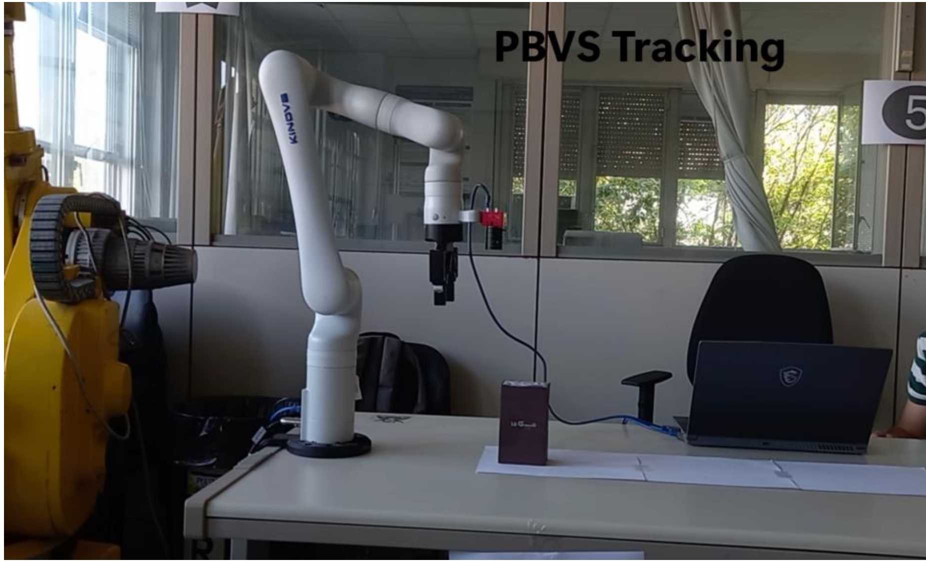
→ **Results:** Proof-of-concept, not a quantitative benchmark



Dynamic Grasping Results (Tracking Only)

Different Directions

Straight Line



System successfully demonstrated:

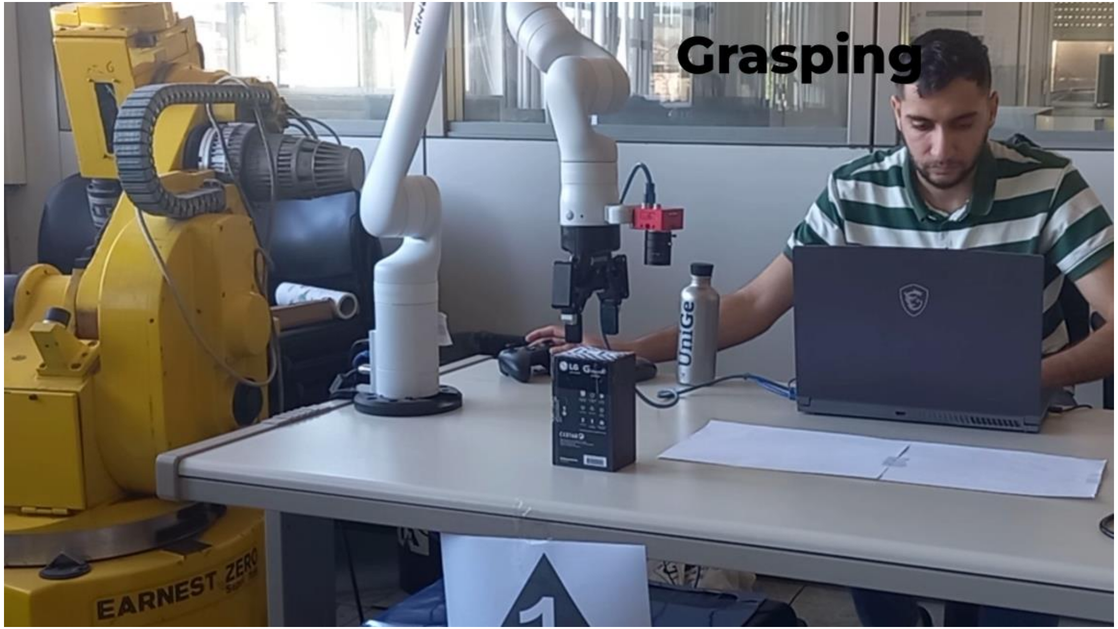
✓ **Real-time PBVS tracking**

Dynamic Grasping: Results

Tracking & Grasp Linear path

System successfully demonstrated:

- ✓ Real-time PBVS tracking
- ✓ velocity estimation
- ✓ Predictive intercept and grasp



Observation:

Gripper closed on moving object, but the contact is often not exactly in the center.

Dynamic Grasping: Results

Tracking & Grasp diagonal path

System successfully demonstrated:

- ✓ Real-time PBVS tracking
- ✓ velocity estimation
- ✓ Predictive intercept and grasp

Observation:

Gripper closed on moving object, but contact often near edge rather than centroid

Dynamic Results Summary: Limitations & Future work

What was Proved:	What was not Proved:
✓ Pipeline provides stable poses for PBVS ✓ Sufficiently velocity estimation ✓ Feed-forward control compensates for latency ✓ System handles real-time tracking and interception	✗ Quantitative performance vs. Speed ✗ Maximum speed capability ✗ Testing different objects shapes

As a future work:

- ➔ Motorized conveyor for quantitative dynamic evaluation.
- ➔ Learned validation for shape generalization.
- ➔ Hybrid clustering pipeline.

Contributions Summary

Reconstruction-Free Pipeline

- 2D clustering + geometric priors
- Simpler, faster than 3D reconstruction
- Suitable for catalogued objects

Dual-Mode System

- Static: Search-then-grasp
- Dynamic: Track-then-predict
- Same perception core

Experimental Validation

- 3.39-8.85 mm accuracy (static)
- Pattern-free capable (5.31mm)
- Proof-of-concept dynamic grasping

Practical pathway for event cameras in industrial manipulation

Thank you for your attention!

References

[1] Huang, X., Halwani, M., Muthusamy, R., Ayyad, A., Swart, D.,Seneviratne, L., Gan, D. & Zweiri, Y. (2022). Real-time grasping strategies using event camera. J. Intell. Manuf., 33, 593–615

[2] Rebecq, H., Gallego, G., Mueggler, E. & Scaramuzza, D. (2018).EMVS: Event-based multi-view stereo—3d reconstruction with an eventcamera in real-time. Int. J. Comput. Vis., 126, 1394–1414