Disability and Rehabilitation: Assistive Technology

ISSN: 1748-3107 (Print) 1748-3115 (Online) Journal homepage: https://www.tandfonline.com/loi/iidt20

A review of computer vision for semi-autonomous control of assistive robotic manipulators (ARMs)

Stefan Hein Bengtson, Thomas Bak, Lotte N. S. Andreasen Struijk & Thomas Baltzer Moeslund

To cite this article: Stefan Hein Bengtson, Thomas Bak, Lotte N. S. Andreasen Struijk & Thomas Baltzer Moeslund (2019): A review of computer vision for semi-autonomous control of assistive robotic manipulators (ARMs), Disability and Rehabilitation: Assistive Technology, DOI: 10.1080/17483107.2019.1615998

To link to this article: https://doi.org/10.1080/17483107.2019.1615998

DISABILITY AND REHABILITATION: ASSISTIVE TECHNOLOGY
https://doi.org/10.1080/17483107.2019.1615998

Taylor & Francis
Taylor & Francis Group

REVIEW

A review of computer vision for semi-autonomous control of assistive robotic manipulators (ARMs)

Stefan Hein Bengtson[a], Thomas Bak[b], Lotte N. S. Andreasen Struijk[c] and Thomas Baltzer Moeslund[a]

[a]Visual Analysis of People (VAP) Laboratory, Department of Architecture, Design, and Media Technology, Aalborg University, Aalborg, Denmark; [b]Automation and Control, Department of Electronic Systems, Aalborg University, Aalborg, Denmark; [c]Department of Health Science and Technology, Aalborg University, Aalborg, Denmark

ABSTRACT

Purpose: The advances in artificial intelligence have started to reach a level where autonomous systems are becoming increasingly popular as a way to aid people in their everyday life. Such intelligent systems may especially be beneficially for people struggling to complete common everyday tasks, such as individuals with movement-related disabilities. The focus of this paper is hence to review recent work in using computer vision for semi-autonomous control of assistive robotic manipulators (ARMs).

Methods: Four databases were searched using a block search, yielding 257 papers which were reduced to 14 papers after applying various filtering criteria. Each paper was reviewed with focus on the hardware used, the autonomous behaviour achieved using computer vision and the scheme for semi-autonomous control of the system. Each of the reviewed systems were also sought characterized by grading their level of autonomy on a pre-defined scale.

Conclusions: A re-occurring issue in the reviewed systems was the inability to handle arbitrary objects. This makes the systems unlikely to perform well outside a controlled environment, such as a lab. This issue could be addressed by having the systems recognize good grasping points or primitive shapes instead of specific pre-defined objects. Most of the reviewed systems did also use a rather simple strategy for the semi-autonomous control, where they switch either between full manual control or full automatic control. An alternative could be a control scheme relying on adaptive blending which could provide a more seamless experience for the user.

➤ IMPLICATIONS FOR REHABILITATION

- Assistive robotic manipulators (ARMs) have the potential to empower individuals with disabilities by enabling them to complete common everyday tasks. This potential can be further enhanced by making the ARM semi-autonomous in order to actively aid the user.
- The scheme used for the semi-autonomous control of the ARM is crucial as it may be a hindrance if done incorrectly. Especially the ability to customize the semi-autonomous behaviour of the ARM is found to be important.
- Further research is needed to make the final move from the lab to the homes of the users. Most of the reviewed systems suffer from a rather fixed scheme for the semi-autonomous control and an inability to handle arbitrary objects.

ARTICLE HISTORY
Received 26 July 2018
Accepted 3 May 2019

KEYWORDS
Computer vision; assistive robotic manipulators; ARM; semi-autonomous control; shared control; robotics; machine learning; exoskeleton

Introduction

Machines are becoming increasingly smarter and the effort invested into research in artificial intelligence is at an all-time high. This large interest in artificial intelligence is triggered by its ability to make smart decision to aid us in our everyday life.

The healthcare sector is one area which could benefit immensely from artificial intelligence by enabling assistive devices to act autonomously. Autonomous machines could, for instance, assist the elderly and disabeled individuals in feeding, getting dressed and other activities of daily living. This is especially of interest given the increasing demand for caregivers [1,2].

Persons suffering from quadriplegia, i.e., total or partial loss of control of all four limbs, would especially benefit from such assistance due to the severity of their disability. For instance, a study found that the use of an assistive robotic manipulator (ARM) could reduce the need for assistance with 1.25 h per day for persons with upper-extremity disabilities [3]. Another study confirmed these findings as their results showed that the use of an ARM could reduce the need for assistance by 41% [4]. This reduced need for assistance would not only be economically beneficial but also increase the users' quality of life by empowering them and providing them with some privacy. Furthermore, a survey on disabled persons found that 86% of the participants would consider purchasing an ARM given the possibility [5].

Another factor making autonomous control of ARMs increasingly interesting is how readily available the necessary hardware is becoming. For instance, the commercially available ARMs, which are specifically targeted at empowering users with movement impairments, such as JACO from Kinova [6] or iARM from Exact Dynamics [7].

CONTACT Stefan Hein Bengtson shbe@create.aau.dk Visual Analysis of People (VAP) Laboratory, Department of Architecture, Design, and Media Technology, Aalborg University, Aalborg, Denmark

However, any system which is to behave autonomously must rely on some sort of input to make an informed decision, for instance, knowledge of its immediate environment. Computer vision is hence often a part of such autonomous systems as it enables the system to capture and understand visual information. For instance, recognizing objects in an image and figuring out how to grasp said object [8–10].

The purpose of introducing autonomous behaviour into these systems is to reduce both the time it takes to execute a task and to reduce the cognitive burden on the user. This research has been expanded to other types of ARMs as well, such as exoskeletons [11,12]. The idea of using an exoskeleton is to provide a more integrated solution than e.g., a robotic arm mounted on a wheelchair.

However, having an ARM act autonomously is not necessarily a bliss for the user, even though it might reduce the time it takes to execute different tasks [13,14]. An important aspect of using this technology is hence how the control is shared between the human and the machine, i.e., the design of a scheme allowing for semi-autonomous control of the ARM.

The contribution of this paper is hence a review of recent efforts in employing computer vision for semi-autonomous control of ARMs, such as exoskeletons or robotic arms. The goal of this review is to: (1) provide an overview of existing efforts in using computer vision for semi-autonomous control of ARMs; (2) highlight the current challenges associated with this area of research; and thereby (3) point out new directions of interest for this field.

Methods

The following outlines how the review was conducted in terms of the literature search and subsequent sorting of found material. The extraction of data from each reviewed paper is described as well.

Data sources

The literature search was based on the following databases: *Engineering Village*, *Web of Science*, *Scopus* and *Embase*. The search was conducted by constructing blocks of keywords related to computer vision, robotic manipulators and people with disabilities. A paper had to match at least one keyword from each of these blocks to show up when searching each database.

The keywords in each of these blocks were as follows:

- **Block 1 – computer vision:** *("computer vision" OR "robot vision" OR "robotic vision" OR "object detection" OR "image-based" OR "grasp detection" OR "vision-based" OR perception*)*
- **Block 2 – robotic manipulators:** *("robot arm" OR "robotic arm" OR "robot manipulator" OR "robotic manipulator" OR exoarm OR exoskeleton OR "personal robot")*
- **Block 3 – people with disabilities:** *(disab* OR impair* OR adl* OR "activities of daily living" OR handicap* OR "personal robot" OR rehabilitat*)*

It should be noted that the asterisk * serves as a wildcard for unknown terms and different inflections of the same word.

Only the titles, abstracts and keywords were used while searching and any results not in English were removed. Publications before 2008 and duplicates were removed as well. Only conference proceedings, reports and journals were included during the literature search and book chapters or book reviews were removed from the list of results. This initial search resulted in 257 results after applying the above filters, as illustrated in Figure 1.

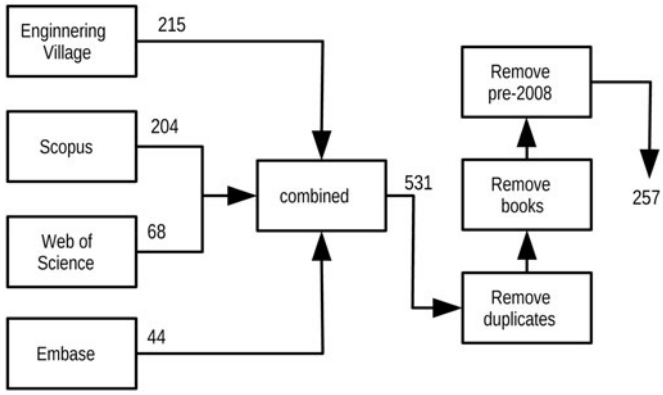

Figure 1. Databases and exclusion criteria used during the initial literature search.

Filtering criteria

Additional criteria were imposed on the initial search to further narrow down the amount of relevant papers. Each paper should fulfill each of the following criteria to be considered relevant:

1. **Purpose:** The intended use of the system described in the paper should be object manipulation tasks. Papers focussing on e.g., rehabilitation and wheelchair navigation were discarded. This criterion was imposed to focus the scope of the review.
2. **Camera:** The system described in the paper should make use of a camera or a similar visual sensor, such as a laser scanner. Any papers failing this criterion are not doing computer vision and are hence outside the scope of this review.
3. **Disabled user:** The intended user of the system described in the paper should be a person suffering some kind of movement impairment, such that they would benefit from an ARM.
4. **Autonomous behaviour:** The papers should describe a system capable of exhibiting some degree of autonomous behaviour. Papers solely describing a way of directly controlling an ARM are discarded.
5. **Details:** The paper should be described in a sufficiently detailed way. A paper is considered sufficiently detailed if it is possible to identify the parameters and information described in the next section.

The initial set of 257 papers was reduced to 14 papers after applying the above criteria. Papers which were of interest, even though they failed the criteria, are included in the discussion part later.

Data extraction

The first set of parameters extracted from each of the included papers relates to the platform and the hardware used in the system, namely: the type of sensor(s) used, the placement of the sensor(s), the type of robotic manipulator and the associated number of degrees of freedom. These parameters are of interest as they impact how both the computer vision and the semi-autonomous control functions. Furthermore, this information could reveal interesting tendencies in terms of hardware selection. It should be noted that the technologies for the user to interface with the different systems are not covered in this review.

The second set of parameters extracted from each paper focuses on the semi-autonomous behaviour of the different systems. This is done by identifying which parts of the system that

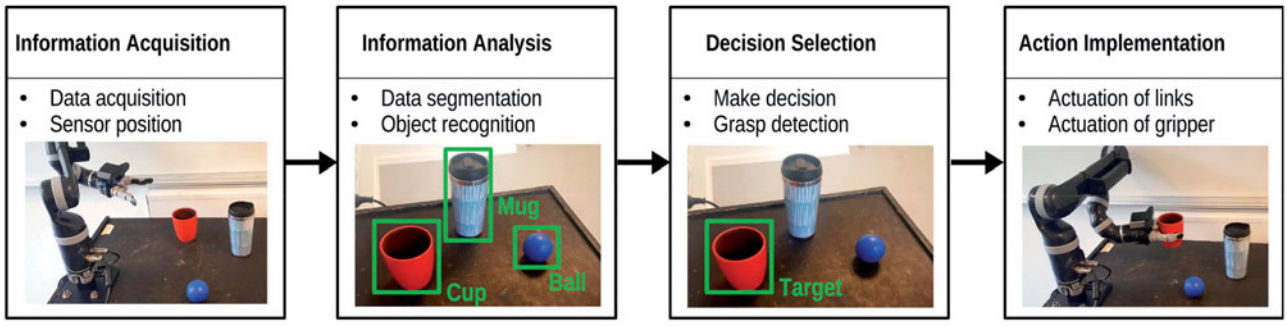

Figure 2. The four-stage model originally proposed by Parasuraman et al. [15], with examples of the tasks associated with each individual stage. The figure is adapted from Pitzer et al. [16].

Table 1. The different levels of autonomy.

Levels of autonomy
1) The system offers no assistance.
2) - offers a complete set of decisions/actions.
3) - narrows down the selection to a few.
4) - suggests one alternative.
5) - executes the suggestion if the human approves.
6) - allows the human a restricted time to veto before executing.
7) - executes automatically, then necessarily informs the human.
8) - informs the human only if asked.
9) - informs the human only if it, the system, decides to.
10) - decides everything, ignoring the human.

Adapted from Parasuraman et al. [15].

acts autonomous, using computer vision, and in which part the human is still in control. These parameters are extracted in a systematic way by using the widely cited framework proposed by Parasuraman et al. [15].

This framework suggests that a semi-autonomous system can be split into four stages, as shown in Figure 2. This model is based on a simple model of the way humans process information and acts on it. The model is hence not intended to be perfect and all-encompassing but rather a simplification making it possible to impose some structure when analyzing a system.

The different stages of this four-stage model are:

- **Stage 1: Information acquisition**
  This stage contains functions related to sensing the environment such as gathering raw data from e.g., a camera. Calculations related to depth estimation can also be considered to belong in this step, for instance, the registration between two cameras in a stereo vision setup. This stage can also include strategies for automatically moving the sensor(s) to better observe certain things. For instance, re-positioning the camera to get a better view of an object.
- **Stage 2: Information analysis**
  This stage is associated with the cognitive functions of the system. This is essentially the stage where the system interprets the information acquired during the previous stage. An example of such could be recognition of an object in an image, i.e., detecting an object's position and classifying the type of object.
- **Stage 3: Decision selection**
  The focus of this stage is to make a decision based on the multiple alternative options identified in the previous stage. The decision could, for instance, be which of the detected objects to pick-up and how to grasp said object. A system with a low level of autonomy would, for instance, offer the user all possible options. A system with a high level of autonomy would, on the other hand, act without user input and grab an object based on some pre-defined measure, e.g., the nearest object.
- **Stage 4: Action implementation**
  The final stage encompasses the actual execution of the necessary actions once a decision has been made. This includes sending the correct signals to the actuators, i.e., motors, of the robot to reach the desired goal such as the position of an object. It is also the stage responsible for actuating the gripper during grasping of objects.

Furthermore, Parasuraman et al. [15] also suggest a continuum when speaking of autonomous behaviour, ranging from a low-level to a high-level of autonomy. The authors specifically suggest 10 levels of autonomy, as outlined in Table 1. This autonomy scale mainly relates to the last two stages of the four-stage model, i.e., decision selection and action implementation, and will hence only be applied in relation to these two stages.

Results

The first part of this section summarizes the different hardware used in each of the reviewed papers. The second part outlines the semi-autonomous behaviour of each reviewed system by following the four-stage model presented earlier.

Hardware selection overview

The hardware associated with each of the reviewed systems are summarized in Table 2. It should be noted that the stated degrees of freedom (DoF) in the table refers to the ARM only. DoFs gained from mounting on mobile platforms, such as wheelchairs, are not included.

Sensor

Looking at the choice of sensor, most of the papers rely on some form of stereo vision to gather depth information with several of the papers using the Kinect v1 from Microsoft. This is a sensible choice given how easy it is to acquire and work with, but it does impose restrictions in terms of possible mounting locations as it has a minimum distance of ≈0.5 m [26]. Any object closer than that is unlikely to be captured by the sensor.

The newer model, Kinect v2, is used in some of the more recent papers [23,24]. This model relies on a time-of-flight (ToF) camera, instead of stereo vision, to acquire depth information but its minimum working distance is identical to the Kinect v1. These restrictions in terms of minimum distance are also clearly visible in the table, as no one mounts their Kinect sensors near the end-effector for this exact same reason.

Table 2. Overview of the hardware used in the different reviewed papers.

	Year	Sensor	Robotic platform
[17]	2008	**Sensor**: Passive stereo vision (custom). **Placement**: End-effector.	**Platform**: MANUS (Exact Dynamics) **Degrees of freedom**: 6
[8]	2009	**Sensor**: Passive stereo vision (custom). **Placement**: End-effector.	**Platform**: MANUS (Exact Dynamics) **Degrees of freedom**: 6
[16]	2011	**Sensor**: Active stereo vision (Kinect v1). **Placement**: Robot's head.	**Platform**: PR2 (Willow Garage) **Degrees of freedom**: 8
[18]	2012	**Sensor**: Passive stereo vision (custom) and force sensor. **Placement**: End-effector.	**Platform**: MANUS (Exact Dynamics) **Degrees of freedom**: 6
[19]	2013	**Sensor**: 2× Monocular RGB cameras. **Placement**: End-effector and overhead.	**Platform**: iARM (Exact Dynamics) **Degrees of freedom**: 6
[9]	2013	**Sensor**: Active stereo vision (2× Kinect v1). **Placement**: Table. Towards user and towards objects.	**Platform**: JACO (Kinova) **Degrees of freedom**: 6
[11]	2014	**Sensor**: Active stereo vision (Kinect v1). **Placement**: Table. Facing the user.	**Platform**: L-Exos, active wrist and hand orthosis (custom) **Degrees of freedom**: 8
[20]	2015	**Sensor**: Active stereo vision (Kinect v1). **Placement**: Overhead.	**Platform**: JACO (Kinova) **Degrees of freedom**: 6
[21]	2016	**Sensor**: Active stereo vision (2× Kinect v1). **Placement**: Table. Towards user and towards objects.	**Platform**: JACO (Kinova) **Degrees of freedom**: 6
[22]	2017	**Sensor**: Active stereo vision (Carmine). **Placement**: End-effector.	**Platform**: Baxter (Rethink Robotics) **Degrees of freedom**: 7
[10]	2017	**Sensor**: Passive stereo vision (Bumblebee). **Placement**: Overhead.	**Platform**: WAM Arm (Barrett Tech) **Degrees of freedom**: 7
[23]	2017	**Sensor**: Time-of-flight camera (2× Kinect v2). **Placement**: Table. Towards user and Towards objects.	**Platform**: JACO (Kinova) **Degrees of freedom**: 6
[24]	2017	**Sensor**: Time-of-flight camera (Kinect v2). **Placement**: Table. Towards user.	**Platform**: JACO (Kinova) **Degrees of freedom**: 6
[25]	2017	**Sensor**: Eye-tracking (EyeX) and RGB camera. **Placement**: Table.	**Platform**: Dobot Magician (Dobot) **Degrees of freedom**: 4

Hardware such as wheelchairs have been omitted from the table as the focus of this review is object manipulation using an ARM.

A few of the reviewed papers, [9,21,23], even use two Kinects with the second one being orientated towards the user of the system. The purpose being either gesture recognition and/or detection of the user's face to move e.g., food to the mouth of the user.

The papers which mount their sensor near the end-effector primarily rely on customized stereo vision setups, likely because it allows them to control the baseline distance and hence their minimum distance. The only exception being [22] utilizing the Carmine camera from PrimeSense but this device is also marketed as having a minimum distance of ≈0.35 m, making it more suitable for such a mounting location than the Kinect.

Another discerning characteristics in terms of sensor choice is whether the active or passive stereo vision is used. Active stereo vision relies on a light source, e.g., infra-red light, to actively illuminate the scene whereas passive stereo vision relies on the ambient light only. Active stereo vision is hence more robust in terms of lacking illumination. Another benefit of active stereo vision is its ability to handle lack of texture as the active light source can be used to introduce texture in the scene. Lack of texture is a general problem in stereo vision as it makes it harder to recognize the same point in two images, which is needed to estimate the depth to said point. However, most of the reviewed systems using passive stereo vision are tested using highly textured objects, for instance [8,17,18], and may hence not experience this problem during the tests.

Furthermore, all the custom stereo vision setups found in the review are of the passive variety. This is not surprising as active stereo vision setups are generally more complicated to implement due to the active light source.

Robotic platform

There is a clear tendency of using robotic arms amongst the selected papers, as only one of them relies on an exoskeleton as their platform. This tendency can likely be explained by the accessibility of robotic arms, as they are generally cheaper than an exoskeleton and more readily available in the market. This observation is further emphasized by the fact that all the robotic arms listed in Table 2 are commercially available.

The single exoskeleton, the L-Exos, is on the other hand custom made by one of the coauthors of the paper as described in Frisoli et al. [27]. This exoskeleton is also notable in the sense of its high number of degrees of freedom in comparison to the other system. It should, however, be noted that Loconsole et al. [11] states the redundancy of some of these DoFs.

Another outlier in terms of platform selection is the PR2 from Willow Garage, which is a full-blown robot featuring two arms and multiple sensors, such as the Kinect. This robot is intended to be used for teleoperation, i.e., being controlled remotely from a distance. Baxter from Rethink Robotics is a full-blown robot as well, intended for industrial purposes. Gualtieri et al. [22] did, however, describe that they salvaged an arm from a Baxter robot, essentially reducing it to an ARM on-par with e.g., JACO from Kinova. A few of the mentioned robotic platforms are shown in Figure 3.

Semi-autonomous control overview

The aspects related to the semi-autonomous behaviour of each reviewed system are summarized in Table 3. The table follows the four-stage model from earlier and seeks to characterize the autonomous behaviour of each system by highlighting how each of them deals with certain aspects associated with each stage. Note that the system presented by Quintero et al. [20] can function in two distint ways when considering the information analysis stage and the decision selection stage. This is signified by

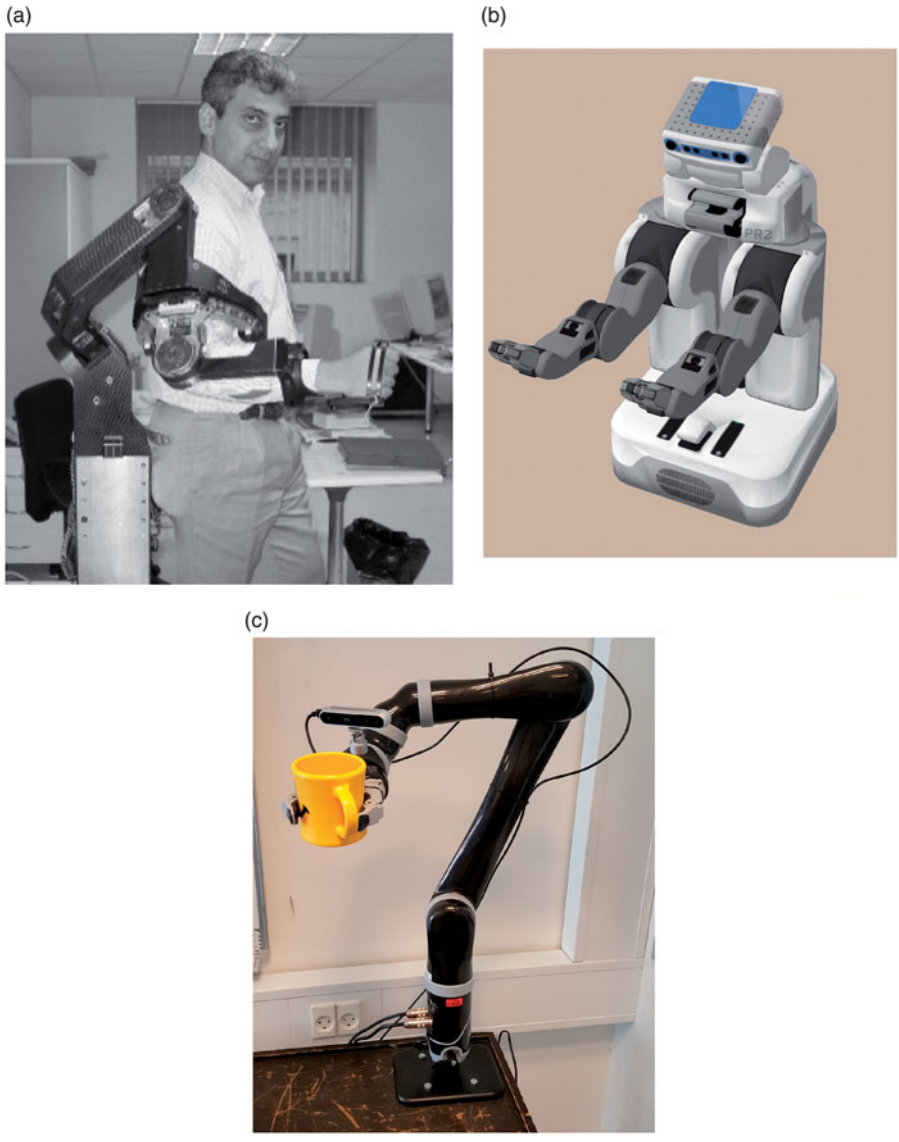

Figure 3. Example of ARMs used in the reviewed papers. (a) L-Exos [27]. (b) PR2 robot. (c) JACO.

the notation [20]a and [20]b which is used to distinguish between these two configurations of the same system.

Information acquisition

Two parameters where emphasized in this stage: (1) the type of data acquired by the system and (2) the ability to adapt the position of the sensors. Knowing the type of data gathering is important as it imposes restrictions on the system later. The ability to change the position is an important factor as well, as it influences what data that can be acquired.

All the reviewed papers collect RGB information, even though some system does not directly use it during the subsequent information analysis stage. This information is however used during the decision selection stage, to visualize different options to the user. Most of the reviewed papers do also acquire depth information, which is sensible given that the systems are expected to navigate in three dimensions to complete their task. The depth information is either represented as a point cloud or as a range image, as shown in Figure 4.

However, a few systems do not gather any depth information at all, which should, in theory, complicate tasks such as grasping objects. Remazeilles et al. [17] solve this issue by employing an optical sensor to detect when an object is inside the gripper and force sensors to ensure sufficient force when picking up the object. Another approach, used by Zeng et al. [25], is to make the simplifying assumption that all objects are cuboids and placed on a table.

Looking at the positioning of the sensors, over half of the reviewed systems employ a fixed position, which could make them susceptible to blind spots. For instance, the ARM occluding the view of the camera. A strategy to avoid such issue is to enable the user to influence the position of the sensor by mounting it on the end-effector as done in several of the reviewed systems (see Table 2 from earlier). A third option is to have the system automatically position the camera [8,18,19,22].

Information analysis

This stage can generally be characterized by two important tasks: (1) segmentation of data into what is of interest and what is not and (2) recognizing patterns in the data to recognize e.g., an object. Computer vision systems will often have well-defined strategies for these tasks which is why it was chosen to focus on these two aspects for this stage.

Looking at segmentation, a quite popular strategy is to remove the main planar surface in the scene, leaving behind objects

placed on e.g., a table. An example of such is shown in Figure 5. This planar surface is often found by using RANSAC (Random Sample Consensus) [28] to fit a plane to the point cloud data. The main drawback of this approach is the underlying assumption that the objects of interest are placed on a single planar surface without much else in the scene.

Another often used strategy amongst the reviewed papers is to rely on the user to manually perform the segmentation task. This is done either by drawing a bounding box around the object of interest or by selecting a single point on said object. In case of the latter, the point is used as the initial seed for segmentation algorithms such as the system describes by Pitzer et al. [16]. However, some of the reviewed papers [9,19], downright skip the segmentation step and are therefore processing information from the entire scene during the subsequent recognition step. This is possible as these systems rely on the SURF keypoint extractor and feature descriptor [29] which are optimized to be fast.

In terms of recognition, many of the reviewed papers rely on matching against a database of known objects. This is commonly done by extracting a set of features from the input data and then applying machine learning to match the features against the database of known objects. The majority of the papers either rely on point cloud or image data during this process, with only Jiang et al. [21] making use of both information sources in this step. Zhang et al. [23] combine both the feature extraction and matching process by training a CNN (Convolutional Neural Network) to distinguish between four pre-defined objects.

The drawback of relying on a pre-defined set of known objects is the inability to deal with objects which are not present in the database. Especially the approach by Arrichiello et al. [24] suffers from this issue as it also requires the objects to be physically marked using pre-defined tags. A much more general approach is to match against primitive shapes, for instance, cylinders [11]. In doing so, the system should be able to handle anything cylinder-shaped.

An even more general approach is used by Gualtieri et al. [22] which relies on identifying good grasping points instead of detecting objects in the scene. This essentially negates the need for both the segmentation and recognition steps for this system. The main problem of this approach is the time it takes to detect the grasping points in the scene, with the authors stating a processing time of two minutes on average.

Decision selection

In this stage, each reviewed system was sought characterized based on: (1) how the system selects what to do, for instance, what object to grasp and (2) its approach when deciding how to

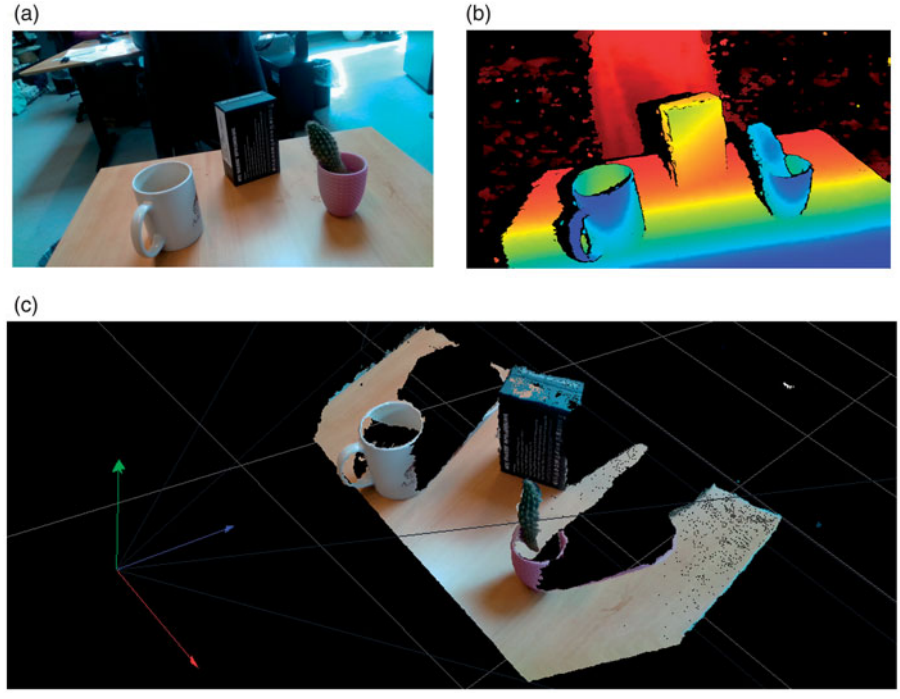

Figure 4. Different types of data from the same scene. (a) Color (RGB). (b) Range image. (c) Point cloud with RGB data.

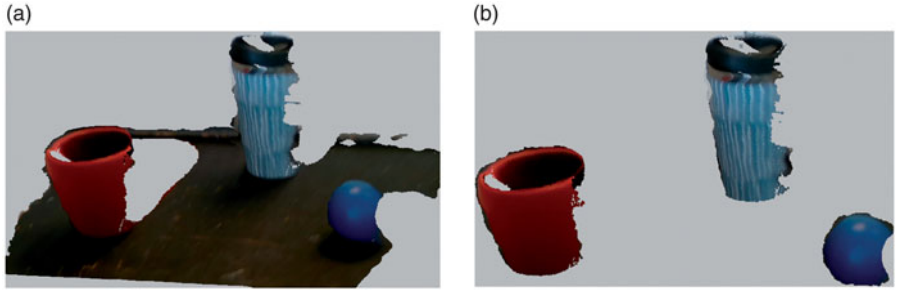

Figure 5. Example of segmentation of objects in a scene. (a) Before segmentation. (b) After segmentation.

Table 3. Overview of the semi-autonomous aspects of the different reviewed papers.

	Year	Information acquisition	Information analysis	Decision selection	Action implementation
[17]	2008	**Data**: RGB. Optical and force sensors in gripper. **Position**: User controlled.	**Segmentation**: Object bounding box from user. **Recognition**: None.	**Selection**: User draws object bounding box. **Grasping**: Estimates position but not orientation.	**Actuation**: Automatic but user can stop it at any time.
[8]	2009	**Data**: RGB and point cloud. Force sensors in gripper. **Position**: Automatic centreing of user's selection.	**Segmentation**: Object point from user. **Recognition**: Template matching against database.	**Selection**: User selects single point on object. **Grasping**: Estimates position and orientation.	**Actuation**: User controls the gripper and the rate of link actuation.
[16]	2011	**Data**: RGB, point cloud and range images. **Position**: User controlled.	**Segmentation**: Object bounding box from user. **Recognition**: Matches point cloud against database.	**Selection**: User draws object bounding box. **Grasping**: Pre-computed for objects in the database.	**Actuation**: Automatic.
[18]	2012	**Data**: RGB and point cloud. Force sensors in gripper. **Position**: User controlled. Automatic centreing of user's selection.	**Segmentation**: Object point from user. **Recognition**: Matches RGB image against database.	**Selection**: User selects single point on object. **Grasping**: Estimates position and orientation.	**Actuation**: Automatic.
[19]	2013	**Data**: RGB and point cloud. **Position**: Overhead camera is fixed. System controls end-effector camera.	**Segmentation**: None. **Recognition**: Matches RGB image against database.	**Selection**: Presents multiple graspable objects. User selects one object. **Grasping**: Pre-defined for objects in the database.	**Actuation**: Automatic.
[9]	2013	**Data**: RGB and range image. **Position**: Fixed.	**Segmentation**: None. **Recognition**: Matches RGB image against database.	**Selection**: Object(s) to be grasped was pre- defined. **Grasping**: Done by the user.	**Actuation**: Coarse control done automatic. Fine control done by user.
[11]	2014	**Data**: RGB and point cloud. **Position**: Fixed.	**Segmentation**: Distance thresholding. **Recognition**: Recognizes cylinder-shaped objects.	**Selection**: Any cylinder- shaped objects detected. **Grasping**: Estimates position and orientation.	**Actuation**: Automatic.
[20]a	2015	**Data**: RGB and point cloud. **Position**: Fixed.	**Segmentation**: Object bounding box from user. **Recognition**: None.	**Selection**: User draws object bounding box. **Grasping**: Estimates position and orientation.	**Actuation**: Either fully automatic or the user controls the rate of actuation.
[20]b	2015	**Data**: RGB and point cloud. **Position**: Fixed.	**Segmentation**: Remove main planar surface. **Recognition**: None.	**Selection**: User Selects from list of detected objects. **Grasping**: Estimates position and orientation.	**Actuation**: Either fully automatic or the user controls the rate of actuation
[21]	2016	**Data**: RGB and point cloud. **Position**: Fixed.	**Segmentation**: Remove main planar surface. **Recognition**: Matches image and point cloud against database.	**Selection**: User selects from a set of pre-defined actions. **Grasping**: Pre-defined for objects in the database.	**Actuation**: Automatic. User input is queued until the robot is done executing.
[22]	2017	**Data**: RGB and point cloud. **Position**: Automatic. System gathers data from multiple viewpoints.	**Segmentation**: None. **Recognition**: None.	**Selection**: User selects object using laser pointer. **Grasping**: Estimates position and orientation.	**Actuation**: Automatic.
[10]	2017	**Data**: RGB and point cloud. **Position**: User controlled.	**Segmentation**: Remove main planar surface. **Recognition**: Matches point cloud against database.	**Selection**: Intent infered using end-effectors position and orientation. **Grasping**: Pre-defined for objects in the database.	**Actuation**: Blending between system and user based on confidence of infered intent.
[23]	2017	**Data**: RGB and point cloud. **Position**: Fixed.	**Segmentation**: Region growing using normals. **Recognition**: Matches image against database.	**Selection**: User selects from a set of pre-defined actions. **Grasping**: Estimates position but not orientation.	**Actuation**: Automatic.

(*continued*)

Table 3. Continued.

	Year	Information acquisition	Information analysis	Decision selection	Action implementation
[24]	2017	**Data**: RGB and point cloud. **Position**: Fixed.	**Segmentation**: None. **Recognition**: Marker-based.	**Selection**: User selects from a set of pre-defined actions. **Grasping**: Estimates position. Orientation is pre-defined.	**Actuation**: Automatic.
[25]	2017	**Data**: RGB and gaze points. **Position**: Fixed.	**Segmentation**: Colour thresholding. **Recognition**: None. All objects are cuboids.	**Selection**: User selects from detected objects using gaze. **Grasping**: Estimates position and orientation.	**Actuation**: Coarse control done automatic. Fine control done by user.

Table 4. The reviewed papers and their level of autonomy based on their decision selection behaviour (stage 3). The indicators (a,b) signifies different configurations of the same system, as outlined in Table 3.

	Decision selection (level of autonomy)
Level 1	[8,16–18,22] and [20]a
Level 2	[25] and [20]b
Level 3	[9,19,21,23,24]
Level 6	[10]
Level 10	[11]

grasp an object, i.e., how to position and orient the end-effector of the ARM for the grasping procedure.

In relation to the decision selection of each reviewed system, it is quite natural to also consider the associated level of autonomy. The scale presented earlier, see Table 1, have hence been used to determine the level of autonomy for each system. The result is shown in Table 4.

Many of the reviewed systems rely on the user directly selecting the object to interact with. This is either done by having the user draw a bounding box around the object or selecting a point on it, as mentioned above. Such approaches rely entirely on the user and can hence be associated with the lowest level of autonomy. A few systems are a bit more restrictive, as they narrow the user's options down, either based on the objects detected in the scene by the system or a pre-defined set of options. These systems are given a rating of 2 and 3, respectively.

The system by Loconsole et al. [11] is however characterized by a high level of autonomy, as it will try to grasp any cylinder-shaped object presented to it. This system has been given a rating of 10, as the user has no say in the matter. Another outlier is the system described by Mülling et al. [10], as it automatically infers the intention of the user based on the end-effectors proximity to objects and how well the end-effectors orientation aligns with these objects. The system will automatically start to act on this estimated intention, but the user can still veto this decision by moving the end-effector in another direction, hence the rating of 6.

An important part of the decision selection stage is to figure out how to grasp an object to manipulate it. This entails figuring out how to position and orient the end-effector for the best grasp. How much to close the gripper is an important step in the grasping procedure as well but this part is not included in this review.

Several of the reviewed systems rely on both grasp positions and poses being pre-defined for a set of known objects. The drawback of relying on pre-defined information for a small set of objects is the inability to handle unknown objects, as stated earlier. Other papers ignore the problem of identifying a proper orientation of the end-effector and only estimate where to position the end-effector for grasping. This approach is possible as these systems make assumptions like the objects always being placed such that their major axis is aligned vertically with the ARM. Such assumptions restrict the system's ability to function in an uncontrolled environment, where the objects are likely to be placed arbitrarily.

A few of the reviewed systems, like Kim et al. [8,18], uses a PCA-based approach (principal component analysis) in order to estimate the major axis of each object. This approach relies on point cloud data for each object and is hence dependent on proper segmentation of the object. Another drawback is that the estimated axis of the object may easily be miscalculated in cases where parts of the object are not present in the point cloud.

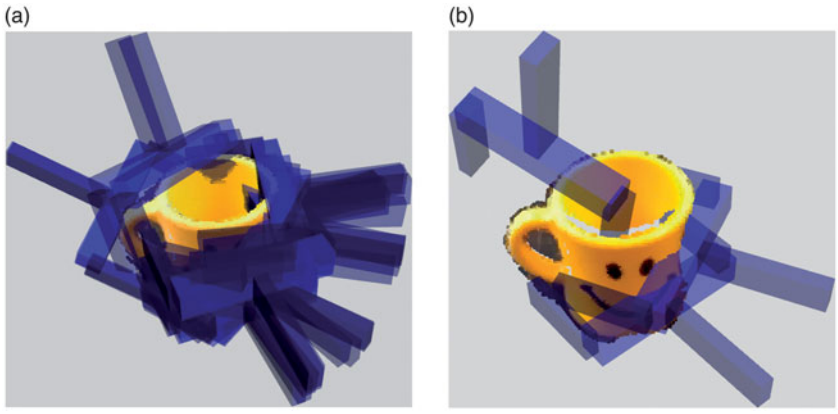

Figure 6. Detecting grasping poses for an arbitrary object using the algorithm by Gualtieri et al. [22]. (a) Grasp synthesis stage. (b) Grasp analysis stage.

Table 5. The reviewed papers and their level of autonomy based on their action implementation behaviour (stage 4).

Action	**implementation** (level of autonomy)
Level 5	[8,20]
Level 6	[17]
Level 7	[9,25]
Level 10	[11,16,18,19,21–24]
Adaptive	[10]

The approach used by Loconsole et al. [11] avoids this issue as it is quite straightforward to extract the major axis of a cylinder, which the system identified during the information analysis step. The disadvantage of this approach is the underlying assumption that every object is cylinder-shaped.

The only reviewed system which is truly able to grasp arbitrary objects is the one described by Gualtieri et al. [22] as it relies on detecting good grasping poses. The process of detecting these grasping poses is split into two stages; grasp synthesis and grasp analysis, as illustrated in Figure 6. The synthesis stage seeks to generate a large number of grasp candidates whereas the analysis seeks to reduce the larger number of candidates to a few good ones. It should be noted that Mülling et al. [10] describes a similar extension of their system in their future works which enables them to handle arbitrary objects as well.

Action implementation

This stage is sought characterized by considering who is in control of the ARM's movement, i.e., who controls the actuation of the ARM's links and its end-effector. Factors like how trajectories are planned could have been considered as well in this stage, but it is deemed outside the scope of this review.

The action implementation stage is also a good candidate for judging a system's level of autonomy and this stage has hence been mapped using the autonomy scale as well. The result is shown in Table 5.

The majority of the reviewed systems are assigned a score of 10, as the actuation of both the ARM's links and end-effector is fully automatic once a decision has been made. Jiang et al. [9] and Zeng et al. [25] are assigned a lower score of 7, as these two systems rely on the idea of dividing the control of the ARM into fine and coarse control. Coarse control entails moving the end-effector to the general position of the object to manipulate and is done automatically. Fine control deals with grasping the object and is initiated by the system, which then asks the user to take over and perform the grasping. It should be noted that Zeng et al. [25] does estimate the orientation of the object to be grasped but this information is only used to guide the user during fine control.

The system by Remazeilles et al. [17] is assigned a rating of 6 as it essentially allows the user to veto the automatic actuation of the ARM. Kim et al. [8] and Quintero et al. [20] employs a scheme where the user continuously has to allow the system to operate automatically, for instance by holding down a button. This results in a score of 5 as the system is essentially limited to only executing actions if the user approves. Finally, the system by Mülling et al. [10] is not assigned a score as the automation level changes depending on the system's confidence in inferring the intention of the user. For instance, the user will be completely in control if the system has no idea about the intention of the user. Furthermore, it should be noted that the authors of Parasuraman et al. [15] do point out that their framework fails to encompass such adaptive automation well.

Level of autonomy summary

The purpose of this section is to summarize the results related to the level of autonomy of the reviewed systems to highlight tendencies. This is done by grouping each of the reviewed papers, as shown in Table 6.

These groups are created by grouping systems where the level of autonomy is identical for both the decision selection and action implementation stage. The resulting groups are then plotted, as shown in Figure 7, with respect to their level of autonomy for the decision selection and action implementation stage.

Looking at the plot in Figure 7, it is quite clear that most of the reviewed systems are placed in the upper left quadrant. Such systems are characterized by having a quite clear-cut strategy for sharing control, as the user decides what to grasp whereas the system performs the actual grasping. These approaches are hence quite similar to e.g., the claw machines found at arcades; the user points the machine towards the object of interest, the user presses a button and the machine takes over. A few of the reviewed systems did, however, allow the user some control in such scenarios, for instance, the systems found in group C and E. These systems rely on constant confirmation from the user, e.g., holding down a button, to continue executing the planned action.

An entirely different approach for semi-autonomous control can be seen in group G consisting of only the system by Mülling et al. [10]. This group differs from the others due to its adaptive

Table 6. Grouping of the reviewed systems based on their level of autonomy for the decision selection and action implementation stage. The indicators (a,b) signifies different configurations of the same system, as outlined in Table 3.

Group	Paper(s)
A	[19,21,23,24]
B	[16,18,22]
C	[8] and [20]a
D	[17]
E	[20]b
F	[25]
G	[10]
H	[9]
I	[11]

nature which is also sought illustrated in Figure 7 by having this group span the entire action implementation continuum.

Another outlier is group I, consisting of the system by Loconsole et al. [11], which have the highest possible level of autonomy for both its decision selection and action implementation stage. It can hence be argued that this system is fully autonomous and hence of no interest when discussing semi-autonomous systems. To be fair, it should be noted that the focus of Loconsole et al. [11] is skewed towards computer vision and not semi-autonomous control.

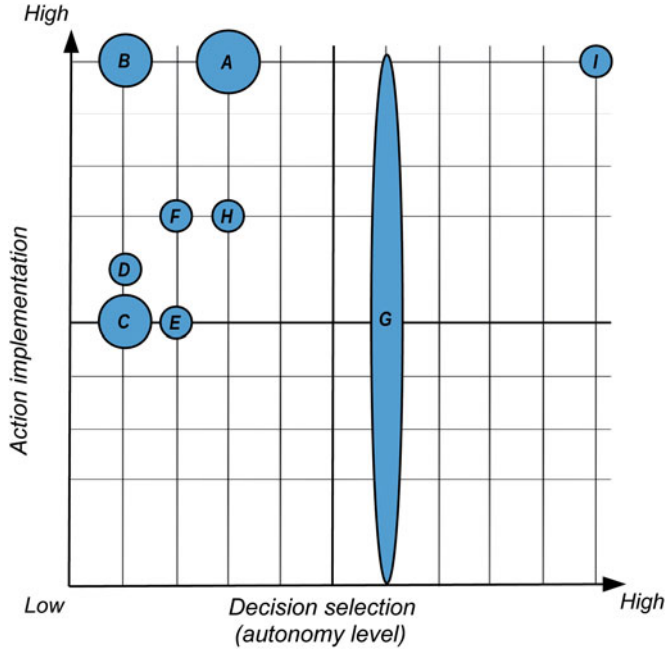

Figure 7. Plot of the groups from the Table 6. The size of each circle increases with the number of members in the group. The large span of group G signifies the adaptive nature of the action implementation stage for the system by Mülling et al. [10].

Discussion

The purpose of this section is to expand upon the findings in the previous sections by pointing out challenges in relation to reviewed systems and suggest further potential avenues to explore. Three challenges will be discussed; ensuring optimal semi-autonomous control, handling arbitrary objects and sensing the environment.

Challenge: optimal semi-autonomous control

Most of the reviewed systems tend to rely on pre-defined roles for respectively the human and the system, as shown earlier in Figure 7. The user decides what to do and the system takes over control, thereby creating this claw machine-like behaviour. The benefit of such schemes is that the user is never in doubt as to who is in control at any time.

However, this behaviour could be problematic as the user has no or very limited control once the system is in charge. This issue was outlined in a study by Chung et al. [13] which found that the users felt less accomplished when relying entirely on the system to complete the task automatically. The participants did, in fact, experience a lower level of satisfaction, despite completing the task faster, due to this lack of accomplishment. A similar observation was made by Kim et al. [14] where individuals with movement impairment appeared less inclined to relinquish control of the ARM than able-bodied persons.

A way to address the above issue could be to rely on adaptive semi-autonomous control, as seen in the system by Mülling et al. [10]. Such a scheme will allow the user some control throughout the entire process, thereby providing the user with some sense of accomplishment when finishing a task, while still aiding the user to some extent.

This form of semi-autonomous control can be viewed as an arbitration of control between the system and the user which can essentially be reduced to a linear blending, controlled by the arbitration factor α, as shown in Figure 8. This is also the approach used by Mülling et al. [10] where α is computed using a sigmoid function dependent on the confidence of the goal predicted by the system.

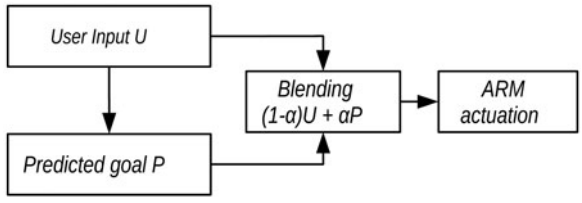

Figure 8. Arbitration between the user U and the goal P predicted by the system using linear blending. The figure is adapted from Dragan et al. [30].

The idea of viewing the arbitration as a blending problem is based on the work by Dragan et al. [30], which also uses the plots of different arbitration factors α to characterize the behaviour of semi-autonomous systems. An example of different arbitration behaviours is shown in Figure 9.

The idea of defining the behaviour of the system using arbitration functions may also make it easier to customize the behaviour of the system to the preference of the user. A lot of different behaviours can be achieved by simply changing the function governing the arbitration factor used during the blending. For instance, the behaviour of the system by Mülling et al. [10] can be characterized by Figure 9(b) whereas the behaviour of e.g., [16,18] can be characterized by Figure 9(d) as the action implementation stage is always fully automatic for these systems. Customization through these arbitration curves may also be beneficial due to their visual nature making it easier to understand for people with a non-technical background.

However, an important prerequisite for adaptive semi-autonomous control is for the system to be able to gauge its confidence. For instance, how confident the system is that the user is reaching for object B and not object A. One way of inferring confidence in this scenario could be to rely on proximity, i.e., how close is the end-effector of the ARM to each object. Such proximity-based approaches are used in the work of Mülling et al., Dragan et al. and Downey et al. [10,30,31]. A possible downside of proximity-

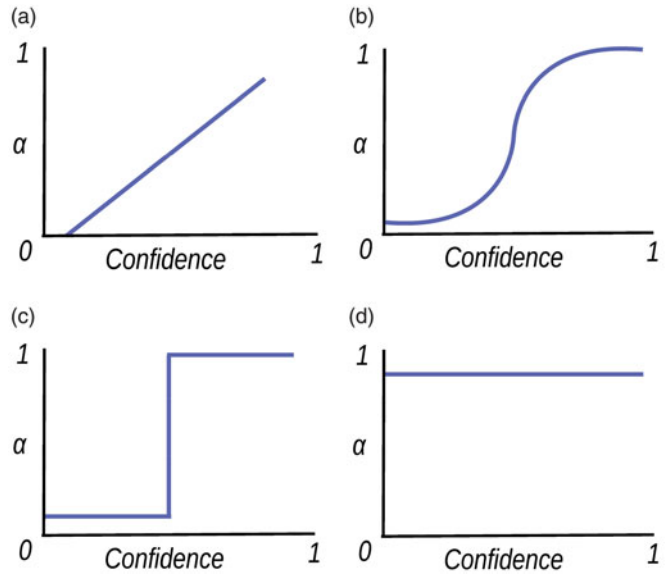

Figure 9. Examples of different functions which can be used to control the arbitration factor. (a) Linear function. (b) Sigmoid function. (c) Unit step function. (d) Constant function.

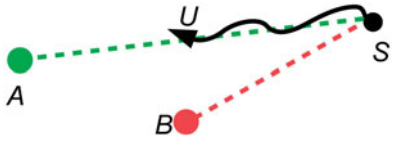

Figure 10. A scenario with two objects, A and B, with an ARM denoted by the current position U and initial position S of its end-effector. The figure is adapted from Dragan et al. [30].

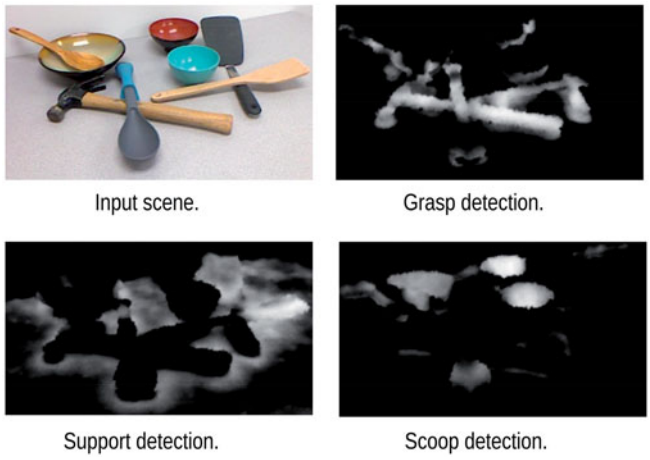

Figure 11. Example of affordances detected in a scene. The input scene is from the dataset published by Myers et al. [33].

based approaches is that they are memory-less, i.e., they only consider the system in its current state. An example of why this lack of memory can be problematic is shown in Figure 10, where the user is reaching for object A but the system misinterprets the user's goal as being object B due to the proximity-based approach.

A way to introduce memory into the process of inferring the intention of the user is to consider the trajectory of the ARM, as done by Dragan et al. and Schultz et al. [30,32]. Looking at Figure 10 again, it is possible to see that considering the trajectories it would have been possible to correctly infer that the user was reaching for object A.

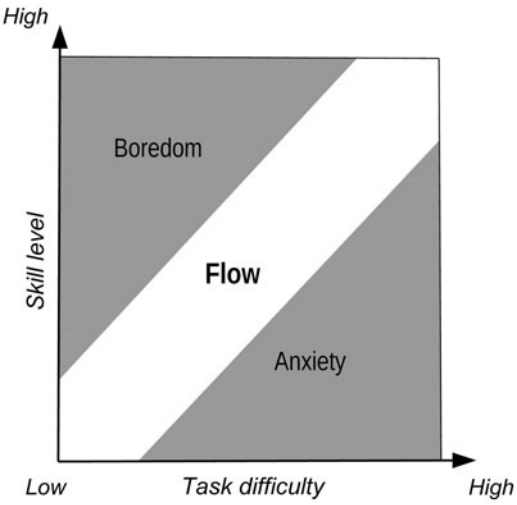

Figure 12. Illustration of how flow can be achieved by matching task difficulty and skill level. The figure is adapted from Csikszentmihalyi et al. [34].

The different measures of confidence, mentioned above, are all related to stage 3, i.e., decision selection. However, it is possible to expand the model to make use of confidences derived from the other stages as well. For instance, the confidence of the sensor data gathered during the information acquisition stage. Another idea could extract a confidence measure from the information analysis stage based on how certain objects are commonly used.

This idea could be achieved through affordance detection, with affordance being the notion that objects "invite" the user to interact with them in certain ways. A handle on a mug would, for instance, be an obvious affordance for grasping. The idea of grasp detection, as discussed earlier, could hence be considered a limited form of affordance detection, which only focuses on the affordance related to grasping. However, multiple other affordances exist, for instance; cutting, scooping, containing, pounding and supporting. These affordances are the focus in the work by for instance Myers et al. [33], which proposes a way of detecting different affordances using RGB and depth information. An example of their results is shown in Figure 11.

Affordance detection could hence be useful in scenario where the user is trying to accomplish a task involving multiple objects. For instance, using a spoon to scoop something or when the user wants to pour a liquid into a container.

Yet another possibility is to incorporate the system's confidence in the user, which is suggested by Dragan et al. [30] as well. An interesting addition to this idea could be for the system to provide a level of assistance which keeps the user in a state of flow or "being in the zone". The idea of flow is described as a mental state where the user would feel a sense of mastery and satisfaction by ensuring that the difficulty of the tasks matches the skill of the user [34].

This idea is often illustrated as shown in Figure 12, where a person is kept in a state of mental flow by matching the difficulty of the task with the skill level of the person. Failure to match these two parameters could cause a person to enter either a state of boredom or anxiety, which is not desirable. The presence of the flow state can hence influence a person's sense of accomplishment and satisfaction which is why it could be interesting to consider it in relation to semi-autonomous control.

Challenge: handling arbitrary objects

Another challenging aspect of using computer vision for semi-autonomous control of ARMs is to be able to handle arbitrary objects, i.e., objects never encountered by the system before. Most of the reviewed papers seem to agree that this is an important issue but only a few of them actually address it.

Looking at the reviewed systems, both Gualtieri et al. [22] and Mülling et al. [10] address this issue by discarding the notion of detecting separate objects and instead rely on detecting good grasping points on an arbitrary object. However, detecting good grasping points can be rather slow [22] due to the synthesis stage which is time-consuming because of the large search space (six variables; three for grasp position and three for grasp orientation).

A way to speed up this process could be to look into approaches [35,36], which rely on a range image instead of a point cloud data during grasp synthesis. Domae et al. [35] report a processing time of 0.31 s or less, making it significantly faster than Gualtieri et al. [22]. The work by Redmon and Angelova [37] and Lenz et al. [38] relies on both RGB and range images to infer grasping points using CNNs, with Redmon and Angelova [37] reporting processing times of ≈77 ms. The low processing time is likely because a GPU is used to accelerate the computations by taking advantage of the highly parallelizable nature of CNNs.

Another way of handling arbitrary objects is to decompose them into primitive shapes like cylinders, cuboids and spheres, as illustrated in Figure 13. This is somewhat similar to the idea used by Loconsole et al. [11], which focussed on cylinder-shaped objects only. The idea is to expand this approach to encompass any object by including more shapes than just cylinders and by allowing these shapes to be combined [39,40].

How to handle arbitrary objects is not necessarily limited to one of the approaches mentioned above. In fact, combining multiple approaches could be a viable solution. An example of such is the work by Ciocarlie et al. [41], which defines a grasping procedure for known objects and a procedure for unknown objects not encountered before. This idea is especially interesting in the scope of semi-autonomous control as an unknown object could be added to the set of known objects by the user showing the system how to grasp said object. This is somewhat similar to the approach by Herzog et al. [42], where the system learns grasp poses through demonstration by the user. The work of Krainin et al. [43] could also help expand this idea as it describes an approach for creating 3 D models of objects once they have been grasped by a robotic manipulator.

Figure 13. Example of an object and its decomposition into primitive shapes. This approach was used by Miller et al. [39].

Challenge: sensing the environment

The last challenge which this paper will touch upon is how to acquire complete and precise data about the environment that the ARM is to operate in. These aspects are important as the subsequent stages in any system will suffer if the information acquisition stage is not up to par.

Roughly half of the reviewed papers decided to mount their sensors near or on the end-effector, a configuration sometimes called eye-in-hand. Such as configuration is advantageous as it is near impossible for the ARM to occlude the view of the sensor and it offers some flexibility, as the sensor can be re-positioned using the ARM. Allowing the user to re-position the sensor, by controlling the end-effector, could also make it easier to infer the intention of the users as they would likely orient the end effector towards the object they are interested in. A few of the reviewed papers [8,18], utilize this option by having the system automatically re-positioning the end-effector such that the user's selection is centred in the view of the camera. The idea is to get a better view of the object to interact with.

The work by Gualtieri et al. [22] takes this approach a step further by re-positioning the sensor to gather information from multiple viewpoints in order to increase the quality of the gathered point cloud. The authors specifically state that doing so have shown an improvement in grasp detection according to their prior work [44]. This idea is somewhat similar to the work by Klingensmith et al. [45] where an end-effector mounted depth sensor is used to map the nearby environment using a SLAM-like approach (Simultaneous Localization And Mapping). The authors demonstrate that their approach improves the quality of the data gathered while continuously estimating the position of the end-effector, i.e., the localization part of SLAM. Employing some strategy for accumulating data from multiple viewpoints may hence be beneficial when dealing with an eye-in-hand configuration [22,45].

Another way to improve upon the depth information acquired by the system could be to use techniques for depth completion [46]. The idea is to use information from a colour image to estimate the missing depth information, as shown in Figure 14. The colour image is used to estimate surface normals for the entire scene which are then combined with the sparse set of depth measurements to infer depth for the entire scene.

The main drawback of this work is the processing time, as the authors state a processing time of between 0.3 and 1.5 s, depending on the hardware used. It can be argued that the processing time is not a big issue as it may not be necessary to use depth completion on every single frame received from the sensor. However, a benefit of this approach is that it will work with sensors mounted in a fixed position, as opposed to the SLAM-like approaches mentioned earlier.

An area which could also improve the system's ability to sense its immediate environment is the actual sensors employed by each system. Most of the reviewed systems rely on stereo vision to gather depth information and it may hence be interesting to explore other options such a ToF camera like the Kinect v2 used by Zhang et al. and Arrichiello et al. [23,24]. The absence of ToF cameras amongst the other reviewed paper can likely be attributed to the available ToF cameras at the time, which were likely expensive and bulky.

A general difference between ToF cameras and stereo vision cameras is that the former does not rely on a baseline to estimate depth. ToF cameras can hence be made more compact, making it possible to mount them in a location not possible for stereo vision cameras. For instance, inside the gripper of the ARM. An

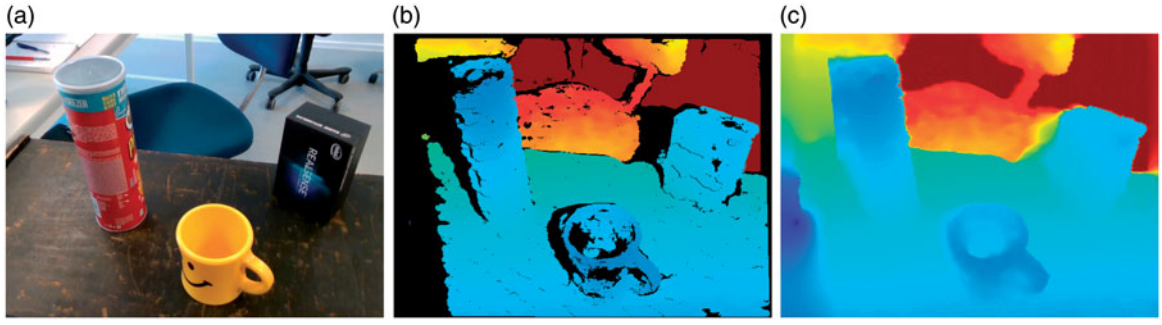

Figure 14. Example of depth completion using the algorithm from Zhang and Funkhouser [46]. (a) Color information. (b) Depth information. (c) Completed depth.

example of such is the CamBoard Pico flexx camera from PMD Technologies [47] which is significantly smaller than the Kinect sensors while featuring a minimum working distance of 0.1 m.

Conclusion

The focus of this review paper was on computer vision systems enabling movement impaired individuals to do object manipulation using an assistive robotic manipulator (ARM). The initial literature search yielded 257 results which were narrowed down to 14 relevant papers. These papers where reviewed in relation to their selection of hardware and their use of computer vision for the semi-autonomous behaviour of the system. Different schemes for the semi-autonomous control were reviewed as well. A four-stage model was used during the review of each system to characterize their behaviours in terms of; information acquisition, information analysis, decision selection and action implementation. A scale, consisting of 10 levels [15], was used to rate the autonomy of each system as well.

The reviewed papers mainly made use of stereo vision-based sensors to capture depth information. Many of the papers used the Kinect from Microsoft which was often mounted near the shoulder or head of the user, viewing the scene from a distance. The second most popular sensor placement was at the end-effector, making it possible for both the user and the system to re-position the sensor. However, only a few of the reviewed systems fully utilized this option by mapping the immediate environment using data from multiple viewpoints. Furthermore, exploring other options in terms of sensor choice may be interesting as well. For instance, a small ToF camera which could be mounted inside the gripper of the ARM.

Handling of arbitrary objects was found to be a general issue with only a few of the reviewed systems being able to do so. The majority made simplifying assumptions such as all objects having a certain shape or all object being in a database of pre-defined objects. A way to approach the issue of handling arbitrary objects could be to reduce it to a problem of detecting good grasping points or decomposing objects into primitive shapes.

Most of the reviewed papers rely on a clear switch between the user and the system for the semi-autonomous control of said system. Adaptable automation, in the form of linear blending, is used in one of the reviewed papers but should be explored further. Such a scheme could be beneficial as it allows the user some control at all times which is especially important for movement-impaired users. A scheme based on linear blending may also allow for easy customization of the semi-autonomous control. Such an adaptive approach may also benefit from the concept of flow, known from psychology, to adjust the level of assistance based on the skill level of the user and the difficulty of the task at hand.

To summarize; there is a substantial amount of on-going research focusing on using computer vision for semi-autonomous control of ARMs. Several working prototypes have demonstrated that this idea can work in a controlled environment, such as a lab. The next big step is to advance the technology to a point where it is possible to move beyond the labs and into the home of the actual user. The benefit of doing so would be priceless for the individual user, and society, in general, may benefit as well due to less demand for caregivers.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

[1] Marasinghe KM. Assistive technologies in reducing caregiver burden among informal caregivers of older adults: a systematic review. Disabil Rehabil Assist Technol. 2016;11:353–360.

[2] Bedaf S, Marti P, Amirabdollahian F, et al. A multi-perspective evaluation of a service robot for seniors: the voice of different stakeholders. Disabil Rehabil Assist Technol. 2017;0:1–8.

[3] Romer G, Stuyt H, Peters A. Cost-savings and economic benefits due to the assistive robotic manipulator (arm). 9th International Conference on Rehabilitation Robotics, 2005 ICORR; 2005 June 28–July 1. Chicago, IL, USA.

[4] Maheu V, Archambault PS, Frappier J, et al. Evaluation of the jaco robotic arm: clinico-economic study for powered wheelchair users with upper-extremity disabilities. IEEE Int Conf Rehabil Robot. 2011;2011:1–5.

[5] Prior SD. An electric wheelchair mounted robotic arm a survey of potential users. J Med Eng Technol. 1990;14:143–154.

[6] KINOVA. Robotic arms series; 2018 [Cited 2019 May 20]. Available from: https://www.kinovarobotics.com/en/products/robotic-arm-series

[7] Exact Dynamics. iARM; 2018. Available from: http://www.exactdynamics.nl/site/?page=iarm. (visited on 05/20/2019)

[8] Kim DJ, Lovelett R, Behal A. An empirical study with simulated adl tasks using a vision-guided assistive robot arm. In: 2009 IEEE International Conference on Rehabilitation Robotics; 2009 June 23–26; Kyoto, Japan. p. 504–509.

[9] Jiang H, Wachs JP, Duerstock BS. Integrated vision-based robotic arm interface for operators with upper limb

mobility impairments. IEEE Int Conf Rehabil Robot. 2013; 2013:6650447.
[10] Mülling K, Venkatraman A, Valois J, et al. Autonomy infused teleoperation with application to brain computer interface controlled manipulation. Autonomous Robots. August 2017.
[11] Loconsole C, Stroppa F, Bevilacqua V, et al. A robust real-time 3d tracking approach for assisted object grasping. In: Auvray M, Duriez C, editors. Haptics: neuroscience, devices, modeling, and applications. Berlin, Heidelberg: Springer Berlin Heidelberg; 2014. p. 400–408.
[12] Oguntosin VW, Mori Y, Kim H, et al. Design and validation of exoskeleton actuated by soft modules toward neurorehabilitation vision-based control for precise reaching motion of upper limb. Front Neurosci. 2017;11:352.
[13] Chung CS, Wang H, Cooper RA. Functional assessment and performance evaluation for assistive robotic manipulators: literature review. J Spinal Cord Med. 2013;36:273–289.
[14] Kim DJ, Hazlett-Knudsen R, Culver-Godfrey H, et al. How autonomy impacts performance and satisfaction: results from a study with spinal cord injured subjects using an assistive robot. IEEE Trans Syst, Man, Cybern A Syst Hum. 2012;42:2–14.
[15] Parasuraman R, Sheridan TB, Wickens CD. A model for types and levels of human interaction with automation. IEEE Trans Syst Man Cybern A Syst Hum. 2000;30:286–297.
[16] Pitzer B, Styer M, Bersch C, et al. Towards perceptual shared autonomy for robotic mobile manipulation. In: 2011 IEEE International Conference on Robotics and Automation; 2011 May 9–13; Shanghai, China. p. 6245–6251.
[17] Remazeilles A, Leroux C, Chalubert G. Sam: A robotic butler for handicapped people. In: RO-MAN 2008 - The 17th IEEE International Symposium on Robot and Human Interactive Communication; 2008 Aug 1–3; Munich, Germany. p. 315–321.
[18] Kim DJ, Wang Z, Behal A. Motion segmentation and control design for ucf-manus;an intelligent assistive robotic manipulator. IEEE/ASME Trans Mechatron. 2012;17:936–948.
[19] Elarbi-Boudihir M, Al-Shalfan KA. Eye-in-hand/eye-to-hand configuration for a wmra control based on visual servoing. In: 2013 IEEE 11th International Workshop of Electronics, Control, Measurement, Signals and their application to Mechatronics; 2013 June 24–26; Toulouse, France. p. 1–6.
[20] Quintero CP, Ramirez O, Jägersand M. Vibi: Assistive vision-based interface for robot manipulation. In: 2015 IEEE International Conference on Robotics and Automation (ICRA); 2015 May 26–30; Seattle, WA, USA. p. 4458–4463.
[21] Jiang H, Zhang T, Wachs JP, et al. Enhanced control of a wheelchair-mounted robotic manipulator using 3-d vision and multimodal interaction. Comput Vis Image Underst. 2016;149:21–31.
[22] Gualtieri M, Kuczynski J, Shultz AM, et al. Open world assistive grasping using laser selection. In: IEEE International Conference on Robotics and Automation (ICRA); IEEE; 2017 May 29–June 3; Singapore, Singapore. p. 4052–4057.
[23] Zhang Z, Huang Y, Chen S, et al. An intention-driven semi-autonomous intelligent robotic system for drinking. Front Neurorobot 2017;11:48.
[24] Arrichiello F, Lillo PD, Vito DD, et al. Assistive robot operated via p300-based brain computer interface. In: 2017 IEEE International Conference on Robotics and Automation (ICRA); 2017 May 29–June 3; Singapore, Singapore. p. 6032–6037.
[25] Zeng H, Wang Y, Wu C, et al. Closed-loop hybrid gaze brain-machine interface based robotic arm control with augmented reality feedback. Front Neurorobot. 2017;11:60.
[26] Microsoft. Kinect Sensor; 2018. Available from: https://msdn.microsoft.com/en-_us/library/hh438998.aspx (visited on 05/20/2019)
[27] Frisoli A, Salsedo F, Bergamasco M, et al. A force-feedback exoskeleton for upper-limb rehabilitation in virtual reality. Applied Bionics Biomech. 2009;6:115–126.
[28] Fischler MA, Bolles RC. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun Acm. 1981;24: 381–395.
[29] Bay H, Ess A, Tuytelaars T, et al. Speeded-up robust features (surf). Comput Vis Image Underst. 2008;110:346–359.
[30] Dragan AD, Srinivasa SS. A policy-blending formalism for shared control. Int J Robot Res. 2013;32:790–805.
[31] Downey JE, Weiss JM, Muelling K, et al. Blending of brain-machine interface and vision-guided autonomous robotics improves neuroprosthetic arm performance during grasping. J NeuroEng Rehabil. 2016;13:28.
[32] Schultz C, Gaurav S, Monfort M, et al. Goal-predictive robotic teleoperation from noisy sensors. In: IEEE International Conference on Robotics and Automation (ICRA); IEEE; 2017 May 29–June 3; Singapore, Singapore.
[33] Myers A, Teo CL, Fermüller C, et al. Affordance detection of tool parts from geometric features. In: IEEE International Conference on Robotics and Automation (ICRA); 2015 May 26–30; Seattle, WA, USA. p. 1374–1381.
[34] Csikszentmihalyi M. Flow and the foundations of positive psychology: the collected works of mihaly csikszentmihalyi. Dordrecht: Springer; 2014.
[35] Domae Y, Okuda H, Taguchi Y, et al. Fast graspability evaluation on single depth maps for bin picking with general grippers. In: IEEE International Conference on Robotics and Automation (ICRA); IEEE; 2014 May 31–June 7; Hong Kong, China. p. 1997–2004.
[36] Klingbeil E, Rao D, Carpenter B, et al. Grasping with application to an autonomous checkout robot. In: IEEE International Conference on Robotics and Automation (ICRA); IEEE; 2011 May 9–13; Shanghai, China. p. 2837–2844.
[37] Redmon J, Angelova A. Real-time grasp detection using convolutional neural networks. In: IEEE International Conference on Robotics and Automation (ICRA); IEEE; 2015 May 26–30; Seattle, WA, USA. p. 1316–1322.
[38] Lenz I, Lee H, Saxena A. Deep learning for detecting robotic grasps. Int J Robot Res. 2015;34:705–724.
[39] Miller AT, Knoop S, Christensen HI, et al. Automatic grasp planning using shape primitives. In: 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422); Vol. 2; 2003 Sept 14–19; Taipei, Taiwan, Taiwan. p. 1824–1829.
[40] Huebner K, Ruthotto S, Kragic D. Minimum volume bounding box decomposition for shape approximation in robot grasping. In: 2008 IEEE International Conference on Robotics and Automation; 19–23 May 2008; Pasadena, CA, USA. p. 1628–1633.
[41] Ciocarlie M, Hsiao K, Jones EG, et al. Towards reliable grasping and manipulation in household environments. In: Khatib O., Kumar V., Sukhatme G, editors. Experimental

robotics. Springer Tracts in Advanced Robotics, vol 79. Berlin, Heidelberg: Springer; 2014.

[42] Herzog A, Pastor P, Kalakrishnan M, et al. Template-based learning of grasp selection. In: 2012 IEEE International Conference on Robotics and Automation; 2012 May 14–18; Saint Paul, MN, USA. p. 2379–2384.

[43] Krainin M, Henry P, Ren X, et al. Manipulator and object tracking for in-hand 3d object modeling. Int J Rob Res. 2011;30:1311–1327.

[44] Gualtieri M, ten Pas A, Saenko K, et al. High precision grasp pose detection in dense clutter. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 9–14 Oct; 2016. Daejeon, South Korea. p. 598–605.

[45] Klingensmith M, Sirinivasa SS, Kaess M. Articulated robot motion for simultaneous localization and mapping (ARM-SLAM). IEEE Robot Autom Lett. 2016;1:1156–1163.

[46] Zhang Y, Funkhouser T. Deep depth completion of a single rgb-d image. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT. 2018. p. 175–185.

[47] PPMDTec. CamBoard Pico Flexx; 2018 [Cited 2019 May 20]. Available from: https://pmdtec.com/picofamily/