IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 6, 2020, accepted July 17, 2020, date of publication July 27, 2020, date of current version August 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3012067

RGB-D Tracking and Optimal Perception of Deformable Objects

IGNACIO CUIRAL-ZUECO, (Member, IEEE), AND GONZALO LÓPEZ-NICOLÁS, (Senior Member, IEEE)
Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, 50018 Zaragoza, Spain

Corresponding author: Ignacio Cuiral-Zueco (ignaciocuiral@unizar.es)

This work was supported by the Spanish Government/European Union through the Interreg Sudoe Programme and European Regional Development (ERDF), under Project PGC2018-098719-B-I00 (MCIU/AEI/FEDER, UE) and Project COMMANDIA SOE2/P1/F0638.

ABSTRACT Addressing the perception problem of texture-less objects that undergo large deformations and movements, this article presents a novel RGB-D learning-free deformable object tracker in combination with a camera position optimisation system for optimal deformable object perception. The approach is based on the discretisation of the object's visible area through the generation of a supervoxel graph that allows weighting new supervoxel candidates between object states over time. Once a deformation state of the object is determined, supervoxels of its associated graph serve as input for the camera position optimisation problem. Satisfactory results have been obtained in real time with a variety of objects that present different deformation characteristics.

INDEX TERMS Computer vision, deformable object tracking, object segmentation, next best view, simultaneous location and mapping.

I. INTRODUCTION

Object tracking in computer vision is a prominent and evolving area. Its applications range from use in mobile phone software to industrial applications for quality control analysis, object handling or transportation. In various fields such as the manufacturing industry, biomedical engineering or the food industry, there is emerging interest in the potential of robustly tracking objects with deformation characteristics since it would provide valuable information about the state of the object that may be used, for instance, as feedback in a manipulation or perception task. The use of visual sensors for these purposes in combination with robotic arms [1] is a common practice prompted by the widespread availability of affordable sensors with relatively good features. Placing static cameras range-covering a working space allows the tracking and characterisation of a deformable object, however, camera synchronisation may be complicated and dealing with occlusions is often impossible. On the other hand, fixing a camera on the end effector of a robot might be advantageous: controlling the robot makes it possible to adjust the position of the camera, thus modifying and enhancing the perception of the object by avoiding object self-occlusions and benefiting from less distant and more precise camera measurements. If a full 3D reconstruction of the object can be dispensed with, controlling a single camera position may be sufficient to adequately perceive the areas of interest of the object during its manipulation. Nonetheless, a vision-based object tracking system may encounter difficulties when dealing with large deformations of the object and considerable variations in camera motion. Overcoming such difficulties by using a single inexpensive sensor, such as an RGB-D camera, would allow the control of deformable object handling tasks performed by robots at reduced cost.

A. RELATED WORK

A variety of methods address the problem of deformable object tracking. Learning-based methods [2], [3] have proven to be robust but they present two disadvantages: (1) they require a significant amount of training with a large variety of data in order to perform properly in generic objects or scenes, and (2), usually, they are not directly related to the geometrical and physical state of the object, which results in potential impediment for designing and implementing a manipulation control action. Regarding learning-free approaches: graph-based methods like [4], [5] achieve robust tracking of deformable objects but, like the correspondence-based method [6], they rely on rich textures and the object analysis remains 2-dimensional. Not relying on textures, [7]

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

processes 2D sequences and extracts contour information from deformable objects to perform multi-object tracking (MOT) and occlusion handling between objects. There is a diversity of publications, namely [8] [9], that propose data-sets and metrics for the evaluation of the performance of object trackers. Benchmarks would traditionally provide data-sets of 2D sequences but recently started including RGB-D and RGB-Thermal sequences as well. Regarding 3D trackers, systems like [10] tackle 3D object-analysis using RGB-D sensors. [11] tracks deformable objects with point clouds and the use of probabilistic methods, it also manages to simulate the object's behaviour in real time using a previous model of the object. Other 3D deformable object trackers accomplish efficiency specialising in specific target objects like faces [12], hands [13] or full bodies [14]. Non-rigid SLAM systems like [15] (RGB-D) and [16] (monocular) manage to solve camera location and non-rigid environment characterisation, nonetheless, they do not tackle the segmentation problem. Other approaches like [17] achieve time coherent object segmentation of a dynamic scene.

Optimising the position of a visual sensor is an important element in a variety of problems. One of the most widespread is the Next Best View (NBV) problem (defined in [18]), which attempts to acquire the complete surface and geometry of an object with the use of a visual sensor, usually with as few frames as possible and following problem formulations like the ones discussed in [19] or [20]. The NBV is usually present in the autonomous generation of complete 3D model of objects [21]. The problem of time-varying scenes requires multi-sensor approaches like [22], where several cameras are optimised in order to perceive a deformable target object. It is also common to find NBV in applications for exploring unknown environments and objects [23]. Solving the NBV problem involves calculating a camera position for optimal perception, which is related to Visual Servoing (VS) [24], [25]. VS attempts to detect and follow the object of interest, therefore a target camera position must be generated to allow the tracking of the object at every moment in time. However, standard VS focuses on rigid objects. VS in non-rigid objects, although it has been addressed in the literature [26], is still an ongoing and researched topic.

B. PROPOSAL OVERVIEW

By using a single RGB-D camera, this article focuses on the problem of perceiving and tracking deformable objects that may be manipulated by hand or with the use of robots. Occlusions caused by the robotic arms, variations in the shape of the object and object self-occlusions can complicate the tracking process, reducing the quality of the information obtained by the camera and even resulting in the loss of the object. Taking this into consideration, and addressing the problem of texture-less objects that undergo large deformations and movements, we present an unsupervised learning-free deformable object tracker in combination with a camera position optimisation system for optimal deformable object perception. Our contribution consists in a graph-based tracking technique combined, for robot manipulation and control purposes, with a modified version of the state of the art simultaneous localisation and mapping (SLAM) system ORB-SLAM2 [27]. Once the deformable object is segmented and tracked, an optimal position for the camera is computed at every frame by solving the minimisation of a cost function that allows the camera's point of view to be adapted to the movements and deformations of the object. Real-time performance and robustness against low density point clouds and noisy input data are achieved.

Although no specific robot or control systems are presented, this proposal is contextualised within the robotics and control field of research as this perception system has been developed with a view to its future application in the handling of deformable objects by robots and thus focusing on providing useful 3D information about the object's visible surface in every frame. This information can be of use in more specific robot applications like, for instance, deformable object shape control. The computation of the optimal camera for object perception is also carried out with a view to future control applications in which consistent and reliable feedback is necessary.

The paper is structured as follows: Section II provides a general structure of the system along with an overview of various preliminary concepts that pave the way for sections III and IV. In section III, the main contribution explanation begins and the object tracking problem is addressed. Section IV focuses on the computation of the optimal camera position for deformable object perception. Several experiments will be presented and analysed in section V leading to the final evaluation and discussion in section VI.

II. SYSTEM STRUCTURE AND DATA ACQUISITION

This section covers the general structure of the system and briefly explains the blocks that support the main scope of the paper and are relevant to the understanding of the core explanation.

A. GENERAL STRUCTURE OF THE SYSTEM

The system takes video frames as input, each frame consists of an RGB image and its associated aligned depth map. Frames are fed to the SLAM system in which the position of the camera is estimated and the dense point map in global reference is generated. The dense map is then downsampled in order to reduce computational cost and reformatted into a point cloud that is over-segmented into supervoxels. While the target object is yet to be selected the scene is segmented into objects using a local convexity method (LCCP) [28]. Once an object is selected its state is updated at each frame by processing the new supervoxel clouds in the Object Tracking module. Using the latest object state a camera position for optimal perception of the object is computed. A comprehensive view of the system is synthesised on the flowchart in Fig. 1 and its associated images in Fig. 2.

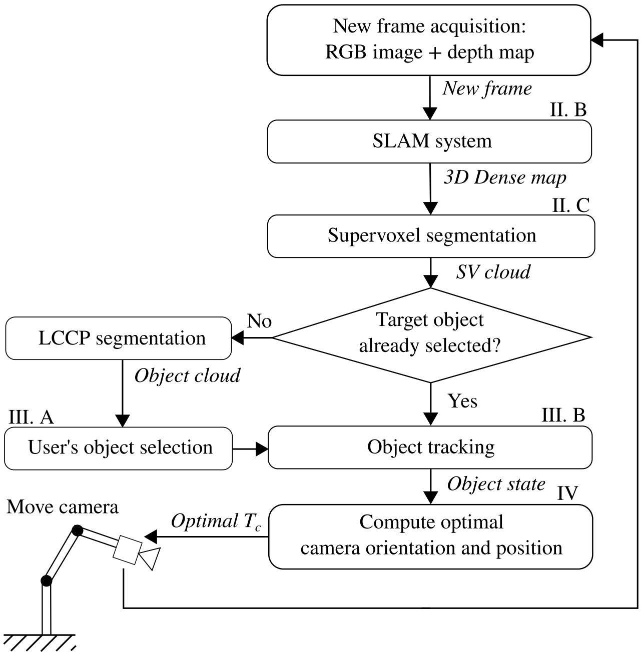


FIGURE 1. **Flowchart representing the general process of the method. Sections of the paper are linked to blocks for ease of reference. *SV* stands for *supervoxel*.**

B. THE SLAM SYSTEM

ORB-SLAM2 is a Bundle Adjustment based visual SLAM system that locates the camera in a sparse 3D map and is able to perform camera relocation with loop detection and closing [27]. Besides the monocular module, the system also features a more robust RGB-D and stereo modules. ORB-SLAM2 provides a sparse map that serves the purpose of locating the camera properly but, unfortunately, a sparse map does not provide enough information about the environment the camera is in. For this reason we have modified the RGB-D module of ORB-SLAM2 so that, for each frame, in addition to the position of the camera and the sparse environment map, a frame-synchronised global-referenced 3D dense map is generated too. It is important to mention that ORB-SLAM2 is designed to work in rigid environments. Nonetheless, it is capable of ignoring those elements of the scene that undergo movements and deformations provided that they do not occlude the camera view excessively. Deformable SLAM is beyond the scope of this project as the SLAM system will only be used as a mere camera location tool.

C. POINT CLOUD OVER-SEGMENTATION INTO SUPERVOXELS

In the fashion of superpixels for 2D images [29], [30], supervoxels over-segment 3D point clouds into small regions based on local low-level features, reducing the number of nodes which must be considered for inference. Voxel Cloud Connectivity Segmentation (VCCS) [31] method is used for this purpose since it segments actual volumes in space, and makes heavy use of the fact that such volumes are not regular or solid (most of the volume is empty space) to aid segmentation. VCCS supervoxels are clusters in the 39 dimensional space

$$\mathbf{F} = [x, y, z, L, a, b, \mathrm{FPFH}_{1..33}], \tag{1}$$

where x, y, z are spatial coordinates, L, a, b are colour in CIELab space, and $\mathrm{FPFH}_{1..33}$ are the 33 elements of Fast Point Feature Histograms (FPFH), a local geometrical feature proposed by Rusu *et al.* [32]. Before the supervoxel segmentation process begins, the input point cloud is transformed into a voxelized space of voxel size R_{voxel}. VCCS supervoxels are then created in $\mathbf{F}$ space with a user-specified scale. Each supervoxel stores an associated cloud of voxels and is defined by a centroid, a normal vector and an RGB value. VCCS supervoxel segmentation distributes seeds uniformly in 3D space creating a grid of step-size R_{seed} and makes supervoxels grow from the seeds gradually until they encounter other supervoxels. Supervoxel growing, which is proposed as an iterative process, proved to converge rapidly in such a way that supervoxel centroids are also uniformly distributed in 3D space.

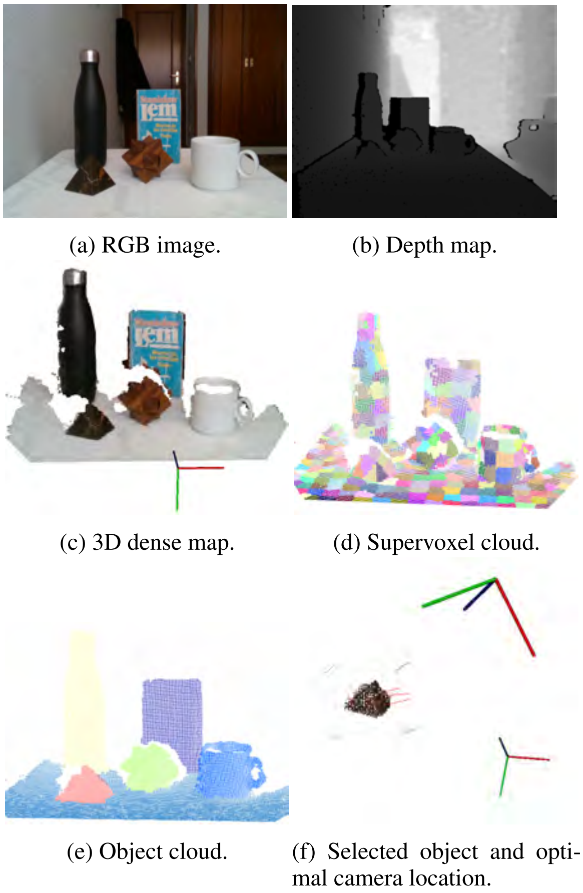

(a) RGB image.
(b) Depth map.
(c) 3D dense map.
(d) Supervoxel cloud.
(e) Object cloud.
(f) Selected object and optimal camera location.

FIGURE 2. **(a), (b) RGB Image and Depth Map, together they constitute the system's input frame. (c) 3D dense map along with the camera location provided by the SLAM system, represented by its frame of reference axes. (d), (e) supervoxel segmentation and LCCP segmentation results. (f) Selected Object (pyramid) together with the current camera location (bottom right) an the optimal camera location (upper right).**

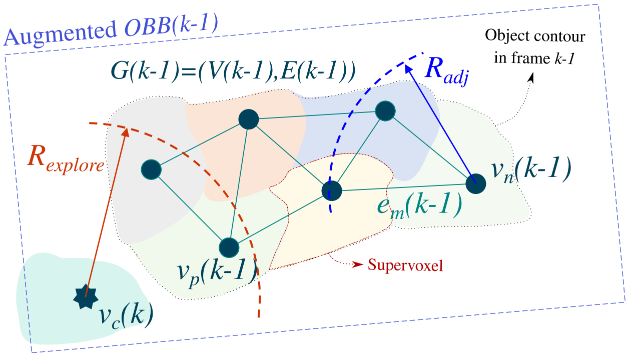


FIGURE 3. Figure that illustrates a supervoxel cloud of a deformable object in frame $k-1$ along with the object's graph $G(k-1) = (V(k-1), E(k-1))$, whose vertices correspond to supervoxel centroids. $R_{explore}$ and R_{adj} are also represented. A supervoxel candidate $v_c(k)$ and one of its neighbouring object vertices $v_p(k-1)$ is labelled as well. Notice how the candidate supervoxel centroid $v_c(k)$ lies inside the augmented $OBB(k-1)$.

III. DEFORMABLE OBJECT TRACKING

A. INITIAL STATE OF THE OBJECT

Incoming frames are processed and their associated 3D point cloud is down-sampled and over-segmented into supervoxels which are then used to perform an LCCP segmentation of the scene [28]. The state of the object $S(k)$ in frame $k \in \mathbb{Z}_{\geq 0}$ is defined by the set of $N(k)$ supervoxels that conform the object. Tracking process begins once one of the LCCP segments is selected as target object. The set of supervoxels that conform the selected LCCP segment determine the initial state of the object $S(0)$, which may be updated on following frames. Although the LCCP method is being used, any other object segmentation method is valid as long as it provides an initial set of supervoxels.

B. OBJECT'S STATE UPDATE

Object's state is updated at each frame. Note that the number of $N(k)$ supervoxels in the object may be different on each state $S(k)$ since the supervoxel distribution is uniform on 3D space and the object may undergo deformations and changes in size. Given an even distribution of the object's supervoxels in frame k, a graph $G(k) = (V(k), E(k))$ is created (Fig. 3). Vertices $V(k) = \{v_n(k),\ n = 1, \ldots, N(k)\}$ are defined by the object's $N(k)$ supervoxel centroids and edges $E(k) = \{e_m(k),\ m = 1, \ldots, M(k)\}$ are defined by the $M(k)$ connections between neighbouring vertices within an R_{adj}, a radius closely related to R_{seed}:

$$R_{adj} = \alpha R_{seed}, \tag{2}$$

where α, $1 \leq \alpha \leq 3/2$, is defined so adjacency between supervoxels is ensured after the supervoxel growing process has taken place. An augmented Oriented Bounding Box (OBB) can be defined using the voxels that conform the object state $S(k)$. The OBB contains the object and adjusts to the object's shape by matching axes with the object's principal axes thus providing a better fit that an $AABB$ (world reference Axis-Aligned Bounding Box). The OBB can be extended by adding a distance d_{OBB} to each of its three dimensions thus defining an exploration volume around the object, for now on we will refer to this extended OBB as the augmented OBB. Those supervoxels of frame k that lie inside the augmented $OBB(k-1)$ become new object candidates $V_{cand}(k) = \{v_c(k),\ c = 1, \ldots, C(k)\}$, with $C(k)$ being the total number of candidates in frame k. Each candidate $v_c(k)$ has a set of P neighbour vertices $V_{neigh}(c) = \{v_p(k-1),\ p = 1, \ldots, P\}$ that belong to the object's graph of the previous frame $V_{neigh}(c) \subseteq V(k-1)$ (see Fig. 3). Neighbour vertices $V_{neigh}(c)$ are those vertices that lie within $R_{explore}$ from candidate $v_c(k)$, $R_{explore}$ is defined by:

$$R_{explore} = \lambda R_{seed}, \tag{3}$$

where λ, $1 \leq \lambda \leq 2$, ensuring $R_{explore} > R_{adj}$ (i.e. $\lambda > \alpha$). Parameter λ acts as a scale factor that defines the size of the exploration zone for object state updates in relation to the supervoxel density established by R_{seed}. The previously mentioned d_{OBB} is set to $d_{OBB} = R_{explore}$.

In order to evaluate the addition of each candidate to the object's state $S(k)$ two criteria have been designed: (1) a Weighted Colour Distance Criterion (WCDC) and (2) a Weighted Normal Angle Criterion (WNAC). These criteria, combined, define the Candidate Acceptance Buffer Criterion (CABC). Both, WCDC and WNAC, depend on a Spatial Weight (Ws) which grows larger when candidate supervoxels get close to the object's core and decreases as candidate supervoxels fall further away from the object's boundaries (Fig. 4). In order to compute Ws, each candidate $v_c(k)$ is evaluated against graph $G(k-1)$ of the previous state:

$$W_s(v_c(k)) = \frac{\sum_{p=1}^{P} (R_{explore} - D_s(k,p))\ deg(v_p(k-1))}{R_{explore}\ \Delta(G(k-1))}, \tag{4}$$

where $D_s(k,p)$ is the 3D spatial Euclidean distance between candidate $v_c(k)$ and a neighbour vertex $v_p(k-1)$, $deg(v_p(k-1))$ is the degree of vertex $v_p(k-1) \in V(k-1)$ and $\Delta(G(k-1))$ the maximum degree in graph $G(k-1)$.

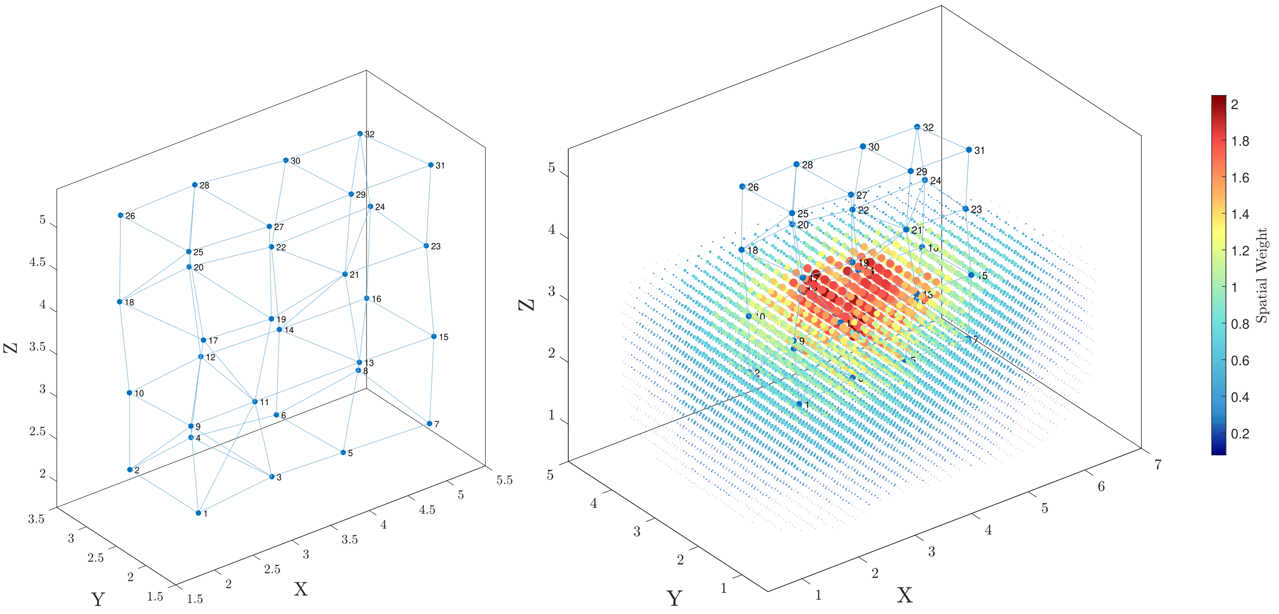
1) WEIGHTED COLOUR DISTANCE CRITERION: WCDC

The WCDC compares, locally, how similar the colour of the candidate and its closest object region are. Their similarity is characterised by the Weighted Colour Distance:

$$\bar{D}_{clr}(v_c(k)) = \frac{1}{W_s(v_c(k))} \frac{\sum_{p=1}^{P} D_{clr}(k,p)\ deg(v_p(k-1))}{\sum_{p=1}^{P} deg(v_p(k-1))}, \tag{5}$$

where $D_{clr}(k,p)$ is the $\Delta E^*(k,p)$ CIELab colour distance between candidate supervoxel $v_c(k)$ and neighbour vertex $v_p(k-1)$. This criterion is satisfied when the distance is less than a threshold γ_{clr}:

$$\text{WCDC}(v_c(k)) := \begin{cases} true & \bar{D}_{clr}(v_c(k)) < \gamma_{clr} \\ false & \text{otherwise.} \end{cases} \tag{6}$$



(a) Object graph

(b) Spatial Weight

FIGURE 4. This figure exemplifies Spatial Weight. A synthetically generated graph of a 3D box-shaped object is presented (a). A full object graph like the one in the figure can be obtained with a multi-camera configuration. The use of a single camera results in a partial perception of the object, therefore the generated graphs are generally planar. The Spatial Weight method is able to perform effectively in multi-camera scenarios as well. In (b), a grid that represents the values of the Spatial Weight around the object is represented along with the object's graph. A plane cuts the Spatial Weight grid at $Z = 3.5$ for ease of visualisation. Note how the value of the Spatial Weight is approximately 1 along the surface of the object. This value increases towards the object's core and decreases as points move away from the object.

where $\gamma_{clr} > 0$ is the maximum acceptable $\Delta E^*(k, p)$ CIELab colour distance. In the event of a background supervoxel happening to lie within the augmented *OBB*, confusion generated by similar object-background colour and intensity is cleared up by the method since it makes an extensive use of the depth information through the spatial weight.

2) WEIGHTED NORMAL ANGLE CRITERION: WNAC

The WNAC analyses how abrupt changes in the direction of the normals are along the object's surface. Similarly to WCDC, this analysis is performed locally defining a Weighted Normal Distance:

$$\bar{D}_{norm}(v_c(k)) = \frac{1}{W_s(v_c(k))} \frac{\sum_{p=1}^{P} D_{norm}(k,p)\ deg(v_p(k-1))}{\sum_{p=1}^{P} deg(v_p(k-1))}, \tag{7}$$

being $D_{norm}(k, p)$ the distance defined by the normalised angle between surface normal vectors $\mathbf{n}_c$, $\mathbf{n}_p$ of candidate supervoxel $v_c(k)$ and neighbour vertex $v_p(k-1)$ respectively:

$$D_{norm}(k,p) = \frac{1}{\pi} \arccos\left(\frac{\mathbf{n}_c^{\intercal}\mathbf{n}_p}{\|\mathbf{n}_c\|\,\|\mathbf{n}_p\|}\right). \tag{8}$$

This criterion is satisfied when the distance is less than a threshold angle γ_{norm}:

$$\mathrm{WNAC}(v_c(k)) := \begin{cases} true & \bar{D}_{norm}(v_c(k)) < \gamma_{norm} \\ false & \text{otherwise.} \end{cases} \tag{9}$$

where γ_{norm}, $0 \leq \gamma_{norm} \leq 1$, is the maximum acceptable angle between normal vectors normalized over π. Note that the Spatial Weight W_s has a strong influence on $\bar{D}_{clr}$ and $\bar{D}_{norm}$ making them lighter when candidate supervoxels are closer to the core of the object and, therefore, chances that they belong to the object increase.

3) CANDIDATE ACCEPTANCE BUFFER CRITERION: CABC

Combining both criteria, WCDC and WNAC, a *Candidate Acceptance Criterion* is defined as:

$$\mathrm{CAC}(v_c(k), S(k-1)) := \begin{cases} true & \mathrm{WCDC}(v_c(k)) \wedge \mathrm{WNAC}(v_c(k)) \\ false & \text{otherwise,} \end{cases} \tag{10}$$

where a candidate $v_c(k)$ is evaluated against a previous object state $S(k-1)$. However, this criterion would lack robustness. If a momentary and rapid occlusion occurred (i.e. occlusions that take place in a time window similar to the period between frames) there could be two possible scenarios:

1) **A partial occlusion of the object occurs:** When the partial occlusion ends, it takes several frames for the tracking method to recover the part of the object that has been re-exposed to the camera.
2) **The object is fully occluded:** The object is completely lost and the user has to re-select the object once the occlusion is finished.

Both issues are addressed in the *Candidate Acceptance Buffer Criterion*. A buffer of $B \in \mathbb{Z}_{>0}$ states of the object enables evaluating a candidate $v_c(k)$, not only against the state of the previous object $S(k-1)$ but also against all the

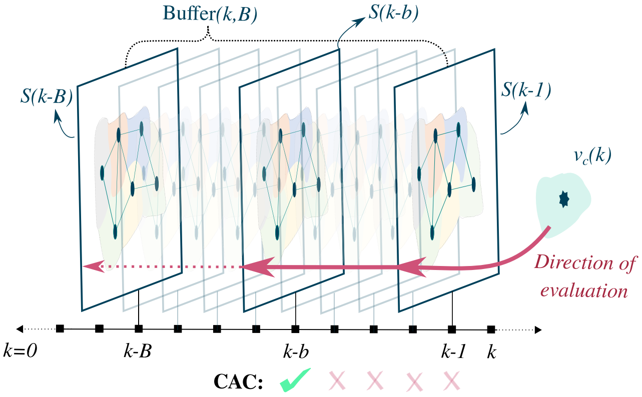


FIGURE 5. Graphic representation of Buffer($\boldsymbol{k}, \boldsymbol{B}$) = {$\boldsymbol{S}(\boldsymbol{k}-\boldsymbol{b})$, $\boldsymbol{b} = 1, \ldots, \boldsymbol{B}$}. Candidate $\boldsymbol{v_c}(\boldsymbol{k})$ is evaluated against criterion CAC with object states $\boldsymbol{S}$ from the Buffer. If CAC is satisfied in any of these states then CABC($\boldsymbol{v_c}(\boldsymbol{k})$) is also satisfied. In this example CAC is satisfied on state $\boldsymbol{S}(\boldsymbol{k}-\boldsymbol{b})$ and thus CABC($\boldsymbol{v_c}(\boldsymbol{k})$) is satisfied as well.

states stored in Buffer$(k, B) = \{S(k-b),\ b = 1, \ldots, B\}$ (Fig. 5).The final criterion is satisfied by evaluating the candidate against previous object states until it satisfies the CAC criterion in any of the Buffer states:

$$\text{CABC}(v_c(k)) := \begin{cases} \mathit{true} & \exists b \leq B \ni \mathit{CAC}(v_c(k), S(k-b)) = \mathit{true} \\ \mathit{false} & \text{otherwise.} \end{cases} \tag{11}$$

The size of the Buffer B, $1 \leq B \leq B_{max}$, should not be too large given that, if the object moves or is deformed during the time covered by the buffer, incorrect matches may occur. Since only one $CAC(v_c(k), S(k-b))$ is required to be true in order for CABC to be satisfied, the verification can be carried out efficiently by finding

$$\min b \ni CAC(v_c(k), S(k-b)) = \mathit{true}. \tag{12}$$

In practice, if there are no occlusions, candidates that belong to the object usually satisfy CABC with $b = 1$.

IV. OPTIMAL CAMERA ORIENTATION AND POSITION

So far, there is an object tracking task being performed. However, if the object were to fall outside the camera's field of view it would be lost. With that in mind, the aim is not only to track the object but also to make sure that it is perceived properly: it is important to move the camera to a position that ensures the perception of the object and keeps it in sight. Furthermore, even if the object remained in the camera's field of view, it would still be advantageous to adapt the camera location to the object's shape and position, which could result in more accurate and better quality information. This would not only prevent the object from being out of sight, but would also enable more precise and efficient object manipulation. Therefore, there is great potential in generating, for each frame, an optimal target position for the camera.

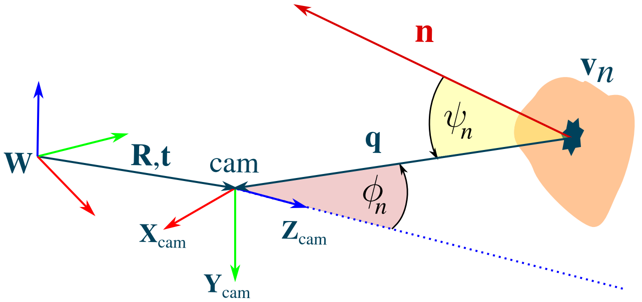

n
$\mathbf{v}_n$
ψ_n
q
ϕ_n
W
R,t
cam
$\mathbf{X}_{\text{cam}}$
$\mathbf{Z}_{\text{cam}}$
$\mathbf{Y}_{\text{cam}}$

FIGURE 6. Elements of the optimisation process. The camera (*cam*) is located with **R,t** with respect the world frame **W**. $\boldsymbol{v_n}$, with normal **n**, is one of the $\boldsymbol{N}$ supervoxels that conform the object.

A. NEXT BEST VIEW FOR DEFORMABLE OBJECT PERCEPTION AND TRACKING

Classic NBV usually tackles the exploration of a rigid or quasi-static environment; on the other hand, this approach focuses on ensuring the proper perception of an object's area of interest that varies through time. A wide variety of criteria may be applied when deciding which view is best for perceiving a deformable object. In this case, in order to achieve generality and considering that the object's physical properties are unknown, a purely geometric optimisation will be performed. The object is intended to appear in the center of the image and to be fully visible. It is also desirable that the camera provides a frontal view of the object's surface and is positioned at an appropriate distance from the object. Given a frame k and an object state $S(k)$, the objective is to determine an optimal World-Camera transformation $\mathbf{T}_{Wcam}(k) = [\mathbf{R} \,|\, \mathbf{t}]$ where camera orientation $\mathbf{R}(k) \in \mathbb{R}^{3\times3}$ and position $\mathbf{t}(k) \in \mathbb{R}^3$ are defined in a fixed global reference frame $\mathbf{W}$. The following elements (Fig. 6) have been used in the optimisation process

- ϕ_n is the angle between the projection ray $\mathbf{q}$ of $\mathbf{v}_n$ and the camera's optical axis ($\mathbf{Z}_{cam}$). It ensures that the camera is pointing towards the object when $\phi_n \to 0$.

$$\phi_n = \text{atan2}\left((\mathbf{q} \times \mathbf{Z}_{cam}) \frac{\mathbf{q} \times \mathbf{Z}_{cam}}{\|\mathbf{q} \times \mathbf{Z}_{cam}\|}, -\mathbf{Z}_{cam}\, \mathbf{q} \right). \tag{13}$$

- ψ_n is the angle between the normal $\mathbf{n}$ associated to $\mathbf{v}_n$'s supervoxel and the camera axis $\mathbf{Z}_{cam}$. This angle serves the purpose of positioning the camera orthogonally to the object's surface when $\psi_n \to 0$.

$$\psi_n = \text{atan2}((\mathbf{n} \times \mathbf{q}) \frac{\mathbf{n} \times \mathbf{q}}{\|\mathbf{n} \times \mathbf{q}\|}, \mathbf{n}\, \mathbf{q}). \tag{14}$$

- $d_e = d_{pref} - d_n(\mathbf{t}, \mathbf{v}_n)$ where

$$d_n(\mathbf{t}, \mathbf{v}_n) = \|\mathbf{q}\|. \tag{15}$$

and d_{pref} is the preferred camera-to-object distance.

All together, these elements conform

$$\gamma_n(k) = \begin{bmatrix} \phi_n(\mathbf{R}(k), \mathbf{v}_n(k)) \\ \psi_n(\mathbf{t}(k), \mathbf{v}_n(k)) \\ e^{|d_e(\mathbf{t}(k), \mathbf{v}_n(k))|} - 1 \end{bmatrix}, \tag{16}$$

which is minimised for every vertex $\mathbf{v}_n$ in the object using least squares:

$$\{\mathbf{R}(k), \mathbf{t}(k)\} = \underset{\mathbf{R}(k),\, \mathbf{t}(k)}{\arg\min} \frac{1}{2} \sum_{n=0}^{N} \rho \left(\|\gamma_n\|^2 \right), \tag{17}$$

where $\|\cdot\|$ is norm 2 and ρ is the robust Hubber Loss function. Orientation $\mathbf{R}(k)$ and position $\mathbf{t}(k)$ of the camera are iterated in order to minimise angles ϕ_n and ψ_n for each vertex supervoxel $\mathbf{v}_n$. Decreasing both angles results in aligning the camera's optical axis with supervoxel's normal vector $\mathbf{n}$. Only focusing on the supervoxel's centroid position (i.e. minimising ϕ_n) would ignore the surface configuration given by the point distribution within the supervoxel. However, taking the supervoxel normals into account (i.e. minimising ψ_n) also considers the distribution that the rest of the object points present inside each supervoxel. Minimising the third term of γ_n is defined as an exponential function that ensures that the camera is positioned at the desired distance from the object allowing some slack. In order to avoid camera-object collisions, d_{pref} should be established keeping in mind the size of the object and the deforming and moving actions the object might undergo. RGB-D cameras have a minimum and maximum sensing range and, generally, the error in their measurements increases along with the distance to the object. This suggests that, if collisions are disregarded and there is no specific desired distance, d_{pref} should be as low as possible (inside range limits).

V. EXPERIMENTAL RESULTS

This section assesses the performance of the proposed system. The results of several experiments carried out with different objects are shown and analysed.

A. EXPERIMENT PROCEDURE AND SETUP

The system's program is run and fed with the information obtained from an RGB-D camera in real time. Although different target objects are used, the procedure is the same for each experiment. First the target object is selected, then is manually manipulated causing movements and deformations on it during some period of time. At some point its manipulation is stopped. Lastly, while the object's position and deformation are kept constant, the RGB-D camera is manually re-positioned to match the computed optimal perception position.

The first four experiments have been carried out using the following objects: a box, a shoe sole, a balloon and a cloth. These objects have been selected to demonstrate the versatility of the method: a simple rigid object that illustrates the basic performance of the method, a deformable object of constant volume, a highly elastic object that has variable volume, and finally, an object that can be freely-shaped. The results of two additional experiments are also included in order to illustrate the robustness and generality of the proposal in resolving more ambitious situations.

TABLE 1. Parameters for supervoxel (SV) segmentation, Deformable Object Tracking (DOT) and camera optimisation.

SV segmentation	R_{voxel} [m]	0.01
	R_{seed} [m]	0.03
DOT	R_{adj} [m]	0.033
	$R_{explore}$ [m]	0.057
	γ_{clr}	12
	γ_{norm}	0.15
	B [frames]	5
Camera optimisation	d_{pref} [m]	0.5

Numeric values of parameters used in all the experiments are shown in Table 1. Experiments have been conducted in a workspace of approximately 1 cubic meter. Sequences were recorded in real time with the Realsense D435 RGB-D camera. The Realsense features an RGB camera and an infrared stereo pair along with an infrared pattern projector (active stereo) used to compute correspondences and generate the depth map. The programming language used is C++ and all measurements have been recorded on an Intel Core i7 1.8 GHz processor.

B. EXPERIMENTS: RESULTS AND ANALYSIS

In the absence of ground truth, the evaluation of object and camera tracking results are based on qualitative criteria: a proper performance of the system can be easily verified in the experiment **videos** (provided as complementary material). On the other hand, a quantitative analysis of system characteristics such as number of supervoxels (SV), optimisation convergence and processing time are analysed. For each experiment, five relevant frames are presented together with the 3D visualisation of the object's voxel cloud, the current camera position and the optimal camera position at the frame's time instant. Specific quantitative information about each sequence's relevant frames is presented on Table 2, where measurements taken throughout each sequence of the first four experiments are presented. For each of the frames selected to be displayed, along with their time stamp, the table provides values related to its number of supervoxels (SV) and optimisation residual R values (17): initial value of the Residual ($R_{initial}$), final value of the residual (R_{final}) and residual increment ($\Delta R = R_{initial} - R_{final}$). It also shows computation times of each of the following processes: SLAM, supervoxel segmentation, deformable object tracking (DOT) and camera position optimisation.

1) BOX

The general experimental procedure is performed with a box as a target object (Fig. 7). This is a base-case scenario where the object is rigid, allowing the analysis of the results to be introduced in a more intuitive manner. The object's voxel cloud can be visualised in Fig. 7 along with its associated supervoxel normals (in red) and augmented OBB. The current camera position provided by the SLAM system is represented by a small reference system having a red cube at its origin.

TABLE 2. In this table measurements taken throughout each sequence of the first four experiments are presented. For each of the frames selected to be displayed, along with their time stamp, the table provides values related to its number of supervoxels (SV) and optimisation process residual (R). It also shows computation times of each of the following processes: SLAM, supervoxel segmentation, deformable object tracking (DOT) and camera position optimisation.

Exp.	Frame	Time stamp [s]	Nº SV		Optimisation				Processing time [ms]				
			Scene	Object	$R_{initial}$	R_{final}	ΔR	Solver time [ms]	SLAM	SV seg.	DOT	Camera opt.	total
Box	1	0.0	76	34	9.692	2.980	6.712	2.84	44	18	10	4	76
	38	7.1	97	34	9.614	2.392	7.222	3.8	44	24	12	5	85
	70	13.0	110	31	23.402	8.443	14.96	11.4	53	22	9	13	97
	118	22.4	103	32	9.029	4.432	4.596	1.7	76	31	12	2	121
	185	37.4	63	28	1.101	0.940	0.161	2.5	54	30	12	4	100
Shoe sole	1	0.0	83	20	2.079	0.986	1.093	1.78	43	13	44	2	69
	61	10.5	67	16	0.837	0.636	0.202	1.82	50	8	6	2	66
	135	21.4	114	19	3.425	2.860	0.566	1.76	43	20	6	2	71
	207	33.2	86	18	8.294	1.181	7.113	1.46	47	15	9	2	73
	264	43.9	46	23	1.672	1.353	0.319	1.94	68	27	11	6	109
Balloon	1	0.0	110	44	16.48	14.481	2.0	2.5	43	22	13	4	82
	15	2.6	103	26	8.48	7.680	0.804	1.53	51	16	12	3	82
	20	3.5	85	7	0.82	0.523	0.296	1.18	40	9	4	2	55
	67	11.6	100	10	0.564	0.253	0.311	1.9	39	15	5	3	62
	169	33.0	60	5	0.657	0.208	0.448	1.48	41	8	3	2	54
Cloth	1	0.0	85	21	6.667	2.966	3.701	1.3	43	16	9	2	70
	44	7.0	120	32	6.647	3.354	3.293	2.15	60	24	18	3	105
	63	10.9	141	47	3.808	3.261	0.546	1.58	51	33	19	2	105
	303	57.6	128	45	17.893	7.0	10.893	2.67	44	26	22	3	95
	369	72.2	129	57	7.787	7.668	0.12	2.87	71	34	15	4	124

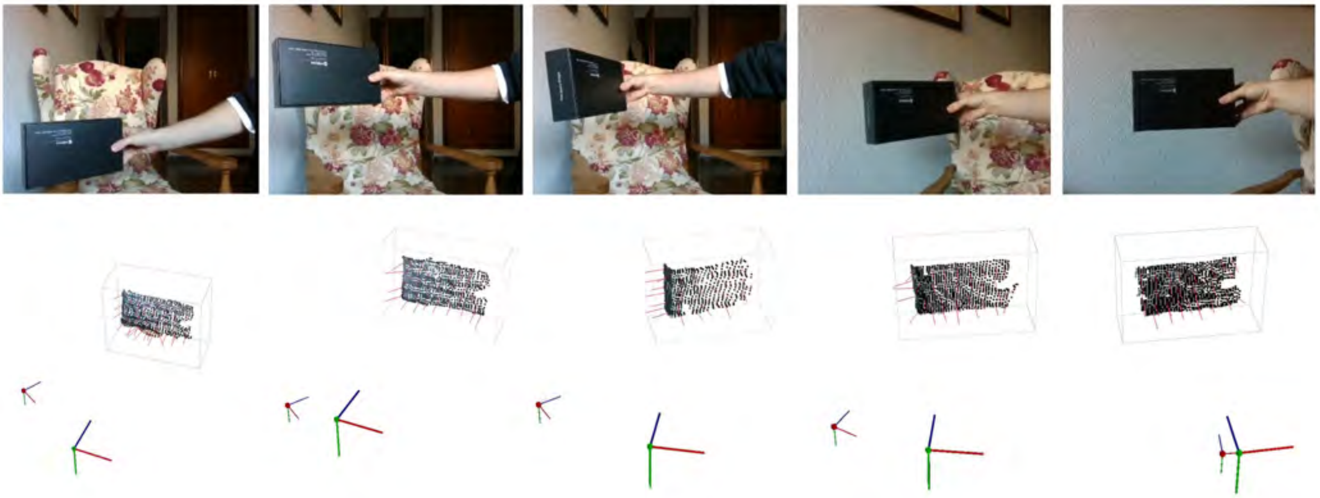


FIGURE 7. Box experiment (rigid object). Five relevant RGB video frames and their associated selected object, current camera position (small reference system with a red cube on its origin) and optimal camera position (large reference system with a green cube on its origin). The object is represented by its voxel cloud along with the normals associated to its supervoxels (red vectors) and its augmented OBB. It becomes clear that the optimal position is obtained in front of the object, at the preferred distance.

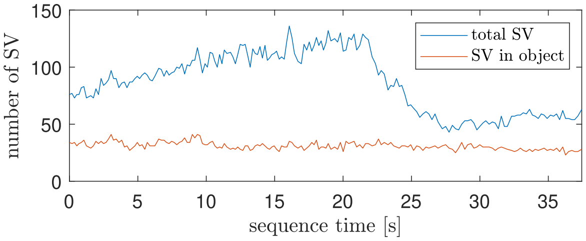


FIGURE 8. Box experiment. Number of supervoxels in the scene (blue) and number of supervoxels that conform the object (orange) over time.

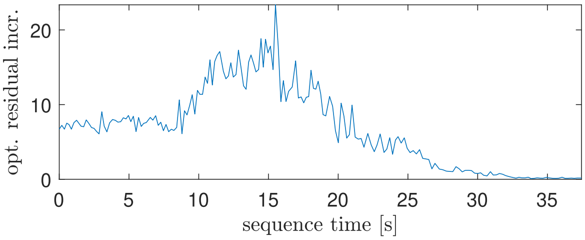


FIGURE 9. Box experiment. Variation of the optimisation residual increment, $\Delta R = R_{initial} - R_{final}$, over time.

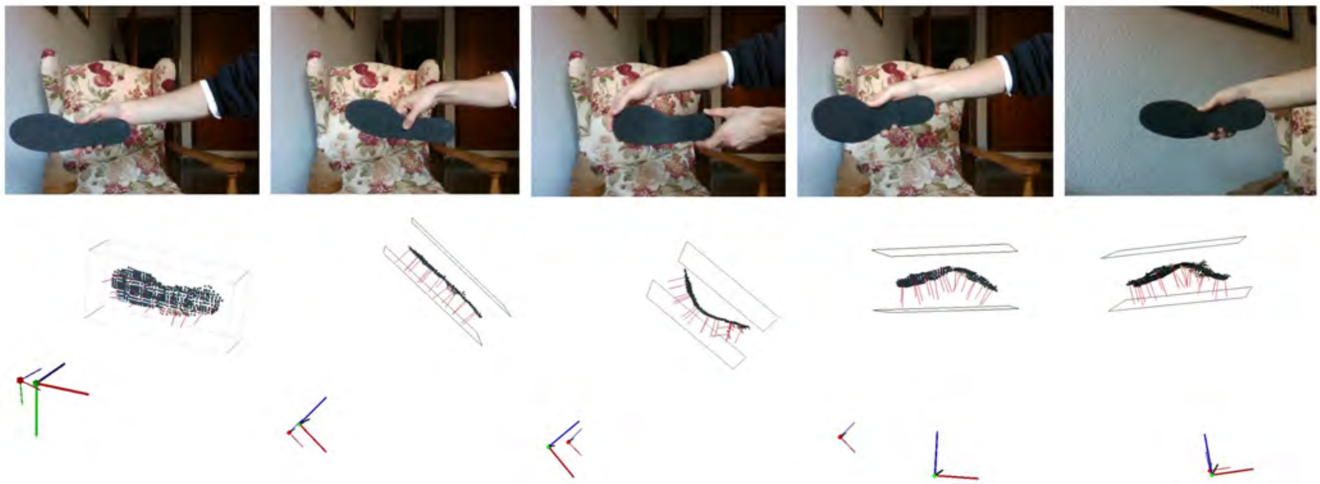


FIGURE 10. Shoe sole experiment (deformable flat object). Five relevant RGB video frames and their associated selected object, current camera position (small reference system with a red cube on its origin) and optimal camera position (large reference system with a green cube on its origin). The shoe sole is represented by its voxel cloud along with the normals associated to its supervoxels (red vectors) and its augmented OBB. Despite the large deformation of the sole, the optimal camera location can be seen to be correctly computed.

Similarly, the optimal camera position is represented by a larger reference system having a green cube at its origin.

Regarding the number of object supervoxels, since the object is rigid and its main face is always visible, it does not vary significantly (Fig. 8). However, when the arm is less visible and the armchair in the background is out of frame, the total number of supervoxels decreases largely (see number of SV in the scene, time stamps 22.4 and 37.4 on Table 2).

In the sequence, the box is first positioned facing the camera on the lower left corner (image reference). Then it is raised and rotated. After the object's rotation, the optimal camera is positioned perpendicularly and centred towards the main face of the box. The initial state of the optimal camera in the optimisation process is always set as the position of the real camera in the same time instant. Therefore, the greater the difference in position between the real and the optimal camera, the greater the residual increment (ΔR). This translates into an increase in ΔR (Fig. 9) right after the box is rotated. The real camera begins to move in frame 85 (around second 17) and when it is placed near the optimal position again, in the second half of the sequence, ΔR decreases. As expected, a poor initial camera position in the optimisation process results in larger solver times (Table 2, optimisation solver time reaches 11.4 ms when ΔR is around its peak value).

An interesting observation in this experiment is how, when the box is rotated, one of it sides becomes visible thus affecting the optimal camera position. However, when modifying the real camera position, at some point the side of the box is not visible and the optimal camera position changes slightly by moving to the right (camera reference). If the system happened to be integrated within a camera position control system, the real camera would always respond to the variations in the optimal camera position and, therefore, chances are that the side of the box would have never been fully visible.

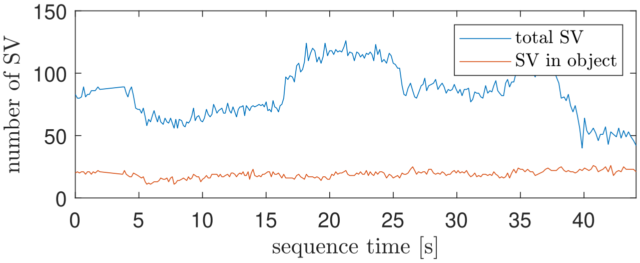


FIGURE 11. Shoe sole experiment. Number of supervoxels in the scene (blue) and number of supervoxels that conform the object (orange) over time.

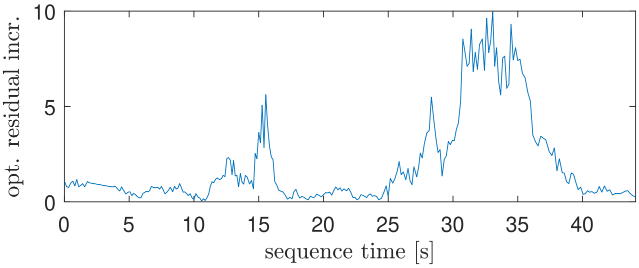


FIGURE 12. Shoe sole experiment. Variation of the optimisation residual increment, $\Delta R = R_{initial} - R_{final}$, over time.

2) SHOE SOLE

The shoe sole is an elastic deformable object that can be deformed to a great extent under bending and torsional stress, but cannot be largely stretched or compressed under tensile and compressing stress. The frames of the sequence (Fig. 10) show the sole being bent into a concave shape (with respect to

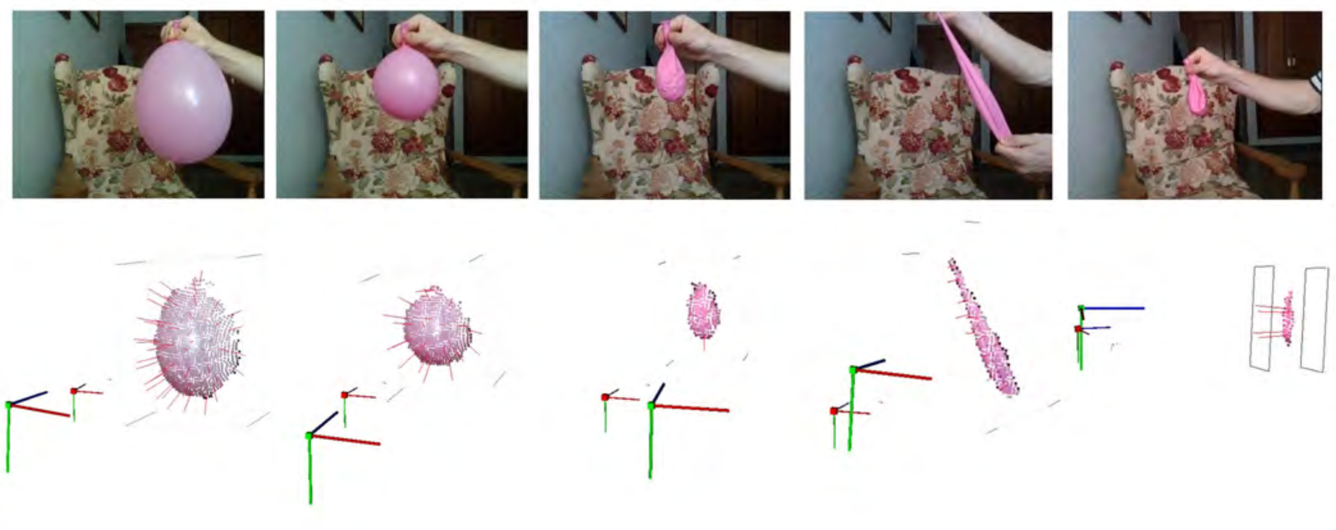


FIGURE 13. Balloon experiment (deformable object with variable volume). Five relevant RGB video frames and their associated selected object, current camera position (small reference system with a red cube on its origin) and optimal camera position (large reference system with a green cube on its origin). The balloon is represented by its voxel cloud along with the normals associated to its supervoxels (red vectors) and its augmented OBB. This example illustrates that the proposed approach is able to deal with large deformations and volume variations with good performance.

the camera) that becomes convex later in the sequence, while being rotated.

As in the first experiment, since the shoe sole cannot be stretched or compressed, its volume does not vary greatly and, therefore, the number of supervoxels it contains remains similarly unchanged (Fig. 11).

When the sole is deformed into a convex shape the optimal camera tends to move away from it in order to gain perpendicularity and reduce ψ_n. This behaviour gives the camera more room to manoeuvre in the event that the complete perception of the object is compromised. However, the camera should not move back indefinitely as the resolution would limit the proper perception of the object. Unlimited optimal camera distancing is counterbalanced by the preferred camera-to-object distance (d_{pref}) optimisation term. On the contrary, when the sole is deformed into a convex shape the optimal camera gets closer to it, thus having a more perpendicular position and tending to avoid object's self-occlusions. Fig. 12 reflects the moments when the bending deformations have occurred by showing peaks on ΔR, which increases even more when deformation and rotation are combined on the sole around second 30. The camera is kept still until frame 207 (second 33.2), notice how at that moment in time $\Delta R = 7.113$ (Table 2) and ten seconds later, when the camera faces the sole perpendicularly, ΔR gets close to 0.

3) BALLOON

The balloon is a highly elastic object that can be easily deformed under tensile, compressing, bending and torsional stress. The sequence (Fig. 13) starts with an inflated balloon that deflates in a time interval of about 3 seconds. The balloon is then re-positioned and largely stretched. Note that, especially when inflated, the balloon has a reflective (and therefore variable) texture. In addition, when it is stretched, its colour becomes less saturated.

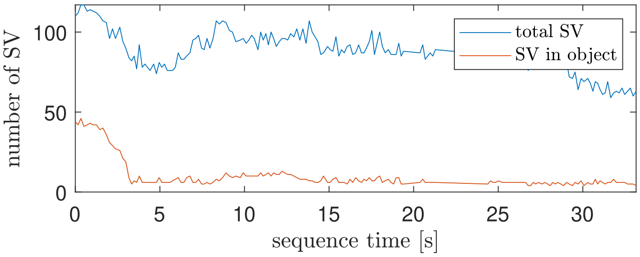


FIGURE 14. Balloon experiment. Number of supervoxels in the scene (blue) and number of supervoxels that conform the object (orange) over time.

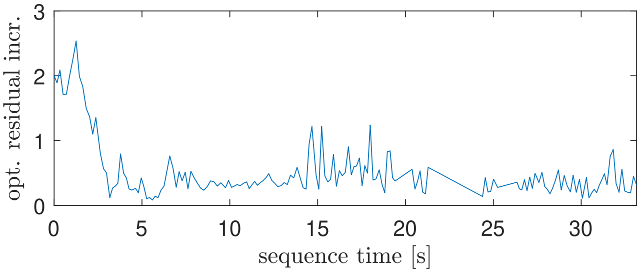


FIGURE 15. Balloon experiment. Variation of the optimisation residual increment, $\Delta R = R_{initial} - R_{final}$, over time.

Due to its high deformability, its surface area can vary greatly. This is noticeable at the beginning of Fig. 14, in which a rapid decrease in the number of supervoxels of the object is registered. The deflating process is also reflected in Table 2, where the number of supervoxels in the object decreases drastically from 44 to 7 between the 3.5 seconds time stamp and the initial frame of the sequence. Although it is somewhat harder to spot, there is a slight increase on the number of

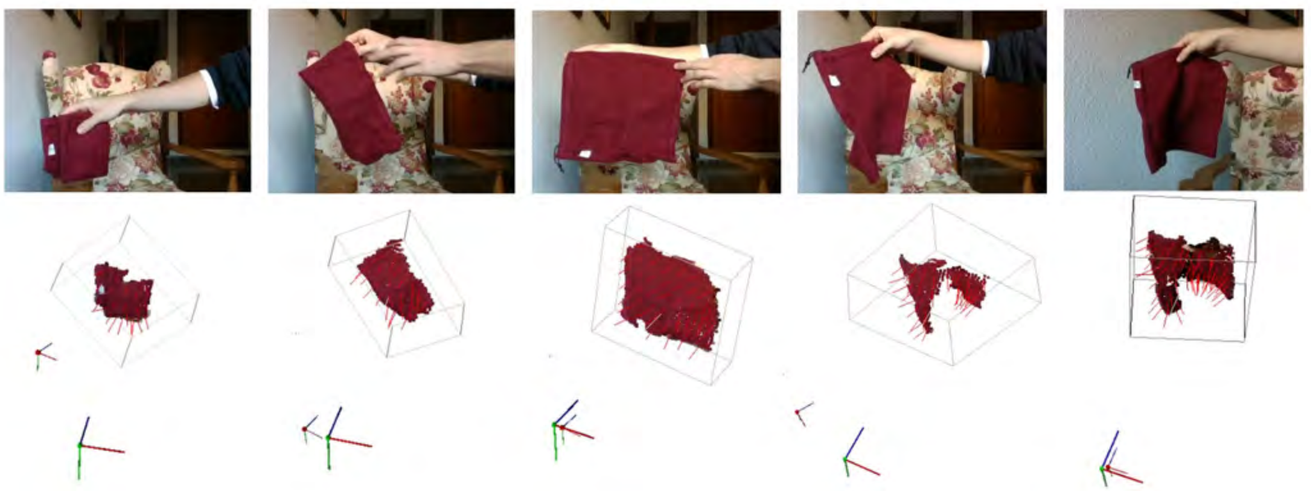


FIGURE 16. Cloth experiment (freely deformable object). Five relevant RGB video frames and their associated selected object, current camera position (small reference system with a red cube on its origin) and optimal camera position (large reference system with a green cube on its origin). The cloth is represented by its voxel cloud along with the normals associated to its supervoxels (red vectors) and its augmented OBB. Note how in the last two shown frames the object presents a self-occlusion that disappears once the real camera is near the computed optimal camera position.

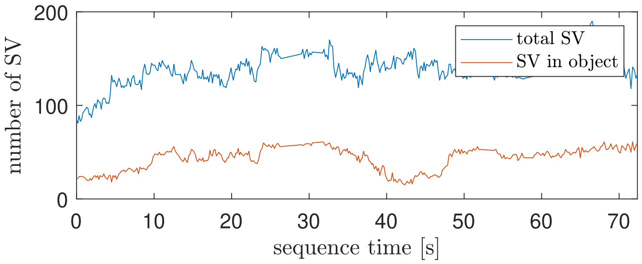


FIGURE 17. Cloth experiment. Number of supervoxels in the scene (blue) and number of supervoxels that conform the object (orange) over time.

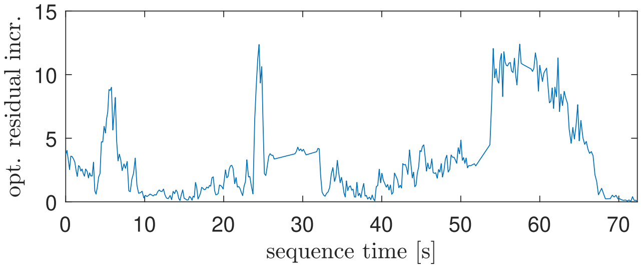


FIGURE 18. Cloth experiment. Variation of the optimisation residual increment, $\Delta R = R_{initial} - R_{final}$, over time.

supervoxels in the object around second 10, corresponding to the stretching of the deflated balloon that can be observed in the fourth frame displayed in Fig. 14.

With regard to optimisation, when the balloon is inflated and tends to have a spherical shape there is a great disparity between the directions of its normals. This translates into a high value of ΔR at the beginning of the sequence (Fig. 15). When the balloon is no longer stretched (around the second 15) the optimisation residuals undergo large variations. As the deflated balloon presents a wrinkled surface, the noise coming from the depth estimation of the RGB-D camera is significantly more prevalent in supervoxel generation and thus propagates greatly to the normal vectors that define the optimisation problem.

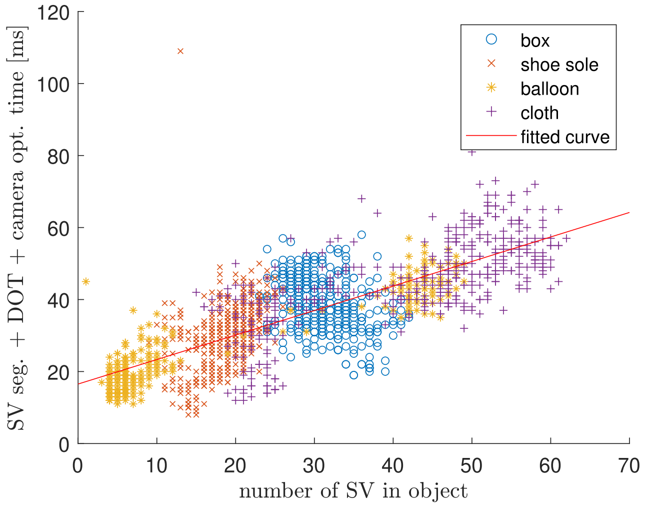


FIGURE 19. Processing time of all object-related tasks as a function of number of SV in objects.

4) CLOTH

In this experiment a cloth is unfolded and freely deformed (Fig. 16). As it is folded and deformed, the region of cloth that is exposed to the camera varies over time, causing a variation in the number of supervoxels that conform it (Fig. 17). Nevertheless, out of the four analysed objects, it is the one that contains more supervoxels and this has noticeable repercussion on the computation time of the Deformable Object Tracking (DOT) process (Table 2). The number of supervoxels in the scene is also high thus making supervoxel (SV) segmentation processing times larger.

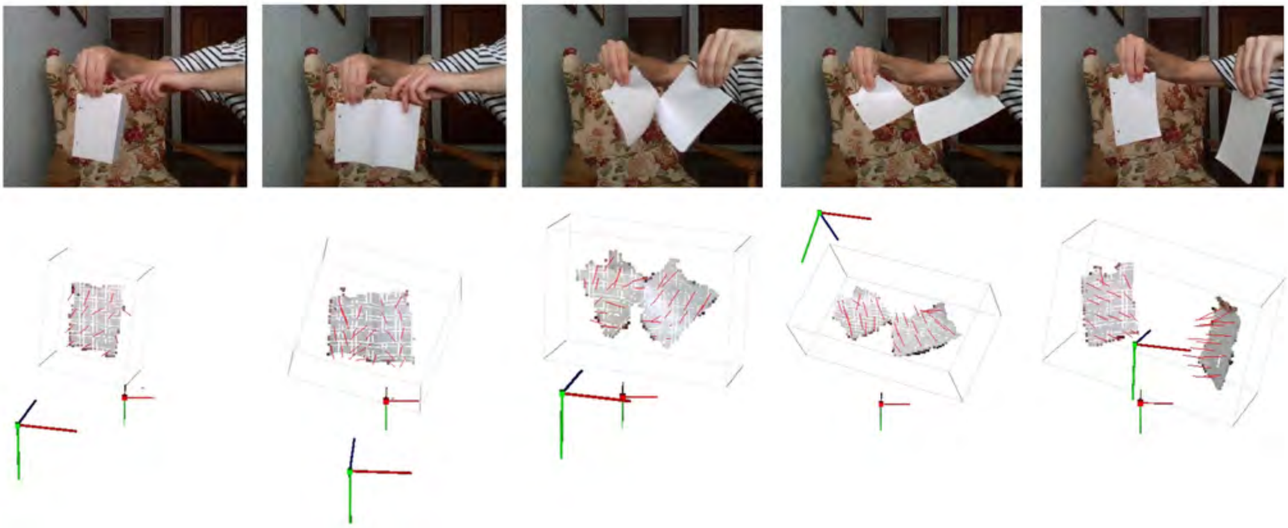


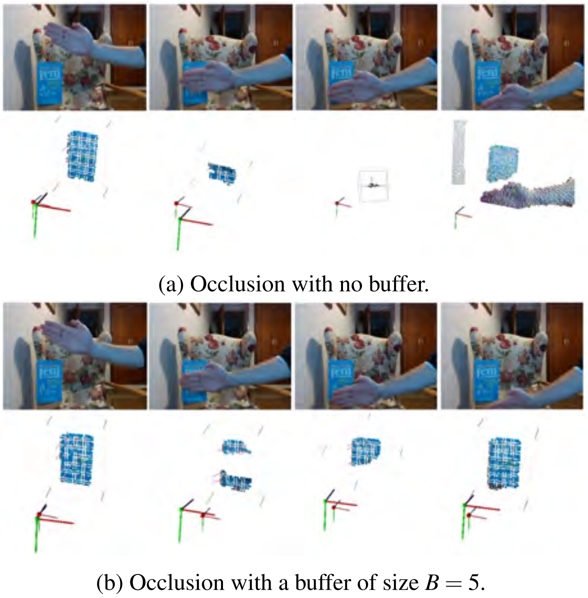
FIGURE 20. Tearing up piece of paper. Five relevant RGB video frames and their associated selected object, current camera position (small reference system with a red cube on its origin) and optimal camera position (large reference system with a green cube on its origin). The paper pieces are represented by their voxel cloud along with the normals associated to their supervoxels (red vectors) and their augmented OBB. The system still manages to track the object even though it has been torn into two pieces and the optimal camera position adapts to the tearing process adequately.

When the handling phase is over, the shape of the cloth, which generates a self-occlusion, is kept constant (time stamp 57). Then, around frame 300, the real camera is manually re-positioned in order to match the optimal position and thus making the self-occluded part visible, resulting in a slight increase in the number of supervoxels that conform the tracked cloth. The number of supervoxels changes from 45 to 57, as shown in Table 2. In the same time interval, as the real camera approaches the optimal position, a sudden drop in ΔR is observed (Fig. 18).

In the four main experiments presented, it becomes clear how the system manages to work with objects of different sizes, which translates into different numbers of supervoxels in each object and/or moment in time. The fact that the number of supervoxels varies leads to an interest in the analysis of the relation between the number of supervoxels in the object and the processing time of the object-concerning tasks. One can intuitively think that the more supervoxels in the object, the longer the processing times will be, and in fact this is the case. This behaviour is demonstrated in Fig. 19, where the number of supervoxels of each object in all time instants are represented and compared to the total processing time of object-related tasks: supervoxel segmentation, deformable object tracking and camera optimisation.

5) PAPER TEARING

This experiment has been carried out in order to demonstrate the robustness of the system against an action for which it has not been specifically designed: being able to track a piece paper that is torn into two pieces. In the sequence shown in Fig. 20 one can observe how the system still considers the pieces as part of the same object and is able to track them. Although the method has not been specifically developed to deal with this scenario, a system to manage this element as separate objects can be considered in the near future.



(a) Occlusion with no buffer.

(b) Occlusion with a buffer of size $B = 5$.

FIGURE 21. Buffer demonstration. In the first sequence (a) the buffer is disabled and the tracking system loses the book: note how in the fourth frame there is no augmented OBB around the book and other elements of the scene become visible in the 3D visualiser. In the second case (b) the buffer size is set to $B = 5$ frames and, even though the hand occludes the book, the tracker is able to locate its visible parts.

6) OCCLUSIONS

This experiment consists of two sequences in which, after selecting a book as target object, a hand is vertically waved between the book and the camera (Fig. 21). In the first case (21a) the buffer is disabled and the tracking system loses the book. When the book is lost note how there is no augmented

OBB around it and other elements of the scene become visible in the 3D visualiser. In the second case (21b) the buffer size is set to $B = 5$ frames and, even when the hand occludes the book, the tracker is able to locate it.

VI. CONCLUSION

A novel tracking method for texture-less deformable objects has been presented. It receives RGB frames and depth maps as inputs and does not require any prior information about the target object. It is also capable of handling rapid occlusions and, given its local approach, it is resilient to colour changes in the object caused by variations in lighting, deformation of the material, light source reflections, etc. Making use of the supervoxel structure of the tracked object, a system for computing the optimal position of the camera has also been designed and implemented so as to allow better perception of the deformable object. Discretisation into supervoxels and synthesis of information into object graphs lightens the computational cost, allowing a run time that is on the order of tens of milliseconds, which suggests that the system is suitable for real-time applications.

It is worth mentioning that this tracker considers a very specific problem that many trackers do not tackle explicitly: The problem of tracking objects that lack texture and undergo large free deformations and movements. In addition, in the problem definition deformable objects are within generic scenes and backgrounds. Being a highly specific and recent problem there are not many methods this new approach can be fairly compared to. However, one of its main advantages is its generality: it can handle most objects and types of deformation combined with large movements in diverse backgrounds. This tracker does not rely on fixing the background's or the object's colour like other approaches, namely [10], [11] as pixel labelling and image segmentation is not in their scope. It does not need prior object knowledge, object model ([10]) or training process ([2]) inasmuch as it is a learning free approach. The method presented performs a local analysis and uses discrete information, which allows run times in the order of tens of milliseconds, as opposed to other methods ([17]) that solve the scene globally but require longer computation times (order of seconds).

Future improvements may include identification and tracking of the occlusive elements of the scene. As in [33], filling occluded areas of the object can be also considered. Regarding optimisation, there may be alternatives for the computation of the optimal camera, however, an optimisation process allows to be easily expanded with new constraints and functions such as: including occlusions in the camera optimisation process, adding workspace size constraints, limiting or fixing camera movement speed, etc. Another improvement would be the dynamic analysis of the object that could use, as nodes for discrete physical models, the supervoxel structure of the graph. A physical model would ease the implementation of object shape control algorithms. RGB-D data-sets keep evolving and providing better performance metrics. Participating in data-set challenges is also considered good practice as it allows the comparison of methods. Challenges like the ones proposed in [8], and more specifically challenges in the lines of robust short-term tracking under appearance variation, occlusion and clutter can be of particular interest for the method presented in this article.

REFERENCES

[1] R. Herguedas, G. Lopez-Nicolas, R. Aragues, and C. Sagues, "Survey on multi-robot manipulation of deformable objects," in *Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2019, pp. 977–984.

[2] G. Ciaparrone, F. Luque Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, Mar. 2020.

[3] G. Olague, D. E. Hernández, P. Llamas, E. Clemente, and J. L. Briseño, "Brain programming as a new strategy to create visual routines for object tracking," *Multimedia Tools Appl.*, vol. 78, no. 5, pp. 5881–5918, Mar. 2019.

[4] Z. Cai, L. Wen, Z. Lei, N. Vasconcelos, and S. Z. Li, "Robust deformable and occluded object tracking with dynamic graph," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5497–5509, Dec. 2014.

[5] D. Du, H. Qi, W. Li, L. Wen, Q. Huang, and S. Lyu, "Online deformable object tracking based on structure-aware hyper-graph," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3572–3584, Aug. 2016.

[6] G. Nebehay and R. Pflugfelder, "Clustering of static-adaptive correspondences for deformable object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2784–2791.

[7] B. Guo and Y. Zhu, "Tracking multiple indistinguishable and deformable objects based on multi-anchor flow with annular sector model," *IEEE Access*, vol. 7, pp. 164265–164275, 2019.

[8] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Cehovin, "A novel performance evaluation methodology for single-target trackers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2137–2155, Nov. 2016.

[9] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, Jul. 2014.

[10] A. Petit, V. Lippiello, and B. Siciliano, "Real-time tracking of 3D elastic objects with an RGB-D sensor," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 3914–3921.

[11] J. Schulman, A. Lee, J. Ho, and P. Abbeel, "Tracking deformable objects with point clouds," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 1130–1137.

[12] F. Alqahtani, J. Banks, V. Chandran, and J. Zhang, "3D face tracking using stereo cameras: A review," *IEEE Access*, vol. 8, pp. 94373–94393, 2020.

[13] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff, A. Topalian, E. Wood, S. Khamis, P. Kohli, S. Izadi, R. Banks, A. Fitzgibbon, and J. Shotton, "Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–12, Jul. 2016.

[14] Y. Wu, D. Kong, S. Wang, J. Li, and B. Yin, "An unsupervised real-time framework of human pose tracking from range image sequences," *IEEE Trans. Multimedia*, vol. 22, no. 8, pp. 2177–2190, Aug. 2020.

[15] R. A. Newcombe, D. Fox, and S. M. Seitz, "DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 343–352.

[16] J. Lamarca and J. M. M. Montiel, "Camera tracking for Slam in deformable maps," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, Sep. 2018, pp. 1–8.

[17] X. Lin, J. R. Casas, and M. Pardas, "Temporally coherent 3D point cloud video segmentation in generic scenes," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 3087–3099, Jun. 2018.

[18] R. Pito, "A solution to the next best view problem for automated surface acquisition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 10, pp. 1016–1030, Oct. 1999.

[19] J. Delmerico, S. Isler, R. Sabzevari, and D. Scaramuzza, "A comparison of volumetric information gain metrics for active 3D object reconstruction," *Auto. Robots*, vol. 42, no. 2, pp. 197–208, Feb. 2018.

[20] G. Olague and R. Mohr, "Optimal camera placement for accurate reconstruction," *Pattern Recognit.*, vol. 35, no. 4, pp. 927–944, Apr. 2002.

[21] M. Krainin, B. Curless, and D. Fox, "Autonomous generation of complete 3D object models using next best view manipulation planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 5031–5037.

[22] R. Herguedas, G. Lopez-Nicolas, and C. Sagues, "Multi-camera coverage of deformable contour shapes," in *Proc. IEEE 15th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2019, pp. 1597–1602.

[23] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon 'next-best-view' planner for 3D exploration," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2016, pp. 1462–1468.

[24] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[25] F. Chaumette and S. Hutchinson, "Visual servo control. II. Advanced approaches [tutorial]," *IEEE Robot. Autom. Mag.*, vol. 14, no. 1, pp. 109–118, Mar. 2007.

[26] D. Santosh and C. V. Jawahar, "Visual servoing in non-rigid environments: A space-time approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 2452–2457.

[27] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[28] S. C. Stein, M. Schoeler, J. Papon, and F. Worgotter, "Object partitioning using local convexity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 304–311.

[29] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, vol. 1, Oct. 2003, pp. 10–17.

[30] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to State-of-the-Art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[31] J. Papon, A. Abramov, M. Schoeler, and F. Worgotter, "Voxel cloud connectivity segmentation–supervoxels for point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2027–2034.

[32] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3212–3217.

[33] Z. Hu, T. Han, P. Sun, J. Pan, and D. Manocha, "3-D deformable object manipulation using deep neural networks," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4255–4261, Oct. 2019.

IGNACIO CUIRAL-ZUECO (Member, IEEE) received the master's degree in industrial engineering majoring in robotics and computer vision from the University of Zaragoza, Zaragoza, Spain, in 2019. He is currently pursuing the Ph.D. degree in systems engineering and computer science. He is a member of the Robotics, Perception and Real-Time Research Group. His current research interests include computer vision, control engineering, and robotics.

GONZALO LÓPEZ-NICOLÁS (Senior Member, IEEE) received the Ph.D. degree in systems engineering and computer science from the University of Zaragoza, Zaragoza, Spain, in 2008. He is currently an Associate Professor with the Department of Computer Science and Systems Engineering, University of Zaragoza. He is a member of the Robotics, Perception and Real-Time Group and the Aragon Institute of Engineering Research (I3A). His current research interests include visual control, autonomous robot navigation, multirobot systems, and the application of computer vision techniques to robotics.