# Event-based 6-DoF Visual Servoing for Dynamic Objects

Yufan Kang[1,2], Arren Glover[3], Guillaume Caron[1,4], Luna Gava[3],
Thomas Duvinage[1], Ryoichi Ishikawa[2], Takeshi Oishi[2]

***Abstract*— Visual servoing of dynamic objects remains challenging for conventional frame-based cameras due to motion blur and limited temporal resolution. Event cameras, with their microsecond-level latency and high dynamic range, are a natural fit for this regime, yet event-based 6-DoF visual servoing has not been demonstrated. Among real-time event-based 6-DoF trackers, registration-based and prediction-based methods trade off differently between per-frame accuracy and update rate: direct registration is accurate over a wide convergence range but costly per step, whereas a discrete hypothesis search runs at a far lower per-step cost, and hence a higher update rate, at the cost of a step-size-limited accuracy. We propose EDFT$^+$, a tracker that evaluates both solvers on a single shared distance-field cost surface, interleaving frequent low-cost hypothesis-search steps, which raise the update rate enough to follow fast motion, with periodic Levenberg–Marquardt registration steps that restore per-step accuracy. Building on these trackers, we present the first event-based 6-DoF visual servoing system for dynamic objects, coupling event-driven object tracking with a pose-based controller on a manipulator. We validate the system in a dual-robot setup, in which a second manipulator moves the target along diverse 6-DoF trajectories at speeds up to 0.4 m/s. EDFT$^+$ sustains stable servoing across objects, establishing the viability of event-based perception for dynamic-object visual servoing.**

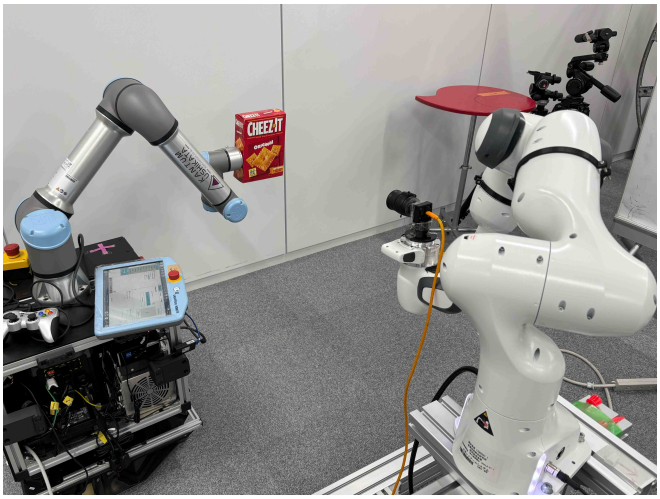



Fig. 1: Dual-robot setup for event-based 6-DoF visual servoing. The UR5e (left) moves the target (Cheez-It box) as a dynamic object; the Franka Panda (right), carrying an event camera, servos to maintain a desired relative pose using event-only 6-DoF pose estimates.

## I. INTRODUCTION

Visual servoing closes the control loop with visual feedback to align a robot with a target object, a foundational capability for robotic manipulation. Servoing on a *moving* target is itself a long-studied problem: classical schemes estimate and predict the target motion, typically by Kalman filtering [1], [2], [3] or predictive control [4], and feed the prediction to the controller to suppress tracking lag. The open difficulty is doing this when the target moves *fast* and unpredictably. Motion models take several cycles to recover from abrupt changes [2], and frame-based cameras produce motion blur and frame-rate-bounded latency precisely in the regime where the target moves fast enough to make servoing nontrivial; precise frame-based servoing on moving objects has accordingly been reported only at low target speeds (e.g., around 0.1 m/s in [5], [6]).

Event cameras, asynchronous sensors that report per-pixel brightness changes with microsecond latency and no motion blur, offer a structurally different modality and have already enabled high-speed perception in flight control, visual odometry, and 6-DoF object tracking [7], [8], [9], [10], [11]. Yet no event-based 6-DoF visual servoing system for dynamic objects has been demonstrated. Bridging this gap is the focus of this paper, closing the loop directly on a low-latency event-based pose estimate rather than on a predicted target motion.

Building such a system reduces to integrating a real-time event-based 6-DoF tracker into a control loop. The two existing real-time trackers sit at complementary points of the speed–accuracy trade-off. EDFT [12] performs per-frame 3D–2D registration by Levenberg–Marquardt optimization on an event-based distance field: it is accurate over a wide convergence range, but each registration step is costly and limits the update rate. EDOPT [10] instead selects, at each update, the best-scoring candidate from a fixed grid of motion hypotheses, giving a low constant per-update cost and a high update rate, but the fixed step caps per-update accuracy. Following a fast target needs both a high update rate, to keep the inter-update displacement small, and

[1]Y. Kang, G. Caron and T. Duvinage are with CNRS-AIST JRL (Joint Robotics Laboratory), IRL, and the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, 305-8568, Japan. kang-yufan@g.ecc.u-tokyo.jp, thomas.duvinage@cnrs.fr

[2]Y. Kang, R. Ishikawa and T. Oishi are with the Institute of Industrial Science, the University of Tokyo, Tokyo 153-8505, Japan. {ishikawa, oishi}@cvl.iis.u-tokyo.ac.jp

[3]A. Glover and L. Gava are with Event-driven Perception for Robotics, Istituto Italiano di Tecnologia, Italy. {arren.glover, luna.gava}@iit.it

[4]G. Caron is also with Université de Picardie Jules Verne, MIS laboratory, 80000 Amiens, France. guillaume.caron@u-picardie.fr

per-step accuracy, to servo precisely. Neither tracker alone meets both requirements.

We propose EDFT$^+$, which evaluates both solvers on a single shared cost surface, interleaving several fast hypothesis-search steps between successive Levenberg–Marquardt registration steps, all operating on the same distance field. Visual servoing also introduces a challenge absent from tracking-only settings. Once the controller drives the relative camera–object motion toward zero, the target stops generating events, so representations built on recent event activity fade at precisely the steady state the controller seeks. We adopt SCARF [13], [14], a velocity-invariant representation that maintains an active set of events at every instant, as the front-end of EDFT$^+$, and show that its velocity invariance is what keeps perception alive through servoing convergence. Together these components form, to our knowledge, the first event-based 6-DoF visual servoing system for dynamic objects. We validate it in a dual-robot setup (Fig. 1), where a second manipulator moves the target along diverse 6-DoF trajectories at speeds up to 0.4 m/s, servoing primarily with EDFT$^+$ and comparing against a prediction-based baseline.

The paper makes the following contributions:

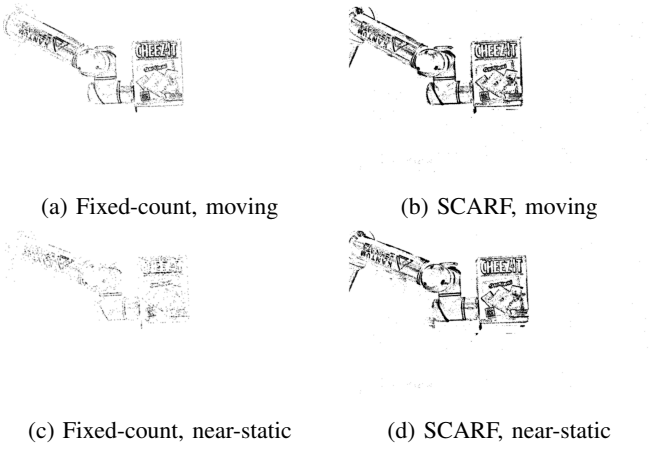
- We propose EDFT$^+$, a cascaded event-based 6-DoF object tracker that fuses registration and prediction-based paradigms, achieving stable, accurate closed-loop tracking that neither paradigm attains on its own.
- We present the first event-based 6-DoF visual servoing system for dynamic objects. Within a pose-based control formulation, we identify steady-state event scarcity as a failure mode specific to the closed-loop regime and resolve it with a velocity-invariant event representation.
- We evaluate the system on a dual-robot benchmark spanning multiple objects and diverse 6-DoF trajectories, demonstrating successful closed-loop servoing at object speeds up to 0.4 m/s.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes EDFT$^+$ and its integration with the SCARF representation. Section IV presents the visual servoing system and its experimental validation. Section V concludes.

## II. Related Work

### A. Event Representations for Tracking and Servoing

Tracking and servoing operate not on raw events but on an intermediate representation that converts the asynchronous event stream into a form a solver can use. Several representations are in common use. Event frames and Time Surfaces [15], [16] accumulate or exponentially weight recent events into a 2D map. EDOPT [10] uses an exponentially-reduced ordinal surface, an image-like map of expected intensity gradients updated asynchronously per event. EDFT [12] converts events into a distance field that endows sparse events with smooth spatial gradients. These representations share an implicit assumption of continuous relative motion. Their support is built from recent event activity and decays once the target stops generating events. SCARF [13], [14] instead maintains an active set of events at every instant and is velocity-invariant, so its output does not decay as relative motion vanishes (Fig. 2). This property is irrelevant for open-loop tracking, where the target is always in motion, but becomes decisive in closed-loop servoing, where the controller drives the relative motion toward zero. We return to it in Section III-A.



(a) Fixed-count, moving (b) SCARF, moving

(c) Fixed-count, near-static (d) SCARF, near-static

Fig. 2: Event-based distance field from fixed-count accumulation (left) versus the velocity-invariant SCARF front-end (right), under relative camera–object motion (top) and as that motion vanishes (bottom). While the target moves, the two representations are comparable. As motion stops, the target generates almost no events, so fixed-count accumulation fills its budget with stale and background events that bury the object edges, whereas SCARF retains the edges. Both columns are shown after the same distance-field transform.

### B. Event-based 6-DoF Object Tracking

Estimating the 6-DoF pose of a rigid target from a known 3D model is a long-standing problem in frame-based vision, addressed by edge-based geometric trackers [17], [18] and, more recently, by learned pose regressors [19], [20]. These methods inherit the temporal limits of frame-based imaging: under fast motion the input blurs and the inter-frame displacement exceeds the convergence basin of the tracker. Event cameras, with microsecond latency and no motion blur, sidestep these limits and have motivated a line of event-based 6-DoF trackers [7], [8], [11]. Several of these retain an auxiliary frame or depth sensor [7], [8], [11]: the events estimate inter-frame motion while a frame anchors absolute pose, so the system inherits the latency floor of the auxiliary sensor.

A separate line of work removes the auxiliary sensor and tracks from events and a 3D model alone. BIAM [9] established this setting with a direct 3D–2D registration between accumulated events and a rendered brightness-increment image, and was the first event-only 6-DoF tracker validated quantitatively on real objects; its

dense per-pixel cost, however, places it roughly two orders of magnitude below real-time. Two subsequent methods reach real time and represent the two dominant paradigms for the task. First, EDOPT [10] is *prediction-based*: it represents incoming events as an exponentially-reduced ordinal surface and, at each update, renders the object model at a small fixed set of pose hypotheses, scoring each by its similarity to the EROS and keeping the best. The hypotheses are placed at the pose increments that produce at most a one-pixel image-plane displacement, computed per axis from the inverse image Jacobian, which yields $2\times6+1$ candidates. This per-update cost is low and constant, so EDOPT runs at a high update rate, but bounding each step to a fixed maximum displacement caps the per-update accuracy and imposes a hard maximum trackable speed. Second, EDFT [12] is *registration-based*: it aligns a sparse, model-derived 3D point set to an event-based distance field by Levenberg–Marquardt optimization, achieving high per-step accuracy over a wide convergence range, but each registration step is computationally heavier and thus sustains a lower update rate.

The two paradigms thus trade off differently between update rate and per-step accuracy. Our tracker brings both into a single framework (Section III), and our servoing system demonstrates that 6-DoF closed-loop control becomes achievable.

### C. Visual Servoing

Visual servoing closes a control loop on visual feedback to drive a robot toward a desired configuration relative to a target [21], [22]. The standard taxonomy distinguishes image-based servoing (IBVS), which regulates an error defined in the image plane, from pose-based servoing (PBVS), which estimates the 6-DoF target pose and regulates the error in Cartesian space. The system in this paper is PBVS, driven by the 6-DoF pose output of the tracker, which allows straighter robot trajectories than IBVS [22].

Servoing on a moving target is itself well studied. Classical schemes estimate and predict the target motion, typically by Kalman filtering [1], [2], [3] or predictive control [4], and feed the prediction to the controller to suppress tracking lag. The open difficulty is doing this when the target moves fast and unpredictably: motion models take several cycles to recover from abrupt changes [2], and for frame-based cameras the speed regime that makes servoing nontrivial is exactly where motion blur and frame-rate-bounded latency appear. High-speed approaches [23] accept large positional error to keep a fast target within the field of view, while precise approaches [5], [6] operate only at low target speed, where motion blur and inter-frame displacement stay manageable. This trade-off follows directly from the temporal characteristics of frame-based imaging and motivates the use of event cameras for servoing on fast-moving targets.

The use of event cameras for visual servoing is recent and remains restricted in scope. Existing systems align an object centroid with the image center for grasping [24], regulate planar or one-dimensional motion for ground robots [25], or apply image-based servoing to a parametric Time Surface model [26]. None addresses a target moving freely in all six degrees of freedom, which is the gap this paper closes.

## III. Method

We build a closed-loop event-based 6-DoF visual servoing system from two parts. The first is a real-time tracker, EDFT$^+$, that estimates the relative pose of the target from events alone. The second is a pose-based controller that drives a camera-carrying manipulator to maintain a desired relative pose. Figure 3 gives an overview of the processing pipeline. The tracker takes two inputs that originate from separate branches. One is an event-based distance field generated from the live event stream. The other is a sparse 3D point set rendered once from the object model. The tracker registers the point set to the distance field with two interleaved solvers. This section describes the shared representation and point set (Section III-A), the cascaded solver that defines EDFT$^+$ (Section III-B), and the integration with the controller (Section III-C). We treat the event-based distance field and its Levenberg–Marquardt registration as established components [12] and focus on what is new in this work: the solver cascade, the single-render point set, and the velocity-invariant front-end that keeps perception alive as servoing converges.

### A. Shared Representation and Point Set

EDFT$^+$ inherits the representation and registration target of EDFT [12]. Incoming events are summarized into an event-based distance field $I_{\mathrm{EDF}}$, a 2D field that endows sparse events with smooth, two-sided spatial gradients around object edges. This field serves as the cost surface for registration. In parallel, a sparse 3D point set $S^* = \{\mathbf{S}_i\}$ is obtained by rendering the object model at a known initial pose, extracting edge pixels, and back-projecting them with the rendered depth. These points mark the locations where events are expected to appear under motion. Registration aligns $S^*$ to $I_{\mathrm{EDF}}$ by estimating a pose increment $\Delta\mathbf{p}$ that minimizes the field sampled at the warped, reprojected points,

$$\Delta\mathbf{p} = \arg\min_{\Delta\mathbf{p}} \sum_{\mathbf{S}_i \in S^*} I_{\mathrm{EDF}}(W(\mathbf{S}_i;\, \Delta\mathbf{p}))\,, \tag{1}$$

where $W(\cdot)$ warps and projects a 3D point onto the image plane. The form of $I_{\mathrm{EDF}}$, the warp $W$, and the analytical Jacobian follow [12].

*a) Single-render point set:* EDFT regenerates $S^*$ whenever the accumulated pose change exceeds a threshold, rendering a new keyframe so that the point set stays aligned with the current view. In the servoing regime this regeneration is unnecessary. The controller actively drives the relative camera–object motion toward zero, so the viewpoint from which $S^*$ was rendered remains representative throughout a servoing episode. We therefore render $S^*$ once, at the initial pose, and reuse it for the entire loop. This removes the rendering cost from the per-update

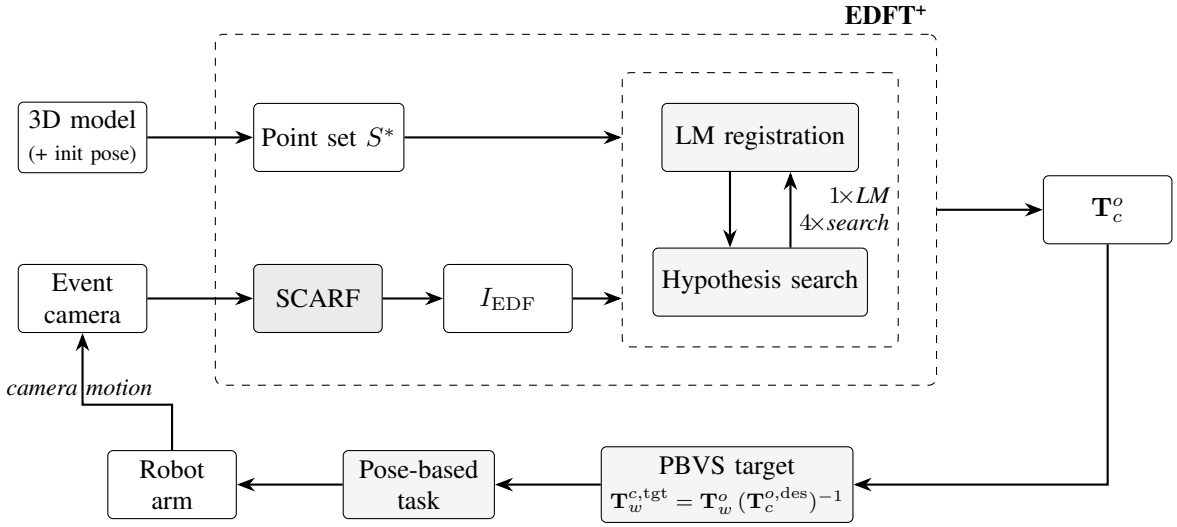



Fig. 3: Overview of the proposed system. The outer dashed box is the novel EDFT+ tracker; the inner dashed box contains its two interleaved solvers. Events are converted to a velocity-invariant SCARF representation, from which the event-based distance field $I_{\mathrm{EDF}}$ is generated. A sparse 3D point set $S^*$ is rendered once from the 3D model and a known initial pose, and reused throughout. The two solvers, one Levenberg–Marquardt step alternating with $N = 4$ discrete hypothesis-search steps, register $S^*$ against $I_{\mathrm{EDF}}$ to produce the relative pose $\mathbf{T}_c^o$. A pose-based controller turns $\mathbf{T}_c^o$ into a Cartesian target for the camera-carrying robot arm, closing the loop as the induced camera motion changes the observed events.

budget and simplifies the cascade described below, without degrading closed-loop tracking in our experiments.

*b) Velocity-invariant front-end:* A second consequence of the servoing regime concerns the representation itself. The event rate is proportional to the relative camera–object motion, so as the controller converges the target generates progressively fewer events. Any representation built on recent event activity then fades at precisely the steady state the controller seeks, and $I_{\mathrm{EDF}}$ as generated in EDFT is no exception. We therefore replace the event accumulation step that feeds $I_{\mathrm{EDF}}$ with the velocity-invariant SCARF representation [13], [14], which maintains an active set of events at every instant and does not decay as the target slows. The distance field is then generated from this active set rather than from a fixed accumulation window. SCARF is the only event-side input to the system and is shared by both solvers. Its velocity invariance is what allows servoing to be sustained as the target slows toward the desired pose.

### B. Cascaded Solver: EDFT+

A single solver is insufficient for servoing on a fast-moving target. On the one hand, EDFT's Levenberg–Marquardt (LM) registration on $I_{\mathrm{EDF}}$ is accurate and converges over a wide range, but each step performs an iterative optimization with Jacobian evaluations and is therefore too expensive to run at the update rate a fast target demands. EDOPT's discrete hypothesis search has the opposite profile: a single step only evaluates a small, fixed set of candidate increments and keeps the best one, at a fraction of the cost of an LM step. So it can run at a much higher rate than EDFT and keep the inter-update displacement small under fast motion; its accuracy, however, is bounded by the spacing of the candidates.

EDFT+ combines the two on the same cost surface of Eq. (1) and the same point set $S^*$, so that the two solvers differ only in how they search for $\Delta\mathbf{p}$, not in what they minimize.

*a) Pyramidal registration:* To widen the convergence range of the registration step, the LM optimization of Eq. (1) is run coarse-to-fine over an image pyramid of $I_{\mathrm{EDF}}$. At level $\ell \in \{2, 1, 0\}$ (quarter, half, and full resolution) the estimate is

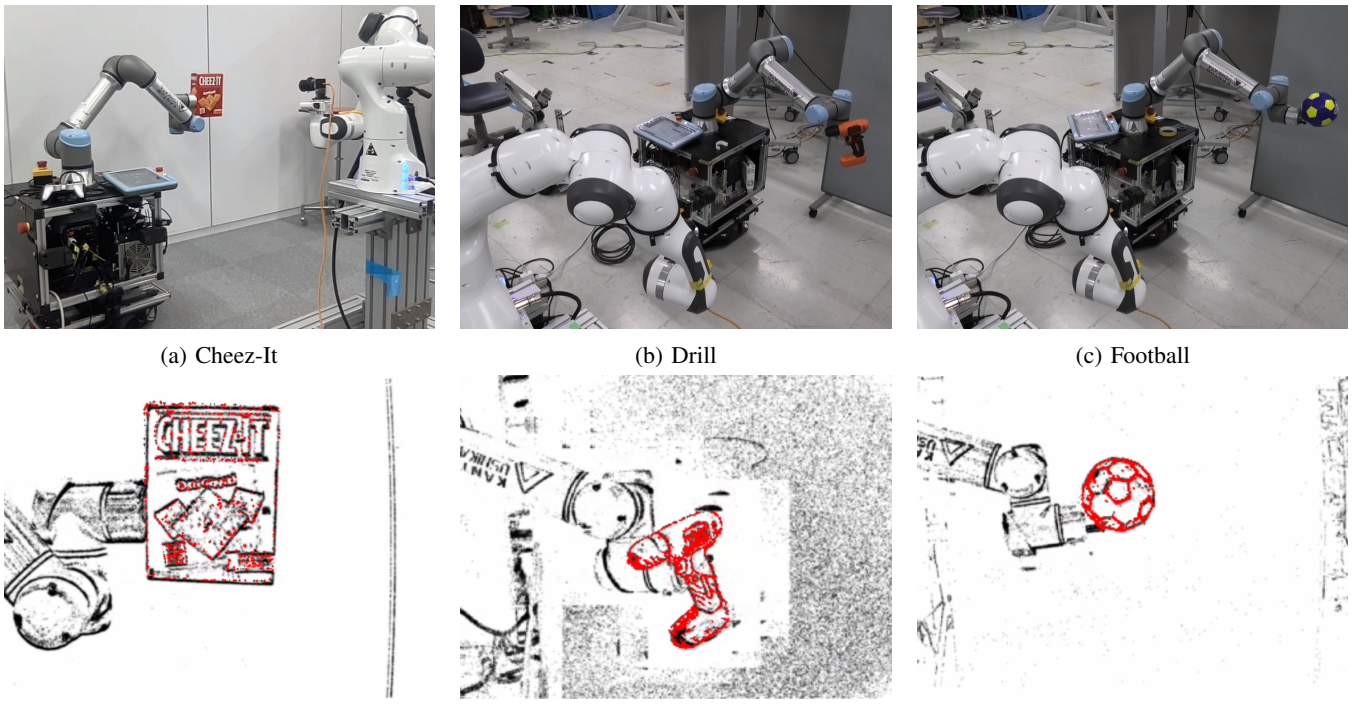
$$\hat{\mathbf{p}}^{(\ell)} = \arg\min_{\Delta\mathbf{p}} \sum_{\mathbf{S}_i \in S^*} I_{\mathrm{EDF}}^{(\ell)}\big(W(\mathbf{S}_i;\, \Delta\mathbf{p})\big), \tag{2}$$

each level initialized from the coarser solution $\hat{\mathbf{p}}^{(\ell+1)}$, and the registration output is the full-resolution result $\hat{\mathbf{p}}_{\mathrm{LM}} = \hat{\mathbf{p}}^{(0)}$. The coarse levels pull large displacements close before the fine levels refine the alignment, extending the range over which registration converges, at the cost of one optimization per level.

*b) Hypothesis-search step:* Following the prediction-based paradigm [10], the hypothesis-search step refines the current estimate $\hat{\mathbf{p}}$ by testing a fixed set of candidate increments centered on it. For each of the six axes we form a pair of increments $\pm\delta_k\mathbf{e}_k$, where $\mathbf{e}_k$ is the unit twist of the $k$-th DoF and $\delta_k$ is the magnitude that produces a one-pixel image-plane displacement, computed per axis from the image Jacobian. Together with the null increment this gives the candidate set

$$\mathcal{C} = \{\mathbf{0}\} \cup \{\pm\delta_k\mathbf{e}_k\}_{k=1}^{6}, \qquad |\mathcal{C}| = 2 \times 6 + 1. \tag{3}$$

Each candidate warps the shared point set $S^*$ by $\hat{\mathbf{p}} \oplus \Delta\mathbf{c}$, reprojects it, and accumulates $I_{\mathrm{EDF}}$ at the reprojected



(a) Cheez-It (b) Drill (c) Football

(d) Cheez-It registration (e) Drill registration (f) Football registration

Fig. 4: Qualitative servoing results. Top: the three objects during closed-loop servoing in the dual-robot setup. Bottom: the corresponding EDFT$^{+}$ registration, with the reprojected model point set (red) overlaid on the event-based distance field. The point set stays aligned with the object edges throughout the servoing runs.

locations as in Eq. (1). The search output is the lowest-cost candidate,

$$\hat{\mathbf{p}}_{\mathrm{HS}} = \hat{\mathbf{p}} \oplus \arg\min_{\Delta\mathbf{c}\in\mathcal{C}} \sum_{\mathbf{S}_i\in S^*} I_{\mathrm{EDF}}\big(W(\mathbf{S}_i;\, \hat{\mathbf{p}} \oplus \Delta\mathbf{c})\big), \quad (4)$$

where $\oplus$ denotes composition on $SE(3)$. Because $S^*$ is rendered only once, every candidate reuses the same point set and the step reduces to sampling $I_{\mathrm{EDF}}$ at warped pixel locations, which is inexpensive.

*c) Cascade schedule:* The two solvers are interleaved at a fixed ratio: each LM step $\hat{\mathbf{p}} \leftarrow \hat{\mathbf{p}}_{\mathrm{LM}}$ is followed by $N$ search steps $\hat{\mathbf{p}} \leftarrow \hat{\mathbf{p}}_{\mathrm{HS}}$ before the next LM step, both updating the same shared estimate $\hat{\mathbf{p}}$. The $N$ search steps advance $\hat{\mathbf{p}}$ in fast, low-cost increments that raise the effective update rate, so the inter-update displacement stays small even under fast motion. The periodic LM step then refines $\hat{\mathbf{p}}$ to sub-candidate accuracy, removing the quantization error left by the discrete grid $\mathcal{C}$. The cascade output is the relative object pose $\mathbf{T}_c^o$ recovered from $\hat{\mathbf{p}}$.

### C. Visual Servoing Integration

The controller is a standard pose-based visual servoing (PBVS) formulation. The contribution of this section is the integration of an asynchronous event-based pose stream into a real-robot control loop, not a new control law. We write $\mathbf{T}_a^b \in SE(3)$ for the pose of frame $b$ expressed in frame $a$, with frames $w$ (world), $c$ (camera optical frame), and $o$ (object); the tracker output $\mathbf{T}_c^o$ is thus the object pose in the camera frame. The camera is mounted on the end effector of the robot arm, and the hand–eye transform is obtained by offline calibration. At the start of a servoing episode the current relative pose is captured as the desired pose $\mathbf{T}_c^{o,\mathrm{des}}$, the object-in-camera pose the controller maintains.

At each control cycle the tracker provides the current relative pose $\mathbf{T}_c^o$. With the current camera pose in the world frame $\mathbf{T}_w^c$ read from forward kinematics, the object is located in the world frame as $\mathbf{T}_w^o = \mathbf{T}_w^c\,\mathbf{T}_c^o$. The camera pose that places the object at the desired relative pose is

$$\mathbf{T}_w^{c,\mathrm{tgt}} = \mathbf{T}_w^o \left(\mathbf{T}_c^{o,\mathrm{des}}\right)^{-1}, \quad (5)$$

issued as the Cartesian target of a pose-based task. Let $\mathbf{e} \in \mathbb{R}^6$ be the pose error between the camera frame and $\mathbf{T}_w^{c,\mathrm{tgt}}$, stacking the translation and the angle–axis rotation of $(\mathbf{T}_w^c)^{-1}\mathbf{T}_w^{c,\mathrm{tgt}}$. The task drives this error with the second-order law

$$\ddot{\mathbf{e}}^* = -k\,\mathbf{e} - 2\sqrt{k}\,\dot{\mathbf{e}}, \quad (6)$$

where the stiffness $k$ is paired with the damping $2\sqrt{k}$ for a critically damped response. The asynchronous tracker output and the fixed-rate controller run on separate threads communicating over a lightweight interface, and the translation of $\mathbf{T}_c^o$ is smoothed by an exponential moving average before forming the target to suppress jitter. As the controller moves the camera, the induced motion changes the observed events and closes the loop of Figure 3.

## IV. Experiments

We evaluate the system on a dual-robot benchmark, in which one manipulator produces repeatable dynamic motion of the target while the other servos to it. We refer to EDOPT running on the same SCARF front-end as *EDOPTS*, so that any comparison isolates the solver paradigm rather than the event representation. The experiments address two questions: (i) how closed-loop 6-DoF servoing behaves with the prediction-based paradigm (EDOPTS) used alone, and (ii) whether EDFT$^+$ sustains servoing across objects of differing geometry. A registration-only baseline is not viable on its own: although EDFT reaches real-time rates in open-loop tracking [12], in the closed loop its per-step registration cost lowers the achievable control rate, and even with our best implementation the servoing is not stable enough to keep the robot locked onto a moving target, so tracking fails shortly after the target accelerates. We therefore report and analyse EDOPTS and EDFT$^+$.

### A. Experimental Setup

The dynamic target is carried by the end effector of a UR5e manipulator, which moves it along prescribed trajectories. The event camera, a Prophesee IMX636, is mounted on the end effector of a Franka Emika Panda manipulator, which servos to maintain a desired relative pose to the target (Fig. 1). We use three objects of differing geometry and texture: a Cheez-It box, a power drill, and a football. The Cheez-It model was built in CAD; the drill and football models were obtained with a 3D reconstruction scanner.

The camera intrinsics and radial distortion were obtained by blinking checkerboard calibration [27] as $f_x = 1163.59$, $f_y = 1173.48$, $c_x = 658.823$, $c_y = 339.619$ pixels and $k_1 = -0.2819$. The hand–eye transform between the camera optical frame and the Panda flange was obtained by offline calibration [28]. The controller is implemented in the mc_rtc framework [29], which issues the Cartesian target of Eq. (5) as a pose-based task and resolves it to joint commands through a QP that respects the robot's constraints. The task stiffness is $k = 20$ with critical damping $2\sqrt{k}$ (Section III-C). The two trackers run on an Intel Core i7-1280P CPU and NVIDIA GeForce RTX 3070 GPU at 107 Hz (EDOPTS) and 68 Hz (EDFT$^+$). The lower rate of EDFT$^+$ reflects the periodic pyramidal LM step, which is more costly than a hypothesis-search step.

### B. Comparison of the Two Paradigms on Cheez-It

We first compare EDFT$^+$ against the prediction-based baseline EDOPTS on the Cheez-It box, the object for which a model in the format required by EDOPTS is available. Both run on the identical SCARF front-end and are evaluated with the same error definition, so the comparison isolates the solver paradigm.

Because the two robot bases are not jointly calibrated, an independent ground-truth relative pose is unavailable. We instead report the closed-loop servoing error, the deviation of the estimated relative pose $\mathbf{T}_c^o$ from the desired relative pose $\mathbf{T}_c^{o,\mathrm{des}}$ captured at the start of each run. Translation error is the Euclidean distance between the two relative positions and rotation error is the geodesic angle between the two relative rotations. This error lives in the tracker's sensor frame and measures how well the desired relative pose is maintained throughout the motion. It is the quantity the loop regulates rather than an absolute accuracy against external ground truth. For each run we report the root-mean-square (RMS) error over the servoing interval.

We run at two camera–object depths $Z$. At the larger $Z = 0.9$ m the target moves under translation at 0.2 to 0.4 m/s; at the closer $Z = 0.6$ m, where the same object speed induces larger image-plane motion, we run both pure translation and combined translational–rotational motion. Table I reports, for

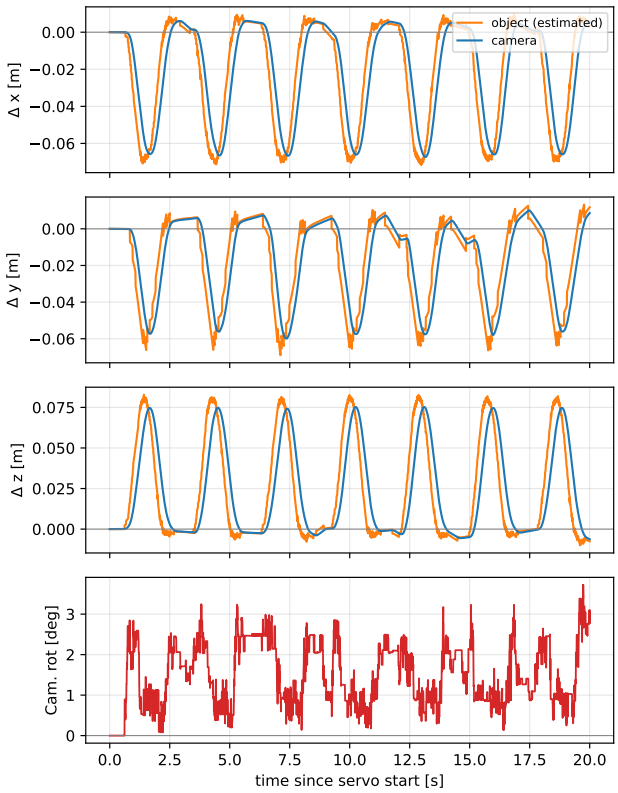



Fig. 5: Camera and target motion during a translational servoing run (Cheez-It, $Z = 0.6$ m), over the first 20 s after servo start. The top three rows show the displacement of the camera (blue) and of the object (orange) along each world axis, relative to the pose at the start of the run; the object trajectory is reconstructed from the camera pose and the estimated relative pose $\mathbf{T}_c^o$ and is therefore not an independent ground-truth reference. The bottom row shows the rotation of the camera in the world frame relative to its starting orientation, a different quantity from the relative-pose error of Table I. The translation curves move together at a near-constant offset while the camera rotation stays near zero, showing that the manipulator compensates the translating target with an almost purely translational motion, at a short servoing lag.

TABLE I: Servoing error on Cheez-It (RMS over the servoing interval), EDFT$^+$ versus the prediction-based baseline EDOPTS on the same SCARF front-end. Translation pos. in mm, rotation rot. in degrees. Best result in each trajectory is in **bold**. EDFT$^+$: mean $\pm$ std over three runs; EDOPTS: single run.

| | EDFT$^+$ | | EDOPTS | |
|---|---|---|---|---|
| Trajectory | pos. | rot. | pos. | rot. |
| $Z$=0.9 m, trans. | $\mathbf{20.57 \pm 3.36}$ | $\mathbf{1.85 \pm 0.57}$ | 21.41 | 3.63 |
| $Z$=0.6 m, trans. | $\mathbf{22.01 \pm 4.93}$ | $\mathbf{5.68 \pm 1.65}$ | 26.27 | 8.38 |
| $Z$=0.6 m, trans.+rot. | $\mathbf{30.78 \pm 2.74}$ | $\mathbf{5.28 \pm 1.00}$ | 58.20 | 5.42 |

each trajectory, the RMS of the relative-pose error defined above, in translation and rotation. EDFT$^+$ achieves the lower error in every case. At $Z = 0.9\,\mathrm{m}$ both methods sustain servoing, but EDFT$^+$ is more accurate: at this speed its update rate already suffices to follow the target, so per-step accuracy rather than update rate governs servoing quality, and the periodic LM step supplies the more accurate per-step estimate. At $Z = 0.6\,\mathrm{m}$ EDFT$^+$ retains its advantage, most markedly under combined motion, where EDOPTS degrades sharply. These results lead us to discard EDOPTS and to carry EDFT$^+$ forward as the tracker for the remaining objects.

Figure 5 plots the camera and reconstructed object trajectories over the first $20\,\mathrm{s}$ of the $Z = 0.6\,\mathrm{m}$ run, in which the UR5e moves the Cheez-It box in pure translation. Maintaining the desired relative pose under a purely translating target requires the camera to translate with it while keeping its orientation fixed, so the two translation trajectories should coincide up to a constant offset only if the camera itself barely rotates. The bottom row of Figure 5 therefore reports a different quantity from the servoing error of Table I: not the relative-pose error $\mathbf{T}_c^o$ versus $\mathbf{T}_c^{o,\mathrm{des}}$, but the rotation of the camera in the world frame relative to its pose at servo start. Its mean over the run is $1.89°$, confirming that the manipulator holds its orientation nearly fixed and compensates the translating target with an almost purely translational motion. The three translation axes then indeed move together at a near-constant offset. All six DoF remain under control throughout, the orientation held in place by small continuous corrections rather than left free. From the recorded video[1] the camera lags the target by roughly $0.15\,\mathrm{s}$.

### C. Servoing Across Objects with EDFT$^+$

Having discarded EDOPTS, we evaluate EDFT$^+$ across the three objects at $Z = 0.9\,\mathrm{m}$ under translational motion at $0.2$ to $0.4\,\mathrm{m/s}$. The drill and the football present less distinct edges than the printed Cheez-It box and are therefore the more demanding targets. Table II reports the RMS relative-pose error. EDFT$^+$ sustains servoing on all three from start to finish, holding the relative pose close to the desired one through large Cartesian excursions of the manipulator. The error is comparable across the three geometries, with a modestly higher rotation error on the football, indicating that servoing does not depend on a particular object shape or texture even when the edges are less pronounced. The qualitative results in Figure 4 show the reprojected model staying aligned with the event representation across the servoing runs.

[1] https://bluecatli.github.io/EVS_page/

TABLE II: EDFT$^+$ servoing error across objects, $Z = 0.9\,\mathrm{m}$ translational motion (RMS, mean $\pm$ std over three runs). Translation pos. in mm, rotation rot. in degrees.

| Object | pos. | rot. |
|---|---|---|
| Cheez-It | $20.57 \pm 3.36$ | $1.85 \pm 0.57$ |
| Drill | $18.90 \pm 2.30$ | $1.88 \pm 0.65$ |
| Football | $21.64 \pm 2.61$ | $3.53 \pm 0.71$ |

## V. Conclusion

We presented the first event-based 6-DoF visual servoing system for dynamic objects. At its core is EDFT$^+$, a new tracker that interleaves frequent low-cost hypothesis-search steps with periodic pyramidal Levenberg–Marquardt registration on a shared distance-field cost surface, combining a high update rate with per-step accuracy. Coupled with a pose-based controller on a manipulator and a velocity-invariant SCARF front-end that keeps the perception signal alive as servoing converges, the system sustains closed-loop servoing across objects of differing geometry and two working distances in a dual-robot benchmark, and we additionally report a comparison against a prediction-based baseline.

The main limitation is the speed of the closed loop. The target speeds reached in our experiments remain modest, and the camera follows the target with a lag on the order of $0.15\,\mathrm{s}$. This lag is not set by perception. The sensing latency, from target motion to data arriving in CPU memory and covering only the sensor and the communication pipeline, we measure at roughly $3.5\,\mathrm{ms}$ for the event camera against about $35\,\mathrm{ms}$ for a frame-based camera; at $0.4\,\mathrm{m/s}$ this is the difference between $1.4\,\mathrm{mm}$ and $14\,\mathrm{mm}$ of unobserved target displacement before a pose is even computed. This sensing latency is an order of magnitude below the observed closed-loop lag. The bottleneck therefore lies in the controller and the sensor-to-actuation pipeline rather than in perception. Future work will address this side directly through predictive control that anticipates target motion across the loop, and will extend the quantitative evaluation with an independent ground-truth reference for the relative pose.

## References


[1] F. Chaumette and A. Santos, "Tracking a moving object by visual servoing," in *12th IFAC World Congress*, 1993, pp. 409–414.

[2] F. Bensalah and F. Chaumette, "Compensation of abrupt motion changes in target tracking by visual servoing," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1995, pp. 181–187.

[3] A. Crétual and F. Chaumette, "Application of motion-based visual servoing to target tracking," *Int. J. of Robotics Research*, vol. 20, no. 11, pp. 878–890, 2001.

[4] G. Allibert, E. Courtial, and F. Chaumette, "Predictive control for constrained image-based visual servoing," *IEEE Trans on Robotics*, vol. 26, no. 5, pp. 933–939, 2010.

[5] J. González Huarte and A. Ibarguren, "Visual servoing architecture of mobile manipulators for precise industrial operations on moving objects," *Robotics*, vol. 13, no. 5, p. 71, 2024.

[6] Q. Zhang et al., "Fusing phase map servoing and MPC for high-precision robotic tracking of dynamic objects," *Actuators*, vol. 15, no. 2, 2026.

[7] E. Dubeau, M. Garon, B. Debaque, R. d. Charette, and J.-F. Lalonde, "RGB-D-E: Event camera calibration for fast 6-DoF object tracking," in *Proc. of IEEE Int. Symposium on Mixed and Augmented Reality*, 2020, pp. 127–135.

[8] H. Li and J. Stueckler, "Tracking 6-DoF object motion from events and frames," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2021, pp. 14 171–14 177.

[9] Y. Kang et al., "Direct 3D model-based object tracking with event camera by motion interpolation," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2024, pp. 2645–2651.

[10] A. Glover, L. Gava, Z. Li, and C. Bartolozzi, "EDOPT: Event-camera 6-DoF dynamic object pose tracking," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2024, pp. 18 200–18 206.

[11] J.-Y. Kang et al., "Event6D: Event-based novel object 6D pose tracking," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2026.

[12] Y. Kang, R. Ishikawa, G. Caron, and T. Oishi, "Event-based 6-DoF object tracking with distance field reaching 130 Hz," *IEEE Robotics and Automation Letters*, 2026.

[13] L. Gava, M. Ikura, C. Bartolozzi, and A. Glover, "SCARF: A set of centre active receptive fields for velocity invariant event representation," in *IROS Workshop on Neuromorphic Perception for Real World Robotics (NeuRobots)*, 2025.

[14] M. Ikura, A. Glover, M. Mizuno, and C. Bartolozzi, "Lattice-allocated real-time line segment feature detection and tracking using only an event-based camera," in *Proc. of Int. Conf. on Computer Vision*, 2025, pp. 4645–4654.

[15] Y. Zhou, G. Gallego, and S. Shen, "Event-based stereo visual odometry," *IEEE Trans on Robotics*, vol. 37, no. 5, pp. 1433–1450, 2021.

[16] J. Niu, S. Zhong, X. Lu, S. Shen, G. Gallego, and Y. Zhou, "ESVO2: Direct visual-inertial odometry with stereo event cameras," *IEEE Trans on Robotics*, 2025.

[17] T. Drummond and R. Cipolla, "Real-time visual tracking of complex structures," *IEEE Trans on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 932–946, 2002.

[18] A. I. Comport, E. Marchand, M. Pressigout, and F. Chaumette, "Real-time markerless tracking for augmented reality: The virtual visual servoing framework," *IEEE Trans on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 615–628, 2006.

[19] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Proc. of Robotics: Science and Systems*, 2018.

[20] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "FoundationPose: Unified 6D pose estimation and tracking of novel objects," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2024, pp. 17 868–17 879.

[21] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," in *Workshop on Geometric Reasoning for Perception and Action*, 1991, pp. 106–136.

[22] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[23] K. Yang, C. Bai, Z. She, and Q. Quan, "High-speed interception multicopter control by image-based visual servoing," *IEEE Trans on Control Systems Technology*, vol. 33, no. 1, pp. 119–135, 2024.

[24] R. Muthusamy et al., "Neuromorphic eye-in-hand visual servoing," *IEEE Access*, vol. 9, pp. 55 853–55 870, 2021.

[25] M. Mordad, K. Behzad, D. Biswas, N. J. Cowan, and M. Siami, "Bio-inspired event-based visual servoing for ground robots," *arXiv preprint arXiv:2603.23672*, 2026.

[26] G. Gong, Q. Wang, D. Navarro-Alarcon, and Z. He, "Event-based photometric Gaussian mixture models for visual servoing," in *IECON – Annual Conference of the IEEE Industrial Electronics Society*, 2025, pp. 1–7.

[27] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[28] R. Y. Tsai, R. K. Lenz, et al., "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Trans on Robotics and Automation*, vol. 5, no. 3, pp. 345–358, 1989.

[29] J. Vaillant et al., *mc_rtc: Real-time robot control framework*, `https://jrl.cnrs.fr/mc_rtc`, Accessed: 2026-06-09.