Journal of Intelligent Manufacturing (2022) 33:593–615
https://doi.org/10.1007/s10845-021-01887-9

Real-time grasping strategies using event camera

Xiaoqian Huang[1] · Mohamad Halwani[1] · Rajkumar Muthusamy[2] · Abdulla Ayyad[1,3] · Dewald Swart[4] · Lakmal Seneviratne[1] · Dongming Gan[5] · Yahya Zweiri[1,6]

Received: 16 July 2021 / Accepted: 23 November 2021 / Published online: 10 January 2022

Abstract
Robotic vision plays a key role for perceiving the environment in grasping applications. However, the conventional framed-based robotic vision, suffering from motion blur and low sampling rate, may not meet the automation needs of evolving industrial requirements. This paper, for the first time, proposes an event-based robotic grasping framework for multiple known and unknown objects in a cluttered scene. With advantages of microsecond-level sampling rate and no motion blur of event camera, the model-based and model-free approaches are developed for known and unknown objects' grasping respectively. The event-based multi-view approach is used to localize the objects in the scene in the model-based approach, and then point cloud processing is utilized to cluster and register the objects. The proposed model-free approach, on the other hand, utilizes the developed event-based object segmentation, visual servoing and grasp planning to localize, align to, and grasp the targeting object. Using a UR10 robot with an eye-in-hand neuromorphic camera and a Barrett hand gripper, the proposed approaches are experimentally validated with objects of different sizes. Furthermore, it demonstrates robustness and a significant advantage over grasping with a traditional frame-based camera in low-light conditions.

Keywords Neuromorphic vision · Model-based grasping · Model-free grasping · Multi-object grasping · Event camera

Introduction

Robots equipped with grippers has become increasingly popular and important for grasping tasks in the industrial field, because they provide the industry with the benefit of cutting manufacturing time while improving throughput. Especially in the 4th industrial revolution, the desire for robots that can perform multiple tasks is significant. Assisted by vision, robots are capable to perceive the surrounding environment such as the attributes and locations of the grasping targets. The vision-based robotic grasping system can be categorized along various criteria (Kleeberger et al. 2020). Generally, it can be summarized into analytic and data-driven methods depending on the analysis of the geometric properties of objects (Bohg et al. 2013; Sahbani et al. 2012). Moreover, according to whether or not building up the object's model,

✉ Xiaoqian Huang
xiaoqian.huang@ku.ac.ae

Mohamad Halwani
100053800@ku.ac.ae

Rajkumar Muthusamy
rajkumar.muthusamy@dubaifuture.gov.AE

Abdulla Ayyad
abdulla.ayyad@ku.ac.ae

Dewald Swart
ASwart@strata.ae

Lakmal Seneviratne
lakmal.seneviratne@ku.ac.ae

Dongming Gan
dgan@purdue.edu

Yahya Zweiri
yahya.zweiri@ku.ac.ae

[1] Khalifa University Center for Autonomous Robotic Systems (KUCARS), Khalifa University, Abu Dhabi, UAE

[2] Dubai Future Labs, Dubai, UAE

[3] Aerospace Research and Innovation Center (ARIC), Khalifa University of Science and Technology, Abu Dhabi, UAE

[4] Research and Development, Strata Manufacturing PJSC, Al Ain, UAE

[5] School of Engineering Technology, Purdue University, West Lafayette, IN 47907, USA

[6] Department of Aerospace Engineering, Khalifa University, Abu Dhabi, UAE

the vision-based grasping can be divided into model-based and model-free approaches (Zaidi et al. 2017; Kleeberger et al. 2020). Model-based approaches are mostly used for known objects due to the requirement of object's prior knowledge. Model-free methods are more flexible for both known and unknown objects by learning geometric parameters of objects based on vision. Lots of standard vision-based robotic grasping systems are explored for many applications, such as garbage sorting (Zhihong et al. 2017), construction (Asadi et al. 2021) and human interaction (Úbeda et al. 2018).

With the development of neuromorphic vision in grasping field, the robotic grasping system can be newly categorized into standard vision-based and neuromorphic vision-based approaches along the different perception methods. Standard vision sensors continue to sense and save picture data as long as the power is on, resulting in significant power consumption and large data storage. Moreover, the grasping quality would be affected severely due to the poor perceiving quality, such as the motion blur and poor observation in low-light condition. For example, it is proved that the quality of the picture taken by standard camera will be affected by the moving speed of the conveyor belt in production line (Zhang and Cheng 2019), due to the motion blur and low sampling rate of the conventional RGB camera. In addition, the actuating speed of the electrical gripper is generally over 100 ms in robotic grasping tasks. Meanwhile, standard cameras commonly have a frame rate of less than 100 per second. Even for the high-speed frame-based camera, the frequency is also generally less than 200 frames per second with a high consumption of both power and storage. Furthermore, computing the complex algorithm for vision processing algorithm will take additional time to slower the grasping process from the vision resource. So the acceleration of vision acquirement and process will contribute to the grasping efficiency. To improve the reacting speed of vision-based grasping, a faster detecting helps to reserve more time for the gripper's actuation. For instance, a high sampling rate will assists the robotic system by providing adequate time take actions to prevent slip in a closed-loop control system. Distinct to the conventional frame-based camera, individual events are triggered asynchronously by event camera with a micro second-level sampling rate (Gallego et al. 2019). Therefore, the unique property of event camera becomes indispensable to improve the performance for grasping tasks.

Neuromorphic vision sensors (Indiveri and Douglas 2000) are inspired by biological systems such as fly eyes, which can sense data in parallel and asynchronously in real time. Initially, the neuromorphic vision sensor was known as silicon retina only utilized for computer vision and robotics researches (Etienne-Cummings and der Spiegel 1996). Then it becomes known as an event-based camera because it captures per-pixel illumination changes as events (Gallego et al. 2019). In contrast to traditional frame-based vision sensors, event-driven neuromorphic vision sensors have low latency, high dynamic range and high temporal resolution. The event camera functions as a neuromorphic vision sensor with the ability to asynchronously measure brightness changes per pixel. It results a stream of events which has microsecond-level time stamp, spatial address, and polarity referring the sign of brightness changes (Gallego et al. 2019). Hence, utilizing events-based segmentation and grasping provides superiorities of no motion blur, low-light operation, a faster response and higher sampling rate. It introduces new opportunities as well as challenges for neuromorphic vision processing and event-based robotic grasping. Recently, the event-based camera has been utilized in a growing number of applications such as object detection and tracking (Mitrokhin et al. 2018), 3D reconstruction (Zhou et al. 2018), and simultaneous localization and mapping (Milford et al. 2015). However, only few works used event camera to address gripping tendencies, such as dynamic force estimation (Naeini et al. 2019, 2020), grasping quality evaluation (Huang et al. 2020), and incipient slippages detection and suppression (Rigi et al. 2018; Muthusamy et al. 2020).

The key tasks in robotic vision-based grasping can be summaries as object localization, object pose estimation, grasp generation, and motion planning (Du et al. 2019). In this work, we assume no obstacles exist between objects and the gripper, so the first three tasks are addressed in the real-time grasping framework. Object localization aims to obtain the target's position, commonly involving object detection and instance segmentation. Using object detection, the object will be classified and located by the bounding box. With the development of deep learning, CNN (Chen and Ling 2019), YOLO (Redmon et al. 2016) and Faster R-CNN (Ren et al. 2015) are popularly utilized for the object detection. Differently, object instance segmentation is pixel-wise for each individual object. Instance segmentation can be achieved by machine learning based clustering methods, such as KNN (Peterson 2009), K-means (Likas et al. 2003), SVM (Cortes and Vapnik 1995) and Mean shift clustering (Fukunaga and Hostetler 1975). In these methods, mean shift clustering is non-parametric method that can be applied without prior knowledge. Local and global masks are another technique generally used in deep learning based instance segmentation, such as YOLACT-the first methods attempting real-time instance segmentation Bolya et al. (2019) and SOLO-the segmentation objects by locations (Wang et al. 2020). However, mostly all the segmentation techniques are applied to standard vision like RGB images and videos. In this work, we develop Multiple-object Event-based Mean-Shift (MEMS) instance segmentation for asynchronous event data in model-free approach, and utilize event-based multi-view clustering method in model-based approach. As the other part of grasping, the object pose can be estimated by 3D point cloud (Zhou et al. 2016) and image coordinate (Hu et al. 2020).

But they are also mostly applied on the standard vision-based grasping, which suffers from motion blur and latency. Grasp generation refers to estimate the grasping pose of gripper, which can be divided into 2D planar grasp and 6DoF Grasp (Zhou and Hauser 2017) based on standard vision. For event-based grasp generation, the author in Bin Li et al. (2020) constructed Event-Grasping dataset by annotating the best and worst grasp pose using LED's flickering marker. But the frequency is constrained up to 1 KHz due to the limitation of LED frequency. Based on the dataset, the grasping angle is learned via deep learning as a "good" or "bad" classification problem. In other words, the gripper is required to adjust pose until the feedback of pose classification achieves "good" class or the stop criteria is reached. Therefore, the grasp pose generation is not efficient since it cannot provide the proper grasp pose directly.

According to these three tasks of grasping, the neuromorphic eye-in-hand 2D visual servoing approach for single object grasping was developed (Muthusamy et al. 2021) in our previous work, which adopts an event-based corner detector, a heatmap based corner filter, an event-based gripper alignment strategy. By comparison with the conventional frame-based image-based visual servoing, our previous work shows a superior performance on both time efficiency and computation under different operating speeds and lighting conditions. To improve our prior work for the multiple 3D objects grasping, there are several challenges including the event-based segmentation of multiple objects, the event-based Visual Servoing (EVS) method adopting depth estimation, and the grasp generation according to segmented information represented by spatial-temporal events. Therefore, two event-based approaches for multiple objects grasping are developed in this paper, involving the Model-Based Approach (MBA) and Model-Free Approach (MFA). Event-based segmentation, event-based visual servoing adopting depth estimation, and grasping generation using Barrett hand are developed. In addition, we quantitatively evaluate and compare the performance of these two approaches experimentally. The primary contributions of the paper are summarized below:

1. We devise an event-based grasping framework for robotic manipulator with neuromorphic eye-in-hand configuration. In particular, we propose a model-based and model-free approach for grasping objects in a cluttered scene.
2. We study the computational performance of the two event-based grasping approaches and assess their applicability for the real-time and evolving industrial requirements. In particular, we evaluate the grasping framework using a robotic manipulator with a neuromorphic eye-in-hand configuration, a robotic end-effector of Barrett hand, and objects of various sizes and geometries.
3. We demonstrate the multiple-object segmentation, grasping and manipulation for multiple-object pick and place applications. In factory automation, the completely event-based strategies can boost production speed.

The remainder of this work is organized as follows: "Overview of the proposed approaches" section introduces the working principle and the data property of event-based camera. "Model-based grasping approach" and "Model-free event-based multiple objects grasping" section elaborate our proposed method for model-based and model-free multi-object grasping using neuromorphic vision, respectively. The validation experiments and results analysis are described in "Experimental validation on multiple-object grasping" section. Based on the experimental performance, two approaches are discussed and their pros and cons are summarized in "Discussion" section. Then the conclusion and future work are presented in the last section. The video of demonstrations is in the link: https://youtu.be/NBHkchnQfLY.

Overview of the proposed approaches

This section introduces the events data and describes the overall description of Model-Based Approach (MBA) and Model-Free Approach (MFA) using neuromorphic vision for robotic grasping, which are elaborated in "Model-based grasping approach" and "Model-free event-based multiple objects grasping" section respectively.

The pixels of event-based camera can respond to logarithmic brightness ($L = log(I)$) variations independently. Once the perceived logarithmic light intensity change exceeds the threshold C, the events will be generated at a pixel (x, y) at time t.

$$\begin{aligned} \triangle L(x, y, t) &= \mid L(x, y, t) - L(x, y, t - \triangle t) \mid \\ &= p * C \Big| C > 0, p \in \{+1, -1\} \end{aligned} \tag{1}$$

where Δt is the interval time between the current and the last event generated at the same pixel, ΔL represents the illumination change, and p describes positive $(+1)$ or negative (-1) polarity of events indicating the brightness increase or decrease. The stream of events has a microsecond temporal resolution with an event represented as $e = (x, y, t, p)$ (Gallego et al. 2019). In this research, DAVIS 346 with a high dynamic range (140 dB), low-power consumption (10 mW) and 346×260 resolution will be used. The block diagram of two designed 3D grasping frameworks utilizing event camera is briefly summarized in Fig. 1.

The gray blocks indicate the common processes of the two approaches, and the green and blue blocks repre-

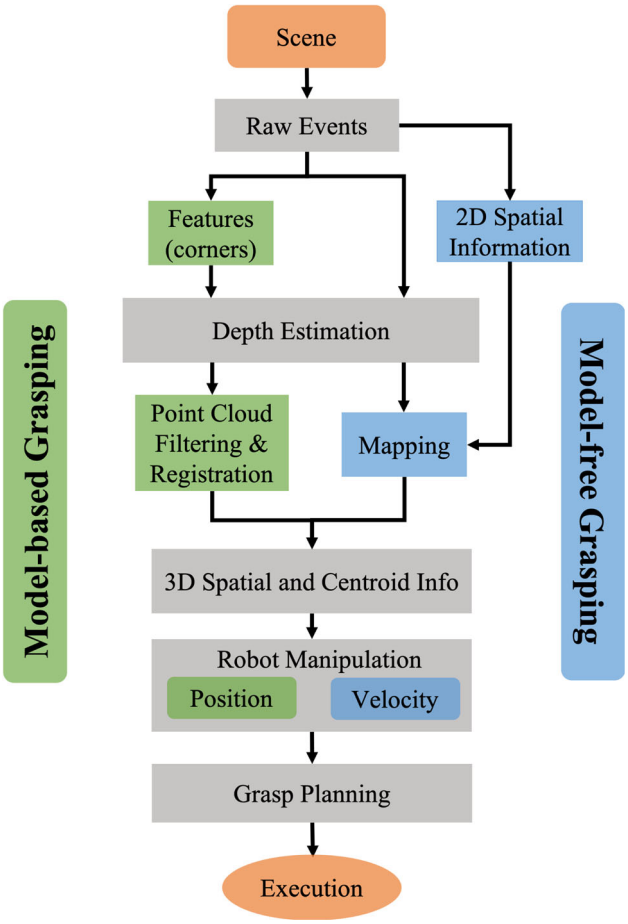


Fig. 1 Model-based grasping (green and gray blocks) and model-free grasping (blue and gray blocks) summary (Color figure online)

sent processes that belong to model-based and model-free approaches, respectively. From the Fig. 1, both approaches acquire the 3D spatial and centroid information of individual objects from raw event and depth estimation. Then execute robot manipulation and grasping according to the obtained object's information and grasp planning. The most significant differences contain the segmentation and robotic manipulation methods. The model-based approach segments objects based on the 3D point cloud of features, but the model-free approach utilizes machine learning technique to segment each instance. Moreover, position-based and velocity-based visual servoing are applied for robotic manipulation in the proposed model-based and model-free approach, respectively.

Model-based grasping approach

This section explains an event-based objects grasping approach that uses an inexact-sized model fitting to estimate the pose of the objects to be manipulated. An event camera mounted on robotic manipulator in an eye-in-hand configuration is used. Using pure events from an object in the scene, high-level corner features are extracted using e-Harris corner detector (Vasco et al. 2016), the use of this detector was justified in our previous work (Muthusamy et al. 2021) which developed 2D event-based visual servoing approach for single object grasping. As we are using an eye-in-hand setting with a monocular camera, we are missing the depth information of the objects we want to grasp. In the following subsection we will introduce the Neuromorphic Multi-View Localization approach used in this study for the localization of the objects in the environment. The overall model-based grasping framework is shown in Fig. 2.

Neuromorphic multi-View localization

Let us consider a moving calibrated event camera observing a rigid static object in the scene. The movement of the camera generates events on the sensor plane of the camera. Event cameras uses the same optica as traditional cameras, so the pinhole projection equation of a 3D point in the environment can be still used. Figure 3 shows the pinhole projection, a 3D point $\mathbf{P} = [x, y, z]$ is mapped into a 2D point $\mathbf{p} = [u, v]$ on the camera sensor plane, which is expressed as:

$$\mathbf{z}\left[u, v, \mathbf{1}\right]^T = \quad \mathbf{K}\left[\mathbf{R}\ \mathbf{t}\right]\left[\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{1}\right]^T \tag{2}$$

where K is a 3×3 camera's intrinsic parameters and R and t are the relative pose between the camera and the object in the environment.

The most common approach that tackles objects localization using event-cameras is by using two or more event cameras with known fixed attachment between them, and sharing a common clock. This method requires to solve for the events correspondence among the two cameras and then localize the point feature in the environment. In our work we used an approach that considers only a single camera with a known camera trajectory. Event-Based Multi-View Stereo (EMVS) introduced in Rebecq and Gallego (2016), was used to estimate the exact pose of the object but utilizing only the high-level corner features and the information of the known trajectory of the event camera. The benefit of this approach is that it considers the asynchronous and sparse nature of the events generated by the event camera to warp them as rays through a discretized volume of interest called the Disparity Space Image (DSI). This method works directly in 3D space, so it is not requried to recover the 3D location by associating individual event to a particular 3D point. The event stream e_k generated by the event camera is the input point features that are warpped into the DSI as rays according to the viewpoint of the event camera at time t_k. Then, the number of rays passing through each voxel is counted and a 3D point is considered present or not in each voxel by voxel voting. The

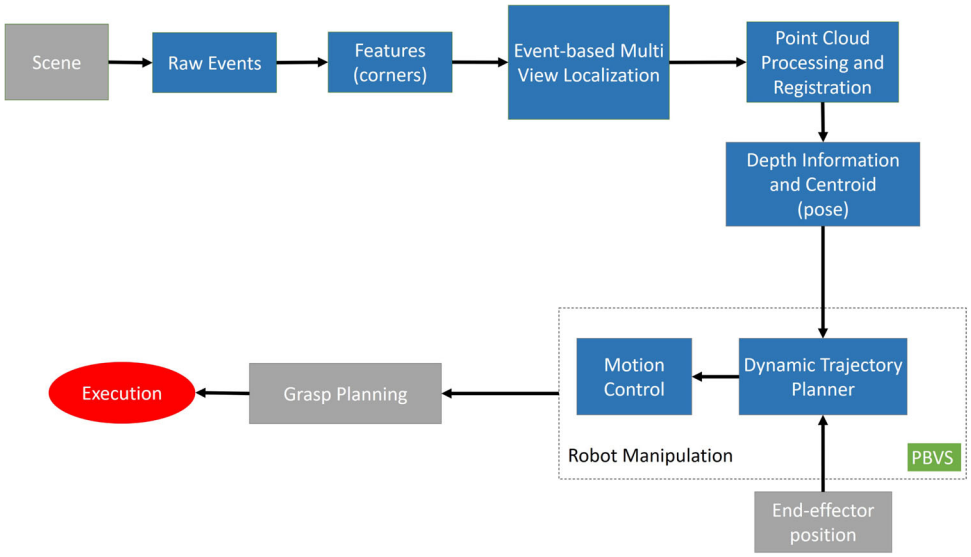


Fig. 2 Model-based multiple-object grasping framework

approach back-projects the events from their corresponding camera view to a reference camera view to find their exact projection location in depth plane Z_i. An event is back projected via two steps: first, it is mapped to the first depth plane Z_0 using the homography H_{Z_0} and then to its correspondent depth plane Z_i using the homography $H_{Z_i} H_{Z_0}^{-1}$, where:

$$H_{Z_i} \sim R + \frac{1}{Z_i} t \mathbf{e_3^T} \tag{3}$$

and

$$\mathbf{e_3} = (0, 0, 1) \tag{4}$$

The back projected viewing rays passing through the DSI are counted for each voxel. First the DSI is defined by the camera pixels and a pre-defined number of depth planes N_z, therefore, it has a size of $width \times height \times N_z$. The amount of viewing rays that intersect each voxel are stored as a score in the DSI:

$$f(\boldsymbol{X}) : V \subset \mathbb{R}^3 \tag{5}$$

where $\boldsymbol{X} = (X, Y, Z)^T$ represents voxel center. Finally, EMVS algorithm produces a semi-dense depth map by thresholding a confidence map $c(x, y)$, which represents the location and magnitude of the optimal score with the maximum value as follows:

$$f(X(x), Y(y), Z^*) =: c(x, y) \tag{6}$$

The depth map can be then converted to a point cloud.

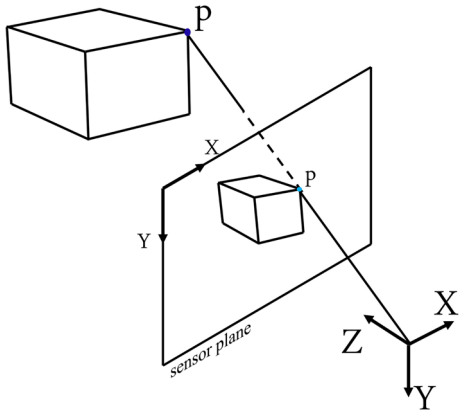


Fig. 3 Relative motion between the camera and the 3D object projects a point (event) to the camera sensor plane

Point cloud processing

The model-based multi-objects grasping is achieved by executing the following: point cloud downsampling, object clustering, model registration, and robot manipulation with grasp planning, which are explained in the following subsections respectively.

Point cloud downsampling

Post processing of the point cloud is performed to remove the outliers in the point cloud. Using a space partitioning data structure for organizing the points in a k-dimensional space in a tree structure (k–d tree), we used a nearest neighbour search to remove points that have a distance higher than a threshold. This helps removing isolated points which are most likely outliers.

Object clustering

Grasping multiple objects requires finding and segmenting the objects in the scene to individual object point clusters. Since our point clouds contain only objects corner points (i.e., maximum 8 points per object), we can use simple data clustering approaches without worrying about the execution speed of the clustering algorithm. We applied an euclidean cluster extraction method, implemented using point cloud library (PCL) (Rusu and Cousins 2011). This method divides an unorganized point cloud model into smaller parts to reduce the processing time. Same as the method used in "Point cloud downsampling" section, a tree data structure is used to subdivide the points and make use of the nearest neighbours to search for points that are within a sphere of radius equal to a threshold and adds them to a point cloud cluster $\{C_i\}_{i=1}^{N}$, where N is the number of objects detected.

Model registration

Because of camera visual constraints and objects geometrical constraints, not all object corners can be viewed considering the linear motion of the eye-in-hand, thus, this issue has to be solved to provide an exact model of the object to be grasped. Object registration aligns two point clouds to find the relative pose between the two point clouds in a global coordinate frame. Iterative Closest Point (ICP) offers a good solution to solve for the un-seen corners and performing model registration, but original ICP needs an exact model of the targeted object to find the transformation between the target model point cloud C_i and the source model point cloud P, thus, it does not handle the case of models with different scales. In addition, to increase the chance of good convergence and successful alignment using ICP, an initial rough estimate of the alignment is required to avoid converging in a local minima (Rusinkiewicz and Levoy 2001). However, finding a rough estimate requires finding feature descriptors to determine point-to-point correspondences, yet its a challenging problem since we operate on a small sized point clouds, and common feature based descriptors (i.e., spin image, PFH, DH, etc.) were designed for dense point clouds.

In our paper, we used an inexact model P (i.e., 8 corners relevant to a cube of length 1 m) to generalize our registration algorithm. According to Sankaranarayanan et al. (2007), the registration problem is divided into 4 steps:

1. Selection: select input point cloud.
2. Matching: estimate correspondences.
3. Rejection: filter to remove outliers.
4. Alignment: find optimal transformation by minimizing an error metric.

As discussed in "Object clustering" section and "Point cloud downsampling" section, the input point cloud is the clustered point cloud C_i after removing the isolated points. We used Singular Value Decomposition (SVD), to find the optimal transformation parameters (rotation R, translation t and scaling c) between the set of points $X = x_1, x_2, \ldots, x_n$ in cluster C_i and the set of points $Y = y_1, y_2, \ldots, y_n$ from the source model after solving the correspondence, where each set of points of m-dimensional space (i.e., 3D in our case). We apply the mean squared error for these two point sets to find the transformation parameters. The derivations and equations below are from Umeyama (1991).

$$e^2(R, t, c) = \frac{1}{n}\sum_{i=1}^{n} \|y_i - (cRx_i + t)\|^2 \tag{7}$$

where

$$\mu_x = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{8}$$

$$\mu_y = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{9}$$

$$\sigma_x^2 = \frac{1}{n}\sum_{i=1}^{n} \|x_i - \mu_x\|^2 \tag{10}$$

$$\sigma_y^2 = \frac{1}{n}\sum_{i=1}^{n} \|y_i - \mu_y\|^2 \tag{11}$$

$$\sum_{xy} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \mu_y)(x_i - \mu_x)^T \tag{12}$$

and let the SVD of Eq. 12 be UDV^T where D is a diagonal matrix of size m and

$$S = \begin{cases} I & \text{if } det\left(\sum_{xy}\right) \geq 0 \\ diag(1, 1, \ldots, 1, -1) & \text{if } det\left(\sum_{xy}\right) < 0. \end{cases} \tag{13}$$

where $\sum_{xy}$ is a covariance matrix of X and Y, μ_x and μ_y are mean vectors of X and Y, and σ_y^2 and σ_y^2 are variances around the mean vectors of X and Y, respectively. Hence, the optimal transformation variables can be computed as follows:

$$R = USV^T \tag{14}$$

$$t = \mu_y - cR\mu_x \tag{15}$$

$$c = \frac{1}{\sigma_x^2} tr(DS) \tag{16}$$

For mathematical proof of the equations you can review (Umeyama 1991).

Robot manipulation and grasp planning

The robot manipulation controller for the model-based object grasping is controlled by a Position Based Robot Controller (PBVS). The PBVS stage guides the end-effector towards the object features using the 6DoF pose estimate from the multi-view detection and the model registration stage explained in "Model registration" section. PBVS considers a known intial and final poses of the robot's end effector. The final pose can be pre-defined as in the detection step, or it is found by the EMVS and the point cloud processing stages, which would represent the object centroid and orientation angle. The desired joint angles vector of the robotic arm $\hat{\theta}$ is found using an inverse kinematic approach of the open-source Kinematic and Dynamics Library. Afterwards, we compute a trajectory for the joint angles on the Open Motion Planning Library, and a PID controller regulates each joint and tracks the joint angles.

Algorithm 1: Model-based grasping framework

Input: Events stream: position (x_i, y_i), timestamp t_i
Output: Objects centroid (X,Y,Z), Object Orientation Angle

```
Set starting point of the gripper P_0
Scan the scene and extract objects corners
Perform the event-based multi-view localization
Perform point cloud downsampling
Perform object Euclidean clustering
for each cluster do
    Perform model registration
    Extract object centroid
    Extract object orientation
    Navigate to object and orient the gripper
    Perform Grasp
```

Model-free event-based multiple objects grasping

Distinct from the model-based approach, the model-free approach acquires object information directly from the asynchronous event data. For 3D object detection and grasping, the depth information can be obtained by EMVS, and mapped into image coordinates for further visual servoing. Building on that, the velocity control will be utilized to manipulate the robot arm, ensuring the end-effector/gripper is aligned with the centroid of the object. While the grasp hypothesis is generated for the gripper according to the event stream, then grasping operation will be executed to enclasp the aiming object. Therefore, the designed event based multi-object grasping consists of three parts: segmentation, visual servoing, and grasping plan, which are explained in the following subsections respectively.

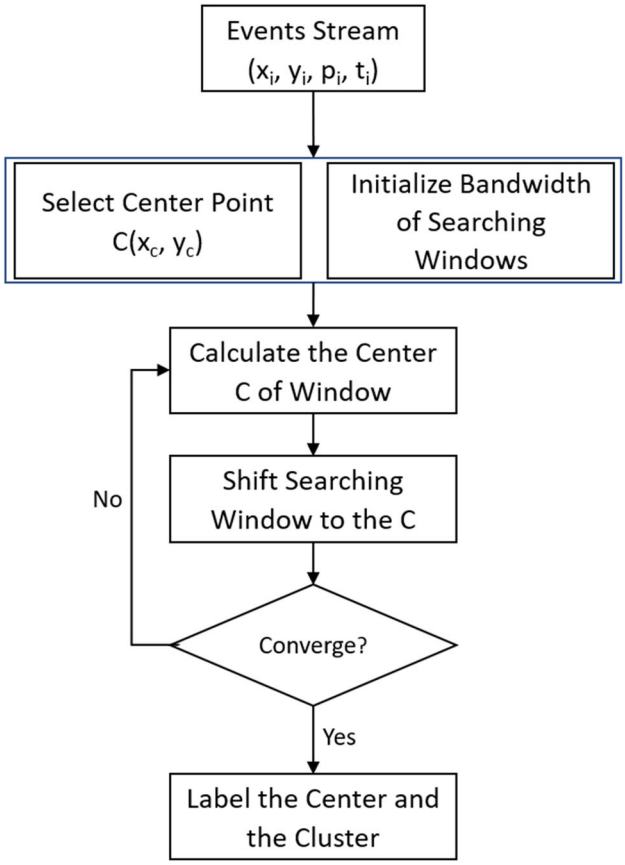


Fig. 4 Event-based mean shift clustering principle

Segmentation

In this work, the Mean Shift (MS) algorithm is employed for object segmentation due to its non-parametric character, which is developed based on the assumption that different clusters of data are of different probability density distributions (Fukunaga and Hostetler 1975). The working principle is that by computing the shifting vectors of one point and all neighbouring points in some range, the mean shift vector can be obtained including shifting magnitude and orientation. Then repeating calculating this mean shift vector until it converges. The main idea of object segmentation based on events data is visualized in Fig. 4.

The probability density distributions can be expressed as the Probability Density Function (PDF) shown in Eq. 17 (Fukunaga and Hostetler 1975):

$$P_x = \frac{1}{nh^2} \sum_{i=1}^{n} K \frac{(x - x_i)}{h^2} \tag{17}$$

where K is the kernel function applied to each data point, n denotes the number of data points, and h presents the bandwidth parameter which means the radius of the kernel.

For dealing with the non-linear datasets, the data is usually reflected into high dimension space by using kernel function $K(x)$. The most used kernel- Gaussian kernel is expressed in Eq. 18.

$$K_G(x) = e^{-\frac{x^2}{2\sigma^2}} \tag{18}$$

where σ is the bandwidth of the window. In mean shift procedure, each point has its own weight and bandwidth which can be an isotropic and diagonal matrix. To simplify the expression, the case that all points have the same and scalar bandwidth σ and the same weight $\frac{1}{nh^2}$ is considered most practically. In addition, the Gaussian kernel is mostly used in PDF because the bandwidth σ will be the only parameter required for MS clustering.

Suppose x is a point to be shifted and $N(x)$ are the sets of points near the point x. $D(x, x_i)$ is the distance from the point x to the point x_i, then the new position x' shifted from x is calculated as Eq. 19 (Fukunaga and Hostetler 1975).

$$x' = \frac{\sum_{x_i \subset N(x)} k(D(x, x_i)^2) x_i}{\sum_{x_i \subset N(x)} k(D(x, x_i)^2)} \tag{19}$$

The new position will keep updating until the MS vector is converged or the maximum iteration step is reached. The MS vector is represented as Eq. 20 (Fukunaga and Hostetler 1975):

$$V_x = \frac{\sum_{x_i \subset N(x)} k(D(x, x_i)^2) x_i}{\sum_{x_i \subset N(x)} k(D(x, x_i)^2)} - x \tag{20}$$

However, the standard MS algorithm only considers the spatial information of data points. For dealing with the asynchronous and sparse evens data, both spatial and temporal information is used for multiple objects in this research as Multi-object Event-based Mean Shift (MEMS). The main idea of instance segmentation based on events data is shown in Fig. 4. By repeatedly updating a given point with the mean of the shifting vectors with respect to all its neighbouring points within a specified range, the process will eventually converge to the distribution mode of the cluster to which the starting point belongs. Currently, MEMS algorithm is applied on 2D spatio-temporal events stream which are represented as (x_i, y_i, t_i), where (x_i, y_i) and t_i are spatial and temporal information respectively. The bandwidth of the searching window is initialized before processing. The spatial center point $c(x_i, y_i)$ will be randomly selected after obtaining events data. Then iterating the procedure of mean computing and shifting until it is converged.

Utilizing both spatial and temporal information, the event-based PDF and Gaussian kernel are expressed as Eqs. 21 and 22 (Barranco et al. 2018):

$$P_{\boldsymbol{x},t} = \frac{1}{n} \sum_{i=1}^{n} K([\boldsymbol{x}, t] - [\boldsymbol{x}_i, t_i]) \tag{21}$$

$$K([\boldsymbol{x}, t] - [\boldsymbol{x}_i, t_i]) = ck\left(\left|\left|\frac{[\boldsymbol{x}, t] - [\boldsymbol{x}_i, t_i]}{\sigma}\right|\right|\right)^2 = ce^{-\frac{(x-x_i)^2+(t-t_i)^2}{2\sigma^2}} \tag{22}$$

where $\boldsymbol{x}_i$ is a 2D vector representing the spatial coordinates, t_i is the time stamp of x_i, and c is the coefficient of Gaussian kernel which is equal to $\frac{1}{\sqrt{2\pi}\sigma}$. Underlying density $P_{\boldsymbol{x},t}$ is to find the modes of this density. The modes are located among the zeros of the gradient $\nabla P_{\boldsymbol{x},t} = 0$ and the mean shift procedure is an elegant way to locate these zeros without estimating the density. The $\nabla P_{\boldsymbol{x},t}$ is obtained as:

$$\nabla P_{\boldsymbol{x},t} = \frac{1}{n} \sum_{i=1}^{n} \nabla K([\boldsymbol{x}, t] - [\boldsymbol{x}_i, t_i]) \tag{23}$$

By substituting Eq. 22 into Eq. 23, $\nabla P_{\boldsymbol{x},t}$ is obtained as:

$$\nabla P_{\boldsymbol{x},t} = \frac{c}{n} \sum_{i=1}^{n} g_i \left[\frac{\sum_{i=1}^{n} [\boldsymbol{x}_i, t]\, g_i}{\sum_{i=1}^{n} g_i} - [\boldsymbol{x}, t]\right], g = -k' \tag{24}$$

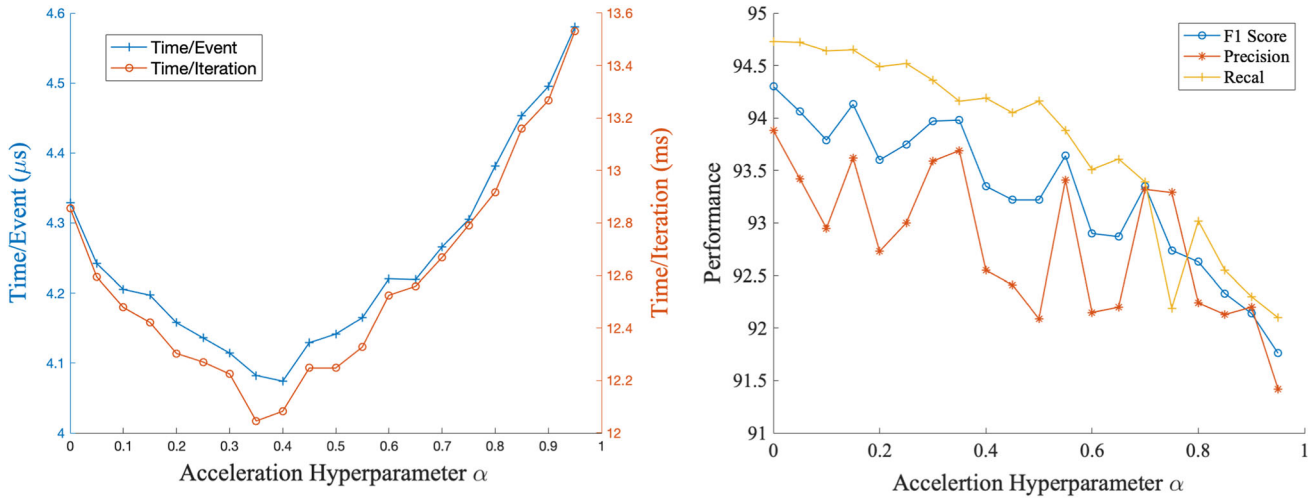
Then the MS vector can be expressed as Eq. 25, which is important for target localization and gripper manipulation.

$$\boldsymbol{m}([\boldsymbol{x}, t]) = \left[\frac{\sum_{i=1}^{n} [\boldsymbol{x}_i, t]\, g\left(\frac{||[\boldsymbol{x},t]-[\boldsymbol{x}_i,t_i]||^2}{h}\right)}{\sum_{i=1}^{n} g\left(\frac{||[\boldsymbol{x},t]-[\boldsymbol{x}_i,t_i]||^2}{h}\right)} - [\boldsymbol{x}, t]\right] \tag{25}$$

Running the MEMS on the event data obtained by the neuromorphic sensor, the execution time is 12.861 ms presenting a dramatic improvement of time efficiency compared to the mean shift algorithm for standard vision (754.977 ms). To further accelerate the segmenting speed of MEMS compared to the event-based mean shift algroithm in Barranco et al. (2018), we applied two strategies: soft speedup term and downsampling data illustrated in "Strategy 1" and "Strategy 2" sections, respectively. For robotic grasping tasks, both efficiency and accuracy are key aspects for evaluating the performance of MEMS. Therefore, the metric E-score is designed as the following equations:

$$E - score = \lambda_1 \cdot Ere + \lambda_2 \cdot Fre \tag{26}$$

where λ_1 and λ_2 are factors indicating the significance of efficiency and accuracy considered in the task, and the sum of



(a) Processing time **(b)** Performance

Fig. 5 Processing time and performance with varying acceleration hyperparameters α

λ_1 and λ_2 is constrained as 1 that $\lambda_1+\lambda_2 = 1$. Compared with the baseline when $\alpha = 0$ in strategy 1 or $\beta = 1$ in strategy 2, Ere (Eq. 27) and Fre (Eq. 28) represent the relative error of processing time per event and F1 score, respectively.

$$Ere = -\frac{T_e(\alpha or \beta) - T_e(\alpha = 0 or \beta = 1)}{T_e(\alpha = 0 or \beta = 1)} \cdot 100 \tag{27}$$

$$Fre = -\frac{F1(\alpha or \beta) - F1(\alpha = 0 or \beta = 1)}{F1(\alpha = 0 or \beta = 1)} \cdot 100 \tag{28}$$

where T_e and $F1$ are the processing time per event and *F1-score* correspondingly. Time efficiency is the main concern in this work, so the core contribution of MEMS is to accelerate the standard mean shift. Therefore, λ_1 and λ_2 are set as 0.6 and 0.4 respectively to assess the overall performance.

Strategy 1

In iterations of MEMS, each event will be shifted along the shifting vector with the magnitude. The speedup term is then added to calculate the final new positions as expressed in Eq. 29.

$$\boldsymbol{x}' = \left[\frac{\sum_{i=1}^{n} [\boldsymbol{x}_i, t]\, g\left(\frac{||[\boldsymbol{x},t]-[\boldsymbol{x}_i,t_i]||^2}{h}\right)}{\sum_{i=1}^{n} g\left(\frac{||[\boldsymbol{x},t]-[\boldsymbol{x}_i,t_i]||^2}{h}\right)} + \alpha \cdot \boldsymbol{m} \right] \tag{29}$$

where α is the acceleration coefficient that controls how much extra distance the shifted points move along the shift vector. While hyperparameter α is set properly, the procedure will be accelerated compared with the standard MS. We tested the mean shift speed for individual events and iteration using different α within 1 as shown in Fig. 5a.

As shown in Fig. 5a, the processing time occupied by each event and iteration demonstrates a similar pattern, that declines when $\alpha \leq 0.35$ and increases while $\alpha > 0.35$, and almost the smallest deviations are demonstrated when $\alpha = 0.35$. In other words, the efficiency of MEMS in event and iteration levels will be improved within some range of α. However, the MEMS will diverge when α reaches 1. Besides, the clustering accuracy of MEMS is assessed by precision, recall and F1 score as recorded in Fig. 5b. Although three metrics demonstrate slightly better when $\alpha = 0$, the average differences are only around 1%.

The overall evaluation considering both efficiency and accuracy is computed in Eq. 26. Figure 6 illustrates the relationship between E-score and α value, that our MEMS with strategy-1 performs better than the standard MS when $\alpha < 0.75$, especially at $\alpha = 0.35$.

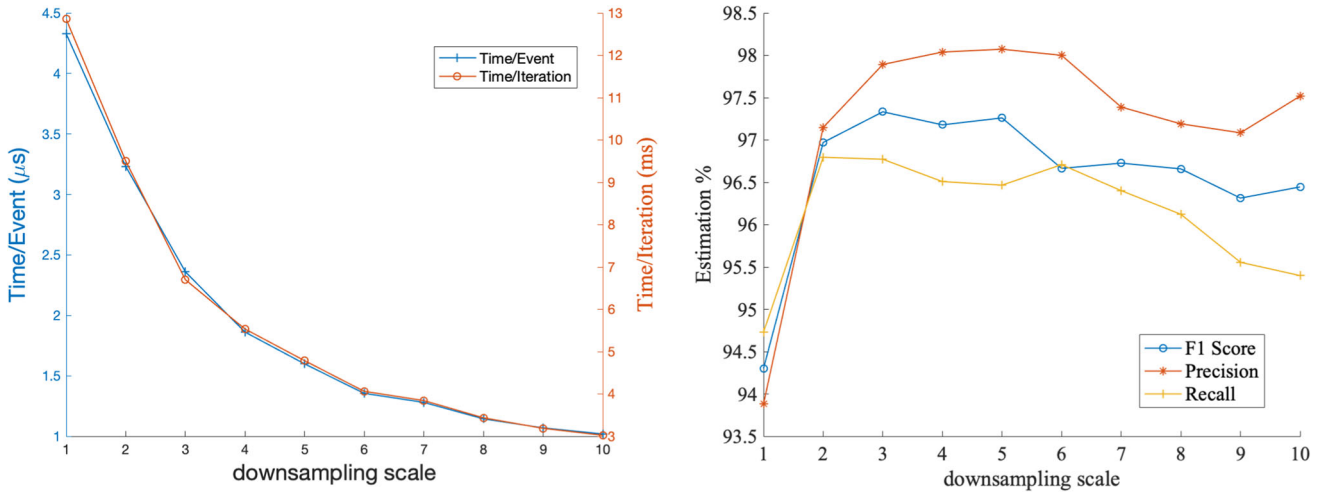
Strategy 2

To accelerate MEMS algorithm, one of the most intuitive approaches is to reduce the processing data. Building on that, the hyperparameter β is introduced to downsample the original events expressed in Eq. 30 evenly. The downsampled events are represented in Eq. 31, that the size will be reduced to $1/\beta$ of the original size.

$$\textit{Original events: } \mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_i\} \tag{30}$$

$$\textit{Downsampled events: } \mathbf{X}' = \left\{\mathbf{x}_1, \mathbf{x}_{2\beta}, ..., \mathbf{x}_{n\beta}\right\}, n\beta \leq i \tag{31}$$

Similarly, the processing time of each event and mean shift iteration are computed to assess the efficiency, as illustrated in Fig. 7.



(a) Processing time **(b)** Performance

Fig. 7 Processing time and performance with varying downsampling rate β

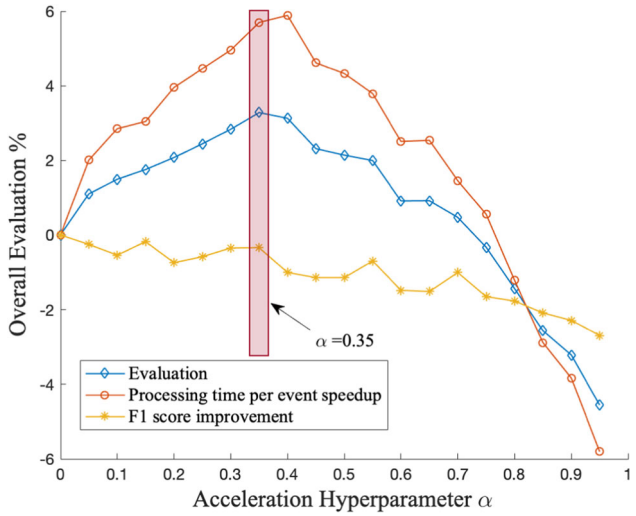


Fig. 6 The relationship between E-score and α value

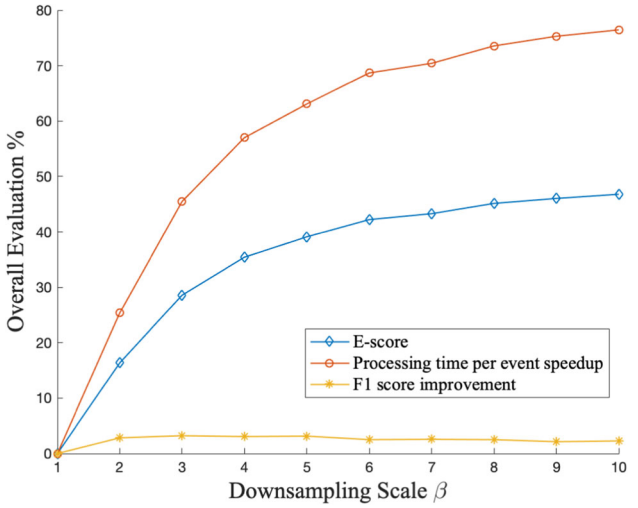


Fig. 8 Overall evaluation with varying downsampling scale β

From Fig. 7a, the processing time evaluated on both single event and iteration demonstrates a decaying trend with increasing β. The smaller β results in the greater reduction slop which means a more significant improvement of efficiency. When β exceeds 4, the processing time per event can even reach around 1 μs. Figure 7b presents the assessment of the clustering accuracy, where recall, precision and F1 score outperforms the standard MS as $\beta = 1$. With the reduction of original events, the noise captured and included in the original data will also be reduced. As a result of the reduced influence of noise disruption, clustering and segmentation will function more accurate. Based on the processing time per event and F1 score, the overall evaluation $E - score$ of MEMS with strategy-2 is calculated with varying β values as illustrated in Fig. 8. It shows a rising improvement with increasing β values, and at least around 16% improvement is reached when $\beta = 2$.

Visual servoing

The traditional Visual Servoing (VS) is based on the information extracted from standard vision sensors, which was first proposed by the SRI International Labs in 1979 (Hill 1979). Distinguishing to the eye-to-hand configuration relying on observing the absolute position of the target and the hand, an eye-in-hand camera is attached on hand and observes the relative position of the target. The major purpose of VS in this study is to manipulate the gripper to the desired position of the target's centroid in a 2D plane. An Event-based Visual Servoing (EVS) method for multiple objects adopting

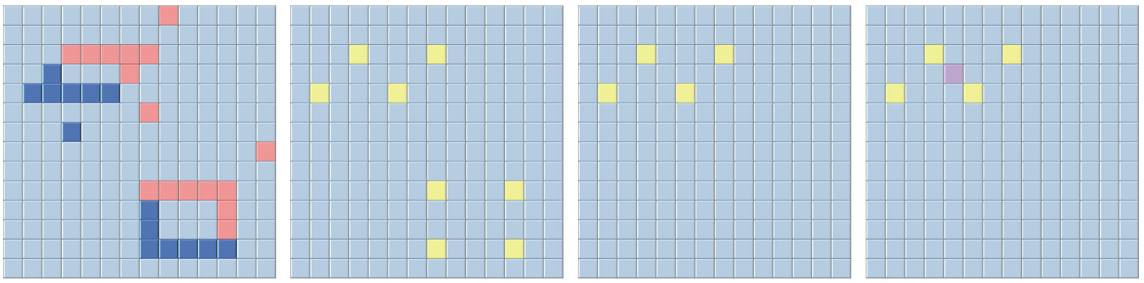


Fig. 9 Four layers of surfaces of active events for robust centroid detection. **a** Raw events on SAE. Red and blue blocks represent negative and positive polarity events. **b** Corner features (yellow blocks) on SAFE. **c** Corners of targeting object on SALE. **d** Virtual robust centroid (purple block) on SAVE (Color figure online)

depth information with the eye-in-hand configuration is proposed based on our previous work (Muthusamy et al. 2021). The centroid information obtained by the proposed MEMS will guide EVS to track the object. Then the robust corner is further calculated using a heatmap to ensure a stable manipulation.

Four layers of active events surfaces are considered as shown in Fig. 9. The Surface of Active Events (SAE) shows all the raw events captured by the event-based camera. By using a feature detector, only the corners will be extracted and projected into Surface of Active Feature Events (SAFE). In this work, eHarris detector is applied to detect corner features of the objects. The mask is applied to remove the corners of other objects, so only the useful features are remained and projected to the Surface of Active Locking Events (SALE). The robust centroid information will be calculated and virtually projected into the Surface of Active Virtual Events (SAVE).

According to the robust centroid and depth information, the robot will be manipulated to track the object. The block diagram for completing the manipulation task by EVS is depicted in Fig. 10. P_d and P_a represent the desired and the actual/current planar position of the object's centroid. θ_d and θ_a indicates the desired and the actual/current orientation of the object. The error e_p and e_θ can be calculated as $P_d - P_a$ and $\theta_d - \theta_a$. According to the position error, a forward and lateral correction will be executed to move the gripper to the proper position. Then the angular correction will be implemented in order according to the angular error. Based on the control and manipulation, the orientation and position error will be eliminated until the robot aligns with the object.

The UR robot utilized in this work provides the secondary velocity control $\vec{v}(\vec{v}_f, \vec{v}_l, \vec{v}_r)$ for end-effectors/grippers, and the relationship between the moving distance $dist$ and planar velocity $\vec{v}_p$ is as $dist = |f(\vec{v}_p)|$. Here $\vec{v}_f$, $\vec{v}_l$ and $\vec{v}_r$ represent the forward, lateral and rotational velocity, respectively. Besides, the planar velocity can be computed as $\vec{v}_p = \vec{v}_f + \vec{v}_l$. The velocity control based visual servoing is illustrated in

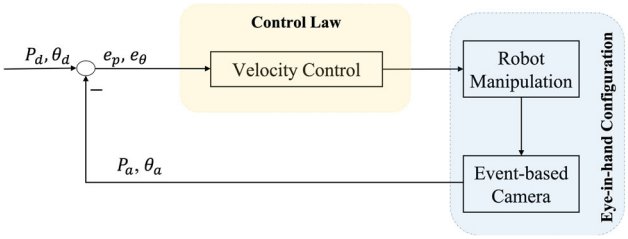


Fig. 10 The block diagram of event-based visual servoing

Fig. 11, that consists of two stages - translational (forward and lateral) correction and angular correction in order.

Figure 11a, b show the sequence of translational correction, and the estimated position error $dist$ in image coordinate will be eliminated by velocity $\vec{v}_p(\vec{v}_f + \vec{v}_l)$ until $dist$ is smaller than the threshold. For Barret hand with three fingers, the grasp hypothesis generated is required to ensure a proper and stable grasp. In this work, grasp is planned based on the principal orientation which will be elaborated in "Grasping plan". As depicted in Fig. 11c, d, the gripper will keep rotating with $\vec{v}_r$ until the angle difference θ is eliminated to near 0. After accomplishing correction, the gripper will move down to pick the object up according to the depth information mapped.

Grasping plan

For model-free object grasping, there is no geometric model or any prior knowledge of the object. A mount of researches rely on exploring the geometrical information such as shapes, edges, and saliency. It suffers from low efficiency, since the exploration by moving the camera around the unknown objects takes time. Another popular approach is using deep learning techniques such as DCNN to train robots to generate a proper grasping hypothesis but it requires a vast amount of manually labeled data for training. Therefore, a fast grasp generation is proposed for unseen objects using Principal Component Analysis (PCA), which is relatively more efficient by avoiding online exploration and offline training.

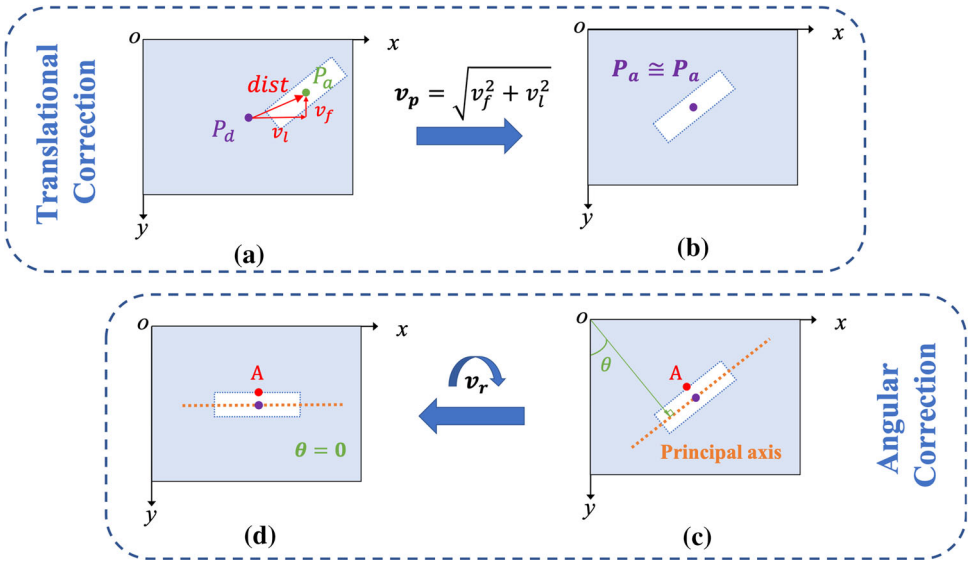


Fig. 11 Velocity control principal of EVS which includes two parts in order: translational correction (**a**, **b**) and angular correction (**c**, **d**)

In this work, the principal axis of objects obtained from PCA is utilized to generate a proper grasp position. The principal component is equivalently defined as a direction that maximizes the variance of the projected data, which can be computed by eigen decomposition of the covariance matrix COV of the data matrix as described in the following equation:

$$COV = \begin{pmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 \\ \sigma_{yx}^2 & \sigma_{yy}^2 \end{pmatrix} \tag{32}$$

where $\sigma_{xx}^2, \sigma_{xy}^2, \sigma_{yx}^2$ and σ_{yy}^2 are the covariance values of 2D coordinate. It is based on calculating the eigenvalues $(\lambda_1 > \lambda_2)$ and the corresponding eigenvectors (u_1, u_2) to find the principal component, where eigenvectors and eigenvalues are used to quantify the direction and the magnitude of the variation captured by each axis. Then $u_1(u_{1x}, u_{1y})$ can approximate the direction θ of the principal axis as:

$$\theta = arctan\frac{u_{1y}}{u_{1x}} \tag{33}$$

The grasping pose will be generated by the centroid C and the direction θ. To ensure a robust principal orientation, all the orientations detected before grasping will be stored in a histogram with 3-degree bins. The final rotation is converted into the range of $[-90, 90]$ to ensure the shortest rotation path, resulting in 61 bins in the histogram. Then the final robust orientation is determined by the bin value with the maximum probability in the histogram.

The Barrett hand used in this work has three fingers of eight joints with only four degrees of freedom. Each finger contains two degrees of freedom controlled by a servo-actuated joint. Two of the fingers have an extra joint which allows them to rotate synchronously around the palm with a certain spread angle relative to the third finger up to 180 degrees. The three fingers are commanded with the same joint value, simplifying the grasp plan and limiting the number of possible configurations. Figure 12 shows the knowledge-based approach that is used in our case to find the appropriate grasp plan, (1) the grasping point on the object (centroid), (2) the principal axis of the object. To perform a grasp, a Tool Center Point (TCP) is defined first on the Barrett hand. The hand is moved to the grasping point on the object and rotate to be perpendicular to the object's principal orientation. Next, the fingers are closed around the object until contacts or joint limits prevent further motion. This configuration is executed after ensuring that the Barret hand moving fingers can achieve stable contact points with the object's side surfaces. The distance between the two moving fingers is pre-measured and compared with the edge length of the side surface to confirm the grasp.

Event-based model-free grasping framework

The whole framework of the proposed model-free neuromorphic vision-based multiple-object grasping is illustrated in Fig. 13.

In this approach, each object will be numbered from one after segmentation and grasped orderly according to their IDs. The depth information obtained by EMVS will be used only once before grasping, since the additional movement and time are required and consumed. Thus the one-time EMVS will be executed in the initial stage before segmentation when the cluster ID is equal to one, and the depth map of the initial position will be frozen. By using the developed MEMS, objects will be clustered with their IDs and centroid position. Through mapping the frozen depth map with the segmented object information, the 3D spatial information of

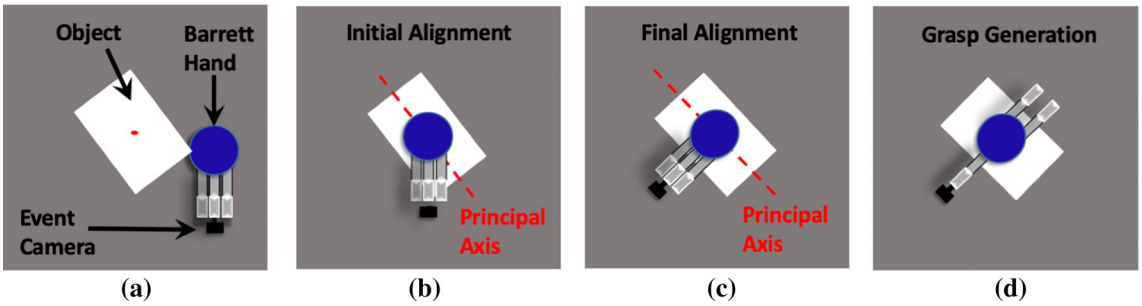


Fig. 12 Barrett hand grasp alignment

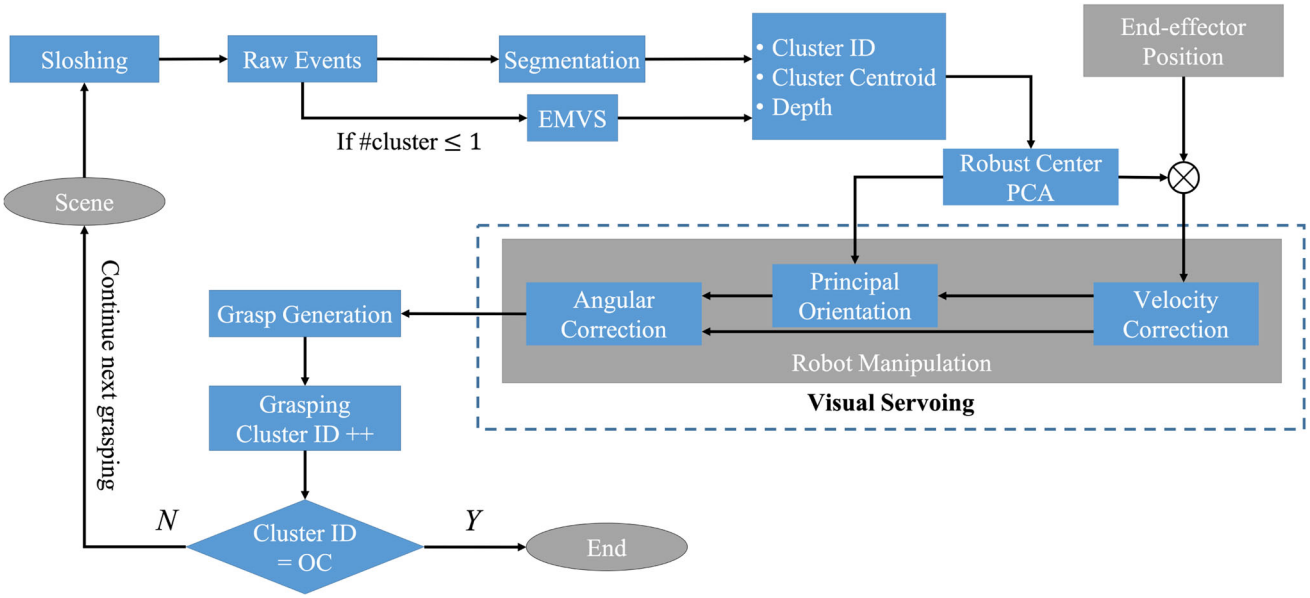


Fig. 13 The whole framework of proposed model-free event-based multiple-object grasping

each object at the initial position will be obtained and locked to provide the depth information for the next grasping. As depicted in "Visual servoing" section, the robust centroid of the current tracking object will be obtained and projected into SAVE. Based on the position error and orientation error, the translational and angular correction will be employed until the gripper is aligned to the targeting object. Then the object will be picked up and placed at the specific dropping area. After that, the cluster ID will be accumulated by 1, and the gripper will return to the initial position and start the next grasping task. The framework of model-free multiple-object grasping is summarized in Algorithm 2.

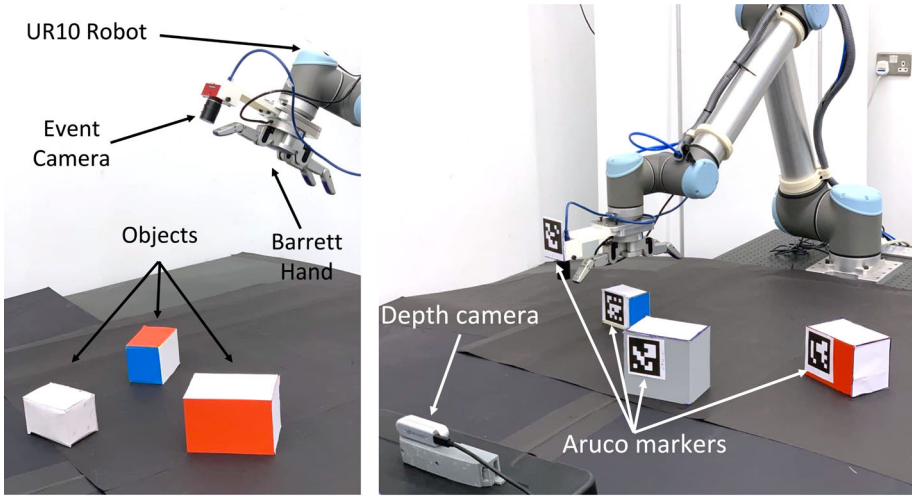
Experimental validation on multiple-object grasping

This section describes the experimental validation of multiple-object grasping and discusses the experimental results.

Experimental setup and protocol

The real experiments are performed to validate the proposed grasping approaches. As demonstrated in Fig. 14, the experimental setup consists of a robotic grasping system and an evaluation system.

The grasping system includes a Universal Robots UR10 6-DOF arm [43], a Barrett hand gripper [44], and a Dynamic and Active pixel VIsion Sensor (DAVIS346) [45] placed in an eye-in-hand configuration. The UR10 arm features a 10 kg weight capacity and 0.1 mm movement accuracy, making it ideal for packaging, assembly, and pick-and-place tasks. The DAVIS346 sensor has an outstanding dynamic range ($>$ 100 dB) and 346 $\times$ 260 resolution. The stream of events encodes time t, position (x, y) and sign of brightness change p. Objects of different sizes and shapes are used as the grasping targets. To perform the proposed approach successfully, it is assumed that the targeting objects are within the gripper's manipulation range since the robot is installed in a fixed



(a) Robotic grasping system **(b)** Evaluation system

Fig. 14 Experiment setup consists of two parts: **a** Robotic grasping system for experimental validation of proposed approach; **b** evaluation system for assessing the grasping performance

Algorithm 2: Model-free Grasping Framework

Input: Events stream: position (x_i, y_i), polarity p_i, timestamp t_i
Output: Cluster ID, cluster centroid (x_c, y_c)

Initialize cluster ID = 1
Set starting point of gripper P_0
while *ID = 1 or ID ≤ The number of objects* **do**
  if *ID = 1* **then**
    Move gripper and record the trajectory
    Perform EMVS and MEMS
    Map and freeze depth with 2D spatial and centroid position of each object
  else
    Perform MEMS to detect the centroid of the targeting object
    Obtain depth, 2D spatial and centroid information of each object
  Detect corners in SAE by applying e-Harris, and project corner events to SACE
  Extract object corners in SACE using heatmaps
  Calculate the robust centroid of the targeting object
  Calculate the position error e_p between the current position and robust centroid position
  if $abs(e_p) > 0$ **then**
    Perform EVS to eliminate e_p
  else
    Perform EVS to eliminate the angular error e_r
    Execute grasping
  if *grasp accomplished* **then**
    Cluster ID +1
    Move back to the initial position

base. Moreover, the sizes of objects are expected to be within the maximum opening of the gripper.

To estimate the grasping performance, we developed an evaluation system that consists of ArUco markers and a standard camera Intel D435. The identity of an ArUco maker is determined by its binary matrix inside of the black border, that facilitates a fast detection and applicability for camera calibration and pose estimation. By conducting 10 experiments of measuring the static object's pose using our evaluation system, the estimation error of angle and position are evaluated as only 1° and 0.1 cm, respectively. In this work, the identified ArUco markers are attached on the lateral side of gripper and targeting objects, so their poses can be determined by detecting and estimating the pose of ArUco markers. According to the evaluation metrics for grasping performance detailed in "Evaluation metrics" section, we focus on the poses in three stages: initialization, optimal grasping and object grasping as demonstrated in Fig. 15. In the beginning, the object's pose will be recorded as P_{obj}. After visual servoing, the gripper will reach the optimal grasping pose as P_{grip}. Since the ArUco marker would be covered by the finger of Barrett hand after grasping, Barrett will hold the object for one second and release it. After opening the gripper, the object pose will be estimated as $P_{obj'}$ to evaluate the object deviation.

Model-free grasping experiment protocol

According to Algorithm 2, the experiments are designed and performed in the following steps:

(1) Depth exploration stage. Move the gripper in a linear trajectory to the initial position and perform EMVS to obtain the depth information. This stage is only activated once at the start of the whole experiment.
(2) Segmentation stage. Slosh gripper to generate some movement for observing the objects, since only illumination change can be captured by the event camera. Then segment each object by the developed MEMS to obtain the centroid information. Meanwhile, the orientation of each object is acquired by PCA. Sort objects according to

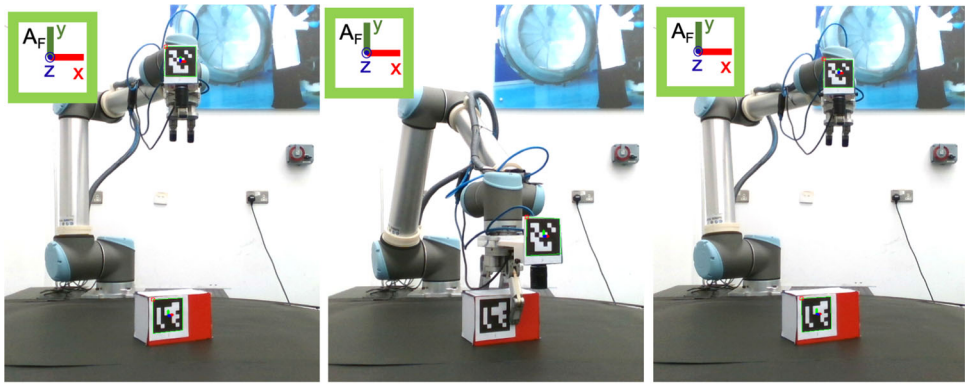


Fig. 15 Pose estimation by developed evaluation system in three steps: **a** Initial pose. **b** Optimal grasping pose. **c** Deviated object pose after grasping. The coordinate of ArUco markers is indicated at the left top corner

(a) Initial pose of object P_{obj} **(b)** Grasping pose of Barrett hand P_{grip} **(c)** Deviated pose of object $P_{obj}{}'$

volume, and update centroid and orientation information of the largest object to visual servoing.

(3) Visual servoing stage. Extract the robust corner feature and virtual object centroid in SAVE, and track the object until the object and camera centers are matched.

(4) Optimal grasping stage. Rotate the Barrett hand to align to the object, and adjust the gripper's position to compensate the installation deviation between the camera and gripper. After rotation and adjustment, Barrett hand will reach the grasping point and hold the object.

(5) Pick and place stage. Barrett hand lifts and places the object into the drop box in this phase.

Model-based grasping experiment protocol

The model-based grasping framework shown in Algorithm 1, shows that the grasping approach is divided in the following steps:

(1) Scene scanning stage: the robot end-effector starts from a known point and scans the scene in a linear trajectory set of movements.

(2) Object Localization stage: detected objects' corners are used as an input for the event-based multi-view localization approach, and the objects are localized.

(3) Point cloud processing stage: point cloud downsampling and object Euclidean clustering is performed to divide the objects in the scene to separate point clouds.

(4) Model registration stage: for each object an inexact model is fitted to to the detected objects, and the transformation matrix is extracted.

(5) Grasping stage: The robot gripper is navigated towards the object using PBVS and grasp is performed with the required manipulation.

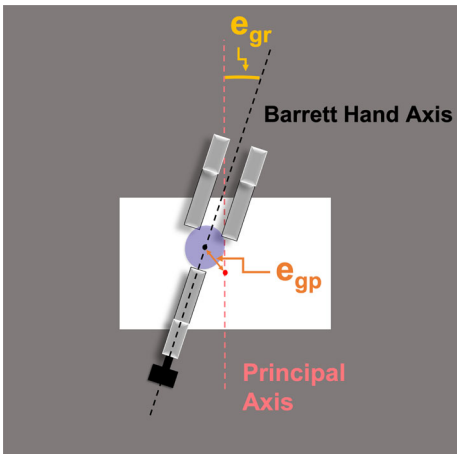


Fig. 16 Two metrics of grasping performance evaluation–positioning error e_{gp} and angular error e_{gr}

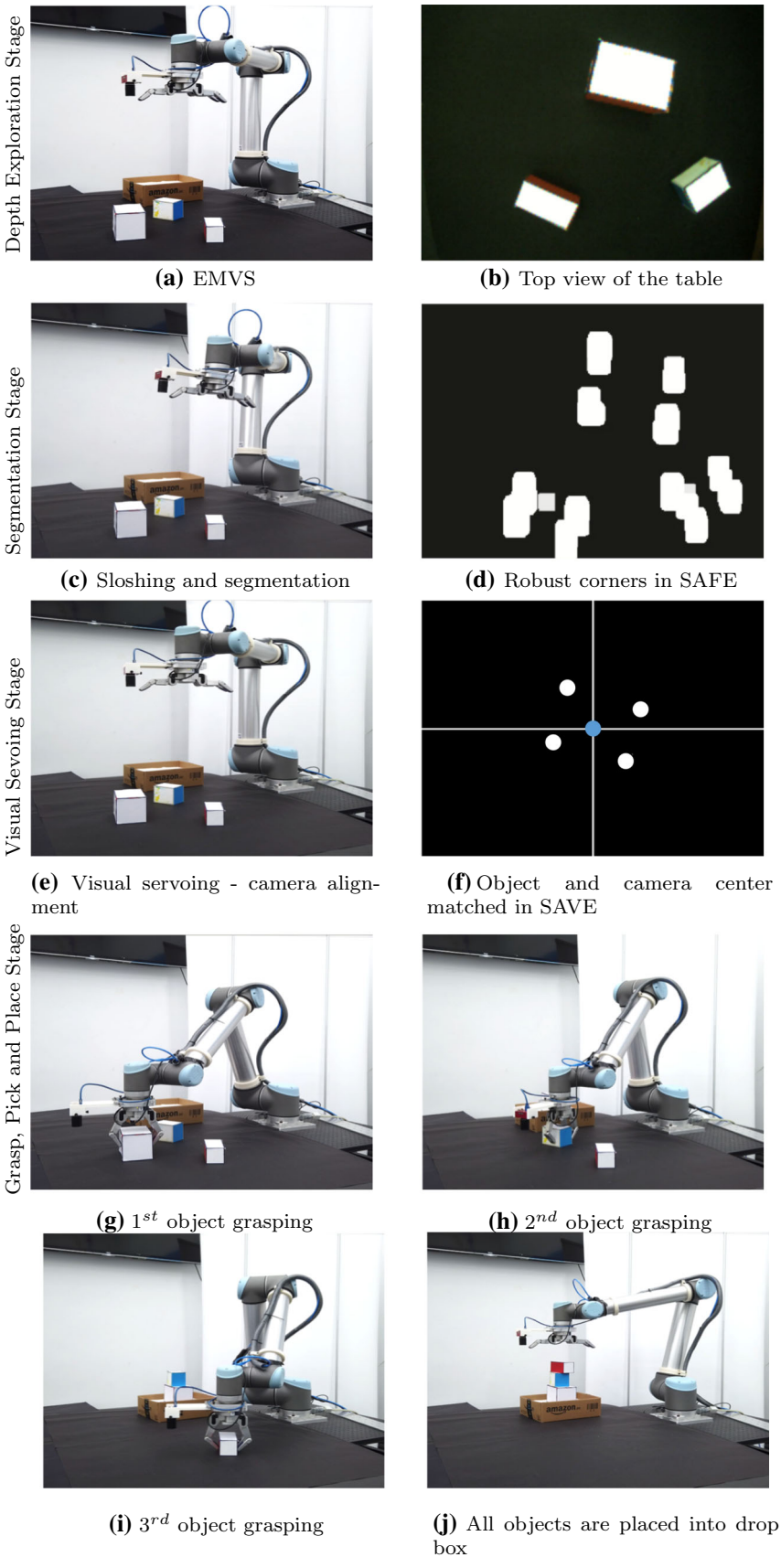
Evaluation metrics

Proper metrics are crucial to quantify the grasping quality and evaluate the performance of real grasping. In this work, the accuracy of grasping is assessed by the position and orientation error in two phase: optimal grasping and object deviation evaluations. Building on that, the success rate of grasping pose and the grasping quality score are computed to indicate the overall grasping performance.

Optimal grasping evaluation

The goal of optimal grasping evaluation is to meaure the difference between the optimal grasping pose and the actual grasping pose of the gripper after the alignment and before enclasping the object, using two components as illustrated in Fig. 16 including the position error e_{gp} and the orientation error e_{gr} of the planning grasp pose.

The position error e_{gp} and the orientation error e_{gr} represent the distance and the angle between the gripper's center



(a) EMVS

(b) Top view of the table

(c) Sloshing and segmentation

(d) Robust corners in SAFE

(e) Visual servoing - camera alignment

(f) Object and camera center matched in SAVE

(g) 1^{st} object grasping

(h) 2^{nd} object grasping

(i) 3^{rd} object grasping

(j) All objects are placed into drop box

Fig. 17 Experimental sequences of proposed neuromorphic vision based multi-object grasping

Table 1 Model-free experimental results of grasping different-size objects using event camera

Object size	e_{gp} (cm)	e_{gr} (°)	SS	D_P (cm)	D_R (°)	Q_G
Small	1.477	2.14	0.800	1.099	2.10	0.655
Medium	1.461	2.46	1.000	1.684	1.47	0.530
Large	1.498	2.62	1.000	1.343	1.46	0.616
Average/overall	1.479	2.41	0.933	1.375	1.68	0.600

Table 2 Model-based experimental results of grasping different-size objects using event camera

Object Size	e_{gp} (cm)	e_{gr} (°)	SS	D_P (cm)	D_R (°)	Q_G
Small	0.891	4.88	1.000	0.821	10.70	0.438
Medium	0.742	3.73	1.000	0.361	0.51	0.893
Large	0.481	3.88	1.000	0.711	2.46	0.740
Average/overall	0.705	4.16	1.000	0.631	4.56	0.690

and the actual object's center, respectively. We set the limitation of position error L_P and orientation error L_R to 2 cm and 15°. Only when both grasping errors are within the limitations, the grasping can be considered as successful as described in Eq. 34, where SS indicates the success sign of the current grasping. Then the overall success rate can be computed as $SS = \sum_N SS_i/N$, where N denotes the total number of grasping performed.

$$SS_i = \begin{cases} 0 & \text{if } (e_{gp} \le L_P \text{ and } e_{gr} \le L_R) \\ 1 & \text{if } (e_{gp} > L_R \text{ or } e_{gr} > L_R) \end{cases} \tag{34}$$

Object deviation evaluation

However, the overall grasping quality can not be estimated only using the planed grasping error before the real grasping. Then the deviation of object pose D is taken into account, reflecting the relative pose before $(\vec{P}_b, R_b)$ and after $(\vec{P}_a, R_a)$ trapped by the fingertips of the gripper. The deviation can be quantified as two parts: the position deviation D_P and the orientation deviation D_R as expressed in Eq. 35.

$$\begin{aligned} D &= \{D_P, D_R\} \\ D_P &= \|\vec{P}_b - \vec{P}_a\| \\ D_R &= |R_b - R_a| \end{aligned} \tag{35}$$

The grasp quality score Q_G is calculated according to the deviations and the predefined limitations as expressed in Eq. 36. Grasping with less object deviation is considered to be of better quality, as the deviation of the object's pose would cause grasping failure.

$$Q_G = \begin{cases} 1 - \frac{D_P}{2*L_P} - \frac{D_R}{2*L_R} & \text{if } (D_P \le L_P \text{ and } D_R \le L_R) \\ 0 & \text{if } (D_P > L_P \text{ or } D_R > L_R) \end{cases} \tag{36}$$

Experimental results and analysis

The five stages of the proposed neuromorphic vision-based robotic grasping approaches with multiple cubic objects of different sizes as demonstrated in Fig. 17.

To quantify the grasping performance, we conducted 15 experiments of hexahedron objects with three different sizes using both model-based and model-free approaches. The size of small, medium and large object is $15 \times 10 \times 12\,\text{cm}^3$, $15 \times 10 \times 10\,\text{cm}^3$ and $10 \times 7 \times 8\,\text{cm}^3$, respectively. For individual object, five experiments are repeated and the errors are averaged. Tables 1 and 2 show the experimental results of grasping error and object deviation of the model-free and the model-based approaches.

Seen from Tables 1 and 2, both proposed neuromorphic vision-based grasping approaches can successfully accomplish the grasping tasks, and all of those evaluation metrics are within the limitations. By analyzing, the source of error is considered coming from several aspects. First, the error is caused by the experimental setup that the camera is not installed exactly parallel to the work plane. As segmentation and tracking are executed at some height, the positioning error will occur after reaching the center at a certain height and amplified while the gripper is moving down for grasping. In addition, there are two manually induced errors in the grasping phase and evaluation stage. Since the object segmentation and visual servoing are accomplished in the camera frame, the position adjustment is executed after visual servoing to compensate the manually measured deviation between the centers of event camera and Barrett hand. The similar deviation exists in the evaluation system, while calculating the optimal grasping pose by transformation from ArUco marker center to object top surface center. Besides, the low spatial resolution of DAVIS 346C utilized can also cause the error.

Table 3 Model-free experimental results of grasping different-size objects using event camera in low-light environment

Object size	e_{gp} (cm)	e_{gr} (°)	SS	D_P (cm)	D_R (°)	Q_G
Small	1.443	2.61	1.000	1.373	2.74	0.565
Medium	1.551	2.91	1.000	1.046	2.32	0.661
Large	1.411	2.88	0.600	1.203	1.96	0.503
Average/overall	1.478	2.80	0.867	1.207	2.34	0.576

Table 4 Model-based experimental results of grasping different-size objects using event camera in low-light environment

Object size	e_{gp} (cm)	e_{gr} (°)	SS	D_P (cm)	D_R (°)	Q_G
Small	1.031	5.98	1.000	0.83	5.01	0.626
Medium	0.951	5.41	1.000	0.40	3.12	0.795
Large	1.120	6.22	1.000	1.05	5.15	0.566
Average/overall	1.034	5.87	1.000	0.76	4.43	0.662

Robustness testing

To test the robustness of the proposed grasping approaches using an event camera, the additional experiments were conducted in low-light condition and using objects of other shapes.

low-light conditions One of the advantages of an event camera is high sensitivity to the change of light intensity, that can observe objects even in the low-light environment. However, more noise will also be captured in low-light condition. So the noise filter is applied to eliminate the noise and capture more meaningful events. We conducted 5 experiments for each cubic/hexahedron object. The experimental results of model-free and model-based approaches are recorded in Tables 3 and 4, including grasping pose error in terms of e_{gp} and e_{gr}, and the object deviation in terms of D_P and D_R. The success rate SS and grasp quality Q_G are also calculated with the same position limitation $L_P = 2$ cm and orientation limitation $L_R = 15°$.

Seen from the results, the average errors are all within the limitations and the success rate and grasp quality score are similar to those in the normal light environment. However, by comparing the standard deviation of the model-free approach as shown in Fig. 18a, the overall performance in the low-light condition is more unstable with a higher standard deviation, even though the value of position error of grasping orientation under normal light is slightly higher. For the model-based approach, it presents a higher standard deviation value of object deviation in low-light condition as depicted in Fig. 18b. On the whole, both of our proposed approaches can reach the grasping goal successfully in low-light environment.

The comparison between the two proposed approaches is depicted in Fig. 19, where MFA and MBA presents the model-free approach and model-based approach, GE and OD indicates the grasping error and object deviation, and LL expresses the low-light condition. From Fig. 19a, the model-free approach reaches a relative smaller orientation error and a larger position error comparing to the model-based approach. In both low-light and normal-light conditions, the model-based approaches reaches a higher successful rate and grasp quality score as indicated in Fig. 19b.

Objects with different shapes In addition, the experiments of grasping different shapes were also conducted to test the robustness. Besides of hexahedron, two octahedrons with different shapes demonstrated in Fig. 20 are utilized as unknown objects to validate the robustness of the proposed approach.

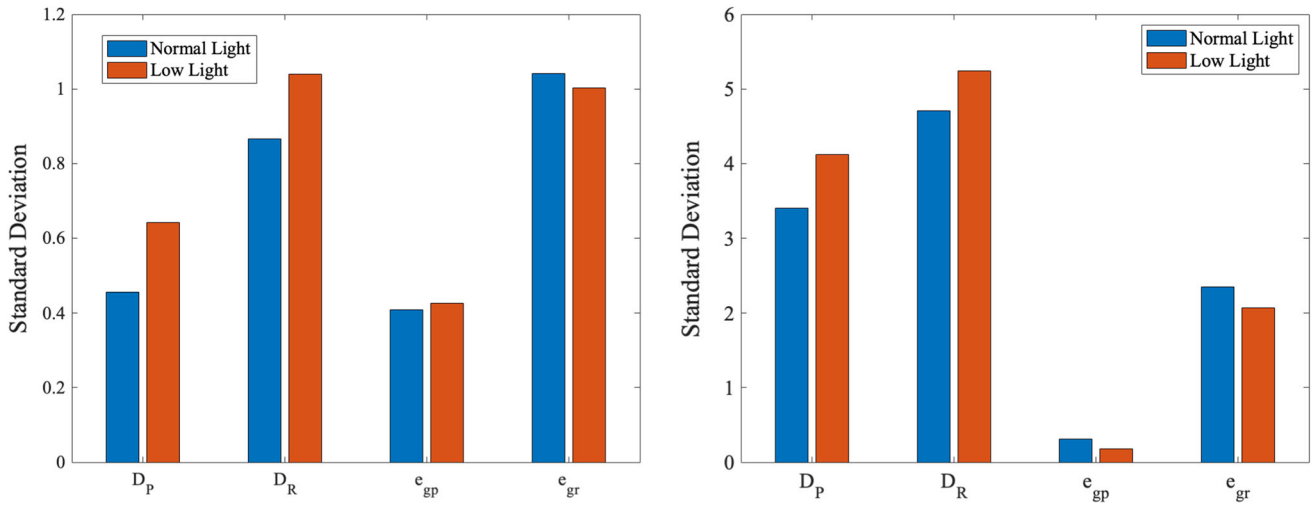
Since the model-based approach is designed for known objects with the prior model, only the model-free approach can achieve this task as the two octahedrons are considered as unknown objects. Table 5 shows the grasping pose error, object deviation, success rate and grasp quality of experiments on objects with varying shapes. Octahedron-6 and octahedron-8 represent the octahedron with six and eight visible corner features from the top view as shown in Fig. 20.

By experimental validation, all those objects with different shapes can be pick and placed effectively. Seen from Table 5, the grasping of three objects of different shapes demonstrates a comparable performance. Furthermore, the proposed model-free approach is also validated on real objects in daily life. The whole pick and place process is demonstrated in Fig. 21, that achieves a successful pick and place task for multiple objects.

Discussion

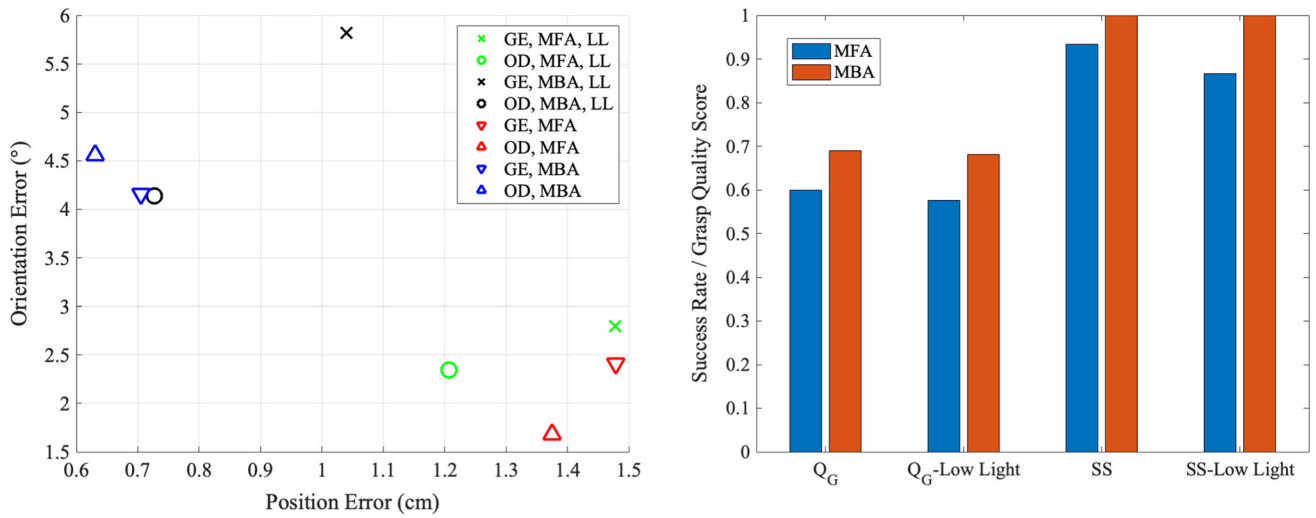
Both model-based and model-free approaches proposed are valid for multiple-object grasping using an event camera. By comparison, the pros and cons of two approaches are concluded in Table 6.

From the experimental results in "Experimental results and analysis" section, the model-based approach shows a slightly better grasping performance with less error because



(a) Model-free approach **(b)** Model-based approach

Fig. 18 Standard deviation of grasping errors and object deviations in normal-light and low-light condition by model-free approach



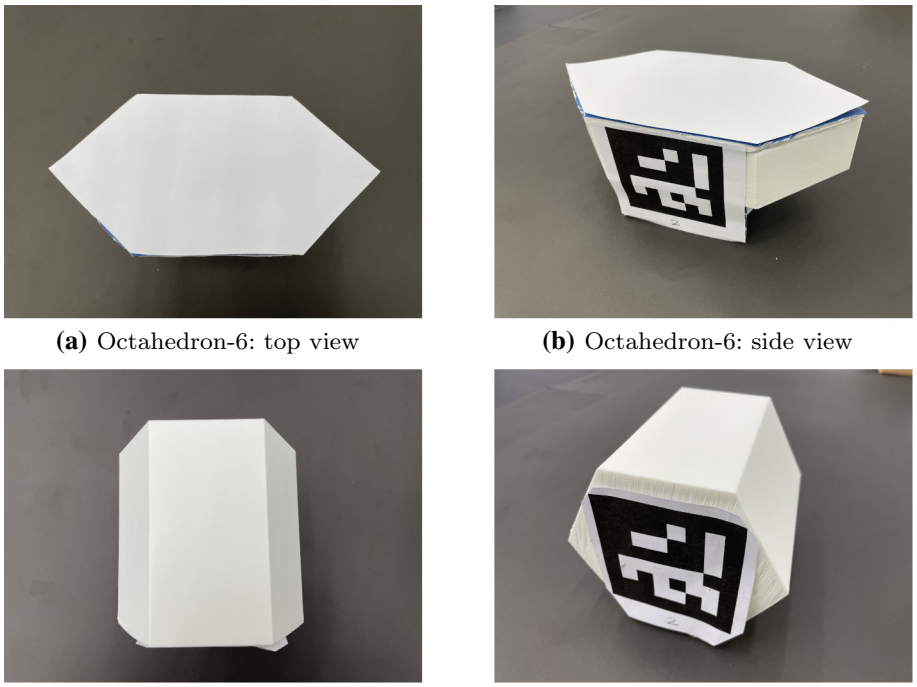
(a) Grasping error and object deviation **(b)** Success rate and grasp quality score

Fig. 19 Comparison of the proposed model-based and model-free approaches

of the position based visual servoing. But the model-based approach is limited to known objects as it requires offline modeling of objects. However, the objects are generally unknown requiring online process in real scenarios. It indicates that the model-based approach more suitable for grasping tasks for the specific or pre-defined objects. By contrast, the proposed model-free approach can obtain the position information of unknown objects without prior knowledge, which shows a great advantage in practical and real applications. Moreover, it is quite less sensitive to the imperfect perception than the model-based approach. In addition, the model-free approach also shows the flexibility and possibility to deal with the moving object besides of the static object.

Conclusion

We proposed an event-based grasping framework for robotic manipulator with neuromorpihc eye-in-hand configuration. Particularly, a model-based and a model-free approaches for multiple-object grasping in a cluttered scene are developed. The model-based approach provides a solution for grasping known objects in the environment, with prior knowledge of the object shape to be grasped. It consists of the 3D reconstruction of the scene, euclidean distance clustering, position-based visual servoing and grasp planning. Differently, the model-free approach can be applied to unknown objects grasping applications in real time, which consists



(a) Octahedron-6: top view **(b)** Octahedron-6: side view

(c) Octahedron-8: top view **(d)** Octahedron-8: side view

Fig. 20 Two octahedrons utilized in grasping experiments. Octahedrons with six corners (Octahedron-6) in the top view (**a**) and in the side view (**b**). Octahedrons with eight corners (Octahedron-8) in the top view (**c**) and in the side view (**d**)

of the developed event-based segmentation, visual servoing adopting depth information and grasp plan.

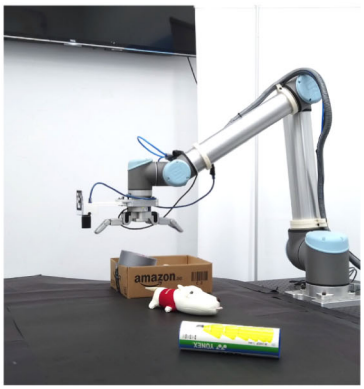
By experimentally validating with objects of different sizes and in different light conditions, both approaches can effectively achieve the multiple-object grasping task successfully. From the quantity evaluation of the grasping pose, success rate, object deviation and grasp quality, the model-based approach presents slightly more accurate because of the position-based visual servoing. However, the model-based approach is constrained to known objects with prior knowledge of models. The model-free approach is more applicable in real scenarios for operating unknown objects, which is experimentally validated with real objects in this paper. To conclude, both model-based and model-free approaches are applicable and effective for neuromorphic vision-based multiple-object grasping applications, which can boost production speed in factory automation. According to their pros and cons, the particular approach can be selected in different specific scenarios. This paper demonstrates grasping for multiple-object in simple scenarios, we will focus on the event-based object segmentation for more complex situations such as objects with occlusion in the future.

Table 5 Experimental results of grasping different-shape objects by model-free approach using event camera

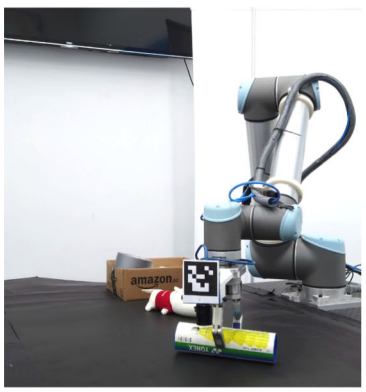
Object Shape	e_{gp} (cm)	e_{gr} °	SS	D_P (cm)	D_R °	Q_G
heptahedron	1.479	2.41	0.933	1.375	1.68	0.600
octahedron-6	1.557	2.82	1.000	0.901	2.04	0.707
octahedron-8	1.530	2.79	0.800	0.993	2.43	0.671
Average/overall	1.522	2.67	0.880	1.090	2.05	0.659

(a) initial state with real objects: soft doll, badminton tube and tape

(b) grasp tape

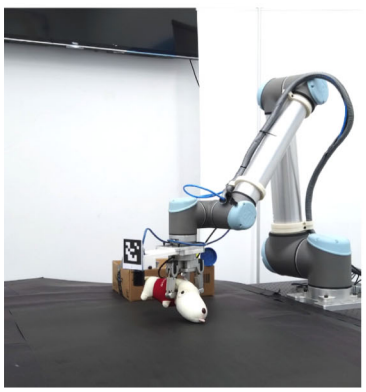


(c) drop tape



(d) grasp badminton tube

(e) drop badminton tube



(f) grasp soft doll

(g) drop soft doll

Fig. 21 Picking and placing process of real objects by the proposed neuromorphic vision based multi-object grasping approach

Table 6 Comparison of the proposed model-based and model-free approaches

Terms	Model-based approach	Model-free approach
Pros	Higher accuracy	Model free Unknown and moving objects Robust to imperfect perception
Cons	Prior knowledge of model is required Sensitive to imperfect perception	Relatively lower accuracy

Acknowledgements This work is supported by the Khalifa University of Science and Technology under Award No. CIRA-2018-55 and RC1-2018-KUCARS, and was performed as part of the Aerospace Research and Innovation Center (ARIC), which is jointly funded by STRATA Manufacturing PJSC (a Mubadala company) and Khalifa University of Science and Technology.

References

Asadi, K., Haritsa, V. R., Han, K., & Ore, J.-P. (2021). Automated object manipulation using vision-based mobile robotic system for construction applications. *Journal of Computing in Civil Engineering, 35*(1), 04020058.

Barranco, F., Fermuller, C.,& Ros, E. (2018). Real-time clustering and multi-target tracking using event-based sensors. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 5764–5769). IEEE.

Bin Li, H., Cao, Z. Q., Yingbai, H., Wang, Z., & Liang, Z. (2020). Event-based robotic grasping detection with neuromorphic vision sensor and event-grasping dataset. *Frontiers in Neurorobotics, 14,* 51.

Bohg, J., Morales, A., Asfour, T., & Kragic, D. (2013). Data-driven grasp synthesis-a survey. *IEEE Transactions on Robotics, 30*(2), 289–309.

Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9157–9166).

Chen, C., & Ling, Q. (2019). Adaptive convolution for object detection. *IEEE Transactions on Multimedia, 21*(12), 3205–3217.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning, 20*(3), 273–297.

Davis 346. https://inivation.com/wp-content/uploads/2019/08/DAVIS346.pdf. Accessed 08 2019.

Du, G., Wang, K., & Lian, S. (2019). Vision-based robotic grasping from object localization pose estimation grasp detection to motion planning: A review. arXiv preprint arXiv:1905.06658.

Etienne-Cummings, R., & der Spiegel, J. V. (1996). Neuromorphic vision sensors. *Sensors and Actuators A: Physical, 56*(1–2), 19–29.

Fukunaga, K., & Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory, 21*(1), 32–40.

Gallego, G., Delbruck, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., et al. (2019). Event-based vision: A survey. arXiv preprint arXiv:1904.08405.

Hill, J. (1979). Real time control of a robot with a mobile camera. In *9th International symposium on industrial robots, 1979* (pp. 233–246).

Hu, Y., Fua, P., Wang, W., & Salzmann, M. (2020). Single-stage 6d object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2930–2939).

Huang, X., Muthusamy, R., Hassan, E., Niu, Z., Seneviratne, L., Gan, D., & Zweiri, Y. (2020). Neuromorphic vision based contact-level classification in robotic grasping applications. *Sensors, 20*(17), 4724.

Indiveri, G., & Douglas, R. (2000). Neuromorphic vision sensors. *Science, 288*(5469), 1189–1190.

Kleeberger, K., Bormann, R., Kraus, W., & Huber, M. F. (2020). A survey on learning-based robotic grasping. *Current Robotics Reports*, 1–11.

Likas, A., Vlassis, N., & Verbeek, J. J. (2003). The global k-means clustering algorithm. *Pattern recognition, 36*(2), 451–461.

Milford, M., Kim, H., Leutenegger, S., & Davison, A. (2015). Towards visual slam with event-based cameras. In *The problem of mobile sensors workshop in conjunction with RSS*.

Mitrokhin, A., Fermüller, C., Parameshwara, C., & Aloimonos, Y. (2018) Event-based moving object detection and tracking. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 1–9). IEEE.

Multi-fingered programmable grasper. https://advanced.barrett.com/barretthand. Accessed 08 2019.

Muthusamy, R., Huang, X., Zweiri, Y., Seneviratne, L., & Gan, D. (2020). Neuromorphic event-based slip detection and suppression in robotic grasping and manipulation. arXiv preprint arXiv:2004.07386.

Muthusamy, R., Ayyad, A., Halwani, M., Swart, D., Gan, D., Seneviratne, L., & Zweiri, Y. (2021). Neuromorphic eye-in-hand visual servoing. *IEEE Access, 9,* 55853–55870.

Naeini, F. B., AlAli, A. M., Al-Husari, R., Rigi, A., Al-Sharman, M. K., Makris, D., & Zweiri, Y. (2019). A novel dynamic-vision-based approach for tactile sensing applications. *IEEE Transactions on Instrumentation and Measurement, 69*(5), 1881–1893.

Naeini, F. B., Makris, D., Gan, D., & Zweiri, Y. (2020). Dynamic-vision-based force measurements using convolutional recurrent neural networks. *Sensors, 20*(16), 4469.

Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia, 4*(2), 1883.

Rebecq, H., Gallego, G. & Davide, S. (2016). Emvs: Event-based multi-view stereo.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779–788).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497.

Rigi, A., Naeini, F. B., Makris, D., & Zweiri, Y. (2018). A novel event-based incipient slip detection using dynamic active-pixel vision sensor (Davis). *Sensors, 18*(2), 333.

Rusinkiewicz, S., & Levoy, M. (2001). Efficient variants of the ICP algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling* (pp. 145–152). IEEE.

Rusu, R. B., & Cousins, S. (2011). 3d is here: Point cloud library (PCL). In *2011 IEEE international conference on robotics and automation*, (pp. 1–4).

Sahbani, A., El-Khoury, S., & Bidaud, P. (2012). An overview of 3d object grasp synthesis algorithms. *Robotics and Autonomous Systems, 60*(3), 326–336.

Sankaranarayanan, J., Samet, H., & Varshney, A. (2007). A fast all nearest neighbor algorithm for applications involving large point-clouds. *Computers & Graphics, 31*(2), 157–174.

Úbeda, A., Zapata-Impata, B. S., Puente, S. T., Gil, P., Candelas, F., & Torres, F. (2018). A vision-driven collaborative robotic grasping system tele-operated by surface electromyography. *Sensors, 18*(7), 2366.

Umeyama, S. (1991). Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence, 13*(04), 376–380.

Ur10 technical specifications. https://www.universal-robots.com/media/50895/ur10_en.pdf. Accessed 09 2016.

Vasco, V., Glover, A., & Bartolozzi, C. (2016). Fast event-based Harris corner detection exploiting the advantages of event-driven cameras. In *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 4144–4149).

Wang, X., Kong, T., Shen, C., Jiang, Y., & Li, L. (2020). Solo: Segmenting objects by locations. In *European Conference on Computer Vision*, (pp. 649–665). Springer.

Zaidi, L., Corrales, J. A., Bouzgarrou, B. C., Mezouar, Y., & Sabourin, L. (2017). Model-based strategy for grasping 3d deformable objects using a multi-fingered robotic hand. *Robotics and Autonomous Systems, 95,* 196–206.

Zhang, Y., & Cheng, W. (2019) Vision-based robot sorting system. In *IOP conference series: Materials science and engineering* (Vol. 592, p. 012154). IOP Publishing.

Zhihong, C., Hebin, Z., Yanbo, W., Binyan, L., & Yu, L. (2017). A vision-based robotic grasping system using deep learning for garbage sorting. In *2017 36th Chinese control conference (CCC)* (pp. 11223–11226). IEEE.

Zhou, Y., & Hauser, K. (2017). 6dof grasp planning by optimizing a deep learning scoring function. In *Robotics: Science and systems (RSS) workshop on revisiting contact-turning a problem into a solution,* (Vol. 2, p. 6).

Zhou, Y., Gallego, G., Rebecq, H., Kneip, L., Li, H., & Scaramuzza, D. (2018). Semi-dense 3d reconstruction with a stereo event camera. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 235–251).

Zhou, Q.-Y., Park, J., & Koltun, V. (2016). Fast global registration. In *European conference on computer vision* (pp. 766–782). Springer.