Citation: R. Herguedas, G. López-Nicolás, C. Sagüés. Coverage of deformable contour shapes with minimal multi-camera system. Measurement, 190 (2022) 110693. 2022.

# Coverage of deformable contour shapes with minimal multi-camera system*

Rafael Herguedas, Gonzalo López-Nicolás, Carlos Sagüés

[a]*Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, C/ María de Luna 1, E-50018 Zaragoza, Spain*

**Abstract**

Perception over time is a critical problem in those cases where deformable objects are manipulated. The goal of this study is to cover the contour of an object along a deformation process and according to a prescribed coverage objective, in terms of visibility and resolution. This task is carried out by a set of limited field-of-view cameras. We propose novel methods for guaranteeing feasibility of the coverage objectives, which include the computation of the maximum visibility and resolution of the contour. Then, we introduce the coverage objectives in an offline constrained optimization problem to compute a priori the minimum number of cameras that achieve the coverage requirements. Finally, we propose an online technique that provides optimized configurations faster than the offline one, even when the object's reference deformation is unknown. We report experimental results in which our method achieves 100% of the coverage in simulation and in a real task.

*This work was supported by projects COMMANDIA SOE2/P1/F0638 (Interreg Sudoe Programme, ERDF), PGC2018-098719-B-I00 (MCIU/AEI/FEDER, UE) and DGA_T45-17R (Gobierno de Aragón). The first author was partially supported by the EU through the European Social Fund (ESF) "Construyendo Europa desde Aragón".

*Email addresses:* rherguedas@unizar.es (Rafael Herguedas), gonlopez@unizar.es (Gonzalo López-Nicolás), csagues@unizar.es (Carlos Sagüés)

*Keywords:* Coverage, Deformable objects, Multi-camera perception, Active perception

## 1. Introduction

Autonomous sensing and manipulation of objects whose shape changes over time are topics of increasing interest in the last years [1]. Today's challenges impel the strategic industries that deal with these kinds of objects to improve the quality of their products, and also the efficiency and flexibility of their processes. Automation of the sensing and manipulation tasks is one of the most promising alternatives for achieving these goals, but it poses important challenges. These challenges apply, in particular, when dealing with objects that are large, fragile or that show complex shapes or small deformations [2]. Control approaches require in these cases a continuous and complete feedback of the object's state [3], [4], which allows to control the object's shape and evaluate the overall quality of the manipulation process. Such control and feedback systems usually employ multiple agents and sensing units [5], [6], [7]. Besides the industrial interest of these perception systems, medical applications related to robot-assisted surgery also demand accurate and flexible sensing solutions [8], [9].

We focus our interest in industrial tasks where the shape of an object is deformed following a prescribed guideline. This is the case for manufacturing processes that transform raw materials (cloths, plastics...) into objects with specific shapes. Guaranteeing a continuous feedback of the object's state in industrial environments may be challenging, owing to the highly dynamic and complex structure of deformable objects and the external perturbations. This is why our research problem consists in developing a multi-camera perception system to cover an ob-

ject whose shape changes over time, according to a set of prescribed requirements. Coverage is defined in terms of completion and accuracy of the detected object's contour. We also want to optimize the performance of the perception system by minimizing the number of sensing resources. The main properties we exploit to define the coverage task are visibility, as a measure of coverage completion, and resolution, as an accuracy indicator. A potential application of our system is the inspection of 3D printed parts, which are created by stacking layers of material. Our techniques could be applied to cover the lateral surface of the parts as they are built, to discover potential cracks and check the quality of the interlayer bonding. The usefulness of this proposal is also significant with tasks involving objects of great dimensions, such as the manipulation of large plastic containers, in which the lateral surface of the contour must be continuously supervised for quality purposes. Other example application would be the inspection of high precision machined parts during manufacturing. These kinds of parts have critical cross sections whose lateral surfaces must be controlled carefully to prevent defects. Our proposal could also be applied to monitor assembly tasks [10].

The perception of areas of interest, according to different requirements and constraints, includes techniques that range from coverage to tracking. Depending on the type of deployment of the sensors, we can classify these approaches as static or dynamic. The deployment criterium is closely related to wheter the perceived environment is time-dependent or not. Particularly, statically positioning the sensors is usually an appropriate strategy when there are no substantial changes over time in the perceived environment. The well-known art-gallery problem is essentially a static coverage strategy where the number of surveillance cameras is minimized. Recent developments include variants of this problem for

tracking of mobile intruders with fixed and mobile guards [11]. In general, those studies based on the general art-gallery problem do not include realistic models with sensing constraints [12]. This problem also shares some aspects with the one we tackle, but classical solutions to the art-gallery problem are not directly applicable to our time-varying scenario. Other static perception strategies with multiple sensors are selected for purposes like industrial inspection of 3D parts [13], 3D reconstruction of pipelines for inspection [14], buildings exterior surveillance [15], volume estimation of merchandise [16] and pose estimation in restricted space scenarios [17]. These kinds of works select the viewpoints with *ad hoc* criteria or from a predefined candidate set, which may be non-optimal, and have limited adaptability. In contrast, our approach computes at every deformation instant the optimal set of cameras in the continuous space, parameterized in polar coordinates (angle and radius).

Approaches where there is a single static sensing unit have been proposed for estimating the deformation properties of deformable objects [18]. In other cases, dynamic deployment may be necessary to obtain complete information of the area of interest. This happens in coverage systems of 3D rigid parts for industrial inspection tasks [19] and in structure from motion approaches [20]. Our target problem shares common points with the structure from motion topic, but in that case the environment is static in general [21], not dynamic as ours. Dynamic deployment of multiple sensing units may also be required for full perception at every time instant. This is the case in centralized approaches where small teams of robots monitor 3D static environments [22] or enclose and track a mobile target [23], and also in decentralized strategies where 2D static environments are monitored by large teams of robots [24].

As for the active perception field [25], [26], works in this domain are dynamic in general. Multi-camera centralized networks, where the position and orientation of each sensor are optimized, are considered in studies of this field for shape recovery of moving deformable objects [27], [28]. Compared to our approach, these works do not minimize the number of cameras that are necessary to recover the shape of the object.

The present work improves and extends the methods of an earlier work [29] with the following main contributions: (a) a new improved formulation of the optimization problems, which is less constrained and more efficient; (b) the computation of the maximum angular resolution of a shape; (c) a fast coverage technique based on local optimization of the cameras; (d) a novel analysis of the geometrical visibility properties of a shape and (e) additional results and analyses from different experiments in both simulated and real scenarios.

The rest of the paper is organized as follows: Section 2 presents the problem setup and the optimization objectives, Section 3 describes a new method for computing the maximum visibility and resolution of the object's contour, Section 4 indicates the proposed procedure to obtain the minimum number of cameras with near-optimal configurations, Section 5 contains the local optimization technique we propose to operate online, Section 6 shows the experimental results, and Section 7 explains the final conclusions and the future research directions.

## 2. Problem statement

### *2.1. Framework and assumptions*

Let us consider an object that undergoes non-negligible deformations due to some specific manipulation process. The object's shape is defined by the 2D con-

tour of the object's vertical projection onto the ground plane. For inspection purposes, the contour must be sampled at each instant $k$ ($k = 1, ..., K$) by a set of 3D cameras. These cameras $\mathbf{c}_i(k)$ ($i = 1, ..., C$) are able to position and orient themselves around the contour's centroid $\mathbf{g}(k)$ (e.g. they are mounted on mobile robots or robot arms). In particular, their degrees of freedom are the following: (i) $\psi_i(k)$, the angle around $\mathbf{g}(k)$ where the center of the camera is located, (ii) $d_i(k)$ ($d_{min} \leq d_i(k) \leq d_{max}$, $\forall i$), the distance between the camera center and $\mathbf{g}(k)$, and (iii) $\phi_i(k)$, the camera orientation. We consider that the cameras' movements obey single-integrator kinematics, and no vision model is selected *a priori*. Instead, we focus on ray tracing from the optical center of the cameras, and angular and range parametric constraints are applied to the rays to approximate the restricted Field of View (FOV). This formulation is compatible with any model based on ray tracing from an optical center, like laser-based sensors, or even with tools like spray guns. Figure 1 shows the overview of the studied system.

The different assumptions we consider in the present study are reported next:

**Assumption 1.** ***Known data.*** We consider the following:

1. Cameras can get accurate 2D point clouds of the environment.
2. We know the state of the cameras with respect to the object $\mathbf{c}_i(k) = [\psi_i, d_i, \phi_i]$ at any time.
3. We know the desired deformation of the object. Therefore, we have at each instant $k$ an approximated shape of the 2D contour of the object, called reference contour, as well as an approximated $\mathbf{g}(k)$.

**Assumption 2.** ***Slow deformations.*** We assume that the object's deformations (i.e. contour shape variations) between each two consecutive deformation instants $k$ and $(k+1)$, $\forall k$, are small. We assume, therefore, that the dynamics of the deformable object are as slow as to allow the cameras to reach their required positions and orientations $\forall k$.

It is not unusual to consider reference models that complete missing information for reconstructing complex setups [14] or deformable objects [30]. In our case, the reference contour gives the desired shape of the object during the deformation process. For example, for inspection of 3D printed parts, the desired contour shape of the 2D layers, which is known *a priori*, would represent the reference contour. Depending on the task, it does not need to be accurate necessarily. If the purpose is, for instance, to inspect a large object, a rough approximation of the contour may suffice. A collection of reference segments is obtained by sampling the reference contour, and the number and size of the reference segments $\mathbf{s}_j(k) = \left[\mathbf{v}_j, \mathbf{v}_{(j+1)}\right]$ $(j = 1, ..., S)$ must be set according to the requirements of the inspection task and the complexity of the contour shape.

For clarity purposes, we will omit in general the $(k)$ term in the rest of the paper.

### *2.2. Optimization objectives*

We set the coverage objectives, in terms of visibility and resolution, from an optimization point of view.

**Definition 1.** ***Visibility cost.*** We define the *visibility cost*

$$\gamma_v = \frac{\sum_{j=1}^{S} \gamma_{v_j}}{\sum_{j=1}^{S} \vartheta_j^*} \text{, with } \gamma_{v_j} = \begin{cases} \vartheta_j^* - \vartheta_j & \text{if} \quad \vartheta_j^* > \vartheta_j \\ 0 & \text{otherwise} \end{cases} . \tag{1}$$

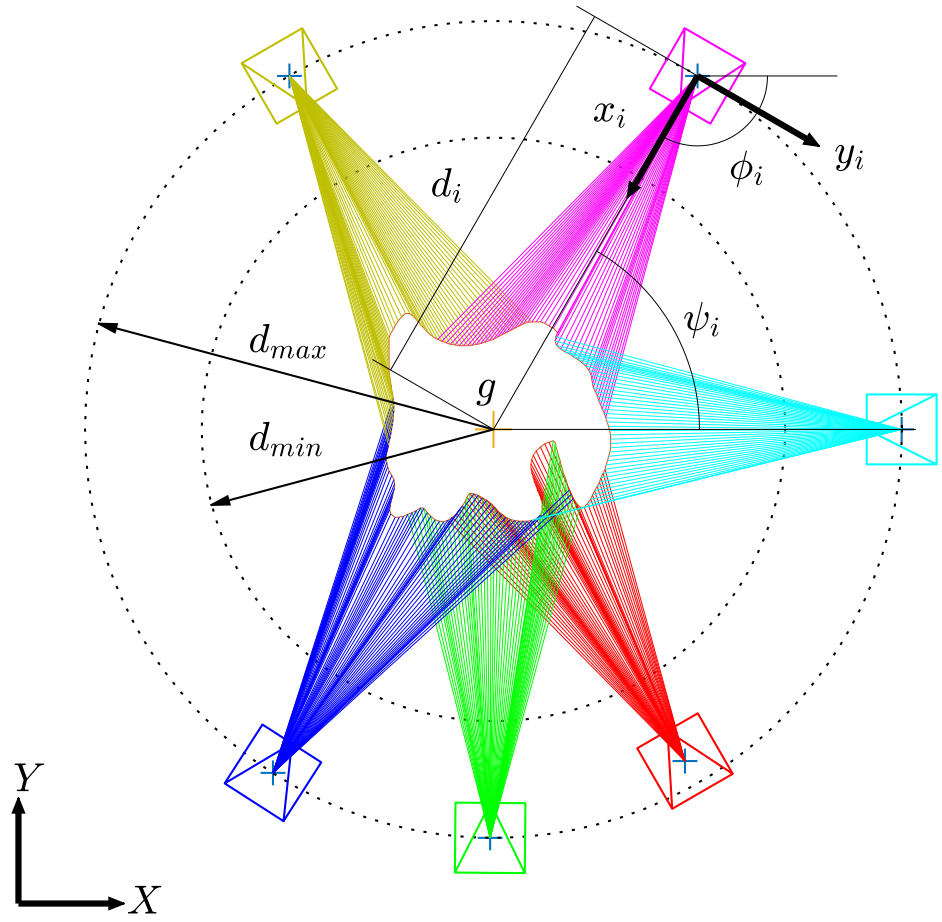



Figure 1: Overview of the system we consider. At the center of the figure, the 2D sampled contour of the object is fully covered by six restricted FOV cameras. As the object deforms, the cameras will synchronously move and rotate so that the coverage is guaranteed at all times. The global axes ($X$, $Y$) and the local optical axes of camera $i$ ($x_i$, $y_i$) are represented. In addition, we show the projection rays that link the center of each camera to the contour vertexes.

$\vartheta_j$ indicates the number of cameras that detect segment $\mathbf{s}_j$ at the $k$ instant, and $\vartheta_j^*$ represents the target number of cameras that must detect $\mathbf{s}_j$ according to the requirements of the inspection task.

The *visibility cost* describes how far the system is from the objective of detecting the contour segments as required, in terms of visibility. Note that detecting a segment with more cameras than necessary does not contribute to decrease the value of the *visibility cost*.

**Definition 2.** ***Resolution cost.*** We define the *resolution cost*

$$\gamma_r = \frac{1}{\beta \sum_{j=1}^{S'} 1/r_j^*} \sum_{j=1}^{S'} \frac{\beta - r_j}{r_j^*} \quad , \tag{2}$$

where $\beta$ is the cameras' angle of view, $r_j$ indicates the maximum angular resolution of segment $\mathbf{s}_j$, $r_j^*$ is the target angular resolution of $\mathbf{s}_j$ and $S'$ is the number of visible segments with $r_j^* > 0$.

The *resolution cost* allows to evaluate the angular resolution of the segments with respect to the maximum resolution. The target resolution should take higher values in those areas where more accurate perception is required, as it will drive the cameras towards front-facing positions with respect to the segments. In case no specific resolution is required in an area, the segments there should be assigned $r_j^* = 0$.

As a consequence, the near-optimal configurations of the cameras $\mathbf{c}_i^* = [\psi_i^*, d_i^*, \phi_i^*]$ are the ones that reduce both costs to, at least, these values:

$$\gamma_v^* = 0\,, \tag{3}$$

$$\gamma_r^*(r_j = r_j^* \text{ if } r_j > r_j^*) = \beta \sum_{j=1}^{S'} \left(\frac{1}{r_j^*}\right) - S'\,. \tag{4}$$

Then, the problem we tackle consists in configuring a minimal set of cameras in a near-optimal manner so that they cover a 2D contour that deforms over time, as specified in terms of visibility and angular resolution. Inter-camera collisions and occlusions must be avoided to guarantee coverage in the whole deformation process. Note that placing a single camera on top of the contour for inspection is not a valid solution in our case. This is due to the fact that lateral views of the contour would not be obtained for inspection.

Properly defining the target visibility and angular resolution is crucial for obtaining the desired behavior of the system. In the next section, some tools are proposed for defining the target visibility and angular resolution with feasibility guarantees.

## 3. Maximum visibility and resolution

In order to define feasible objectives we need to identify the segments that are inevitably occluded, and also the maximum angular resolution of each detected segment that a camera placed around $\mathbf{g}$ can obtain.

**Definition 3.** ***Maximum visibility.*** The maximum visibility $\vartheta_j^{max}$ of a segment is a value that indicates the maximum number of cameras that detect $\mathbf{s}_j$ from the distance $d$. If $\mathbf{s}_j$ can be detected, $\vartheta_j^{max} > 0$, and $\vartheta_j^{max} = 0$ otherwise.

The *maximum visibility* of convex contour shapes is known *a priori* in an obstacle-free framework ($\vartheta_j^{max} > 0, \forall j$). However, when dealing with non-convex shapes non-visible zones may appear due to auto-occlusions.

Previously to computing this property, we need a way to determine whether a segment $\mathbf{s}_j$ can be detected or not from certain camera position. For this purpose, we consider a system based on *bi-partite visibility graphs*.

**Definition 4.** ***Bi-partite visibility graph.*** The *bi-partite visibility graph*

$$G_v = (\mathbf{C}_T, \mathbf{S}_{\mathbf{ref}}, \mathbf{E}) , \tag{5}$$

$$\mathbf{C}_T = \{\mathbf{c}_1, ..., \mathbf{c}_i, ..., \mathbf{c}_C\},\ \mathbf{S}_{\mathbf{ref}} = \{\mathbf{s}_1, ..., \mathbf{s}_j, ..., \mathbf{s}_S\} , \tag{6}$$

is the graph where an edge $\mathbf{e}_{i,j} \in \mathbf{E}$ connects the camera center $\mathbf{c}_i$ with the vertex $\mathbf{v}_j \in \mathbf{s}_j$ if the virtual line that links them in the 2D space does not intersect any obstacle, i.e. if $\mathbf{v}_j$ is visible from $\mathbf{c}_i$. Each edge's weight is equal to the Euclidean distance $d_E(\mathbf{c}_i, \mathbf{v}_j)$.

Then, if two edges $\mathbf{e}_{i,j}$ and $\mathbf{e}_{i,j+1}$ exist, such that $\mathbf{e}_{i,j} = \left[\mathbf{c}_i, \mathbf{v}_j\right]$ and $\mathbf{e}_{i,j+1} = \left[\mathbf{c}_i, \mathbf{v}_{(j+1)}\right]$, we determine that segment $\mathbf{s}_j = \left[\mathbf{v}_j, \mathbf{v}_{(j+1)}\right]$ is visible from $\mathbf{c}_i$. Figure 2 shows a visible segment $\mathbf{s}_j$ with its angular resolution $r_j$. We formulate next the method to compute the *maximum visibility* of a 2D discretized contour from a prescribed distance to $\mathbf{g}$.

**Proposition 1.** Consider the *bi-partite visibility graph* of an scenario with no obstacles where the object is surrounded by infinite omnidirectional cameras ($C \rightarrow \infty$) with unlimited FOV range. The cameras completely cover a circumference of radius $d$ centered at $\mathbf{g}$. In such situation, the graph provides exactly the number of segments that are visible from that circumference, with $\vartheta_j^{max} \rightarrow \infty$ if $\mathbf{s}_j$ is visible and $\vartheta_j^{max} = 0$ otherwise.

PROOF. By definition 4, $\mathbf{s}_j$ is visible from $\mathbf{c}_i$ if its two vertexes, $\mathbf{v}_j$ and $\mathbf{v}_{(j+1)}$, are connected by edges to $\mathbf{c}_i$. Thus, $\vartheta_j^{max}$ of the setup described in Proposition 1 can be computed as follows:

$$\vartheta_j^{max} = \sum_{i=1}^{C} (\exists \mathbf{e}_{i,j} \wedge \exists \mathbf{e}_{i,j+1}) , \tag{7}$$

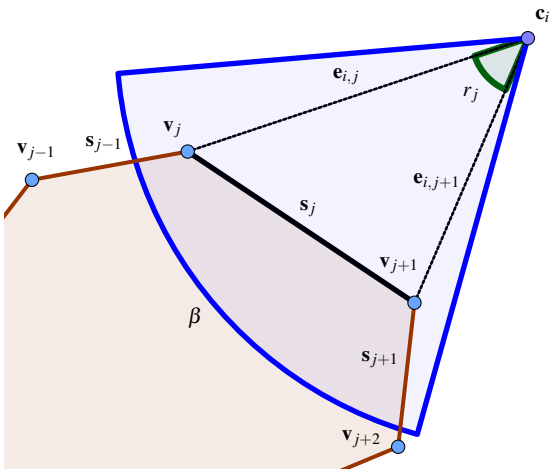



Figure 2: Example of visible segment. Segment $\mathbf{s}_j$ is visible from $\mathbf{c}_i$ with angular resolution $r_j$ (in green), under the angle of view $\beta$ (in blue).

where we treat the boolean outputs *true* and *false* as integers 1 and 0, respectively. Given that every possible sensor location at a distance $d$ around $\mathbf{g}$ is considered and that omnidirectional cameras are able to detect in any possible direction, every possible valid edge will be determined. Therefore, every visible segment will be obtained with $\vartheta_j^{max} \to \infty$, as well as the occluded segments with $\vartheta_j^{max} = 0$.

**Remark 1.** Note that in practice the number of cameras $C$ is a finite value. For this reason, an exact accuracy for identifying occluded segments, when computing $\vartheta_j^{max}$, may not be achieved (infinite positions for detection are not evaluated). Despite this, we obtain a useful measure: the closer $\vartheta_j^{max}$ is to $C$, the wider the range of possible locations from which $\mathbf{s}_j$ can be detected. This measure allow us to know the difficulty of detecting a segment compared to the rest.

Next Proposition includes additional reasoning about the *maximum visibility*, concerning the distance from which this one is obtained.

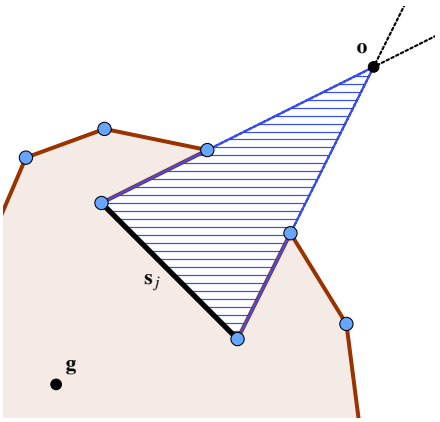



Figure 3: Example of contour shape with a dovetail slot. Segment $\mathbf{s}_j$ will be fully detected only by those cameras that are placed within the blue dashed area.

**Proposition 2.** Consider a *bi-partite visibility graph* with infinite omnidirectional and unlimited FOV range cameras, which are placed at a distance $d$ from $\mathbf{g}$. Then, by reducing the distance $d$ the visibility of the contour segments will remain constant or it will increase for any possible shape, but it will never decrease.

PROOF. If we consider the paradigmatic case of a contour that includes a dovetail shaped slot, whose inner part consists of a single segment $\mathbf{s}_j$, only those cameras that are placed inside an specific area in front of the slot will be able to detect the segment. This area is contained between $\mathbf{s}_j$ and the two adjacent lines that start at both vertexes of $\mathbf{s}_j$ and intersect at the point $\mathbf{o}$ (Fig. 3). Thus, only if $d \leq \overline{\mathbf{go}}$ the segment $\mathbf{s}_j$ will be detectable. In addition, no geometric construction exists such that the opposite effect occurs: the area in which the camera must be placed to detect the segment will always start at the segment, and it will extend from there on.

**Remark 2.** An interesting observation arises when applying the test of Proposition 1 to a continuous contour, e.g. given by a parametric curve $\{x(\psi), y(\psi)\}$. As long as the infinite cameras lie outside the convex hull of the contour, $\vartheta_j^{max}$ will remain constant for every segment independently of the distance $d$. This is explained by the fact that when obtaining $\vartheta_j^{max}$, each edge connecting a camera center to a contour vertex will determine a semi-infinite projection ray. These rays start at the contour vertexes and have the direction of the edges, and their main property is that any camera placed on them will be able to detect the contour vertex. Given that the cameras can be placed on any circumference outside the convex hull centered at $\mathbf{g}$, and that all projection rays cross the convex hull, any circumference of infinite cameras will intersect all projection rays. Thus, $\vartheta_j^{max}$ values will remain constant independently of $d$.

**Definition 5.** ***Maximum resolution.*** The *maximum resolution* $r_j^{max}$ of a segment is a value that indicates the maximum angular resolution of $\mathbf{s}_j$ that can be obtained from the distance $d$ to $\mathbf{g}$ by a camera. If $\mathbf{s}_j$ can be detected, $r_j^{max} > 0$, and $r_j^{max} = 0$ otherwise.

The setup described in Proposition 1, which allows us to obtain the *maximum visibility*, is also appropriate without modifications for obtaining the *maximum resolution* in the following manner:

$$r_j^{max} = \max_i \left[ \cos^{-1} \left( \frac{\mathbf{e}_{i,j} \cdot \mathbf{e}_{i,j+1}}{\|\mathbf{e}_{i,j}\|_2 \cdot \|\mathbf{e}_{i,j+1}\|_2} \right) \right] . \tag{8}$$

Note that the shorter the distance $d$, the greater $r_j^{max}$ values are.

The *maximum visibility* and the *maximum resolution* establish upper bounds that cannot be surpassed in terms of visibility and resolution of the object's contour. It is important to mention that we consider point cameras when computing

the *maximum visibility* and *the maximum resolution*. This may imply that the real system, with physical constraints, cannot obtain the maximum values for all the segments at the same time.

## 4. Offline optimization

### *4.1. Limits of the number of cameras*

In addition to the maximum visibility and resolution, we are going to compute the lower and upper bounds of the number of cameras to perform the required coverage. The minimum number of cameras can be obtained with additional formulation from the setup described in Proposition 1. Approaches dealing with coverage problems often include a visibility map or visibility function, which relates the point of view of a sensor with the parts of the object it is covering at certain time instant [15], [28]. From Proposition 1 we can also consider a visibility matrix $\mathbf{V}_{S\times C} = [V_{ji}]$ that indicates which segments are detected by each camera, i.e. $V_{ji} = 1$ means that segment $\mathbf{s}_j$ is visible for the camera $\mathbf{c}_i$, and $V_{ji} = 0$ otherwise. Then, we define the following integer linear programming problem:

$$\begin{aligned} \text{Given} \quad & \mathbf{V}, [\vartheta_1^*, \vartheta_2^*, ..., \vartheta_S^*]^T \\ \text{minimize} \quad & \sum_{i=1}^{C} \varepsilon_i \\ \text{subject to} \quad & \mathbf{V} \cdot [\varepsilon_1, \varepsilon_2, ..., \varepsilon_C]^T \geq [\vartheta_1^*, \vartheta_2^*, ..., \vartheta_S^*]^T \,, \\ & \varepsilon_i \in \{0, 1\}, \; \forall i \,, \end{aligned} \tag{9}$$

where $\varepsilon_i = 1$ *activates* the camera $\mathbf{c}_i$ and $\varepsilon_i = 0$ *deactivates* it. The final cost, after solving (9), represents the minimum number of omnidirectional cameras, with unlimited range, placed at a distance $d$ from $\mathbf{g}$ that achieve the target visibility.

Here we only consider the visibility objective, so that we get a more conservative solution than considering both visibility and resolution.

As for obtaining the upper bound of the number of cameras, we follow a conservative area division strategy. We compute the maximum number of cameras as

$$\text{truncation}\left(\frac{d_{max}^2 - d_{min}^2}{d_c^2/4}\right) \geq C\,, \tag{10}$$

where radii $d_{min}$ and $d_{max}$ represent the space where the cameras are allowed to move, and $d_c/2$ is the radius of the minimum bounding circle of a camera. Essentially, we obtain with this equation the maximum number of cameras fitting in the allowable space in an approximate but conservative manner.

### *4.2. Minimization problem*

Once we know how to set feasible $\vartheta_j^*$ and $r_j^*$, and also the range where to search the minimum number of cameras, we can define the optimization problem at instant $k$ we are interested in.

Firstly, we define a new type of *bi-partite visibility graph* with the purpose of including the FOV constraints of the cameras into the problem.

**Definition 6.** ***Restricted visibility graph.*** The *restricted visibility graph* is a *bi-partite visibility graph* where all edges connecting a camera with the contour vertexes are contained within the cameras FOV limits $\beta$ and $L_{max}$.

This new graph is created by comparing the weight of each edge $\mathbf{e}_{i,j}$, given by the Euclidean distance $d_E(\mathbf{c}_i, \mathbf{v}_j)$, with the cameras' maximum range $L_{max}$, and the angle between the edge and the camera's local axis $x_i$ (Fig. 1) with the cameras' angle of view $\beta$. If the weight is greater than $L_{max}$ or the angle is greater than $\beta/2$,

we consider that $\mathbf{v}_j$ is not visible from $\mathbf{c}_i$, and therefore we remove the edge from the graph. After the comparison and filtering, the intersections checking is applied to the remaining edges. In this case, the set of obstacle segments includes not only the set of segments of the object $\mathbf{S_{ref}}$, but also the set of camera segments. This is due to the fact that the FOV of a camera may be occluded by other neighboring cameras. The set of camera segments is obtained by sampling the contour of the geometric 2D camera model, which represents the 2D shape of the cameras.

With these new constraints for computing $\vartheta_j$ and $r_j$, the optimization problem is defined as follows:

$$\begin{aligned} &\text{Given} \quad \mathbf{C}_T, \mathbf{S_{ref}} \\ &\underset{\mathbf{C}_T}{\text{minimize}} \quad \gamma = F \cdot \gamma_v + (1-F) \cdot \gamma_r + \gamma_\sigma \\ &\text{subject to} \quad d_E(\mathbf{c}_i, \mathbf{c}_l) \geq d_c \, , \\ &\qquad\qquad \forall i,l = 1,...,C, \; i \neq l \, . \end{aligned} \tag{11}$$

$\gamma$ is the cost function of the problem, that includes a linear combination of $\gamma_v$ and $\gamma_r$ through the interest factor $F \in [0,1]$. $F$ is a user-defined constant that regulates the relative contribution of the visibility and resolution terms, as required by the specific coverage task. The remaining term, $\gamma_\sigma$, exploits the *potential field approach* to prevent collisions in those scenarios that require two cameras to be close to each other:

$$\gamma_\sigma = \frac{1}{(2 \min_{\forall i,l, i \neq l} \{d_E(\mathbf{c}_i, \mathbf{c}_l)\}/d_c - 1)^w} \tag{12}$$

where the 'min' operator selects the minimal distance between two neighboring cameras, $d_c$ is the minimal safety distance between two neighboring cameras, and $w$ is an exponent that regulates the distance where $\gamma_\sigma$ is effective. This exponent

$w > 0$ is tuned according to the safety requirements, so that $w$ is small when large separations are demanded, and large otherwise. Thus, the $\gamma_\sigma$ term introduces flexibility for collision avoidance, with a positive contribution that shows near zero values when the cameras are widely spread and near 1 values when the distance between two of them approaches to $d_c$. In addition to $\gamma_\sigma$, the set of non-linear constraints that we include at the end of (11) prevents two cameras from getting closer than $d_c$ under any circumstances. These constraints also avoid considering near-infinite values of $\gamma_\sigma$ that can lead to a malfunctioning of the *pattern search* optimizer.

We propose an iterative technique to solve (11) at the deformation instant $k$. In the first iteration the problem is solved for the minimum number of cameras computed in (9), and the values of $\gamma_v$ and $\gamma_r$ are checked. If values in (3) and (4) are satisfied, the process stops, and the near-optimal configurations $\mathbf{C}_T^*$ and the near-optimal number of cameras $C^*$ are obtained. Otherwise, the problem is iteratively solved for $C := C + 1$ until equations (3) and (4) are satisfied or the maximum number of cameras from eq. (10) is reached. We solve (11) with the *pattern search method* [31], which is a derivative-free approach compatible with our cost function $\gamma$. Although global optimality is not guaranteed, this method always provides, at least, a local optimum.

### *4.3. Minimization problem of the complete deformation process*

We extend the result obtained in the previous section, the minimum number of cameras $C^*(k)$ at the deformation instant $k$, to the complete deformation process. The first step consists in solving (11) for a set of instants such that Assumption 2 is still satisfied at each two consecutive instants. When we have the solution for each selected $k$, we compute the minimum number of cameras that are required to

**Algorithm 1** Compute $C^*$.

Set $k \leftarrow 0$

$C^* \leftarrow C_{min}(k)$ # obtained from (9)

**while** $k \leq K$ **do**

  Initialize $\gamma_v > \gamma_{v_{stop}}$ and $\gamma_r > \gamma_{r_{stop}}$

  $C^*(k) \leftarrow C_{min}(k) - 1$

  **while** $((\gamma_v > \gamma_v^*)$ or $(\gamma_r > \gamma_r^*))$ and $(C(k)^* < C_{max})$ **do**

    # $C_{max}$ is obtained from (10)

    $C^*(k) \leftarrow C^*(k) + 1$

    Minimize $\gamma(k, \psi_i, d_i, \phi_i)$ # defined in (11)

  **end while**

  **if** $C^* < C^*(k)$ **then**

    $C^* \leftarrow C^*(k)$

  **end if**

  $k \leftarrow k + 1$

**end while**

**return** $C^*$

perform the coverage task as $C^* = \max_{\forall k} C^*(k)$. This conservative measure improves the robustness against unexpected situations, although it may overfeed the system with unnecessary cameras at certain instants. After that, the optimization is performed again with $C(k) = C^*, \forall k$, and the reference positions and orientations are computed. The main steps involved in the minimization of the number of cameras for the full deformation process, and the sequence in which they are performed, are shown in Algorithm 1.

It is important to highlight that this process is offline, i.e. it is performed over the reference contours and before the real task takes place. If the evolution of the object's deformation is similar to the planned one during the real coverage task and the optimization objectives are still achieved, these reference configurations are the ones the cameras will adopt. The target visibility and resolution are evaluated online by comparing the detected segments with the ones in $\mathbf{S_{ref}}$. During the real coverage task, if one of the points detected by the cameras, at least, lies on an area containing a reference segment, we consider that the reference segment is detected. The accuracy of the selected cameras determines the dimensions of this area. Then, if equations (3) and (4) are satisfied at $k$, each camera will follow the shortest path from $\mathbf{c}_i(k)$ to $\mathbf{c}_i(k+1)$, while avoiding collisions with the neighboring cameras to reach the next optimal configuration. In case deformation evolves in a different manner than the planned one, we propose in the next section an online reactive strategy that improves the achievement of the optimization objectives in a setup with high uncertainty.

## 5. Online optimization

During the real coverage task the object may undergo unexpected deformations. In order to properly capture these deformations and continue to cover the object as required, the system must be able to reconfigure itself in an online manner if necessary.

In case abnormal deformations (over a threshold) are detected, the object's reference contour is updated with the cameras data. Then, we propose an online repositioning technique that performs local optimizations at the camera level, as opposed to the offline method, where optimizations occur at the set level. This method provides near-optimal configurations of lower quality than the global optimization, but much faster than the former one. According to Assumption 2, deformations are small between $k$ and $(k+1)$, and therefore we consider that $\mathbf{c}_i(k)$ and $\mathbf{c}_i(k+1)$ will be close. Then, we can restrict the values of the optimization parameters and perform individual optimizations with each camera. The new local optimization problem at the deformation instant $(k+1)$ is defined as follows:

$$
\begin{aligned}
&\text{Given} \quad \mathbf{C}_T,\, \mathbf{S} \\
&\underset{\mathbf{c}_i}{\text{minimize}} \quad \gamma = F\cdot\gamma_v + (1-F)\cdot\gamma_r + \gamma_\sigma \\
&\text{subject to} \quad d_E(\mathbf{c}_i(k+1), \mathbf{c}_l(k+1)) \geq d_c\,, \\
&\qquad\qquad \mathbf{c}_i(k)\cdot(1-m) \leq \mathbf{c}_i(k+1) \leq \mathbf{c}_i(k)\cdot(1+m)\,, \\
&\qquad\qquad \forall i,l = 1,...,C,\; i \neq l\,,
\end{aligned}
\tag{13}
$$

where $\mathbf{S}$ is the set of detected segments and $m$ is a user-defined threshold that sets the allowed margin to perform the local optimization. It is worth mentioning that, despite the fact that only the configuration of one camera is optimized each time, the complete visibility graph with all the cameras is considered. The local

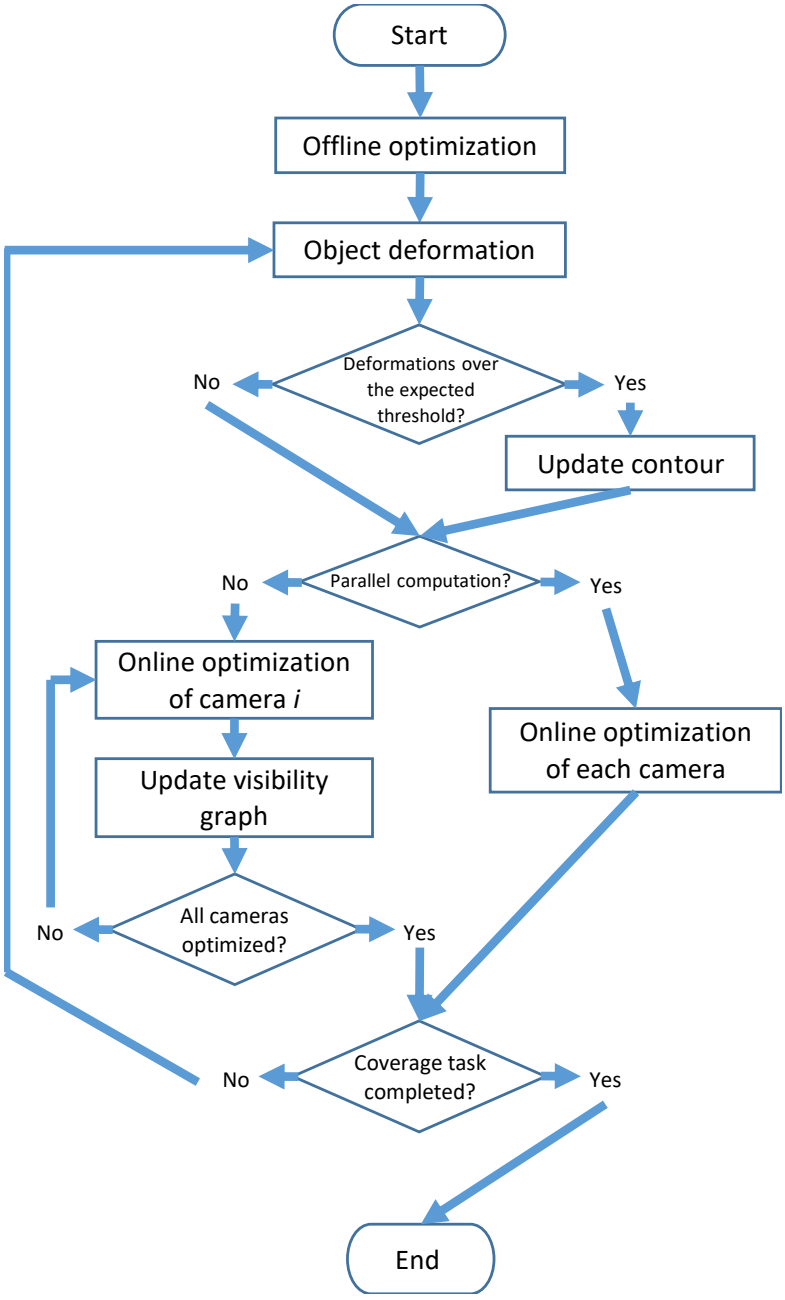



Figure 4: Flow chart including the main elements in the online optimization method.

optimizations can be solved in parallel, but better results in terms of optimality can be obtained if the problem is solved sequentially for each camera, at the expense of a higher computational cost. After each local optimization, the visibility graph is updated with the newly computed configurations.

The local optimization method also allows to relax Assumption 1.3 as follows: The deformation of the object is unknown, but an approximated shape of the 2D contour of the object, called reference contour, as well as an approximated $\mathbf{g}(k)$ are known *at the initial deformation instant* $k = 1$.

This means that we can utilize this technique even when the deformation evolution is unknown *a priori*, provided that the initial contour shape is known. As long as the detection of segments with coverage requirements ($\vartheta_j^* > 0$ or $r_j^* > 0$) is maintained from $k$ to $(k+1)$, the local optimization can be directly applied over the updated contour. At the initial deformation instant we still need to apply the iterative optimization, in order to obtain the set of $C^*$ minimal cameras with their initial near-optimal configurations. $C^*$ will remain constant for the whole deformation process, as well as the target visibility and resolution. Note that maintaining the target values over time requires tracking and matching of the contour. We consider that the development of these techniques is out of the scope of this study, and we assume they are given. Figure 4 shows a flow chart with the main processes of the online optimization method.

It may happen that the local optimizations are not able to achieve the target visibility and resolution at some $k$. In such situations, the global optimization, with the complete set of cameras, can be executed online.

## 6. Experimental results

In this section, we present different experiments focused on validating the theoretical results and techniques we have developed. We evaluate in Matlab® a general set of cases, with diverse shapes and deformation processes. In the absence of a specific data set with 2D objects that undergo deformation, the 2D shapes we use in the simulations have been obtained from the MPEG-7 data set [32]. This data set contains many different images in bitmap format. From these images, we have extracted a sampled contour by using a regularly spaced grid. Firstly, we will test the effectiveness of our methods for obtaining the desired results and behaviors, i.e. how well they meet the expected operational requirements. Secondly, we will evaluate their performance and utility for being applied on real perception tasks.

### *6.1. Maximum visibility and resolution tests*

We will start by testing the effectiveness of our methods for obtaining the *maximum visibility* and *maximum resolution* of contours from the data set. As we previously mentioned, this information is needed for setting feasible coverage objectives, and it will we exploited in the next sections. Given a 2D image from the data set, we extract and sample the contour to create a set of reference segments $\mathbf{S_{ref}}$. Then, we cover the sampled contour with a set of evenly spaced omnidirectional cameras with unlimited FOV range at a distance $d$ from $\mathbf{g}$, and afterwards we create and analyze *bi-partite visibility graph*. We have applied the method to the 1400 images of the MPEG-7 data set with the following common parameters, for uniformity purposes: $S = 200$ segments, $C = 500$ cameras and $d$ is equal to the greater side of each contour bounding box. Figure 5 shows some examples that

have been tested with our method, including their visibility ratios $v_r$, defined as

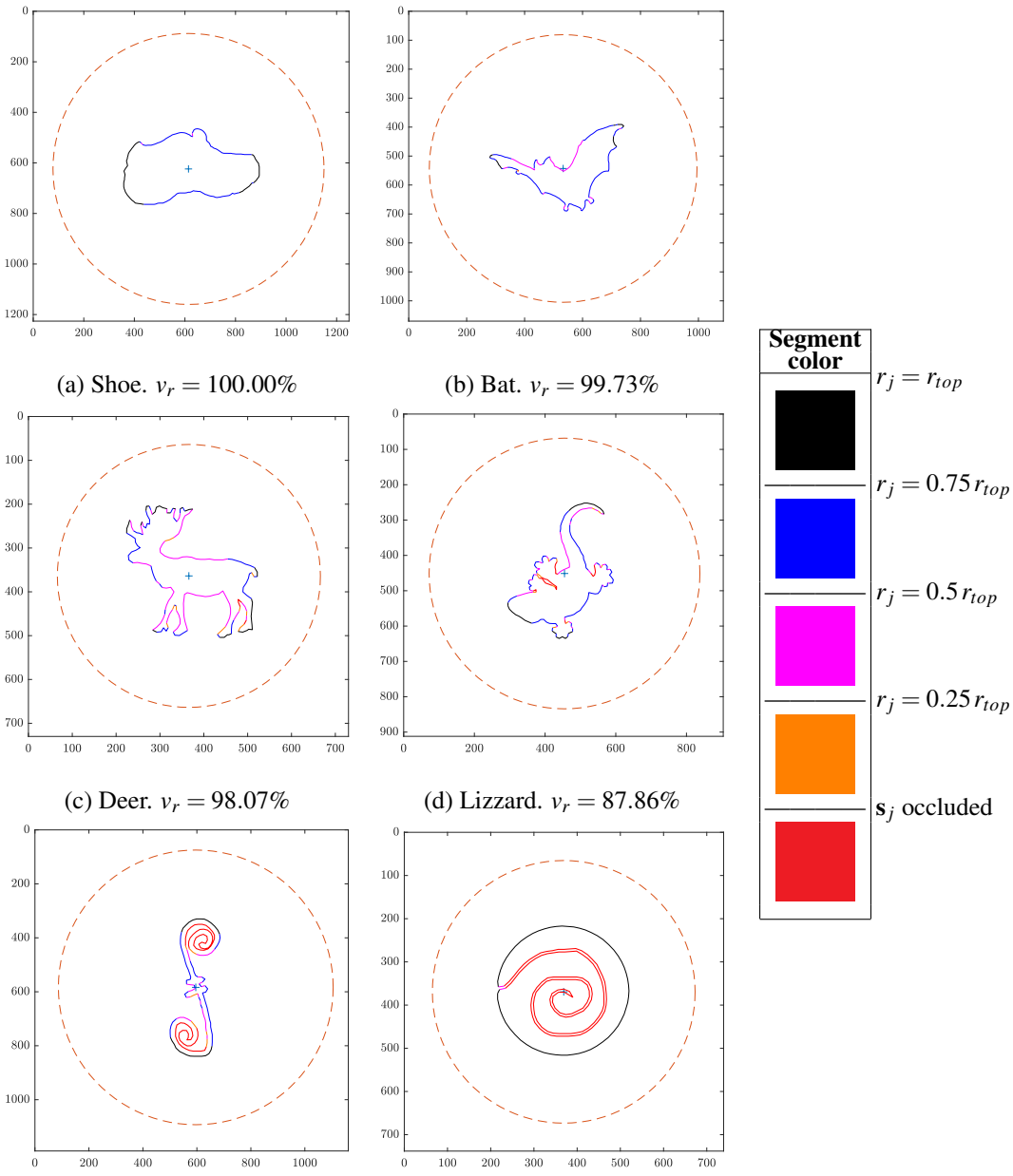
$$v_r(\%) = 100 \cdot visible\ perimeter\ /\ total\ perimeter\ . \tag{14}$$

The segments are colored depending on their maximum resolution $r_j$ with respect to $r_{top}$, the absolute maximum resolution, in five different intervals. Figure 6 shows a box plot of the $v_r$ values of all the images from the data set, grouped in six different categories according to the total contour perimeter. Results seem reasonable for different reasons. As expected, lower visibility ratios are obtained as the contour perimeter increases, since shapes with longer perimeters in the data set are usually more complex. We can see also that as the contour perimeter increases, so does scattering in the $v_r$ values. Besides, in Figure 5 the closer to perpendicularity the segments are to the radii of the circumference where the cameras are placed, the higher the $r_j$ values are. Additional statistical measures include an average visibility ratio of 96.98%, a 65.29% of the contours being fully visible, and a minimal visibility ratio of 32.70% (Fig. 5(f)). As no previous analyses about visibility and resolution of 2D shapes such as the one we present have been performed, this study may serve as a baseline for further research.

### *6.2. Cost function tests*

Now, we evaluate the effectiveness and performance of $\gamma$ (11). Given a 2D object, we analyze how the system is configured in different situations. In order to properly highlight the effects of each individual term in $\gamma$, we perform the following simulations over a squared contour shape:

a) $\gamma = \gamma_v$. Only visibility properties are analyzed. $\vartheta_j^* = 1$ for all contour segments, which corresponds to a coverage task where we want the full contour to be inspected by, at least, one camera.




(a) Shoe. $v_r = 100.00\%$ (b) Bat. $v_r = 99.73\%$

(c) Deer. $v_r = 98.07\%$ (d) Lizzard. $v_r = 87.86\%$

(e) Spring. $v_r = 59.16\%$ (f) Device. $v_r = 32.70\%$

Figure 5: Shapes from the MPEG-7 data set [32] with their visibility ratios. Segments are colored depending on their maximum resolution $r_j$ with respect to $r_{top}$, the absolute maximum resolution. Occluded segments are depicted in red (best seen in color and with zoom). The dashed line circumference indicates the distance $d$ from $\mathbf{g}$ where the omnidirectional cameras with unlimited FOV range are placed.

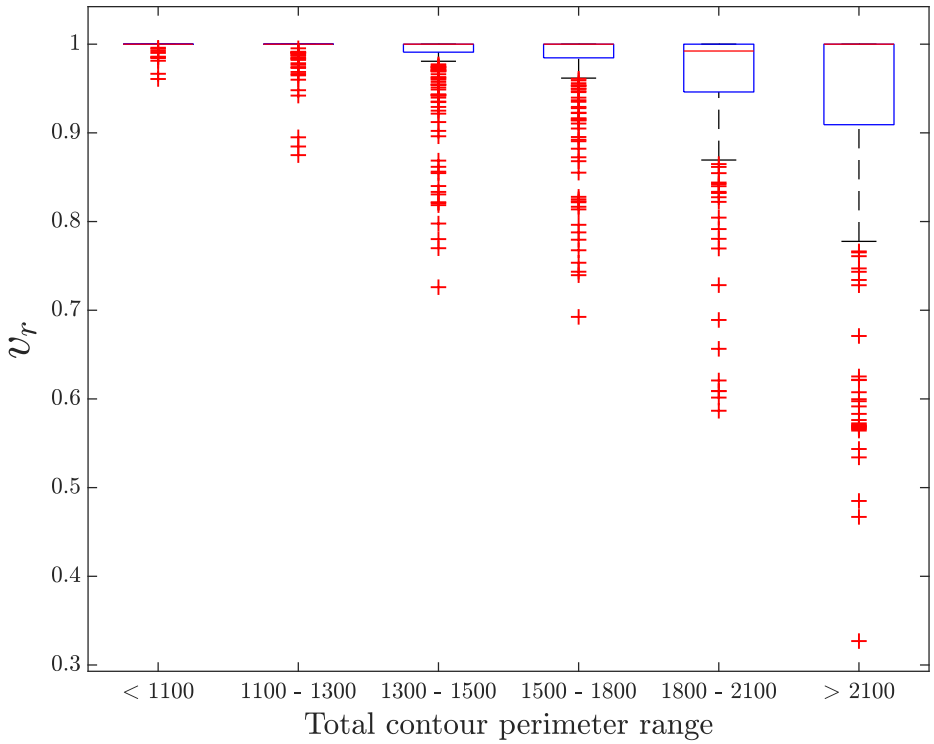



Figure 6: Visibility ratios of the 1400 contours from the MPEG-7 data set [32], contained in a single box plot. We can see that as the total perimeter increases, so does scattering, and also that the median is near 1 (full visibility) in all categories.

b) $\gamma = \gamma_r$. Only resolution properties are analyzed. $r_j^* = r_j^{max}/2$ for all contour segments, which corresponds to a coverage task where we want the full contour to be detected with, at least, the half of the maximum resolution.

c) $\gamma = (\gamma_v + \gamma_r)/2$. The effects of combining the visibility and resolution terms are studied. $\vartheta_j^* = 1$ for the segments of the top and bottom sides of the square, and $\vartheta_j^* = 0$ for the rest. $r_j^* = r_j^{max}/2$ for the segments of the right and left sides of the square, and $r_j^* = 0$ for the rest. It corresponds to a coverage task where we want some parts of the object to be simply inspected, while others must be inspected more accurately.

d) $\gamma = (\gamma_v + \gamma_r)/2 + \gamma_\sigma$. All terms are considered, including the $\gamma_\sigma$ term. The target visibility and resolution are set in the same manner as in statement c). It corresponds to a coverage task similar to the one of the previous case, but here the cameras must be more separated for safety reasons.

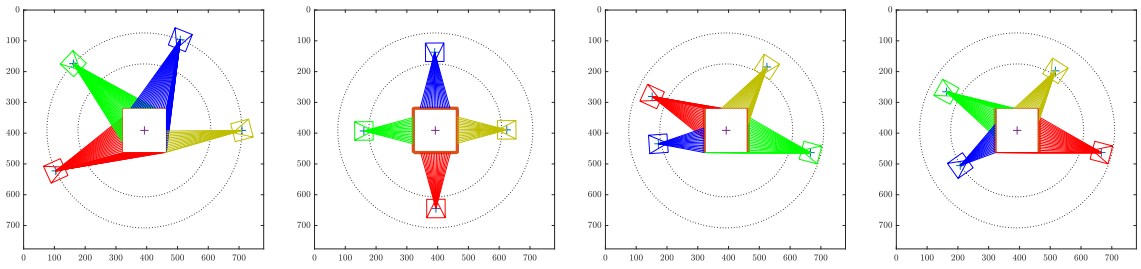
For these tests we set $C = 4$ cameras, $S = 100$ segments, $\beta = 30°$, $L_{max} = 400$ units, and the $\gamma_\sigma$ exponent is set to $w = 5$. These cameras are radially constrained to the space between the distances $d_{min}$ and $d_{max}$, which have been set to 1.5 and 2.2 times the greater side of the contour bounding box, respectively. Figure 7 shows the *restricted visibility graphs* optimized at each simulation, and Table 1 contains the values each term of $\gamma$ takes. We can see clearly how $\gamma_v$ induces the cameras to move towards the radial limit $d_{max}$, so that the FOV is exploited to the highest extent and the maximum number of segments are detected (in fact, in Fig. 7(a) the yellow camera could be omitted without affecting the result). A different tendency is driven by $\gamma_r$ in Fig. 7(b), which pushes the cameras to be closer to the object and as perpendicular as possible to the faces of the square, so

that higher angular resolutions are obtained. The combined effects of $\gamma_v$ and $\gamma_r$ reach the optimized state shown in Fig. 7(c), which is modified in Fig. 7(d) when adding $\gamma_\sigma$ to include wider separations between the cameras.

Additionally to these tests, we study the influence of $w$ in the effects that $\gamma_\sigma$ produces. The iterative optimization is applied to a circular contour with $S = 100$ segments, covered by a set of $C = 6$ cameras with FOV restricted to $\beta = 45^\circ$ and $L_{max} = 800$ units. The radial constraints of the cameras are the same as in the previous case. In this example, we consider a critical section of the contour that must be inspected by at least 4 cameras. Therefore, the target visibility of 20 consecutive segments is set to $\vartheta_j^* = 4$, while for the rest of the segments $\vartheta_j^* = 1$. This configuration produces a dense concentration of cameras. For better understanding the simulation results, the user-defined factor $F$ has been set to 1 to disable the resolution term. Then, there are opposite effects produced by $\gamma_v$ and $\gamma_\sigma$ in such setup: while the cameras tend to be close by the effect of $\gamma_v$ and the achievement of the required coverage, $\gamma_\sigma$ forces them to be separated. The balance here is modified by assigning decreasing values to the $w$ exponent. This extends the effective distance of $\gamma_\sigma$, which forces the cameras to be more and more segregated.

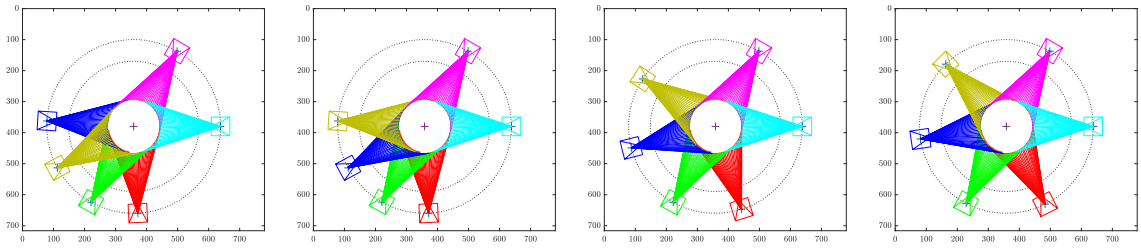
Figure 8 shows the graphical results of these optimizations for $w = 5, 4, 3, 2$, and Table 2 summarizes, for each $w$, the values the terms in $\gamma$ take. It can be seen that at the beginning, as $w$ decreases, $\gamma_v$ remains constant and $\gamma_\sigma$ increases. Then, when $w = 3$, $\gamma_v$ starts to grow and $\gamma_\sigma$ decreases, and finally for $w = 2$ both values increase.

These results show how adaptive the cost function is to different coverage tasks. By adjusting the configuration parameters (namely, $F$ and $w$) we can give preference to the visibility goal over the resolution goal or vice versa, or favor



(a) $\gamma = \gamma_v$ (b) $\gamma = \gamma_r$ (c) $\gamma = (\gamma_v + \gamma_r)/2$ (d) $\gamma = (\gamma_v + \gamma_r)/2 + \gamma_\sigma$

Figure 7: Four *restricted visibility graphs* resulting from different test with $C = 4$ cameras, over a squared shape. $\vartheta_j^* = 1$ in the thin orange line segments and $\vartheta_j^* = 0$ in the rest, while $r_j^* = r_j^{max}/2$ in the thick orange line segments and $r_j^* = 0$ in the rest.



(a) $w = 5$ (b) $w = 4$ (c) $w = 3$ (d) $w = 2$

Figure 8: Four *restricted visibility graphs* resulting from test cases with different values of $w$ in the $\gamma_\sigma$ term, over a circular shape. Thick orange lines represent those segments that must be detected by 4 cameras ($\vartheta_j^* = 4$).

Table 1: Evaluation of the terms of $\gamma$ with the squared shape, at each one of the four simulations (Fig. 7).

| **Test** | $F$ | $F\gamma_v$ | $(1-F)\gamma_r$ | $\gamma_\sigma$ | $\gamma$ |
|---|---|---|---|---|---|
| Fig. 7 (a) | 1 | 0 | 0 | 8.9956E-5 | 8.9956E-5 |
| Fig. 7 (b) | 0 | 0 | 9.5917E-1 | 1.0458E-4 | 9.5928E-1 |
| Fig. 7 (c) | 0.5 | 0 | 4.7290E-1 | 1.6684E-2 | 4.8958E-1 |
| Fig. 7 (d) | 0.5 | 0 | 4.7404E-1 | 8.1569E-4 | 4.7485E-1 |

Table 2: For each simulation with a different value of $w$, over the circular shape, values of the terms in $\gamma$ (Fig. 8).

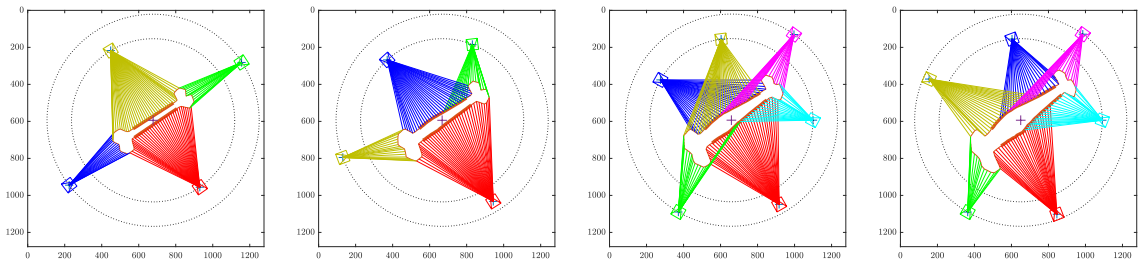
| **Test** | $\gamma_v$ | $\gamma_\sigma$ | $\gamma$ |
|---|---|---|---|
| $w = 5$ | 3.7500E-2 | 1.6720E-2 | 5.4220E-2 |
| $w = 4$ | 3.7500E-2 | 3.8068E-2 | 7.5568E-2 |
| $w = 3$ | 1.1875E-1 | 2.0267E-2 | 1.3902E-1 |
| $w = 2$ | 1.5625E-1 | 5.2453E-2 | 2.0870E-1 |

greater separations between the cameras in case higher levels of safety are needed.

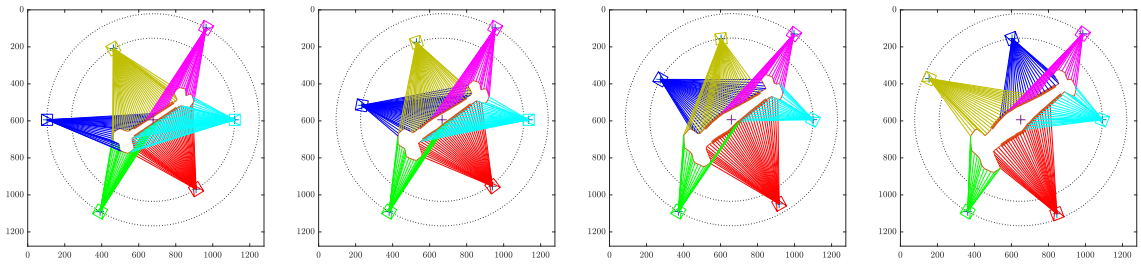
### *6.3. Offline optimization tests*

Our offline techniques for solving the minimization problem over a complete deformation process, are evaluated next. In this case, we consider a bone shape taken from the MPEG-7 data set and we sample it in S = 100 segments. We model deformation by applying different exponential distortion functions to the initial shape. Cameras' FOV are here constrained to $\beta = 60^\circ$ and $L_{max} = 600$ units, the minimum radial limit $d_{min}$ is equal to the greater side of the contour bounding

box, and the maximum limit $d_{max} = 1.3\,d_{min}$. With respect to the optimization objectives, a combination of visibility and resolution is established: $r_j^* = r_j^{max}/2$ and $\vartheta_j^* = 0$ for the segments of the top and bottom straight sides of the bone, and $r_j^* = 0$ and $\vartheta_j^* = 1$ for the rest. This definition simulates an inspection task where the top and bottom sections of the bone must be detected more accurately than the rest, but full contour coverage is still required. The coverage task could be applied, for instance, to a 3D printing process where a precise quality control is needed. Then, Algorithm 1 is executed to obtain the minimum number of cameras $C^*$ of the deformation process. Afterwards, the system is optimized at each $k$ with the already obtained $C^*$. Figure 9 shows the *restricted visibility graphs* resulting from the iterative optimization at different $k$ ($k = 0, 100, 200, 300$) from (a) to (d), and from (e) to (h) the *restricted visibility graphs* computed with $C^* = 6$ cameras. Table 3 includes, for three different cases we have selected for comparison, the values of each term in $\gamma$. The first row of Table 3 represents the case where the cameras remain fixed at evenly spaced positions (0, 90, 180 and 270 degrees around $\mathbf{g}(k = 0)$ and $d_i = (d_{max} + d_{min})/2, \forall i$), oriented towards $\mathbf{g}(k = 0)$. The second row shows the results of fixing the near-optimal configurations at $k = 0$, and the third one contains the values of each term at the cases shown in Fig. 9, from (e) to (h). Even when the system is overpopulated with more cameras than the strictly necessary at some $k$, it can be seen how the optimized solution with mobile cameras improves both resolution and visibility results of the coverage task. The value of this results lies in the fact that the coverage task is successfully executed by means of a minimal set of sensors. Therefore, we can state that the system of offline near-optimal configured mobile cameras is useful for inspection tasks of deformable objects whose prescribed deformation is approximately



(a) $k = 0$ (b) $k = 100$ (c) $k = 200$ (d) $k = 300$



(e) $k = 0$ (f) $k = 100$ (g) $k = 200$ (h) $k = 300$

Figure 9: From (a) to (d), *restricted visibility graphs* at four different deformation instants, resulting from the iterative optimization over a bone shape. The *restricted visibility graphs* resulting from the optimization with a set of $C^* = 6$ cameras, over the same shape and at the same four deformation instants, are shown from (e) to (h). $\vartheta_j^* = 1$ in the thin orange line segments and $\vartheta_j^* = 0$ in the rest, while $r_j^* = r_j^{max}/2$ in the thick orange line segments and $r_j^* = 0$ in the rest.

known.

### *6.4. Online optimization tests*

Now we test the local optimization method. In this case, we consider a turtle shape from the MPEG-7 data set, with $S = 50$ segments, increasingly distorted to simulate deformation. Deformation is modeled by applying exponential distortion and a total counter clockwise rotation of 45 degrees. The cameras are modeled with $\beta = 60^\circ$ and $L_{max} = 2000$ units, the minimum radial limit $d_{min}$ is set approx-

Table 3: Values of each term in $\gamma$, at four deformation instants for comparison of three different setups. The setup of the near-optimal configured mobile cameras corresponds to the cases (e)-(h) in Fig. 9.

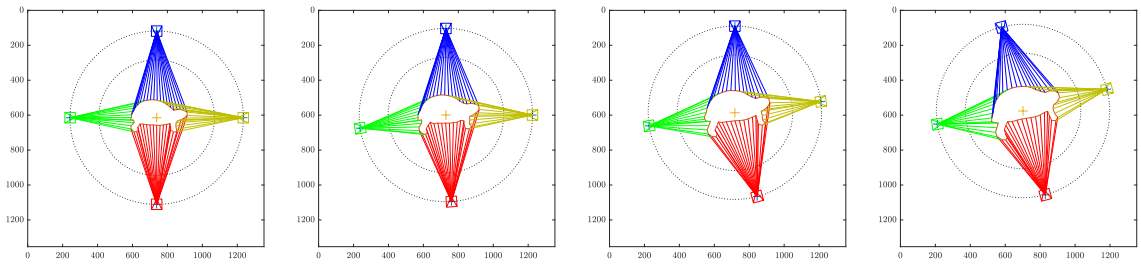
| | | **k = 0** | **k = 100** | **k = 200** | **k = 300** |
|---|---|---|---|---|---|
| **Evenly spaced fixed cameras** ($\psi_i : 0°, 90°, 180°, 270°$) | $\gamma_v$ | 0 | 0 | 8.1818E-2 | 1.9231E-1 |
| | $\gamma_r$ | 4.8683E-1 | 4.8559E-1 | 4.8389E-1 | 4.8221E-1 |
| | $\gamma_\sigma$ | 1.1685E-5 | 1.1685E-5 | 1.1685E-5 | 1.1685E-5 |
| | $\gamma$ | 4.8684E-1 | 4.8560E-1 | 5.6572E-1 | 6.7453E-1 |
| **Near-optimal configured cameras fixed at** $k = 0$ | $\gamma_v$ | 0 | 1.7241E-2 | 1.1818E-1 | 1.9231E-1 |
| | $\gamma_r$ | 4.8485E-1 | 4.8339E-1 | 4.8151E-1 | 4.7997E-1 |
| | $\gamma_\sigma$ | 2.9585E-5 | 2.9585E-5 | 2.9585E-5 | 2.9585E-5 |
| | $\gamma$ | 4.8488E-1 | 5.0066E-1 | 5.9972E-1 | 6.7231E-1 |
| **Near-optimal configured mobile cameras** | $\gamma_v$ | 0 | 0 | 0 | 0 |
| | $\gamma_r$ | 4.8485E-1 | 4.8330E-1 | 4.8130E-1 | 4.7969E-1 |
| | $\gamma_\sigma$ | 2.9585E-5 | 3.8946E-5 | 4.7513E-5 | 5.6068E-5 |
| | $\gamma$ | 4.8488E-1 | 4.8334E-1 | 4.8135E-1 | 4.7975E-1 |

imately equal to the greater side of the contour bounding box, and the maximum limit $d_{max} = 1.5\,d_{min}$. As for the optimization objectives, the resolution term is omitted for simplicity ($F = 1$), and $\vartheta_j^* = 1$ for all the visible contour segments. We execute the iterative optimization at the first deformation instant, that provides $C^* = 4$ cameras and the initial near-optimal configurations. Afterwards, we apply the local optimization sequentially with each camera at each instant, for a maximum allowed relative difference between $\mathbf{c}_i(k)$ and $\mathbf{c}_i(k+1)$ of 0.05. Figure 10 contains the visibility graphs of this test at eight different $k$ ($k = 0, 100, ..., 700$). From (a) to (d) the entire contour is successfully detected, and from (e) to (h) only two segments below the turtle's tail are lost. In comparison with a local optimization, which takes in this case 0.4 seconds on average to solve, the last global optimization with four cameras is 7.5 times slower, taking around 3 seconds. The

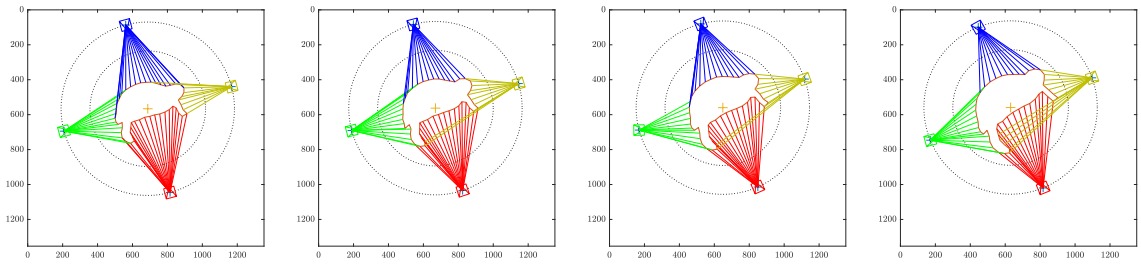
relevance this results show is related to the achievement of the coverage objectives in a much faster manner than the offline method, which is fast enough so that the approach can reconfigure the system online, and also to the fact that deformation of the object over time is unknown.

### *6.5. Global and local optimizations comparison tests*

We perform an additional test in order to compare the offline global and online local optimizations in more detail. A 'C' shape, with ends that progressively approach to each other, is sampled with $S = 100$ segments. The FOV of the cameras is set as in the previous test, as well as the radial limits and the optimization objectives. Then, the same deformation sequence is evaluated with the local optimization technique (see Fig. 11 (a)-(d)), with a threshold of the 5% of allowed deviation between $\mathbf{c}_i(k)$ and $\mathbf{c}_i(k+1)$, and the global optimization one (see Fig. 11 (e)-(h)). Results show that the global optimizations perform slightly better than the local ones, with an improvement of 0.02 in $\gamma_v$ at the last two deformation instants. However, the computational cost is much higher: a global optimization takes 13.96 seconds on average to solve the system, while a local one takes just 1.04 seconds. Thus, we would select the online strategy for those inspection tasks where the deformation of the object over time is unknown. Otherwise, we would opt for the offline method, which is not as fast as the online one but provides better quality results. It is worth mentioning that the order of the cameras may change from $k$ to $(k+1)$ in the global optimization (e.g., from Fig. 11(f) to Fig. 11(h) the blue camera changes twice its order into the cameras sequence). This does not represent a problem, due to the fact that the near-optimal configurations are assigned to the cameras of the real-world system so that the total traveled distance is minimized, while the angular order of them is preserved.



(a) $k = 0$ , $\gamma_v = 0$ (b) $k = 100$ , $\gamma_v = 0$ (c) $k = 200$ , $\gamma_v = 0$ (d) $k = 300$ , $\gamma_v = 0$



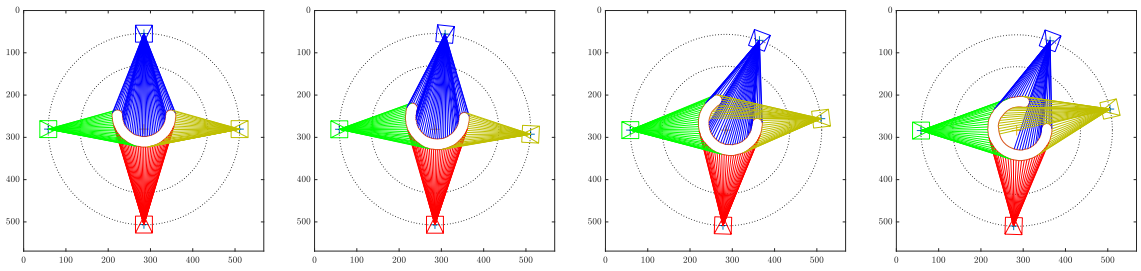
(e) $k = 400$ , $\gamma_v = 0.04$ (f) $k = 500$ , $\gamma_v = 0.04$ (g) $k = 600$ , $\gamma_v = 0.04$ (h) $k = 700$ , $\gamma_v = 0.04$

Figure 10: *Restricted visibility graphs* at different deformation instants, with the corresponding values of $\gamma_v$. After the iterative optimization at $k = 0$, local optimizations are performed with a set of $C^* = 4$ cameras. $\vartheta_j^* = 1$ and $r_j^* = 0$ for all the visible contour segments.

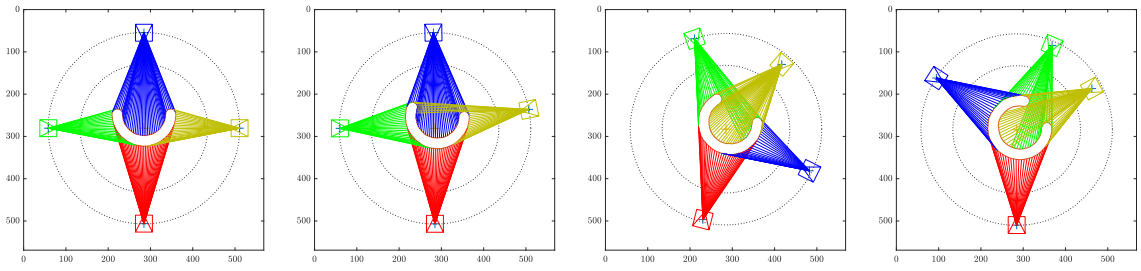
A video that illustrates the main aspects of the methods in different simulation scenarios is included as additional multimedia material.

### *6.6. Real experiment*

Finally, we test the global optimization in a real setup. Figure 12 shows an experimental setup where the goal is to perform full visibility coverage with three cameras, of a sole and last assembly task. In this illustrative example, the evaluation of such setup may be performed visually, but numerical analysis of the result cannot be provided accurately due to the complexity of the shape. Therefore, we propose an additional experimental setup to facilitate the quantitative validation



(a) $k = 0$ , $\gamma_v = 0$ (b) $k = 100$ , $\gamma_v = 0$ (c) $k = 300$ , $\gamma_v = 0.02$ (d) $k = 500$ , $\gamma_v = 0.1$



(e) $k = 0$ , $\gamma_v = 0$ (f) $k = 100$ , $\gamma_v = 0$ (g) $k = 300$ , $\gamma_v = 0$ (h) $k = 500$ , $\gamma_v = 0.08$

Figure 11: *Restricted visibility graphs* at different deformation instants of a 'C' shape, with the corresponding values of $\gamma_v$. $\vartheta_j^* = 1$ and $r_j^* = 0$ for all the visible contour segments. After the global optimizations at $k = 0$ in (a) and (e), with $C = 4$ cameras, the results of applying local ((b)-(d)) and global ((f)-(h)) optimizations are shown for comparison. We can see how global optimizations obtain better configurations for minimizing $\gamma_v$, at the expense of a higher computational cost.

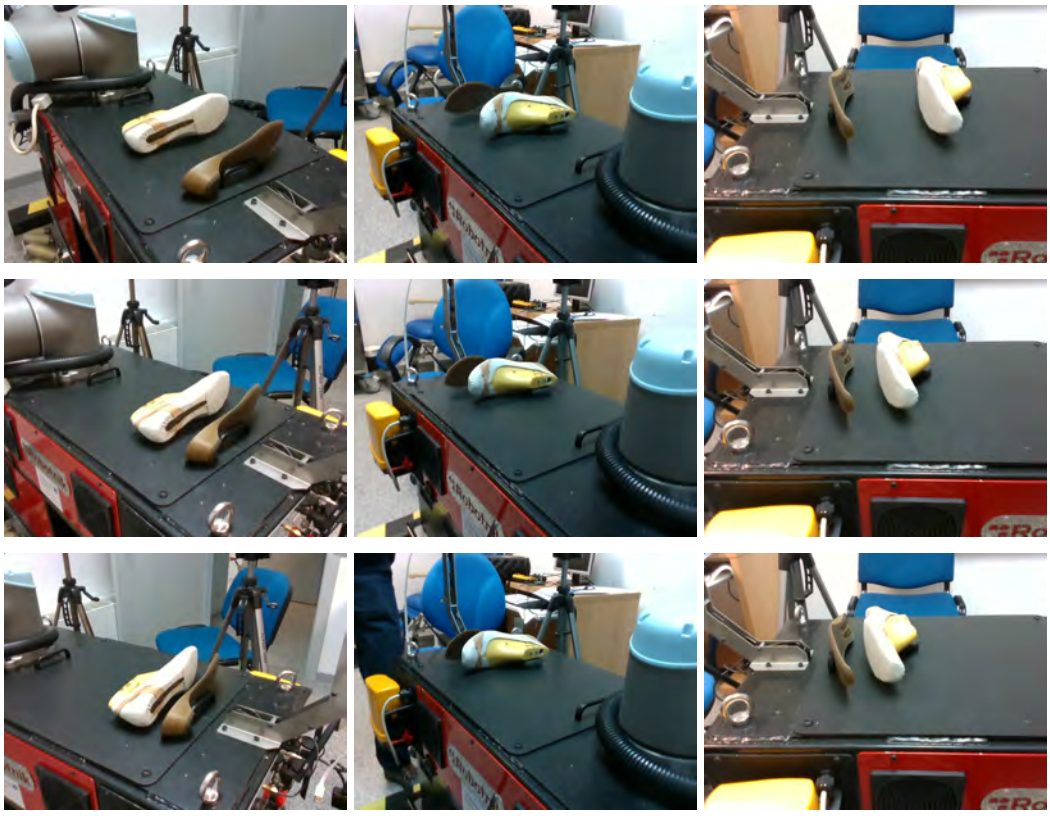


Figure 12: Views of three cameras (left to right) in a full visibility coverage experimental setup, where we represent three steps (top to bottom) of a sole-last assembly task.

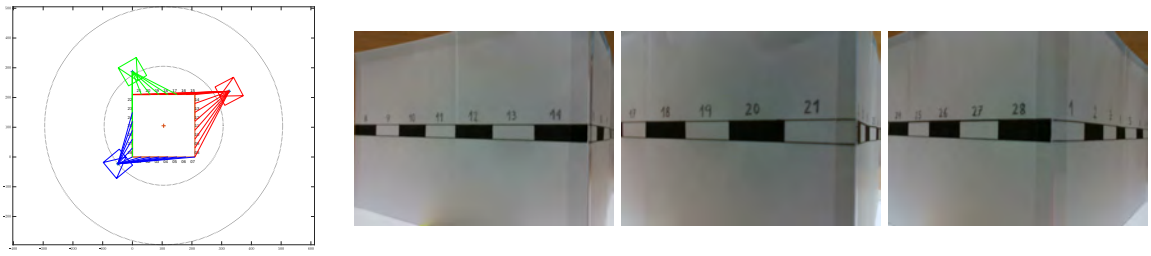

Figure 13: View of the real experiment setup. The Intel RealSense D435 module is on a tripod that is positioned and oriented to get the different optimal configurations of the cameras.

of the coverage system. A $21 \times 21$ [cm] paper structure is sequentially folded to simulate a three-steps deformation process (Figure 13). This structure is horizontally divided into 28 segments of 3 [cm] in length, which are imprinted and numbered in the faces of the structure. Due to the fact that the object itself is segmented, we can perform the coverage task with a standard color camera whose optical center's position is known. With the segments directly imprinted in the object, we also facilitate the interpretation of the results. In this case, we consider an Intel RealSense D435 module installed on a tripod, with a color camera of $69 \pm 1$ degrees horizontal FOV ($\beta = 69°$). We assume that there are no range constraints in this setup ($L_{max} \leftarrow \infty$). The optimization objective is set as $r_j^* = r_j^{max}/4$ and $\vartheta_j^* = 0$ for the segments $\{1,2,3,4,12,13,14,20,21,25,26,27,28\}$ and $r_j^* = 0$ and $\vartheta_j^* = 1$ for the rest, with $F = 0.5$. We execute algorithm 1 to obtain the minimum number of cameras $C^*$ of the deformation process, and we obtain $C^* = 3$. Then, the system is optimized with $C = 3$ at every deformation instant, and we get the optimal configuration of the cameras. Figure 14 shows the simulation results and the images taken by the Intel RealSense color camera from the different optimal configurations at each deformation instant. We can see that all segments are successfully detected, which means the visibility coverage is achieved. Figure 15 shows the results concerning the angular resolution of the selected contour segments with $r_j^* \neq 0$ (the resolution of the rest of the segments is not measured). We measure the angular resolution directly from the images, by mapping the horizontal resolution of the segments in pixels to the corresponding section of the FOV (in this case, the horizontal resolution of the images is 640 pixels, which corresponds to the 69 degrees FOV of the color camera). We see that the target resolution is achieved and improved in all segments, which means that the resolution cover-

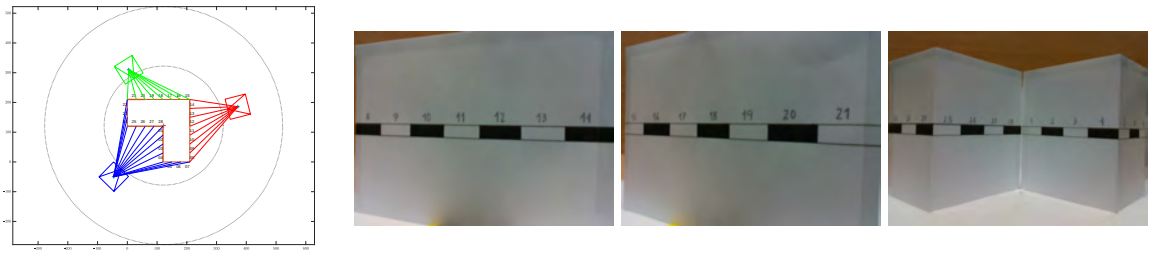
age is achieved. These results confirm the validity and utility of the methods we propose for the application in real perception tasks.

## 7. Conclusion

In this paper, we have presented a multi-camera system for coverage of deformable contour shapes, according to a prescribed objective in terms of visibility and resolution. Firstly, we develop a technique that allows to obtain the theoretical maximum visibility and resolution of a 2D shape at each deformation instant, with an infinite set of omnidirectional and unlimited FOV range cameras. Afterwards, we use this information to define feasible coverage objectives along deformation, and then we apply an iterative optimization technique that provides the minimum necessary number of cameras for the coverage task and their near-optimal configurations. Therefore, by obtaining a minimal number of cameras we are able to reduce the computational cost of the data processing, and also the synchronization complexity. Finally, we propose a method based on local optimizations that outputs near optimal configurations much faster than the global optimization. This method is also applicable online in the case where deformation evolves in an unknown manner. We have tested our approaches by means of different experiments, and the results show they are effective and useful for performing coverage tasks in a wide range of cases: for different objects, deformation processes, coverage objectives, safety conditions and spatial constraints. We want to highlight that our optimization algorithms guarantee the achievement of the target coverage. However, depending on the specific sensors that are considered for the real perception task, the final error of the recovered object shape may vary. In this respect, it is important to properly define the coverage task to obtain the required information



(a) Configuration of the cameras at the first deformation instant



(b) Configuration of the cameras at the second deformation instant

(c) Configuration of the cameras at the third deformation instant

Figure 14: Real experiment results. At the left, the first images correspond to the *restricted visibility graphs* that result after executing the offline optimization for $C = 3$. The next three images at the right side correspond to the views of the Intel RealSense color camera in the red, green and blue configurations respectively.

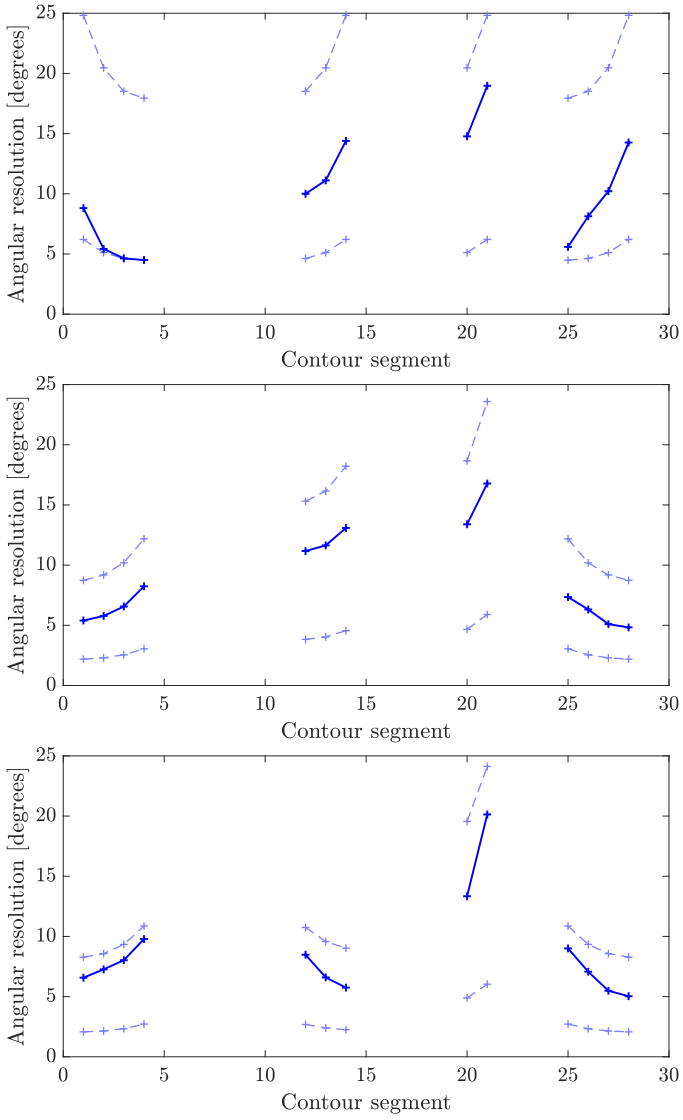



Figure 15: From top to bottom, we show the angular resolution results in the first, second and third deformation instants of the real coverage tasks. The top dashed lines correspond to the maximum angular resolution of the segments, the bottom dashed line to the target resolution and the continuous line to the resolution measured from the images of the Intel RealSense cameras. We can see that the target resolution is achieved and improved in all segments, even getting some results which are close to the maximum values.

of the object. Future research directions include extending our method to perform coverage in 3D, adapting the target coverage online according to the sensors data and allowing the cameras to move in and out of the formation along the deformation process, depending on the coverage requirements.

## References


[1] J. Sanchez, J. A. Corrales, B. C. Bouzgarrou, Y. Mezouar, Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey, The International Journal of Robotics Research 37 (2018) 688–716.

[2] Y. Tang, L. Li, W. Feng, F. Liu, X. Zou, M. Chen, Binocular vision measurement and its application in full-field convex deformation of concrete-filled steel tubular columns, Measurement 130 (2018) 372–383.

[3] J. Das, N. Sarkar, Autonomous shape control of a deformable object by multiple manipulators, Journal of Intelligent and Robotic Systems: Theory and Applications 62 (2011) 3–27.

[4] F. Ruggiero, A. Petit, D. Serra, A. C. Satici, J. Cacace, A. Donaire, F. Ficuciello, L. R. Buonocore, G. A. Fontanelli, V. Lippiello, L. Villani, B. Siciliano, Nonprehensile manipulation of deformable objects: achievements and perspectives from the RoDyMan project, IEEE Robotics and Automation Magazine 25 (2018) 83–92.

[5] R. Herguedas, G. Lopez-Nicolas, R. Aragues, C. Sagues, Survey on multirobot manipulation of deformable objects, in: IEEE Conference on Emerging Technologies and Factory Automation (ETFA), 2019, pp. 977–984.

[6] M. Chen, Y. Tang, X. Zou, K. Huang, L. Li, Y. He, High-accuracy multi-camera reconstruction enhanced by adaptive point cloud correction algorithm, Optics and Lasers in Engineering 122 (2019) 170–183.

[7] Y. Tang, L. Li, C. Wang, M. Chen, W. Feng, X. Zou, K. Huang, Real-time detection of surface deformation and strain in recycled aggregate concrete-filled steel tubular columns via four-ocular vision, Robotics and Computer-Integrated Manufacturing 59 (2019) 36–46.

[8] D. Navarro-Alarcon, Y.-H. Liu, Fourier-based shape servoing: a new feedback method to actively deform soft objects into desired 2-D image contours, IEEE Transactions on Robotics 34 (2018) 272–279.

[9] F. Alambeigi, Z. Wang, R. Hegeman, Y.-H. Liu, M. Armand, Autonomous data-driven manipulation of unknown anisotropic deformable tissues using unmodelled continuum manipulators, IEEE Robotics and Automation Letters 4 (2019) 254–261.

[10] R. Herguedas, G. Lopez-Nicolas, C. Sagues, Experimental multi-camera setup for perception of dynamic objects, in: Workshop ROMADO, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 11874–11878.

[11] H. Emadi, T. Gao, S. Bhattacharya, Visibility-Based Target-Tracking Game: Bounds and Tracking Strategies, IEEE Robotics and Automation Letters 2 (2017) 1917–1924.

[12] C. Robin, S. Lacroix, Multi-robot target detection and tracking: taxonomy and survey, Autonomous Robots 40 (2016) 729–760.

[13] X. Zhang, X. Chen, J. L. Alarcon-Herrera, Y. Fang, 3-d model-based multi-camera deployment: A recursive convex optimization approach, IEEE/ASME Transactions on Mechatronics 20 (2015) 3157–3169.

[14] T. Zhang, J. Liu, S. Liu, C. Tang, P. Jin, A 3d reconstruction method for pipeline inspection based on multi-vision, Measurement 98 (2017) 35–48.

[15] R. Strauss, F. Isvoranu, G. Elber, Geometric multi-covering, Computers and Graphics 38 (2014) 222–229.

[16] P. Artaso, G. Lopez-Nicolas, Volume estimation of merchandise using multiple range cameras, Measurement 89 (2016) 223–238.

[17] Z. Deng, A. Li, X. Liu, Equivalent virtual cameras to estimate a six-degree-of-freedom pose in restricted-space scenarios, Measurement 184 (2021) 109903.

[18] P. Guler, K. Pauwels, A. Pieropan, H. Kjellström, D. Kragic, Estimating the deformability of elastic materials using optical flow and position-based dynamics, in: IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), 2015, pp. 965–971.

[19] W. Jing, J. Polden, P. Y. Tao, C. F. Goh, W. Lin, K. Shimada, Model-based coverage motion planning for industrial 3D shape inspection applications, in: IEEE Conference on Automation Science and Engineering (CASE), 2017, pp. 1293–1300.

[20] J. Delmerico, S. Isler, R. Sabzevari, D. Scaramuzza, A comparison of volumetric information gain metrics for active 3d object reconstruction, Autonomous Robots 42 (2018) 197–208.

[21] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2000.

[22] F. M. De Rainville, J. P. Mercier, C. Gagne, P. Giguere, D. Laurendeau, Multisensor placement in 3D environments via visibility estimation and derivative-free optimization, in: IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 3327–3334.

[23] G. Lopez-Nicolas, M. Aranda, Y. Mezouar, Adaptive Multirobot Formation Planning to Enclose and Track a Target With Motion and Visibility Constraints, IEEE Transactions on Robotics 36 (2020) 142–156.

[24] M. Schwager, B. J. Julian, M. Angermann, D. Rus, Eyes in the sky: Decentralized control for the deployment of robotic camera networks, Proceedings of the IEEE 99 (2011) 1541–1561.

[25] C. Piciarelli, L. Esterle, A. Khan, B. Rinner, G. L. Foresti, Dynamic reconfiguration in camera networks: A short survey, IEEE Transactions on Circuits and Systems for Video Technology 26 (2016) 965–977.

[26] R. Bajcsy, Y. Aloimonos, J. K. Tsotsos, Revisiting active perception, Autonomous Robots 42 (2018) 177–196.

[27] D. S. Schacter, M. Donnici, E. Nuger, M. Mackay, B. Benhabib, A multi-camera active-vision system for deformable-object-motion capture, Journal of Intelligent and Robotic Systems: Theory and Applications 75 (2014) 413–441.

[28] E. Nuger, B. Benhabib, Multi-camera active-vision for markerless shape

recovery of unknown deforming objects, Journal of Intelligent and Robotic Systems: Theory and Applications 92 (2018) 223–264.

[29] R. Herguedas, G. Lopez-Nicolas, C. Sagues, Multi-camera coverage of deformable contour shapes, in: IEEE International Conference on Automation Science and Engineering (CASE), 2019, pp. 1597–1602.

[30] M. Aranda, J. A. Corrales Ramon, Y. Mezouar, A. Bartoli, E. Ozgur, Monocular visual shape tracking and servoing for isometrically deforming objects, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 7542–7549.

[31] C. Audet, J. E. Dennis, Analysis of generalized pattern searches, SIAM Journal on Optimization 13 (2002) 889–903.

[32] L. Latecki, R. Lakamper, T. Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour, in: Proceedings IEEE CVPR, 2000, pp. 424–429.