IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 8, 2021, accepted March 31, 2021, date of publication April 5, 2021, date of current version April 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3071261

Neuromorphic Eye-in-Hand Visual Servoing

RAJKUMAR MUTHUSAMY[1], ABDULLA AYYAD[1,2], (Member, IEEE), MOHAMAD HALWANI[1], DEWALD SWART[3], DONGMING GAN[4], LAKMAL SENEVIRATNE[1], AND YAHYA ZWEIRI[1,5]

[1]Khalifa University Center for Autonomous Robotic Systems (KUCARS), Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates
[2]Aerospace Research and Innovation Center (ARIC), Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates
[3]Research and Development, Strata Manufacturing PJSC, Al Ain, United Arab Emirates
[4]School of Engineering Technology, Purdue University, West Lafayette, IN 47907, USA
[5]Faculty of Science, Engineering and Computing, Kingston University, London SW15 3DW, U.K.

Corresponding authors: Rajkumar Muthusamy (rajkumar.muthusamy@ku.ac.ae) and Yahya Zweiri (yahya.zweiri@ku.ac.ae)

This work was supported in part by the Khalifa University of Science and Technology under Award CIRA-2018-55 and Award RC1-2018-KUCARS, and in part by the Aerospace Research and Innovation Center (ARIC), through STRATA Manufacturing PJSC (a Mubadala company) and Khalifa University of Science and Technology.

ABSTRACT Robotic vision plays a major role in factory automation to service robot applications. However, the traditional use of frame-based cameras sets a limitation on continuous visual feedback due to their low sampling rate, poor performance in low light conditions and redundant data in real-time image processing, especially in the case of high-speed tasks. Neuromorphic event-based vision is a recent technology that gives human-like vision capabilities such as observing the dynamic changes asynchronously at a high temporal resolution ($1\mu s$) with low latency and wide dynamic range. In this paper, for the first time, we present a purely event-based visual servoing method using a neuromorphic camera in an eye-in-hand configuration for the grasping pipeline of a robotic manipulator. We devise three surface layers of active events to directly process the incoming stream of events from relative motion. A purely event-based approach is used to detect corner features, localize them robustly using heatmaps and generate virtual features for tracking and grasp alignment. Based on the visual feedback, the motion of the robot is controlled to make the temporal upcoming event features converge to the desired event in Spatio-temporal space. The controller switches its operation such that it explores the workspace, reaches the target object and achieves a stable grasp. The event-based visual servoing (EBVS) method is comprehensively studied and validated experimentally using a commercial robot manipulator in an eye-in-hand configuration for both static and dynamic targets. Experimental results show superior performance of the EBVS method over frame-based vision, especially in high-speed operations and poor lighting conditions. As such, EBVS overcomes the issues of motion blur, lighting and exposure timing that exist in conventional frame-based visual servoing methods.

INDEX TERMS Neuromorphic vision sensor, event camera, event-based visual servoing, robotic vision, robotic manipulator, neuromorphic vision-based robot control, vacuum gripper, pick and place task.

I. INTRODUCTION

In robotics, visual servoing is a well-studied research topic [1]–[3] and a well-known real-time technique to control the motion of a robot using continuous visual feedback. Such vision-based closed-loop control enhances accuracy, safety, flexibility, reliability, functionality and efficiency in robotic automation while also reducing the need for complex fixtures. Visual servoing has been adopted in a wide range of robotic applications such as pick and place [4], sorting [5], inspection [6], monitoring [7], parts assembly and disassembly [8], harvesting [9], assistive surgery [10] etc. Moreover, visual servoing have been deployed in robotic manipulators [11], unmanned ground vehicles (UGV) [12], unmanned ariel vehicles (UAV) [13], [14], unmanned underwater vehicles (UUV) [15], space robots [16], human-robot interaction (HRI) [17] and multi-robot systems (MRS) [18]. Camera technologies plays a crucial role in visual servoing and the recent neuromorphic vision sensor (also known as event camera or dynamic vision sensor (DVS)) [19] has the potential to bring dramatic change in system performance, efficiency and capability.

In conventional visual servoing, frame-based cameras are mainly used to detect, track and match visual features by processing intensity images at consecutive frames; which in practice, causes delays in visual processing and the

The associate editor coordinating the review of this manuscript and approving it for publication was Saeid Nahavandi.

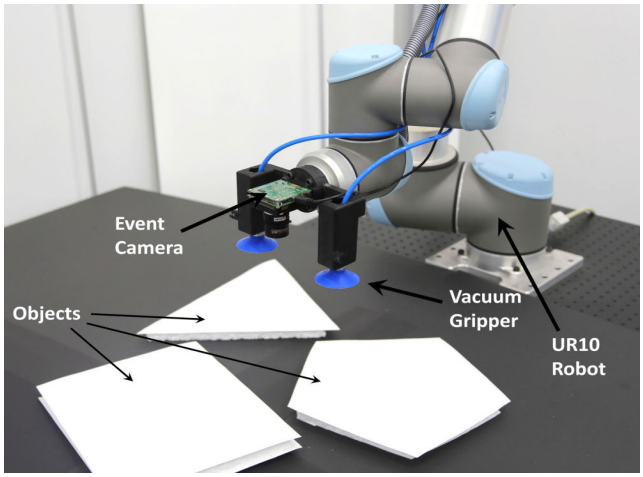


FIGURE 1. Event-based visual servoing for robotic object manipulation task.

consequent robot action. Moreover, they face issues of motion blur and often require increased ambient illumination [20], [21]; which undermines their capabilities in high speed operations and varying light conditions. These shortcomings of frame-based cameras can limit their usage/applicability for visual servoing in both structured and unstructured environments. For instance, in structured environments such as robotic automation in factory settings, employing conventional visual servoing methods can restrict production speeds and jeopardize time-critical operations. Alternatively, autonomous robots operating in hazardous, dangerous and extreme conditions may face poor lighting conditions due to the unstructured environment. Employing conventional frame-based visual servoing in such scenarios may result in unsafe and unreliable operation. As such, robust visual servoing requires that visual information is accurate, reliable and computationally efficient for real-time operation. Recently, neuromorphic vision sensors that mimic the neuro-biological architecture of a human retina overcome the conventional challenges of frame-based vision and can yield faster and more robust visual servoing to meet the evolving industrial requirements.

Unlike conventional vision sensors which are frame-based and clock driven, neuromorphic vision sensors [22], [23] are event driven and provide low latency, high temporal resolution and wide dynamic range. Moreover, the independent sensor pixels operate asynchronously and respond to varying illumination in continuous time. We exploit these inherent properties of the sensor to achieve fast, reliable and efficient visual servoing to facilitate robotic object manipulation in a pick and place task. An event camera attached to the robot's end-effector to perform visual servoing to stage pick and place tasks is depicted in Fig. 1.

A. RELATED WORK

In modern industries ranging from e-commerce warehouses [24] and production/assembly lines [5] to domestic assistive robots for daily living [25], there has been an increased use of vision based robot control to perform manipulation tasks on static and moving objects. In the literature, several visual servoing pipelines have been devised to address these problems in both indoors and outdoors settings using a variety of control strategies and manipulator designs [24]–[30]. Vision based robotic control approaches often differ by visual processing algorithms, control objectives, sensor placement, camera type, number of cameras, 2D or 3D settings, and kinematic or dynamic control strategies. Based on the control objective, classical approaches are mainly divided into position-based visual servoing (PBVS) and image-based visual servoing (IBVS). PBVS employs the target's 3D pose, estimated using a calibrated camera, as the control objective. As such, PBVS is sensitive to calibration and reconstruction errors and often requires knowledge of the 3D object model. IBVS on the other hand directly uses the 2D image features as a control objective and excludes the extrinsic calibration and 3D estimation process; making it more robust and computationally efficient [3], [31] but less suitable for tasks where depth information is required. In terms of camera placement, visual servoing approaches usually adopt one of two configurations: eye-to-hand or eye-in-hand. In eye-to-hand settings, the camera is fixed in the workspace and concurrently observes both the target object and the robot's end-effector. In contrast, eye-in-hand cameras are embedded within the end-effector, experiencing camera motion as the robot moves. The eye-in-hand configuration has several advantages as it enables more flexible and precise viewing of target objects in the workspace, thus augmenting the versatility and accuracy of robotic manipulation [32]. However, as eye-in-hand camera moves with the robot, conventional frame-based cameras suffer from motion blur, which imposes constraints on ambient illumination and maximum operational speeds [20], [21]. event-based vision has the potential to address these challenges in conventional robotic visual servoing. The recent work in [33] have demonstrated the robustness and performance improvements by utilizing a neuromorphic camera for aerial robotic maneuvering applications, where a line tracking method fuses event-by-event processing together with event-image processing. In this work, we propose a purely event-based visual servoing approach for robotic manipulation. In particular, we devise a full pipeline to detect, track and grasp static and dynamic targets using exclusively the event driven information.

B. CONTRIBUTIONS

Similar to IBVS approaches but in the line of event-based vision research, we present an event-based visual servoing (EBVS) method that adopts an eye-in-hand configuration and processes event stream to control the motion of the robot manipulator. Assuming a static environment, the event cameras in such configurations under need to act to perceive and perceive to act. We define EBVS as a way to control the motion of the robot using instantaneous spatio-temporal event information as feedback. Our approach employs high level event features such as corners and object centroids as

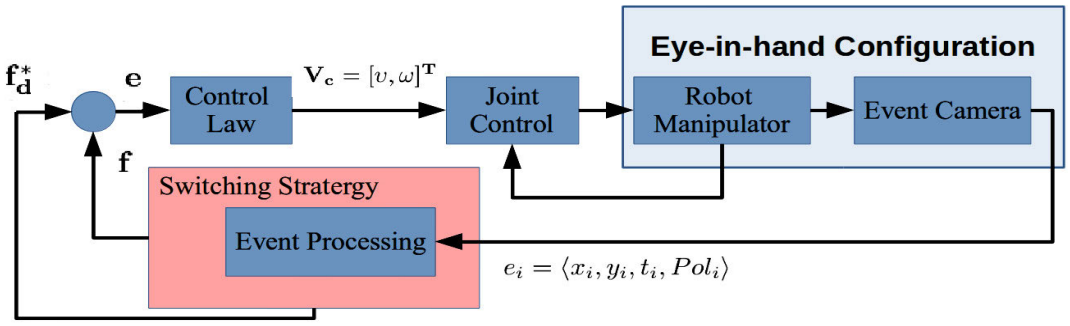


FIGURE 2. Block diagram of the proposed purely event-based visual control scheme (EBVS).

control objective; and relies on the robust extraction and robust tracking of these features to reach the desired robot pose and align the gripper to achieve a stable top-down grasp.

The sensing mechanism of event-based cameras allows detecting transient microsecond level changes in dynamic scene without a global shutter, which is fundamentally different from frame-based cameras. In our recent works [34]–[36], we exploited such potential of event cameras (1) to passively detect incipient and gross slips of a grasped object at a 2KHz sampling rate and suppressed such slips with intelligent grasp controller [34]. (2) to measure contact level forces irrespective of the object size using deep learning methods [35]; (3) to classify grasped objects in the contact level along with machine learning methods for sorting applications [36]. Prior to object grasp, EBVS can assist the robotic manipulation pipeline with precise and stable alignment of the gripper with respect to the target object. In this work, we use a custom-made vacuum gripper to perform a top-down grasp of static and dynamic objects/targets of different geometrical shapes.

A rich survey on event-based vision is available in [19] where several areas relating to robotic applications such as pose tracking, object recognition and tracking, SLAM, etc. are reviewed. In the line of event-based vision research, we address the classic problem in robotic grasping and manipulation that is visual servoing.

In the following, the primary contributions of this paper are summarized:

1) For the first time, we present a purely event-based visual servoing method using a neuromorphic camera in an eye-in-hand configuration for the grasping pipeline of a robotic manipulator. The method processes the event-stream on three layers of active event surfaces to detect, filter and track high level features of an unknown target in the scene and enables the robot to explore, reach, grasp and recover from lost tracking.
2) For the method, we devise an EBVS pipeline where we adopt a robust event-based corner detector; propose a heatmap based corner events filter, a moving average approach for event-based tracker (EBT) and an event driven gripper alignment strategy and switching control strategy.
3) A comprehensive experimental study of the EBVS method and each individual component of the EBVS pipeline.
   a) We have demonstrated the EBVS method by performing top-down pick and place task with different object geometries using a vacuum gripper.
   b) We study and compare the performance of EBVS and conventional frame-based IBVS for static and dynamic objects under different operational speeds and distinct lighting conditions.
   c) We evaluate the EBVS method with the state-of-the-art event-based corner detectors (e-Harris, eFAST, FA-Harris, Arc) and benchmark the event-based tracker (EBT) against the intensity-based KLT under distinct lighting conditions and high speed operation.
   d) We study the computational performance of the EBVS pipeline to evaluate the real-time capability of the method.
4) Experimental results evidence that our method overcomes the issues of conventional frame-based visual servoing methods and achieves superior performance under high speed robot operation and low lighting conditions.

II. EVENT-BASED VISUAL SERVOING METHOD

An event-based visual control scheme for a robotic manipulator with an eye-in-hand configuration to achieve a manipulation task is illustrated in Fig. 2. Instead of a frame-based camera, an event camera is mounted on the robot's end-effector maintaining the relative position with the vacuum gripper. Such setting offers flexibility in viewing the workspace and high precision for grasping objective. Employing a double-loop structure; first, the event stream from a neuromorphic vision sensor caused by relative motion is processed to extract high level features. The switching strategy changes the modes of operation (explore, reach, and align) in event-based visual servoing and regulates the feature stream accordingly. Then, these features are used to estimate the error signal between the goal event state and the current state of the feature events. A simple control law ensuring the

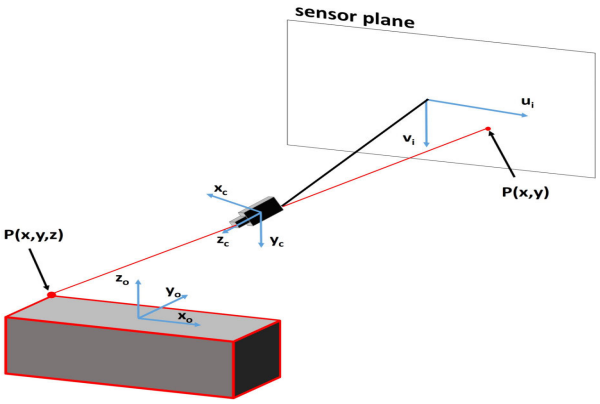


FIGURE 3. A moving event camera projects a point illumination change from a 3D object to the camera's sensor place.

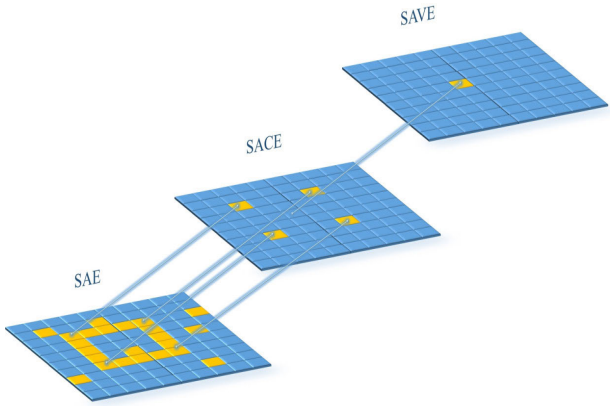


FIGURE 4. Three surface layers for active event processing.

minimization of the feature error outputs a control signal in the form of a command velocity for the robot's end-effector. A second loop locally controls and stabilizes the joints of the robotic manipulator to achieve the commanded velocity. The step-by-step processing of events, control law and switching strategy is detailed in the following.

A. EVENT PROCESSING

Let us consider a moving event camera observing a rigid object placed in a workspace. The movement of the camera generates a stream of events on the camera's sensor plane. The standard pinhole model can still be applied for event cameras since they use similar optics as traditional frame-based cameras. The pinhole projection is shown in Fig. 3, mapping a 3D point $\boldsymbol{\chi} = [x, y, z]$ into a 2D point $\boldsymbol{p} = [u, v]$ on the camera's sensor plane which is expressed in homogeneous coordinates as:

$$z\begin{bmatrix} u, v, 1 \end{bmatrix}^T = \quad K\begin{bmatrix} R & t \end{bmatrix}\begin{bmatrix} x, y, z, 1 \end{bmatrix}^T \tag{1}$$

where K accounts for the camera's intrinsic components and R and t refer to the extrinsic rotational and translation components.

Event camera represents visual information in terms of time with respect to a spatial reference in the camera pixel arrays. Pixels in the neuromorphic vision sensor respond independently and asynchronously to logarithmic brightness changes in the scene. For a relative motion, a stream of events with a microsecond (μs) temporal resolution and latency is generated, where an event $e = \langle \boldsymbol{p}, t, Pol \rangle$ is a compactly represented tuple which describes the point $\boldsymbol{p} = (u, v)$ in the sensor plane coordinate at time t detailing the brightness increase and decrease by polarity Pol. However, analysing a single latest event does not give much information in the operational level and exploring all past events is not scalable.

In this work, we consider three sequential layers of surfaces of active events shown in Fig. 4 for performing operations on the evolving temporal data in the camera pixel space to achieve EBVS. The first layer is known as the surface of active events (SAE), where the surface represents the timestamp of a latest event at each pixel from the raw event stream. For each upcoming event, the function $\Sigma_{SAE} : \mathbb{N}^2 \mapsto \mathbb{R}$ takes the pixel position of a triggered event and assigns to its timestamp:

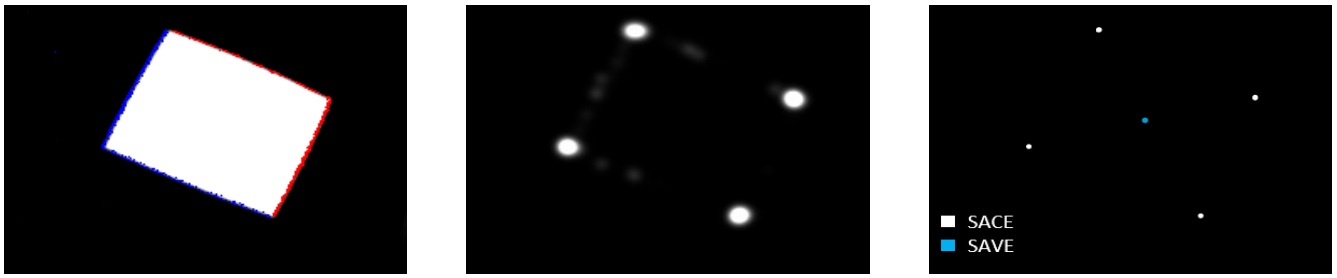
$$\Sigma_{SAE} : (u, v) \mapsto t \mid (u, v) \in \mathbb{R} \times \mathbb{R} \tag{2}$$

In SAE, we apply feature-based algorithms to filter out insignificant events and extract highly informative events such as corners. The second layer is the surface of active corner events (SACE) which maps the pixel position of recent corner events to its timestamp, where we extract the center of the object by robustly localizing the corner events. The object center is the extracted high level feature that is a virtual event and not an actual event used in visual servoing. Moreover, we introduce random and goal state events and consider them as virtual events. The third layer is the surface of active virtual events (SAVE) that maps the extracted and artificially induced virtual events pixel position to its timestamp, where the contiguity of the high level feature is analysed for switching the control objectives. EBVS modes of operations such as exploration, reaching, and grasping are determined by the SAVE.

B. EVENT-BASED FEATURE DETECTION

1) CORNER DETECTION

In conventional image processing, Harris detector [37] is one of the most widely used techniques to detect features such as corners, edges, and flat points based on strong intensity variation in a local neighborhood. This feature detector is known for its efficiency, simplicity, and invariance under scaling, rotation, and illumination. Unlike conventional camera that records large amount of redundant data in a sequence of frames, the event camera records only changes in the visual scene as a stream of events characterized by the pixel positions and its timestamps and does not include intensity measures. Therefore the frame-based Harris detector cannot be directly applied on the SAE. An event-based adaptation of the Harris detector (e-Harris) was proposed in [38], where each event is directly processed based on a binarization of the SAE by the newest N events. A spatially adaptive e-Harris implementation was proposed in [39] where the



(a) (b) (c)

FIGURE 5. Event-based Feature Detection: (a) Rendered event frame (30ms) (b) Corner events heatmap (c) Features events extracted in SACE and SAVE.

newest events are only sampled from a local patch around the event under consideration, making the detection independent of the scene and sensor size. Let Σ_b be a binary surface locally centered around the latest event where 1 and 0 indicates the presence and absence of an event. The gradient $\nabla I_e = (I_x, I_y)$ is computed on the binary surface with 5×5 sobel operator G_x and $G_y = G_x^T$ as:

$$I_x = \Sigma_b G_x; I_y = \Sigma_b G_y \tag{3}$$

The Harris matrix can then be computed as:

$$H_m = \sum_{e \in \Sigma_b} Gu(e) \nabla I_e \nabla I_e^T \tag{4}$$

where Gu(e) represents the Gaussian window function. Finally, the Harris score can be calculated as:

$$H_s = det(H_m) - k.trace(H_m)^2 \tag{5}$$

where k is the empirical constant. The Harris feature detector mainly relies on the analysis of the eigenvalues of the auto-correlation matrix. If the Harris score is a large positive value, the event is classified as a corner whereas a negative value is considered as an edge. The rest of the events which are in between are considered as flat points. In our case, we have adopted the spatially adaptive e-Harris detector introduced in [39]; and have selected a local patch of size 9×9 pixels, a buffer of latest events $N = 20$, and a corner threshold of $HC_{th} = 5$, as they resulted in the best performance in our experimental settings.

In addition to e-Harris, several event-based corner detection algorithms are proposed in the literature, such as: SILC [40], eFAST [39], Arc, [41] and FA-Harris [42]. Our EBVS method could utilize any of state-of-the-art corner detectors available in the literature and is not limited to the use of e-Harris. In this work, we have also studied and compared the performance of EBVS with different event-based corner detectors (see section III-C). The selection of event-based corner detector can be based on the trade-off between the event processing speed and robust performance. After an extensive experimental study, the adoption of e-Harris corner detector in EBVS pipeline showed an overall better performance under high operational speeds and poor lighting condition for a given grasp target within the experimental settings.

2) HEATMAP BASED CORNER FILTER

Whenever a corner event $p_{hc} : (u_{hc}, v_{hc}, t_c, Pol_c)$ is detected, it is projected in the SACE. To cluster these events into object corners and minimize the influence of noisy events, consecutive corner points are concatenated to form a heatmap of corner events. A heatmap matrix $H \in \mathbb{R} \times \mathbb{R}$ is introduced for this purpose. Whenever a new corner event is received, the elements of H are updated as:

$$H_{i,j}(t^+) = H_{i,j}(t^-) + \frac{\alpha}{\sqrt{2\pi}\sigma} e^{-0.5 \times ((u_{hc}-i)^2 + (v_{hc}-j)^2)/\sigma^2} \tag{6}$$

where u_{hc} and v_{hc} represent the coordinates of the most recent corner event, α is a user-defined scaling factor, and σ is the standard deviation of the incoming corner event which dictates the area of effect each event has on the heatmap.

To keep only the recent corner events relevant to the process of detecting the object corners, the heatmap is continuously updated with time as indicated by eq. 7, where τ is a time constant dictating the period of influence for each corner event and t_c is the timestamp of the last received corner event.

$$H(t^+) = e^{-\tau(t_c - t)} \times H(t^-) \tag{7}$$

As such, the heatmap H represents spatio-temporal patterns of corner events. The corners of the object are then obtained from these patterns by detecting the local maxima if these maxima exceed a minimum threshold denoted by HO_{th}. Fig. 5 shows an example corner event heatmap along with its local maxima for a sample object placed in the camera's field of view. Let $S = \{(p_c^0 : (u_c^0, v_c^0)), \ldots, p_c^n : ((u_c^n, v_c^n))\}$ be the set of n local maxima extracted from the heatmap, the object's centroid p_{voc} in pixel coordinates can be computed as:

$$p_{voc} : (u_{voc}, v_{voc}) = (\frac{1}{n}\sum_{i=0}^{n} u_c^i, \frac{1}{n}\sum_{i=0}^{n} v_c^i) \tag{8}$$

p_{voc} is then projected to SAVE and is considered as the control objective for tracking operations in visual servoing. Upon experimental trials, we have selected the heatmap filter parameters as: $\alpha = 0.15, \sigma = 10, \tau = 0.6$, and $HO_{th} = 0.65$.

C. EVENT-BASED FEATURE TRACKING

1) MOTION CONTROL

Let $f \in \mathbb{R}^2$ denote the pixel coordinates of the detected high level features in SAVE (e.g., the object's centeroid), and $f_d^* \in \mathbb{R}^2$ denote the desired coordinates of these high level features (e.g. the center of the sensor plane). The primary goal of EBVS is to compute the camera velocity $\vec{V}_c$ such that L2 norm the error $\vec{\zeta} = f - f_d^*$ is minimized. In our case, we constrain the camera's motion to a linear 2D movement parallel to the sensor plane. As such, the camera velocity vector can be represented as $\vec{V}_c = (v_x, v_y)$. The relationship between the camera's velocity and the error rate in the sensor plane is given by:

$$\dot{\vec{\zeta}} = L\vec{V}_c \tag{9}$$

where $L \in \mathbb{R}^{2\times 2}$ is the image Jacobian given by:

$$L = \begin{bmatrix} \frac{-F}{Z} & 0 \\ 0 & \frac{-F}{Z} \end{bmatrix} \tag{10}$$

in which F is the focal length and Z is the depth of the object.

Following the above formulation, a control law that ensures an exponential decrease of the feature error ($\dot{\vec{\zeta}} = -\lambda\vec{\zeta}$) is expressed as:

$$V_c = -L^{\dagger}\lambda\vec{\zeta} \tag{11}$$

where $\lambda \in \mathbb{R}^{2\times 2}$ is a positive-definite gain matrix, and $L^{\dagger}$ is the inverse of the image jacobian L. Since L is diagonal as shown in (10), the computation of $L^{\dagger}$ is efficient and straightforward.

It must be noted that the control law in (11) is dependent on the depth value Z. We assume this value to be known, which is often the case in many industrial applications where the height of the workspace plane (e.g. table, conveyor belt) is fixed and the end-effector's 6-DOF position $\vec{r}$ can be easily computed from measured joint angles θ as:

$$\vec{r} = g(\theta) \in \mathbb{R}^6, \theta \in \mathbb{R}^{n_j} \tag{12}$$

where $g(\theta)$ is a nonlinear function representing the robot's kinematics, and n_j is the number of joints.

Once the camera's reference velocity is computed from (11), reference joint velocities $\dot{\theta}$ can be obtained by solving the following:

$$\dot{\theta} = J(\theta)^{\dagger}\dot{\vec{r}} = J(\theta)^{\dagger}\begin{bmatrix} v_x & v_y & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{13}$$

in which $J(\theta)^{\dagger}$ represents the inverse of the robot Jacobian defined as:

$$J(\theta) \triangleq \frac{\partial g}{\partial \theta} \in \mathbb{R}^{6\times n_j} \tag{14}$$

Finally, the target velocity for each joint from (13) is regulated independently by a low-level PID controller.

2) EVENT-BASED TRACKER (EBT)

As the end-effector moves towards the object, the location of the object's corners and centroid in the sensor plane must be updated. Several works in the literature have studied the event-based corner tracking problem; including purely event-driven approaches [41], and hybrid approaches that fuse the event stream with intensity images [43]. For our application, we adopt a simple and computationally efficient moving average approach to track the corner features. For every new corner point detected by the e-Harris algorithm denoted by p_{hc}, the closest object corner $p_c^i : (x_c^i, y_c^i) \in S$ is determined and then updated as:

$$p_c^i(t^+) = \beta \times p_c^i(t^-) + (1-\beta) \times p_{hc} \tag{15}$$

Whenever the SACE is updated, the SAVE is also updated accordingly, leading to a refined estimate of the object's centroid.

Due to its simplicity, tracking corners using the moving average approach is much faster than the heatmap corner filter in section II-B2; making it more suitable for high speed applications. However, it is prone to errors if tracking of one corner is lost. To account for such cases, EBT is checked against the corner event heatmap at a regular interval; if considerable discrepancies were found over multiple iterations, the system reverts to the exploration phase.

D. GRIPPER ALIGNMENT TO GRASP

Once the robot tracks and reaches the object's center, the orientation of the gripper is adjusted to achieve a stable grasp. A target orientation θ is defined such that the two gripping points are aligned with a virtual line connecting the object centroid $p_v : (x_v, y_v)$ in the SAVE with a corner point p_c^i in the SACE. To maximize the stability of the grip, p_c^i is selected as the corner point furthest from the centeroid. θ is hence computed as:

$$\theta = atan2(v_c^i - v_{voc}, u_c^i - u_{voc}) \tag{16}$$

Fig. 6 shows the alignment process where the grippers are rotated at a constant angular velocity until θ is within an admissible range.

E. SWITCHING CONTROL STRATEGY

The switching strategy is an essential part of the visual servoing pipeline and is critical to the success of the task. In static environment, the event camera has the requirement to act to perceive and perceive to act, we devised a control strategy to accommodate such requirement and allow switching between different phases to detect and track target objects and recover from potential failures. Overall, the switching strategy enables the robot to explore, reach, grasp and recover from lost tracking in the process of EBVS. In Fig. 7, the switching operation is illustrated in the surface of active virtual events. First, a random virtual event p_{vr} is triggered in SAVE to motivate the robot to explore the workspace and detect the true object feature p_{voc}. The highlighted yellow

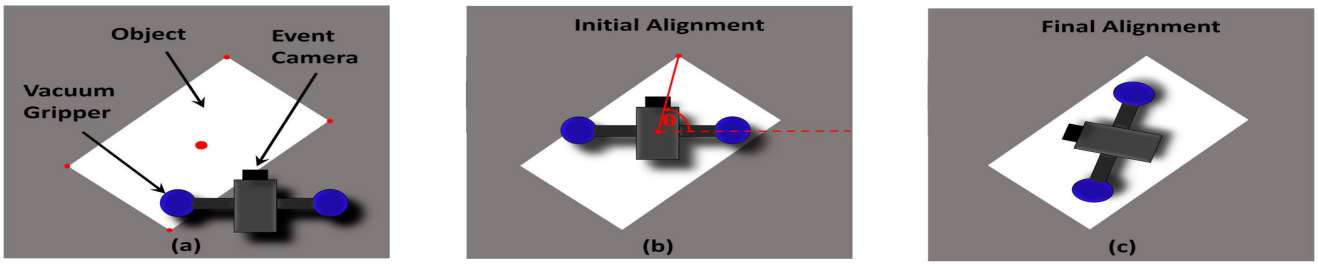


FIGURE 6. Constraint set for gripper alignment after servoing.

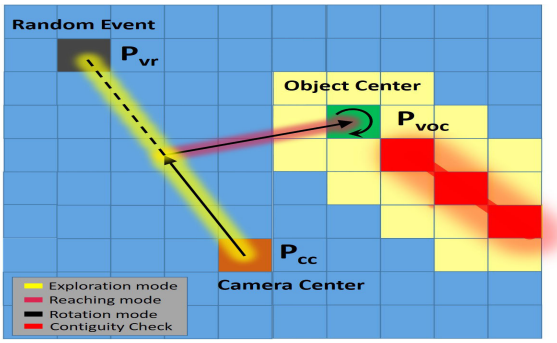


FIGURE 7. Illustration of switching strategy that explores to detect, tracks to reach and aligns to grasp. In our strategy, we generate a virtual random event (p_{vr}) and an event (p_{cc}) at the center of the sensor plane in the surface of active virtual events (SAVE) to explore the scene and detect object features to determine a virtual goal event p_{voc}. Moreover, we check the contiguity of the triggered events in a spatial level to determine the goal in the exploration mode and robustly track (p_{cc} - p_{voc} → 0) them in the reaching mode. In particular, the strategy gives the robot the capability to recover (in cases where tracking is lost) and reach the desired goal.

color indicates the pathway chosen by the robot in the exploration phase. During exploration, the contiguity of p_{voc} is analysed. Once the count of contiguous pixels crosses a predetermined threshold C_{th}, the robot enters the reaching phase and tracks p_{voc} following the procedure in section II-C. The highlighted pink color indicates the new pathway to reach the object center. Switching can occur even in the reaching phase to account for tracking errors and contiguity breakdown; which gives the robot the ability to recover from tracking failures. The switching function can hence be expressed as:

$$f(t) = \begin{cases} p^i_{vr}, & \text{if contiguity in } \mathbf{p}^{j,\ldots,n}_{voc} \mid j < 3 \\ p^j_{voc}, & \text{if contiguity in } \mathbf{p}^{j,\ldots,n}_{voc} \mid j > 3 \end{cases} \tag{17}$$

Finally, when p_{voc} converges to p_{cc}, the robot switches to the alignment phase where the grippers are rotated to perform a stable grasp as discussed in section II-D. Algorithm 1 summarizes the three-layer event feature extraction and control strategy for EBVS.

Algorithm 1: Event-based Visual Servoing

Input: Stream of events $e_i = \langle x_i, y_i, t_i, Pol_i \rangle$, Contiguity threshold C_{th}
Output: Command velocity $\vec{V}_c$.

```
Initialize three layers of surface of active events SAE, SACE and SAVE.
Initialize a desired feature event p_cc (eg: center of the sensor plane) in SAVE
Initialize switching strategy
for each e_i do
    Detect corners in SAE by applying e-Harris, and project corner events to SACE.
    Extract object corners in SACE using heatmaps.
    Compute object centroid p_voc from object corners events in SACE, and project p_voc to SAVE.
    if Contiguity count < C_th then
        Initialize a random desired event p_vr in the SAVE.
        Engage visual servoing to the random feature event.
        Detect and track object feature events in SAVE.
    if Contiguity count > C_th then
        Track the object centroid event p_voc.
    if camera center = object center then
        Align gripper orientation for a stable grasp.
        Move to the pre-grasp pose and execute grasp.
```

III. EXPERIMENTAL RESULTS

We implemented the EBVS pipeline on a top-down robotic grasping platform to assess proposed method's performance and demonstrate its appropriateness for real world smart manufacturing applications. This section presents a comprehensive experimental study of the EBVS method and highlights its functional advantages in practical scenarios. First, we demonstrate the sequence of EBVS operations for a pick and place task and validate the method for different object geometries. Second, we evaluate the performance of each component of the EBVS pipeline for static and dynamic targets under distinct lighting conditions and different robot operational speeds. Third, We benchmark EBVS results against a conventional intensity image-based visual servoing solution; and identify the practical advantages of event cameras in industrial applications. Lastly, we analyse the computational performance of EBVS and verify its real-time capabilities.

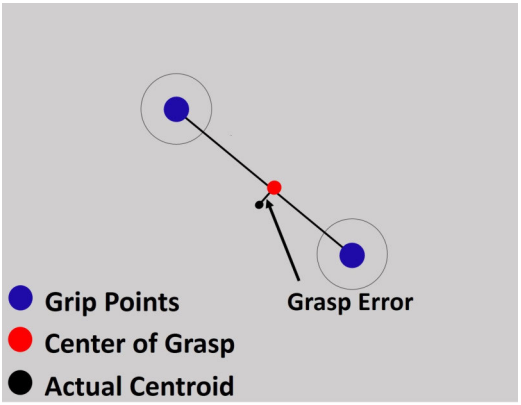


FIGURE 8. The grasp error e_{grasp} is defined as the distance between the center of grasping and the true object centroid.

A. EXPERIMENTAL SETUP AND PROTOCOL

The experimental setup consists of a Universal Robots UR10 6-DOF arm [44], a custom-made vacuum gripper, and a Dynamic and active pixel vision sensor (DAVIS240) placed in an eye-in-hand configuration as displayed in Fig 1. The DAVIS240 provides both a neuoromorphic event stream and frame-based intensity images at a spatial resolution of 240 × 180. The DAVIS240's event stream has a minimum latency of 12 microseconds, a bandwidth of 12×10^6 events per second (eps) and a dynamic range of 120 dB.

To successfully pick and place an object, the robot's end-effector is first driven into alignment with the target object using the EBVS process described in section II. During the exploration and reaching phase, the end-effector's movement is constrained to a 2D plane perpendicular to the camera's optic axis. Once the end-effector is aligned with the target object, it is translated in the camera's optic axis direction until contact with the object is achieved. Subsequently, the vacuum grippers are activated to grasp the target and relocate it to a desired location. Given the limitations of the UR10's reach and the camera's field of view, the workspace of the experiments was limited to a 1.2 × 1.0 m virtual rectangle in front of the robotic platform.

To quantitatively evaluate EBVS results, we use two different measures of errors: grasping error e_{grasp} and tracking error e_{track}. e_{grasp} represents the error at the terminal state upon grasping the object and is defined as the distance between the center of the two gripping points and the true object's centroid as illustrated in Fig. 8. On the other hand, e_{track} quantifies the errors throughout the full tracking stage of EBVS as the gripper is being aligned with the target; and is defined as the Mean Absolute Error as below:

$$\vec{e}(t) = \vec{r}(t) - \vec{p}(t) \tag{18}$$

$$e_{track} = \int_0^T \|\vec{e}(t)\|\, dt \tag{19}$$

where $\vec{p}(t)$ and $\vec{r}(t)$ are subsequently the target object and the end-effector's location at time t, $\vec{e}(t)$ is the error vector in homogeneous coordinates, and T is the overall time for EBVS to grasp the object. We also consider the number of times tracking was lost and the algorithm switches back to detection mode, denoted by N_{switch}.

TABLE 1. EBVS experimental results across multiple trials of grasping a static rectangular shape.

Trial	e_{grasp} (cm)	N_{switch}
1	0.68	1
2	0.15	0
3	0.16	0
4	0.86	0
5	0.54	0
Average	0.48	0.2

TABLE 2. EBVS results for grasping static objects of different geometrical shapes averaged across multiple trials.

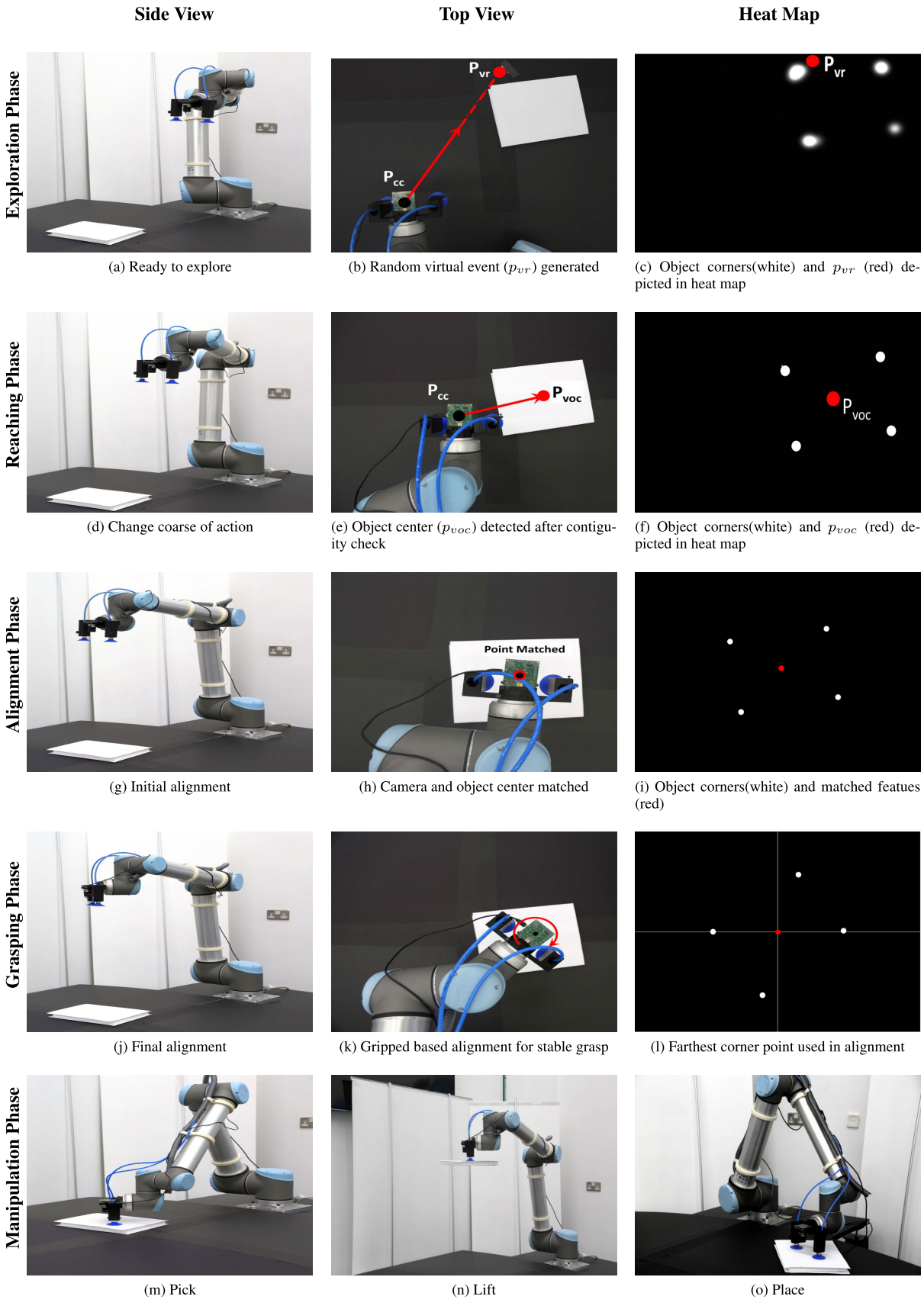
Shape	Trials	Mean e_{grasp} (cm)	Maximum e_{grasp} (cm)	Maximum N_{switch}
Triangle	5	1.22	1.96	0
Rectangle	5	0.48	0.86	1
Pentagon	5	0.44	1.03	3
All Shapes	15	0.71	1.96	3

B. EBVS METHOD VALIDATION

This section presents qualitative and quantitative EBVS results for grasping static objects of different geometrical shapes. Fig. 9 shows the various stages of EBVS for a visual servoing trial with a rectangular object. For each stage, the robotic platform is displayed along with the corresponding heatmap of corner events and SAVE. During the exploration phase, the end-effector first moves towards a random virtual event p_{vr} to trigger events in the scene and update the corner-events heatmap. Based on the heatmap, the EBVS algorithm detects the object's high level features (e.g. corners and centroids). Once contiguity is verified for these features, the robot switches to the reaching phase where it moves towards the object's centroid p_{voc}. The robot then enters the alignment phase where the grippers are rotated to achieve a stable grasp. Finally, the robot enters the grasping and manipulation phase to pick the object and place it in a desired location.

The same experiment in Fig. 9 was repeated five times with different placement of the rectangular object in the workspace. Table. 1 shows the results of these experiments in terms of e_{grasp} and N_{switch}. Similar experiments were also carried out with different geometrical shapes including triangular and pentagonal objects. Table 2 shows experimental results across five trials for each of the three different geometrical shapes.

In all the conducted experiments, EBVS was capable of successfully tracking and grasping the target object with both vacuum grippers adhering to the object. The average grasp error of the overall experiments was 0.71cm. The main source



(a) Ready to explore

(b) Random virtual event (p_{vr}) generated

(c) Object corners(white) and p_{vr} (red) depicted in heat map

(d) Change coarse of action

(e) Object center (p_{voc}) detected after contiguity check

(f) Object corners(white) and p_{voc} (red) depicted in heat map

(g) Initial alignment

(h) Camera and object center matched

(i) Object corners(white) and matched featues (red)

(j) Final alignment

(k) Gripped based alignment for stable grasp

(l) Farthest corner point used in alignment

(m) Pick

(n) Lift

(o) Place

FIGURE 9. **Sequence of visual servoing operations to perform object manipulation task.**

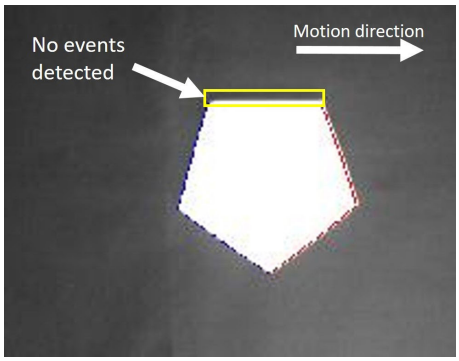


FIGURE 10. The event camera does not trigger events (blue and red) when moving parallel to an edge, this can cause EBVS to lose track of certain features in the sensor plane.

of grasping errors is the low resolution of the DAVIS240 camera. Considering the resolution of the DAVIS240 camera, its 59° diagonal field of view, and that the 2D visual servoing is performed at a height of 0.75m off the workspace plane; an error of 0.71cm corresponds to less than three pixels in the sensor plane. A straightforward workaround would be adjusting the initial height of the camera, which introduces a trade-off between the possible workspace area and grasping accuracy. Other sources of error include design imperfections such as the misalignment of the camera optic axis with the workspace plane and the skewed positioning of the camera with respect to the center of the vacuum grippers. Such errors can be effectively reduced with an accurate calibration process. Enhancing the robustness of EBVS for such irregularities would be a major objective for future work.

The conducted experiments show that the proposed algorithm loses track more often with the pentagon shape; which can cause longer convergence times. As shown in Fig. 10, when the event camera moves parallel to an edge, it is less-likely to trigger events corresponding to this edge. As a result, the event-based corner detection fails to detect corners associated with edges parallel to the camera's movement; causing EBVS to lose track. As a pentagon shape has more edges with varied slopes than a rectangle or a triangle, it is a more probable case for EBVS to encounter this shortcoming. In our future work, we plan to reduce the EBVS parallel edge detection problem by 1) using an optimal motion planner considering the event camera's vision constraints. 2) incorporating motions along the camera's optic axis, which would trigger more events due to the scaling effect. 3) adding an oscillatory motion in the end-effector's motion during the reaching phase.

C. EBVS EVALUATION WITH STATE-OF-THE-ART CORNER DETECTORS

Corner detection is a critical component of the EBVS pipeline. In section II-B, we proposed the use of e-Harris [38] for the corner detection task, however, the EBVS pipeline allows for the use of any event-based corner detection algorithm. In this section, we evaluate the applicability of different state-of-the-art event-based corner detectors for EBVS and justify the selection of e-Harris [38] as the most appropriate corner detector within our experimental configuration. We consider four different corner detection algorithms: e-Harris [38], FA-Harris [42], eFAST [39], and Arc [41]. In our analysis, we emphasize the effects each corner detector has on the visual servoing performance taking into consideration the high operational speed (max. 1.5m/s) of the robot and two distinct light intensity levels E_v.

TABLE 3. Tracking error e_{track} in cm for high speed EBVS using different event-based corner detectors under varied lighting conditions.

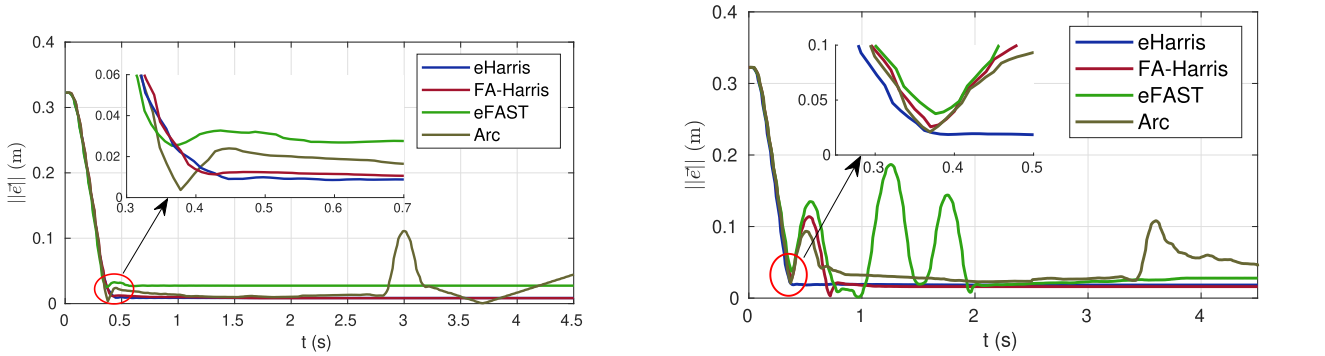
Corner Detector	Light intensity E_v	
	Adequate (365 lx)	Low (<10 lx)
e-Harris	8.39	10.19
FA-Harris	8.69	11.34
eFAST	10.45	26.02
Arc	15.45	18.84

Fig. 11 shows the error profile $\vec{e}(t)$ obtained from multiple high speed visual servoing experiments using the different corner detectors under two distinct lighting conditions, and Table 3 presents the corresponding tracking errors e_{track}. e-Harris results in the most consistent servoing performance in both lighting conditions where the error converges in less than 0.5 seconds, and the resultant e_{track} is the lowest compared to the other corner detectors. Visual servoing with FA-Harris shows slightly lower performance than e-Harris in adequate lighting conditions, but suffers from overshoot and larger convergence time at low light operation. The FA-Harris overshoot behaviour is due to partial failure in detecting the target object's corners at low lighting conditions, where less number of events are being generated (see Fig. 17 (c)). eFAST maintained a steady state error in adequate lighting conditions and diverged several times in low lighting condition, which in turn leads to larger e_{track} errors. Finally, the Arc corner detector is very sensitive to noise and results in considerable false positive corner detection; this impedes EBVS from converging and can cause undesired movements of the robot's end-effector. As for computational efficiency, all the studied corner detectors provide ample performance for real-time operation within the considered experimental configuration, which is discussed in detail in section III-G.

Considering these results, we have chosen e-Harris as the most appropriate corner detector for the proposed EBVS pipeline due to its accuracy, robustness, and tolerance to variations in lighting conditions. It must be noted that the performance of all corner detectors is subject to several tunable parameters. In our experiments, these parameters were manually tuned and fixed across all the conducted experiments.

D. EVENT-BASED TRACKER (EBT) EVALUATION AND COMPARISON

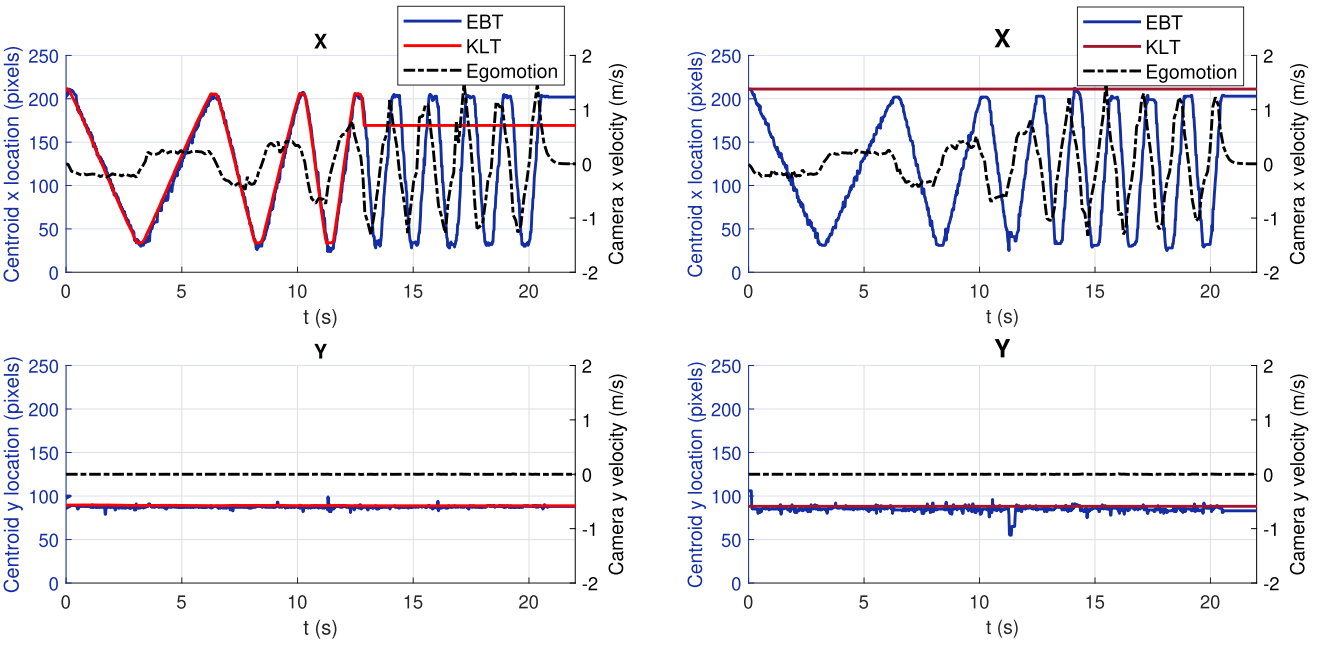
This section evaluates the EBT component of the EBVS pipeline and compares it's performance with an intensity



(a) High speed visual servoing experiment in adequate lighting conditions (365 lx).

(b) High speed visual servoing experiment in low lighting conditions (<10 lx).

FIGURE 11. Error profile of EBVS for high speed visual servoing using different event-based corner detectors under two distinct lighting condition. e-Harris shows superior performance over other detectors, especially in low light conditions.



(a) Object centroid tracking performance under adequate light condition (365 lx).

(b) Object centroid tracking performance under low light condition (<10 lx).

FIGURE 12. EBT and intensity-based KLT performance under varying egomotion velocities and lighting conditions. EBT shows superior performance at high speed operation and low light conditions.

image-based tracking approach. Analogous to the analysis presented in [41], we evaluate the tracking accuracy using the mean minimum distance between tracks generated by EBT and those generated using intensity images. To generate intensity-based tracks, we detect corners from the DAVIS240's intensity images using the original frame-based Harris detector [37], and track these corners using the well-known Kanade-Lucas-Tomase (KLT) [45] tracker. We limit our analysis to tracking a single object in the scene. Table 4 reports the EBT performance across different egomotion speeds for both the corner tracking and centroid tracking objectives. Results show consistent performance of EBT with an error lower than 2 pixels across all experiments. However, the intensity-based KLT tracker fails at higher speeds due to excessive motion blur. As such, we were not able to compute the mean minimum distance criteria for high speed experiments.

To better analyze the advantages of EBT and highlight the failure cases of intensity-based tracking, Fig. 12 shows the object tracking results from both EBT and KLT under varying egomotion speeds and lighting conditions. At lower speeds and adequate lighting conditions, EBT and KLT exhibit a similar behaviour. However, at speeds higher than 1 m/s or at low light conditions, KLT fails to track the object while

TABLE 4. Mean Minimum Distance in pixels between EBT and intensity-based KLT at different egomotion speeds v_{max}. We were not able to compute errors at high speed experiments due to the constant failure of KLT with excessive motion blur.

v_{max}	Mean Minimum Distance	
	Corner Tracking	Center Tracking
0.075	1.5	1.1
0.35	1.6	1.7
0.75	1.9	1.6
1.5	NA	NA

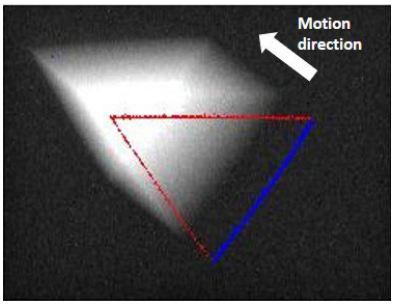


FIGURE 13. Comparison between the event stream (blue and red) and intensity images of DAVIS240 at high speed motion and low light conditions. The intensity image suffers from motion blur due to high exposure time, and is considerably delayed compared to the event stream.

EBT remains persistent. The main reason for KLT's failure is the severe motion blur and latency observed in the camera's intensity images, which causes KLT to lose track of features whenever high speed movements are present. These shortcomings of intensity-based tracking become more severe for low light operation, where the camera's exposure time must be increased to enable perception. In contrast, event cameras do not suffer from motion blur and do not require a trade-off between image clarity and latency. The main effect low light conditions have on event cameras is an increase in noise, which has been studied by many works in the literature [46]–[48]. EBT is robust to such noise and performs persistently across different speeds and lighting conditions. Fig. 13 provides a visualization of these advantages of event cameras, where the event stream remains sharp and timely while the intensity image is blurred and delayed.

E. PERFORMANCE COMPARISON BETWEEN EBVS AND IBVS

Building upon the advantages of EBT shown in section III-D, this section highlights the performance improvements of using event cameras in visual servoing applications. We benchmark EBVS against a conventional intensity image-based visual servoing (IBVS) technique. For IBVS experiments, we utilize the same pipeline in Section II with replacing the event-based Harris corner detector with the original frame-based formulation [37]. Experiments similar to those in section III-B were conducted for a triangular object under two different light intensity levels E_v and with different maximum speed constraints v_{max} imposed on the movement

TABLE 5. Comparison of grasping errors e_{grasp} between EBVS and IBVS under different light intensity levels E_v and maximum operation speed v_{max}. Results are averaged across five trials.

v_{max} (m/s) \ E_v (lx)	365		<10	
0.075	$1.22cm$	$2.14cm$	$1.75cm$	$1.86cm$
0.35	$1.27cm$	$1.52cm$	$1.58cm$	Fail
1.5	$0.91cm$	Fail	$2.01cm$	Fail
Type	EBVS	IBVS	EBVS	IBVS

of the robotic arm. Table 5 reports EBVS and IBVS results averaged across five repetitions of each experimental scenario. Fig. 14 shows the time profile of the robotic end-effector's position throughout a single EBVS and IBVS trial for each of the six conditions.

Results in Table 5 and Fig. 14 show similar performance between EBVS and IBVS at low-speed operation with adequate lighting. Performance improvements of EBVS start appearing as we deviate from these nominal conditions towards faster movements and lower light conditions, where it can be clearly seen that IBVS fails to converge towards the target object's location. IBVS's failure is mainly attributed to the performance deterioration of the intensity-based tracker due to latency and motion blur as discussed in detail in Section III-D. By comparison, EBVS's performance remains persistent across all operational conditions; which highlights its robustness and appropriateness for a wide variety of industrial scenarios.

F. DYNAMIC OBJECT TRACKING USING EBVS

EBVS has also been validated for tracking dynamic objects in the environment. The experimental setup for the dynamic object tracking tests are shown in Fig. 15, where a triangular object was fixed to a rotating disk causing it to move in a circular path of a radius $r = 0.245\ m$. The speed of rotation was incrementally increased from 2 rpm to 7 rpm. For comparison purposes, tests were conducted using both EBVS and IBVS under two different lighting conditions. Fig. 16 shows the object's reference trajectory and the trajectory traveled by the robotic gripper throughout all experiments, and provides the average tracking error e_{track} for each case. Results show that EBVS outperforms IBVS, and is not affected under low light conditions as opposed to the failure observed in the case of IBVS. These results validate EBVS's appropriateness to a wide set of industrial assembly lines, which often involve objects moving on conveyor systems under varying illumination levels.

G. EBVS PIPELINE COMPUTATIONAL PERFORMANCE

All the required computations for EBVS experiments were performed on an Intel i7-8650U CPU with 1.8GHz and 16GB of RAM. The EBVS pipeline uses a multithreaded

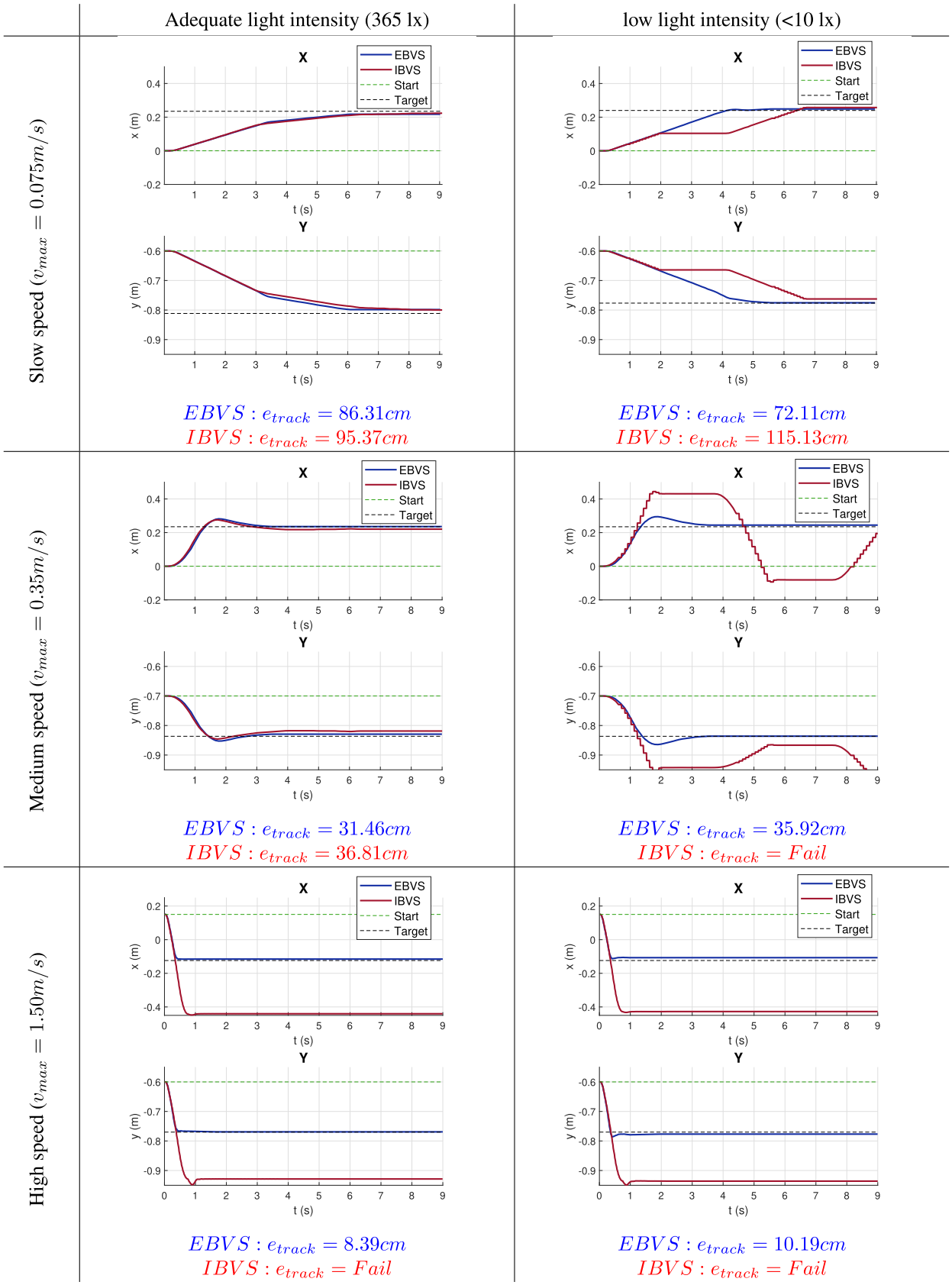


FIGURE 14. Time profile for the x-y planar motion of the manipulator during top-down grasping experiments using both EBVS and IBVS under different lighting conditions and operation speeds. EBVS successfully drives the manipulator to the target object's location in all experiments while IBVS fails to do so under low light and high speed operations.

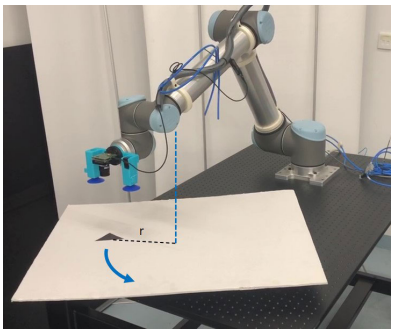


FIGURE 15. Experimental setup for dynamic object tracking trials.

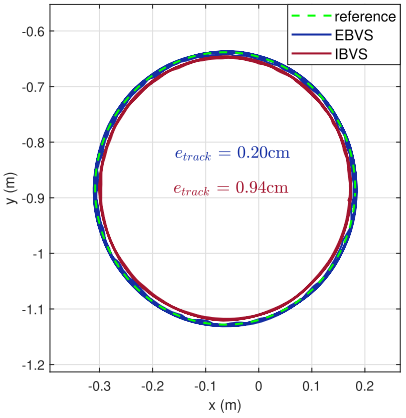
C++ implementation and Robot Operating System (ROS) as a middle-ware. Each component of the pipeline - namely the event-based corner detector, heatmap based corner filter, and event-based tracker run on separate threads. In Fig. 17, we report a comprehensive computational performance study of the EBVS method illustrated with multiple plots, tables, and a picture for the sake of clarity and easy read.

The picture in Fig. 17-(a) shows the experimental setting and the two distinct lighting condition used for the EBVS pipeline computational evaluation. The end-effector with the mounted event camera is moved 0.5 meter back and forth in the Y plane with four maximum operational speeds under adequate and low light conditions such that the event stream is acquired from the scene. The recorded event stream is used for further computational performance study.

Table 6 summarizes the computational performance for each component of the EBVS pipeline in terms of the average μs time required to process a single event and the corresponding maximum event processing rate in Millions of events per second (Meps). For the corner detection component, we have evaluated and quantified the computational performance of the state-of-the-art event-based corner detectors with our system. On average, the Arc runs about 11× faster than e-Harris, 4.5× faster than FA-Harris, and 3.7× faster than eFast. This order of the computational performance between different corner detectors is consistent with the earlier works [41], [42] . The heatmap based corner filtering is the most time-consuming process when compared to any other component in the EBVS pipeline while EBT takes the least amount of processing time.

To demonstrate computational appropriateness of the presented EBVS approach for practical applications, we analyse the raw and corner event generation rates with the maximum event rate of each component in the EBVS pipeline. The raw event generation rate is highly variable depending on many factors such as movement speed, camera resolution, and scene properties. Moreover, the corner generation rate is further dependant on the corner detection algorithm and the amount of generated raw events. Table 7 presents the number of raw events generated from the DAVIS240 camera and corner events generated from four different event-based corner detectors under four different operational speeds ranging

(a) adequate light intensity (365 lx)



(b) low light intensity (<10 lx)

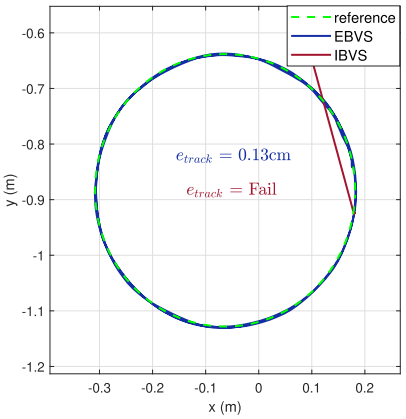


FIGURE 16. EBVS and IBVS results for tracking an object moving in circular motion under different lighting conditions. EBVS's performance is consistent under both cases, while IBVS fails under low light conditions.

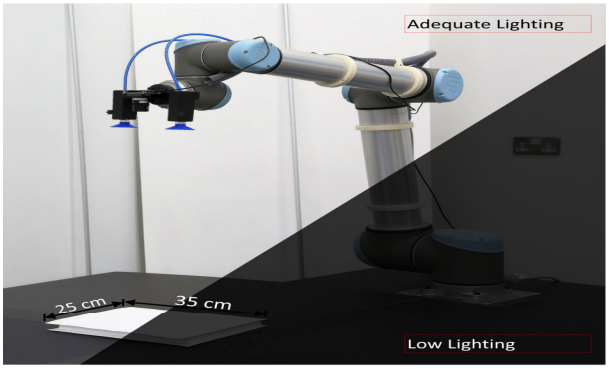
TABLE 6. Computational performance of different components of the EBVS pipeline.

Process		Time per event [μs/event]	Max. event rate [Meps]
Corner Detection	e-Harris	3.81	0.26
	FA-Harris	1.50	0.67
	eFAST	1.21	0.82
	Arc	0.32	3.1
Heatmap based corner filtering		15.33	0.065
Event-based Tracking		1.35	0.74

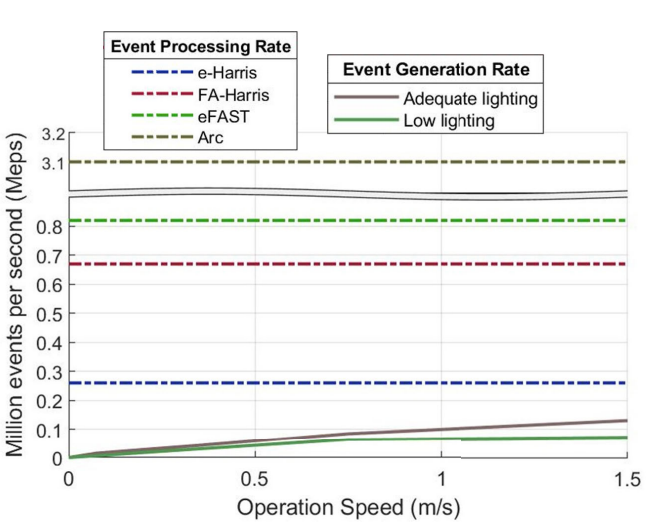
from 0.075-1.5 m/s and two different lighting conditions (365 and <10 lx). For the same object in the scene, the raw events generated from high speed robot operation that is 1.5 m/s under adequate and poor lighting condition is 8× and 10× the amount of raw events generated at 0.075 m/s. On average, for varied speeds, the amount of raw events generated under adequate lighting condition is 1.7× more

TABLE 7. Given the experimental setting (a), Events and Corner-Events generated from the DAVIS240 camera and different event-based corner detectors under four different operational speed and two distinct lighting condition.

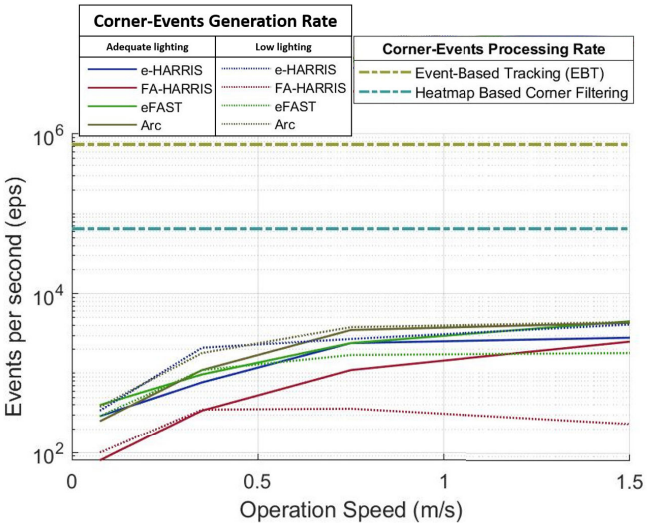
Robot Operating Speed v_{max} [m/s]	Scene: Adequate Lighting Condition (365 lx)					Scene: low lighting Condition (<10 lx)				
	Event Generation Rate [Meps]	Corner Event Generation Rate [Meps]				Event Generation Rate [Meps]	Corner Event Generation Rate [Meps]			
		e-Harris	FA-Harris	eFAST	Arc		e-Harris	FA-Harris	eFAST	Arc
0.075	0.015	0.00029	0.00008	0.0004	0.00025	0.0065	0.00034	0.0001	0.00029	0.00038
0.35	0.042	0.00077	0.00034	0.00097	0.0011	0.03	0.0021	0.00035	0.0011	0.0018
0.75	0.084	0.0024	0.0011	0.0024	0.0035	0.063	0.0027	0.00036	0.0017	0.0038
1.5	0.13	0.0028	0.0025	0.0045	0.0043	0.071	0.0041	0.00023	0.0018	0.0044



(a) Illustration of adequate and low light conditions in the scene and the experimental setting to evaluate computational performance.



(b)



(c)

FIGURE 17. EBVS Computational performance report.

than the events generated at low lighting condition. Fig. 17-(b) illustrates the event generation rate under varying speeds and distinct lighting conditions and the maximum event processing rate of the different corner detectors. The graphs emphasize that the number of generated events in both lighting conditions are within the processing limits of the different corner detectors. Fig. 17-(c) illustrates the rate at which the corners are generated by four different corner detectors under different speed and illumination. Compared to the events generated under adequate lighting condition, Arc, eFast, and E-Harris consistently generate more events under low lighting and varying speeds. Only the FA-Harris generates less

number of corner events under low lighting and varying speeds.

For all the considered cases, the event generation rate is lower than the maximum event processing rate of the corner detectors and the rate at which the corners are generated is lower than the maximum event rate of the filter and tracker reported in Tables 6 and 7; which indicates the real-time capability of EBVS for high speed servoing applications. In cases where higher event processing rates are required, faster event-based corner detection methods such as eFAST, FA-Harris, and ARC can be used in place of the e-harris algorithm without the need for any additional changes to the EBVS pipeline. However, this will have an impact on the visual servoing performance as discussed in III-C. In this work, we used the raw event-stream for corner detection. However, by applying a suitable event filter [41] prior to corner detection, the computational performance of the EBVS could be further improved.

IV. CONCLUSION AND FUTURE WORK

In this work, we presented a purely event-based visual servoing method to detect, track and match object features in an uncluttered scene and a control strategy to explore, reach, and align the end-effector with the target object to achieve pick and place tasks. In particular, we devised three surface layers of active events for performing operations on the evolving temporal data in the event camera pixel-space to acquire robust and efficient visual event information and drive the robot to achieve EBVS. To study the proposed method, a commercial robot manipulator, a custom-made vacuum gripper, and an event camera in an eye-in-hand configuration were utilized and constrained to perform visual servoing in a 2D plane and execute top-down object grasps.

A comprehensive experimental study was conducted, which validates the use of EBVS for manipulation tasks in a wide variety of settings including: high speed operations, varying illumination conditions, static and dynamic objects, and different object geometries. In all the conducted experiments, EBVS was able to guide the robot to a precise grasp without the need for retuning or adaptation. Furthermore, each component of the EBVS pipeline was analyzed, taking into consideration state-of-the-art event processing techniques. The results of EBVS were benchmarked against a conventional frame-based solution and demonstrate the effectiveness of EBVS in solving the blurring, speed, and lighting challenges commonly faced in visual servoing applications. The findings of this paper identify neuromorphic vision as a suitable candidate to meet the evolving industrial requirements for vision-based robot control.

In our future work, we plan to advance the EBVS pipeline by including: (1) event-based object recognition module to segment multiple objects in a cluttered scene using event data and recognise objects using event-driven machine/deep learning methods. (2) Intelligent control module to generate optimal motion plans and robustly track errors for quick convergence along with stability proof.

ACKNOWLEDGMENT

The authors would like to thank the lab engineers, Mr. Pradeep George from the Aerospace Research and Innovation Center (ARIC) and Mr. Rajesh Ganithi from the mechanical engineering department at Khalifa University for their logistic and technical assistance with the experiments. Moreover, they would also like to thank the reviewers for their comments and valuable suggestions which helped to improve the quality of the article.

(Rajkumar Muthusamy and Abdulla Ayyad contributed equally to this work.)

REFERENCES

[1] F. Chaumette, S. Hutchinson, and P. Corke, "Visual servoing," in *Springer Handbook of Robotics*. Berlin, Germany: Springer, 2016, pp. 841–866.
[2] D. Kragic and H. I. Christensen, "Survey on visual servoing for manipulation," *Comput. Vis. Act. Perception Lab., Fiskartorpsv*, vol. 15, p. 2002, Jan. 2002.
[3] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 651–670, Oct. 1996.
[4] W. S. Barbosa, M. M. Gioia, V. G. Natividade, R. F. Wanderley, M. R. Chaves, F. C. Gouvea, and F. M. Gonçalves, "Industry 4.0: Examples of the use of the robotic arm for digital manufacturing processes," *Int. J. Interact. Design Manuf.*, vol. 14, no. 4, pp. 1–7, 2020.
[5] W. Zhang, J. Mei, and Y. Ding, "Design and development of a high speed sorting system based on machine vision guiding," *Phys. Procedia*, vol. 25, pp. 1955–1965, Jan. 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1875389212007511
[6] O. Araar and N. Aouf, "Visual servoing of a quadrotor UAV for autonomous power lines inspection," in *Proc. 22nd Medit. Conf. Control Autom.*, Jun. 2014, pp. 1418–1424.
[7] A. Ahmadi, L. Nardi, N. Chebrolu, and C. Stachniss, "Visual servoing-based navigation for monitoring row-crop fields," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 4920–4926.
[8] P. St, F. Torres, and C. Díaz, "Assembly/disassembly strategies for service applications," *IFAC Proc. Volumes*, vol. 41, no. 2, pp. 15805–15810, 2008.
[9] R. Barth, J. Hemming, and E. J. van Henten, "Design of an eye-in-hand sensing and servo control framework for harvesting robotics in dense vegetation," *Biosyst. Eng.*, vol. 146, pp. 71–84, Jun. 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1537511015001816
[10] C. Molnar, T. D. Nagy, R. N. Elek, and T. Haidegger, "Visual servoing-based camera control for the da vinci surgical system," in *Proc. IEEE 18th Int. Symp. Intell. Syst. Informat. (SISY)*, Sep. 2020, pp. 107–112.
[11] J. Garcia, A. Rodriguez, J. Estremera, B. Santamaria, D. Gonzalez, and M. Armendia, "Visual servoing and impedance control in robotic manipulators for on-orbit servicing," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, vol. 1, Sep. 2020, pp. 734–741.
[12] A. Cherubini, F. Chaumette, and G. Oriolo, "An image-based visual servoing scheme for following paths with nonholonomic mobile robots," in *Proc. 10th Int. Conf. Control, Autom., Robot. Vis.*, Dec. 2008, pp. 108–113.
[13] T. F. Gonçalves, J. R. Azinheira, and P. Rives, "Vision-based autonomous approach and landing for an aircraft using a direct visual tracking method," in *Proc. ICINCO-RA*, 2009, pp. 94–101.
[14] D. Zheng, H. Wang, J. Wang, S. Chen, W. Chen, and X. Liang, "Image-based visual servoing of a quadrotor using virtual camera approach," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 2, pp. 972–982, Apr. 2017.
[15] S. Krupinski, G. Allibert, M.-D. Hua, and T. Hamel, "An inertial-aided homography-based visual servo control approach for (almost) fully actuated autonomous underwater vehicles," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1041–1060, Oct. 2017.
[16] N. Inaba, M. Oda, and M. Hayashi, "Visual servoing of space robot for autonomous satellite capture," *Trans. Jpn. Soc. Aeronaut. SPACE Sci.*, vol. 46, no. 153, pp. 173–179, 2003.
[17] M. Gridseth, C. P. Quintero, R. Fomena, O. Ramirez, and M. Jagersand, "Bringing visual servoing into real world applications," in *Proc. Hum. Robot Collaboration Workshop RSS*, vol. 13, 2013, pp. 1–6.
[18] B. Huang, M. Ye, S.-L. Lee, and G.-Z. Yang, "A vision-guided multi-robot cooperation framework for learning-by-demonstration and task reproduction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 4797–4804.

[19] G. Gallego, T. Delbruck, G. M. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, p. 1, Jul. 2020.

[20] P. Corke, *Visual Control of Robots: High-Performance Visual Servoing* (Robotics and Mechatronics Series), vol. 2. New York, NY, USA: Research Studies Press, 1996. [Online]. Available: https://eprints.qut.edu.au/34313/

[21] X. Wang, G. Fang, K. Wang, X. Xie, K.-H. Lee, J. D. L. Ho, W. L. Tang, J. Lam, and K.-W. Kwok, "Eye-in-hand visual servoing enhanced with sparse strain measurement for soft continuum robots," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2161–2168, Apr. 2020.

[22] A. Vanarse, A. Osseiran, and A. Rassau, "A review of current neuromorphic approaches for vision, auditory, and olfactory sensors," *Frontiers Neurosci.*, vol. 10, p. 115, Mar. 2016.

[23] G. Indiveri and R. Douglas, "Neuromorphic vision sensors," *Science*, vol. 288, no. 5469, pp. 1189–1190, May 2000.

[24] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, "Analysis and observations from the first Amazon picking challenge," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 1, pp. 172–188, Jan. 2018.

[25] W. G. Pence, F. Farelo, R. Alqasemi, Y. Sun, and R. Dubey, "Visual servoing control of a 9-DoF WMRA to perform ADL tasks," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 916–922.

[26] M. Laiacker, F. Huber, and K. Kondak, "High accuracy visual servoing for aerial manipulation using a 7 degrees of freedom industrial manipulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 1631–1636.

[27] A. Plebe and G. Grasso, "Localization of spherical fruits for robotic harvesting," *Mach. Vis. Appl.*, vol. 13, no. 2, pp. 70–79, Nov. 2001.

[28] S. S. Mehta and T. F. Burks, "Adaptive visual servo control of robotic harvesting systems," *IFAC-PapersOnLine*, vol. 49, no. 16, pp. 287–292, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896316316160

[29] L. Cui, H. Wang, X. Liang, J. Wang, and W. Chen, "Visual servoing of a flexible aerial refueling boom with an eye-in-hand camera," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Jan. 8, 2020, doi: 10.1109/TSMC.2019.2957992.

[30] H. Wang, B. Yang, Y. Liu, W. Chen, X. Liang, and R. Pfeifer, "Visual servoing of soft robot manipulator in constrained environments with an adaptive controller," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 1, pp. 41–50, Feb. 2017.

[31] A. Taherian, A. H. Mazinan, and M. Aliyari-Shoorehdeli, "Image-based visual servoing improvement through utilization of adaptive control gain and pseudo-inverse of the weighted mean of the Jacobians," *Comput. Electr. Eng.*, vol. 83, May 2020, Art. no. 106580. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790619319159

[32] G. Dong and Z. H. Zhu, "Position-based visual servo control of autonomous robotic manipulators," *Acta Astronautica*, vol. 115, pp. 291–302, Oct. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576515002313

[33] A. G. Eguıluz, J. P. Rodrıguez-Gómez, J. R. Martınez-de Dios, and A. Ollero, "Asynchronous event-based line tracking for time-to-contact maneuvers in UAS," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5978–5985.

[34] R. Muthusamy, X. Huang, Y. Zweiri, L. Seneviratne, and D. Gan, "Neuromorphic event-based slip detection and suppression in robotic grasping and manipulation," *IEEE Access*, vol. 8, pp. 153364–153384, 2020.

[35] F. Baghaei Naeini, D. Makris, D. Gan, and Y. Zweiri, "Dynamic-vision-based force measurements using convolutional recurrent neural networks," *Sensors*, vol. 20, no. 16, p. 4469, Aug. 2020.

[36] X. Huang, R. Muthusamy, E. Hassan, Z. Niu, L. Seneviratne, D. Gan, and Y. Zweiri, "Neuromorphic vision based contact-level classification in robotic grasping applications," *Sensors*, vol. 20, no. 17, p. 4724, Aug. 2020.

[37] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988, vol. 15, no. 50, pp. 10–5244.

[38] V. Vasco, A. Glover, and C. Bartolozzi, "Fast event-based Harris corner detection exploiting the advantages of event-driven cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4144–4149.

[39] E. Mueggler, C. Bartolozzi, and D. Scaramuzza, "Fast event-based corner detection," in *Proc. 28th Brit. Mach. Vis. Conf. (BMVC)*. Zürich, Switzerland: Univ. of Zurich, 2017, pp. 1–11.

[40] J. Manderscheid, A. Sironi, N. Bourdis, D. Migliore, and V. Lepetit, "Speed invariant time surface for learning to detect corner points with event-based cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10237–10246.

[41] I. Alzugaray and M. Chli, "Asynchronous corner detection and tracking for event cameras in real time," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3177–3184, Oct. 2018.

[42] R. Li, D. Shi, Y. Zhang, K. Li, and R. Li, "FA-Harris: A fast and asynchronous corner detector for event cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 6223–6229.

[43] D. Gehrig, H. Rebecq, G. Gallego, and D. Scaramuzza, "Eklt: Asynchronous photometric feature tracking using events and frames," *Int. J. Comput. Vis.*, vol. 128, pp. 1–18, Mar. 2020.

[44] U. Robots. (2020). *Universal Robot UR10*. Accessed: Aug. 31, 2020. [Online]. Available: https://www.universal-robots.com/products/ur10-robot/

[45] C. Tomasi and T. Kanade, "Detection and tracking of point features," School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CS-91-132, 1991.

[46] A. Khodamoradi and R. Kastner, "O(N)-space spatiotemporal filter for reducing noise in neuromorphic vision sensors," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 1, pp. 15–23, Jan./Mar. 2018.

[47] V. Padala, A. Basu, and G. Orchard, "A noise filtering algorithm for event-based asynchronous change detection image sensors on TrueNorth and its implementation on TrueNorth," *Frontiers Neurosci.*, vol. 12, p. 118, Mar. 2018. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnins.2018.00118

[48] R. W. Baldwin, M. Almatrafi, V. Asari, and K. Hirakawa, "Event probability mask (EPM) and event denoising convolutional neural network (EDnCNN) for neuromorphic cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1698–1707.

RAJKUMAR MUTHUSAMY received the B.E. degree in electrical and electronics engineering from Anna University, India, in 2009, the M.S. degree in electrical engineering with a gold medal for academic excellence from Yuan Ze University, Taiwan, in 2013, and the D.Sc.(Tech) degree in automation, systems and control engineering from Aalto University, Finland, in 2018. After that, he contributed to the development of self-driving vehicles, as a Senior Robotics Engineer at Sensible4 Oy, Finland. Since December 2018, he has been a Postdoctoral Fellow at Khalifa University, United Arab Emirates. His current research activities are in the area of neuromorphic sensing, robotic grasping and manipulation, soft grippers, compliant manipulators, vision-based tactile sensing, visual servoing, machine learning, and intelligent control. His research interests include AI, autonomous vehicles and systems, assistive, collaborative, and general robotics. He is one of the recipient of the Finnish Engineering Award 2020.

ABDULLA AYYAD (Member, IEEE) received the M.Sc. degree in electrical engineering from The University of Tokyo, in 2019 where he conducted research with the Spacecraft Control and Robotics Laboratory. He is currently a Research Associate with the Khalifa University Center for Autonomous Robotic Systems (KUCARS) working on several robot autonomy projects. His current research interest includes the application of AI in the fields of perception, navigation, and control.

MOHAMAD HALWANI received the B.Sc. degree in mechanical engineering from United Arab Emirates University, in 2018. He is currently pursuing the M.Sc. degree in mechanical engineering with Khalifa University, United Arab Emirates. His research interest includes the application of dynamic and active vision sensors in the fields of visual servoing, robotic manipulation, and grasping.

DEWALD SWART received the M.Sc. degree in electrical engineering from the University of Stellenbosch, in 2012, where he conducted research with the Electronics Systems Laboratory. He is currently working as a Senior Development Engineer with the Research and Development, Strata Manufacturing. He focuses on the deployment of robotics on the shop floor to automate conventional assembly procedures.

DONGMING GAN received the dual Ph.D. degree in robotics and mechanical engineering from King's College London and the Beijing University of Posts and Telecommunications. He is currently an Assistant Professor with the School of Engineering Technology, Purdue University, West Lafayette, IN, USA. His main research interests include robotics, mechanism and machine theory with focus on design, modeling, control, and development of intelligent reconfigurable robotic systems, compliant robot manipulators, and flexible light-weight wearable devices for assisting humans in manufacturing, health care, and domestic human–robot co-existing scenarios with safe physical interactions and collaborations. He is also an Associate Editor of *Mechanism and Machine Theory*, IMechE Part C: *Journal of Mechanical Engineering Science*, a Topic Editor of *Mechanical Sciences*, a Review Editor on the Editorial Board of *Bionics and Biomimetics* (Frontiers), and a Guest Editor in *Applied Bionics and Biomechanics*. He is an Elected Committee Member of the ASME DED Mechanism and Robotics Committee and a member of ASME. He has served on program committees and symposiums of several international conferences, including IEEE Cyber 2019, IEEE/ASME ReMAR 2012-2021, Parallel 2014/2020, and ASME IDETC 2012-2020.

LAKMAL SENEVIRATNE received the B.Sc.(Eng.) and Ph.D. degrees in mechanical engineering from King's College London (KCL), London, U.K. He is currently a Professor in mechanical engineering and the Director of the Robotics Institute, Khalifa University. He is also an Emeritus Professor with King's College London. He has published over 300 refereed research articles related to these topics. His research interests include robotics and autonomous systems.

YAHYA ZWEIRI received the Ph.D. degree from King's College London, in 2003. He is currently the School Director with the Research and Enterprise, Kingston University London, U.K. He is also an Associate Professor with the Department of Aerospace, Khalifa University, United Arab Emirates. He was involved in defense and security research projects in the last 20 years with the Defence Science and Technology Laboratory, King's College London, and King Abdullah II Design and Development Bureau, Jordan. He has published over 100 refereed journal and conference papers, and filed six patents in USA and U.K., in unmanned systems field. His central research interests include interaction dynamics between unmanned systems and unknown environments by means of deep learning, machine intelligence, constrained optimization, and advanced control.