Multi-camera coverage of deformable contour shapes

Rafael Herguedas, Gonzalo López-Nicolás and Carlos Sagüés

Abstract— **Perception of deformation is a key problem when dealing with autonomous manipulation of deformable objects. Particularly, this work is motivated by tasks where the manipulated object follows a prescribed known deformation with the goal of performing a desired coverage of the objects contour along its deformation. The main contribution is a simple yet effective novel perception system in which a team of robots equipped with limited field-of-view cameras covers the object's contour according to a prescribed visibility objective. In order to define a feasible visibility objective, we propose a new method for obtaining the maximum achievable visibility of a contour from a circumference around its centroid. Then, we define a constrained optimization problem and we solve it iteratively to compute the minimum number of cameras and their near-optimal positions around the object that guarantee the visibility objective, over the entire deformation process.**

I. Introduction

There is an increasing interest on controlling deformation of solid objects nowadays, specially for domestic and industrial purposes [1], [2], [3]. A requirement in this kind of application is that perception of the object deformation must be continuously feed back to the system, which may be challenging due to the time-varying and highly-dynamic behavior of deformable objects. We envision an application in the context of industrial manipulation of deformable objects, where the deformation guidelines are provided by the particular task and known in advance. The perception system must be able to check if the deformation is performed as specified in the guidelines and evaluate the quality of the process, adapting the perception to the evolution of deformation with an appropriate number of sensors placed at the most convenient locations with respect to the object. Here, we bring visibility into focus as the main tool for defining how the perception task must be performed. Visibility problems have been addressed from many different perspectives, but in general they involve detecting some specific zones of the environment according to a set of requirements, in terms of accuracy and sensing resources.

A main classification of the approaches that deal with visibility refers to the static or dynamic deployment of the points of view and the time-invariant or time-dependent nature of the perceived environment. Static deployment is specially intended for applications in which the perceived environment, and therefore its visibility properties, does not change substantially in time. Covering of 3D static surfaces with a minimal set of observers is performed in [4] for mold design, in [5] for surveillance and inspection tasks, and in [6] for automated manufacturing. Many of these works are based on selecting those points of view that achieve the visibility objectives from a predefined set of candidate view directions or viewpoints, which can be non-optimal.

Sensing with a single agent may require the system to be dynamically deployed in 3D covering or shape reconstruction tasks of static objects [7], [8], [9], and also in cooperative coverage and target enclosing scenarios. In cooperative dynamic coverage one can find decentralized approaches, where a generally large group of robots is requested to collectively monitor a 2D static environment [10], or centralized approaches for smaller teams of robots that perform monitoring of 3D static environments [11], [12]. Concerning target enclosing, centralized [13] and decentralized [14] approaches have been developed for multi-robot teams to perform, either moving or static, targets enclosing. In particular, these strategies pertain to the formation control field, and they do not consider the minimization of the required sensors in the case of a time-varying scenario.

Works in the active perception field are usually dynamic in the sense that in order to get more information about the environment, whose properties may change over time, the sensors must reconfigure their position and/or orientation depending on the requirements of the perception task. The multi-camera centralized networks in [15] and [16] are optimized in position and orientation so that they are able to recover the shape of a moving and deformable target object, but they do not consider robust perception aspects such as occlusion avoidance.

Here, we present a method for performing a desired coverage of a deforming 2D closed contour shape with a minimal set of cameras. The main contributions of this work are a new technique for obtaining the *maximum achievable visibility* of a sampled contour from a prescribed distance, the continuous search space formulation of the visibility optimization problem with safe and robust perception criteria, and the cameras set minimization over a complete deformation process. To the best of our knowledge, this is the first time a multi-camera perception system for deformable objects is designed using this methodology.

II. Problem Statement

A. Framework

Let us consider a certain object of interest ('object' from now on for brevity) in 2D which undergoes deformation over time. With the purpose of sampling the object's contour at

The authors are with Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Spain. {rherguedas, gonlopez, csagues} @unizar.es

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".

N_d time instants t_k ($k = 1, ..., N_d$), a set of N_{ck} holonomic mobile robots equipped with onboard cameras (i.e. the number of robots N_{ck} is also the number of cameras) is placed around the object's contour centroid $\mathbf{g}_k = (g_{Xk}, g_{Yk})$ at a safety distance d_s from it. Each robot/camera $c_{ik} \in \mathbf{C}_k$ ($i = 1, ..., N_{ck}$), where $\mathbf{C}_k$ is the total set of cameras, is represented by the tuple (ψ_{ik}, ϕ_{ik}). ψ_{ik} is the angle around $\mathbf{g}_k$ in which the camera's center is located and ϕ_{ik} is the orientation of the camera (see Fig. 1). Cameras are modeled with restricted Field of View (FOV), with range and angular constraints.

The following assumptions are made throughout the study:

Assumption *1:* **Object integrity.** *The object's contour continuity is preserved along the whole deformation process, i.e. the process does not divide the object into different parts.*

Assumption *2:* **Known information.** *The following information is given or can be obtained:*

1) *Cameras are calibrated, and they acquire point clouds $P_{ik} \subset \mathbb{R}^2$ of the environment. Object detection and segmentation are given from the point cloud $\mathbf{P}_{\mathbf{T}k}$, which is the fusion of all cameras data.*
2) *Positions of the robots with respect to a global reference $c_{ik} = (\psi_{ik}, \phi_{ik})$ are known at any time.*
3) *The prescribed 2D reference contour of the object as well as $\mathbf{g}_k$ are known in advance at each t_k deformation instant.*

Assumption *3:* **Small deformation.** *During the deformation process, it is assumed that the object's deformation (i.e. contour variations) between consecutive sampled instants is small, and also the difference between the object's deformation with respect to the reference object's deformation (i.e. reference contour variations).*

Assumption *4:* **Slow deformation.** *Deformations are slow enough so that the robots with the cameras have time enough to react and follow the sequence of configuration positions around the object.*

These assumptions allow to create a set of reference segments $\mathbf{S_{ref}}$ by sampling the expected object contour into N_s segments of length L_s, which will be utilized for obtaining the distance of the detected points in $\mathbf{P}_{\mathbf{T}k}$ to it and computing the perception error. Polygonal contour reference segments $s_{jk} = \{v_{jk}, v_{(j+1)k}\}$, where $j = 1, ..., N_s - 1$ and for $j = N_s$, $s_{N_s k} = \{v_{N_s k}, v_{1k}\}$, are defined by the position of their two vertexes v_{jk} and $v_{(j+1)k}$. From now on, the k subscript will be omitted in general for clarity purposes.

B. Optimization objectives of the visibility task

Given the previous setup and assumptions, the main objective is to obtain the near-optimal positions and orientations of the robots in the circumference of radius d_s centered at $\mathbf{g}$ such that the *visibility cost* of the object is zero or near zero at each optimization instant.

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

$$\gamma_v = \sum_{j=1}^{N_s} \gamma_{v_j} \ , \ \gamma_{v_j} = \begin{cases} \vartheta_j^t - \vartheta_j & \text{if} \quad \vartheta_j^t \geq \vartheta_j \\ 0 & \text{otherwise} \end{cases} . \quad (1)$$

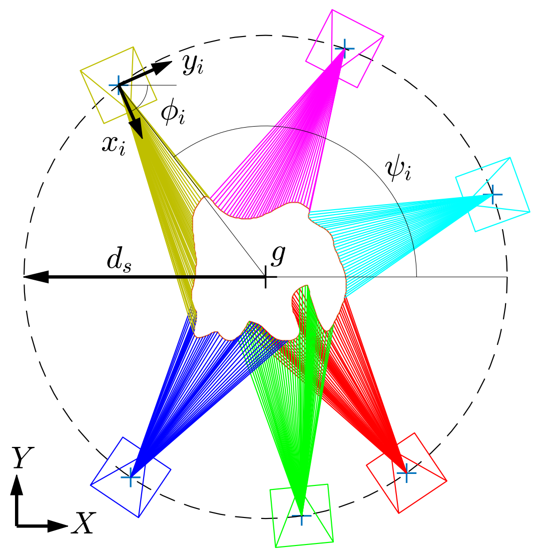


Fig. 1. Overview of the framework with the main geometric parameters. The target object is at the center of the circumference, surrounded by six different cameras with restricted FOV. Projection rays linking each camera center to the vertexes of the object's contour are drawn.

ϑ is a set in which each term ϑ_j provides the number of cameras detecting each segment s_j for a specific time instant, and ϑ^t is the target visibility set in which each term ϑ_j^t indicates the number of cameras that must detect each reference segment s_j according to the task requirements.

Our definition of the visibility objective (or target visibility) allows to configure which contour segments must be detected at each t_k. For example, in case the whole contour must be perceived by at least one camera $\forall t_k$, then $\vartheta_j^t = 1, \ \forall j$. If a higher quality or redundancy of sensing is required in some areas of interest, we can require additional sensing by increasing the value of ϑ_j^t for a certain subset of segments. Additionally to the main objective, we are also interested in determining the minimum value of N_c that accomplishes the desired visibility of the contour.

III. Maximum Achievable Visibility

Before setting a feasible visibility objective, we need information about which segments can actually be detected by cameras placed at a distance d_s from $\mathbf{g}$.

Definition *2:* **Maximum achievable visibility.** *The maximum achievable visibility ϑ_{max} of the object's contour is a set in which each term ϑ_{max_j} indicates whether the segment s_j can be detected or not by some specifically oriented camera placed at a distance d_s from* $\mathbf{g}$.

Given the prescribed evolution of each contour shape segment $s_j \in \mathbf{S_{ref}}$ along deformation, if no external occluders are present, ϑ_{max} is *a priori* known when the shape is convex (full visibility), and *a priori* unknown when the shape is non-convex. We present next the procedure to check whether a segment s_j is visible or not from a specific camera position.

Definition *3:* **Bi-partite visibility graph.** *We define the bi-partite visibility graph [17]*

$$G_v = (\mathbf{C}, \mathbf{S_{ref}}, \mathbf{E}) \ , \quad (2)$$

$$\mathbf{C} = \{c_1, ..., c_i, ..., N_c\},\ \mathbf{S_{ref}} = \{s_1, ..., s_j, ..., N_s\}\ , \quad (3)$$

as the graph where a camera c_i and each vertex of $s_j(v_j, v_{(j+1)})$ are linked with edges e_{ij} and $e_{i(j+1)} \in \mathbf{E}$ only if the edges, represented in the space as the lines connecting c_i to the vertexes v_j and $v_{(j+1)}$, do not intersect any obstacle's segment, i.e. if the vertexes are visible from c_i.

In the following, we propose a method to solve the problem of obtaining the *maximum achievable visibility*.

Proposition *1: The bi-partite visibility graph of the obstacle-free framework where an infinite number of omnidirectional cameras ($N_c = \infty$) completely covers the circumference of radius d_s centered at $\mathbf{g}$ provides the exact number of object's segments which can be detected from that distance. Each term of ϑ_{max} takes then the values of ∞ or 0 depending on whether a segment is visible or not.*

Proof: According to Definition 3, two edges e_{ij} and $e_{i(j+1)}$ linking $c_i \in \mathbf{C}$ and the vertexes v_j and $v_{(j+1)}$ of $s_j \in \mathbf{S}$ exist only if the vertexes are visible from c_i. Thus, if a couple of edges exist such that $e_{ij} = \{c_i, v_j\}$ and $e_{i(j+1)} = \{c_i, v_{(j+1)}\}$, then segment s_j is visible. ϑ_{max_j} can be computed as

$$\vartheta_{max_j} = \sum_{i=1}^{N_c} \{\exists e_{ij} \wedge \exists e_{i(j+1)}\}\ , \quad (4)$$

where the boolean outputs $true$ and $false$ are interpreted as the integer values 1 and 0. Given that an omnidirectional camera is able to cover any possible direction, that infinite possible camera positions around the object are considered, and that vertex positions are fixed, every possible edge will be obtained. Therefore, every segment that can be detected will be detected with this procedure. ∎

Remark *1: Notice that a finite set of cameras must be considered in practice to compute ϑ_{max}. However, this limitation is useful in order to obtain a measure of how difficult it is to detect a certain segment in comparison with the rest: the closer the value of ϑ_{max_j} to the cardinality of the cameras set, the more possible locations can detect s_j.*

IV. Minimization of the number of cameras

A. Minimization problem

Once we have the contour visibility upper bound ϑ_{max} and therefore the knowledge about which segments can be detected, we are able to define the *target visibility set* ϑ^t of the coverage task. Then, the goal is obtaining the minimal set of N_c^* cameras and their locations ψ_i^* and ϕ_i^* such that γ_v is also minimal (ideally zero), for a time instant t_k.

We define the following optimization problem:

$$\begin{array}{lll} \text{Given} & \mathbf{C}(N_c), \mathbf{S_{ref}} & \\ \underset{\mathbf{C}(N_c)}{\text{minimize}} & \gamma = \gamma_v + \gamma_\sigma + \gamma_\beta & , \\ \text{subject to} & R & \end{array} \quad (5)$$

where γ is the cost function of the problem and R is the set of camera and geometric restrictions. The additional terms γ_σ and γ_β will be explained in detail over the next lines, and incorporate additional constraints that improve the performance of the system.

The proposed pipeline is as follows. We set $N_c = 2$ (the minimum required number of cameras for completely detecting a closed contour, without the help of external elements) and check γ_v after solving the optimization problem. If $\gamma_v > 0$, we solve iteratively the problem for $N_c = N_c + 1$ until $\gamma_v = 0$ or the maximum allowed number of robots $N_{c_{max}}$ is reached. We define $N_{c_{max}}$ as the maximun number of robots that can be physically placed without collisions in the circumference around the object. Thus, it depends on d_s and the size of the robots. The optimization method we utilize to solve this problem is the *pattern search method* [18], which is a derivative-free technique compatible with our function γ that always provides a minimum of the cost function, though it may be a local one.

Contour segments detection is performed again building the *bi-partite visibility graph*. However, we have to include the cameras range and angular constraints into the optimization problem. These constraints are introduced by including additional conditions when building the *bi-partite visibility graph*, as explained next.

Definition *4:* **Restricted visibility graph.** *We define the restricted visibility graph as the bi-partite visibility graph in which each edge lies within the FOV limits.*

Thus, if the edge linking a vertex of the contour to a camera c_i has a length which is larger than L_{max}, the maximum range of the FOV, or forms an angle greater than $\frac{\beta_{max}}{2}$ with the local axis x_i, where β_{max} is the FOV's maximum angle, we determine that the vertex is not visible from c_i, and therefore the edge is not valid and excluded from the graph.

In some cases, two robots are required to be close to each other in order to detect certain segments of the contour, and collisions may happen if its physical limits are surpassed. The *potential field* approach is applied to penalize this as:

$$\gamma_\sigma = \frac{1}{(2d/d_{min} - 1)^w} \sum_{j=1}^{N_s} \vartheta_j^t\ , \quad (6)$$

where d is the minimal distance between whichever two neighboring robots of the system and d_{min} is the minimal prescribed distance they should maintain to operate with safety guarantees. This exponential function allows us to introduce a positive contribution that takes values near zero in case the robot is far enough from the rest and rises quickly to the maximum value of γ_v when two of them approach to d_{min}. The $w > 0$ exponent regulates the influence distance of γ_σ, and its value is tuned depending on the safety requirements of the task. Thus, if large camera separations are preferred, small values of w must be selected. In addition to this term, the following set of linear constraints is applied to the optimization:

$$|\psi_{i_1} - \psi_{i_2}| \geq \psi_{min},\ \forall i_1, i_2 = 1, ..., N_c,\ i_1 \neq i_2\ , \quad (7)$$

where

$$\psi_{min} = 4\ \arcsin\left(d_{min}\ /\ (4 d_s)\right)\ . \quad (8)$$

This prevents two neighboring robots to be closer than d_{min} under any circumstances, and limits the maximum number of agents the system can admit to:

$$N_{c_{max}} = \text{truncation}\left(2\pi \;/\; \psi_{min}\right) \ . \tag{9}$$

Robust perception, in terms of a stable and persistent detection of the contour in the presence of deformation and small unexpected movements of the robots or the object, is a desired requirement of the system. Due to the fact that surfaces that will be occluded first are the ones closer to be aligned with the edges of the graph, we consider that the greater the angle between two consecutive edges of the *restricted visibility graph*, the more robust the perception is. We incorporate this notion to the cost function as:

$$\gamma_\beta = 1 - \frac{1}{N_c \cdot \beta_{max}} \sum_{i=1}^{N_c} \beta_{min_i} \ , \tag{10}$$

where β_{min_i} is obtained by computing the minimum angle between each pair of consecutive edges connecting the camera c_i to the vertexes of the segments it detects in the *restricted visibility graph*. The maximum contribution of γ_β is limited by the contribution of not detecting a single contour segment, due to the fact that detecting a segment is considered always more important than detecting more robustly or with a higher quality. Therefore, γ_β can be considered a fine tuning term.

Minimizing γ_β implies that the minimum angles, and therefore the shortest projected lengths of the segments, will be increased, which will improve the quality of perception, and it will also prevent to a higher extent the possible occlusions that could appear during the deformation process.

B. Minimization algorithm for the deformation process

The optimization problem described in the previous section provides a minimum number of cameras and their configurations at the t_k deformation instant. With the purpose of obtaining N_c^*, the minimum required number of cameras for the entire deformation process, the problem must be solved for a set of t_k such that Assumption 3 is still valid from each instant to the following one. Once the problem is solved and N_{ck}^* is obtained at each selected t_k, we set N_c^* to be equal to the maximum N_{ck}^*. This is a conservative decision to get a trade-off between resources optimization and robustness of the system against unexpected behaviors. The final positions ψ_r and orientations ϕ_r of the cameras are computed by optimizing at each selected t_k instant with the set of N_c^* cameras. These desired configurations are achieved by the robots following a motion strategy that consists in maintaining always the safety distance d_s to $\mathbf{g}_k$, so that collisions with the object are avoided. It is worth mentioning that the initial input values for the optimization with N_c^* cameras at the t_k deformation instant are the positions and orientations output by the optimization at the previous one.

Algorithm 1 shows the main steps of the cameras set minimization over the entire deformation process. R_k, as mentioned before, integrates the cameras FOV constraints and the robots physical dimensions.

Algorithm 1 Estimate the minimum number of cameras.

$N_c^* = 2$
Set k to an initial value ($k = 0$)
while $k \leq N_d$ **do**
  Initialize $\gamma_k > 0$
  $N_{ck}^* = 1$
  while ($\gamma_k > 0$) and ($N_{ck}^* < N_{c_{max}}$) **do**
    $N_{ck}^* = N_{ck}^* + 1$
    Minimize $\gamma_k(\psi_{ik}, \phi_{ik})$ s.t. R_k
  end while
  if $N_{ck}^* > N_c^*$ **then**
    $N_c^* = N_{ck}^*$
  end if
  Increase k
end while
return N_c^*

So far, our method is performed mainly offline, but we are currently studying a new technique that will allow the cameras to reposition themselves online in case the deformation substantially differs from the planned one. This will prevent visibility gaps (i.e. discontinuities larger than a threshold in $\mathbf{P_T}$) and will extend, therefore, the validity of the method.

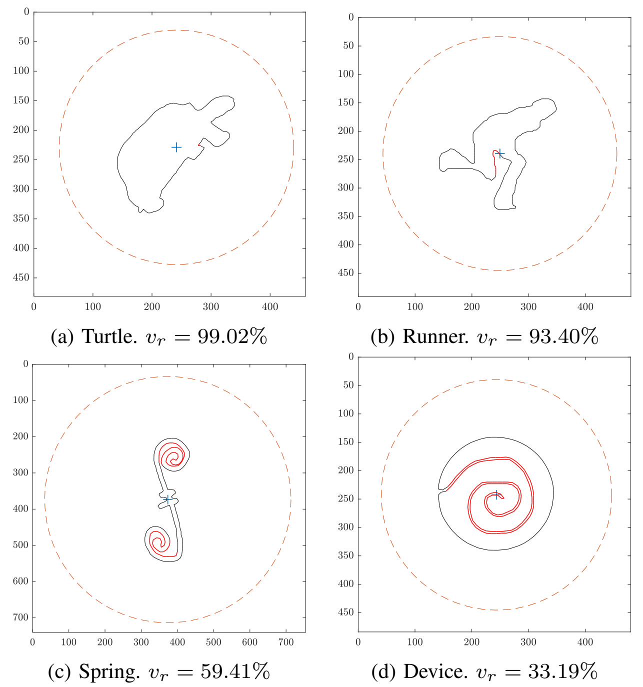
V. Simulation Results

In order to validate our approach, we have performed a series of simulations focusing on each of its main aspects. The simulations are implemented in Matlab over different shapes extracted from the well-known data set MPEG-7. MPEG-7 data set includes 1400 silhouettes of 2D objects grouped in 70 classes, 20 objects each [19], and it is very popular to test shape descriptors among other applications.

We first evaluate the performance of our method for obtaining the *maximum achievable visibility* of 2D sampled shapes. The contour of the shape in each image is extracted and sampled in 200 points. Then, we can represent each contour with $N_s = 200$ segments. After this, we place a set of N_c omnidirectional cameras around the contour centroid at constant intervals, at a distance d_s equal to the maximum side of the contour's bounding box (for uniformity purposes). Finally, in order to accurately estimate the *maximum achievable visibility* we compute an initial ϑ_{max} with the set of N_c cameras, we increase N_c and compute ϑ_{max} again and we repeat this process until no further segments are detected. Figure 2 shows several complex shapes in which this method has been applied with their visibility ratio v_r, defined as

$$v_r(\%) = 100 \cdot \mathit{visible\,perimeter} \,/\, \mathit{total\,perimeter} \ . \tag{11}$$

We have computed the *maximum achievable visibility* of the 1400 images from the MPEG-7 data set. As expected, lower visibility ratios have been obtained for longer perimeters, which are usually more complex than the shorter ones. We have found that the minimal visibility ratio is equal to 33.19% (see plot (d) in Fig. 2), that the average visibility ratio of the data set is 96.90% and we have detected also that the 66.07% of the contours are fully visible.



(a) Turtle. $v_r = 99.02\%$ (b) Runner. $v_r = 93.40\%$

(c) Spring. $v_r = 59.41\%$ (d) Device. $v_r = 33.19\%$

Fig. 2. Example shapes with their v_r values, where segments that are visible from the safety perimeter (dashed line circumference), for at least one camera, are represented in thin black lines, whereas occludes segments appear in thick red lines (best seen in color).

Regarding the optimization procedure, the convenience and effectiveness of each term in γ is evaluated. Over a simple undeformed squared shape, which allows us to illustrate the differences we want to highlight, we have performed the iterative optimization method four times with $\vartheta_j^t = 1, \ \forall j$ (i.e. each segment must be detected by at least one camera): the first test with $\gamma = \gamma_v$, the second one in which γ_σ is added to γ, a third one with $\gamma = \gamma_v + \gamma_\beta$ and a last one considering all terms. The shape has been sampled in 100 segment, cameras' FOV is restricted to an angle of 30 degrees and a maximum range of 350 units, and a value of $w = 5$ has been set for γ_σ. The resulting *restricted visibility graphs* after the optimization are shown in Fig. 3, and Table I indicates the values that each term in γ takes.

TABLE I

TERMS OF γ EVALUATED WITH THE UNDEFORMED SQUARED SHAPE, FOR THE DIFFERENT TERM COMBINATIONS.

Simulation	γ_v	γ_σ	γ_β	γ
$\gamma = \gamma_v$	0	5.2108E-4	0.9883	0.9889
$\gamma = \gamma_v + \gamma_\sigma$	0	9.0296E-5	0.9875	0.9876
$\gamma = \gamma_v + \gamma_\beta$	0	9.1176E-5	0.9784	0.9785
$\gamma = \gamma_v + \gamma_\sigma + \gamma_\beta$	0	9.1176E-5	0.9784	0.9785

One can see from these results that γ_v effectively drives the system to accomplish ϑ^t, that γ_σ tends to separate the robots and that γ_β maximizes the minimum angles between consecutive edges. Clearly, γ_β is the dominant term over γ_σ since in this configuration the robots are considerably separated. It is worth mentioning that in the last two rows of Table I the same result is shown due to this fact, and γ_β already tends to separate the cameras in this case.

The method is also tested with a contour that deforms over time. We have selected for this purpose a bone image from the MPEG-7 data set, and we have distorted it with increasing distortion factors to model a deformation process. We have sampled the shape in 100 segments, and the cameras' FOV has been restricted in angle and range to 60 degrees and 400 units, respectively. Figure 4 shows from (a) to (d) the *restricted visibility graphs* after the iterative optimization at several time steps, and from (e) to (h) the one after optimizing with a set of $N_c^* = N_{ck} = 6$ cameras, $\forall k$. It can be seen in Fig. 4 (e) how our conservative method may provide less efficient configurations when more cameras than necessary are accounted: the dark blue camera barely exploits its FOV and it only detects already detected segments.

VI. CONCLUSION AND FUTURE WORK

The method we propose allows to cover the 2D contour of an object that undergoes deformation with a minimal set of cameras, so that a visibility objective is accomplished. The near-optimal positions of the cameras are computed by means of an optimization process that minimizes a cost function which includes inter-robot collision avoidance and robust visibility aspects. Our approach is limited in different ways: the proposed system is unable to work with unknown objects, it is restricted to smooth deformations and it can not guarantee the global optimal configurations. Future research directions include extending our method to 3D deformable objects and 3D spaces, considering scenarios with occluding obstacles, allowing the cameras to go into and out of the formation along the deformation process (depending on the coverage requirements), introducing object resolution in the image as an optimization index, and performing experiments in real-world scenarios.

ACKNOWLEDGMENT

The authors would like to thank Leonardo Fermín and Edgar Ramírez for their invaluable help.

REFERENCES

[1] J. D. Langsfeld, A. M. Kabir, K. N. Kaipa, and S. K. Gupta, "Robotic bimanual cleaning of deformable objects with online learning of part and tool models," in *IEEE CASE*, Nov. 2016, pp. 626–632.

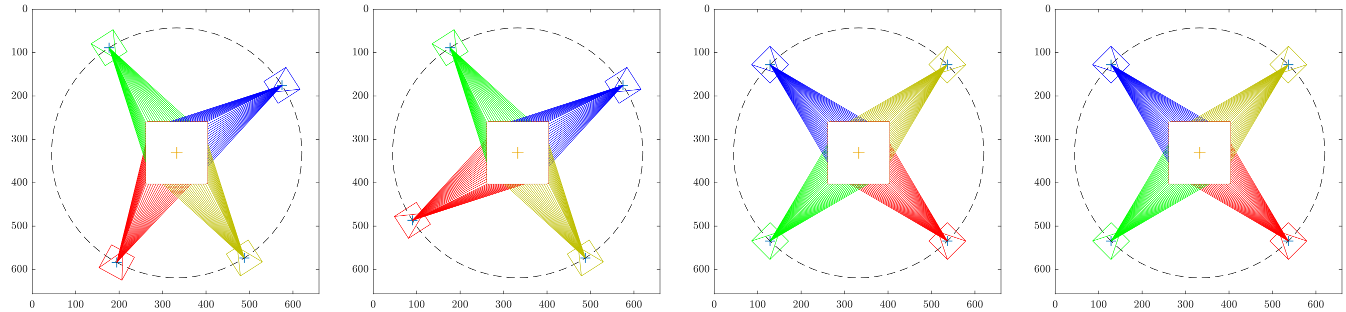
[2] D. Navarro-Alarcon, H. M. Yip, Z. Wang, Y. H. Liu, F. Zhong, T. Zhang, and P. Li, "Automatic 3-d manipulation of soft objects by robotic arms with an adaptive deformation model," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 429–441, 2016.

[3] J. Sanchez, J. A. Corrales, B. C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, 2018.

[4] M. Liu and K. Ramani, "On minimal orthographic view covers for polyhedra," in *IEEE International Conference on Shape Modeling and Applications*, June 2009, pp. 96–102.

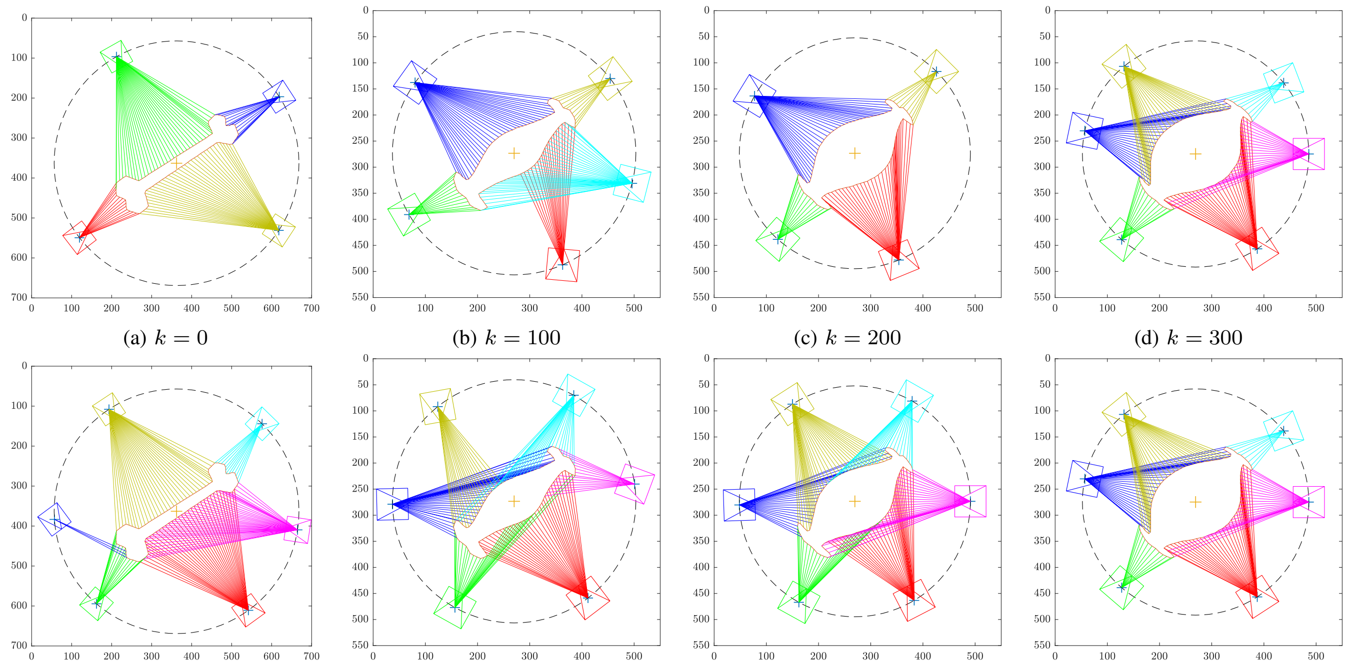
[5] R. Strauss, F. Isvoranu, and G. Elber, "Geometric multi-covering," *Computers and Graphics*, vol. 38, no. 1, pp. 222–229, 2014.

[6] R. Kalawoun, S. Lengagne, and Y. Mezouar, "Optimal robot base placements for coverage tasks," in *IEEE CASE*, Aug. 2018, pp. 235–240.



(a) $\gamma = \gamma_v$ (b) $\gamma = \gamma_v + \gamma_\sigma$ (c) $\gamma = \gamma_v + \gamma_\beta$ (d) $\gamma = \gamma_v + \gamma_\sigma + \gamma_\beta$

Fig. 3. *Restricted visibility graph* after the optimization over an undeformed squared shape, with four different combinations of the cost function γ terms. ϑ_j^t has been set equal to 1 for all segments.



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

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

Fig. 4. From (a) to (d), *restricted visibility graph* after the iterative optimization at different deformation instants, over the bone shape. From (e) to (h), *restricted visibility graph* after the optimization with a set of $N_c^* = 6$ cameras at different deformation instants over the same shape.

[7] L. J. Zhang, C. C. Gu, K. J. Wu, Y. Huang, and X. P. Guan, “Model-based active viewpoint transfer for purposive perception,” in *IEEE CASE*, Aug. 2017, pp. 1085–1089.

[8] W. Jing, J. Polden, P. Y. Tao, C. F. Goh, W. Lin, and K. Shimada, “Model-based coverage motion planning for industrial 3D shape inspection applications,” in *IEEE CASE*, Aug. 2018, pp. 1293–1300.

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

[10] M. Schwager, B. J. Julian, M. Angermann, and D. Rus, “Eyes in the sky: Decentralized control for the deployment of robotic camera networks,” *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1541–1561, 2011.

[11] F. M. De Rainville, J. P. Mercier, C. Gagne, P. Giguere, and D. Laurendeau, “Multisensor placement in 3D environments via visibility estimation and derivative-free optimization,” in *IEEE ICRA*, June 2015, pp. 3327–3334.

[12] A. Adaldo, S. S. Mansouri, C. Kanellakis, D. V. Dimarogonas, K. H. Johansson, and G. Nikolakopoulos, “Cooperative coverage for surveillance of 3D structures,” in *IEEE IROS*, Sept. 2017, pp. 1838–1845.

[13] G. Lopez-Nicolas, M. Aranda, and Y. Mezouar, “Formation of differential-drive vehicles with field-of-view constraints for enclosing a moving target,” in *IEEE ICRA*, 2017, pp. 261–266.

[14] M. Aranda, G. López-Nicolás, C. Sagüés, and M. M. Zavlanos, “Three-dimensional multirobot formation control for target enclosing,” in *IEEE IROS*, Sept. 2014, pp. 357–362.

[15] D. S. Schacter, M. Donnici, E. Nuger, M. Mackay, and B. Benhabib, “A multi-camera active-vision system for deformable-object-motion capture,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 75, no. 3-4, pp. 413–441, 2014.

[16] E. Nuger and B. Benhabib, “Multi-Camera Active-Vision for Markerless Shape Recovery of Unknown Deforming Objects,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 92, no. 2, pp. 223–264, 2018.

[17] R. Diestel, *Graph Theory*, 4th ed., ser. Springer Graduate Texts in Mathematics (GTM). Springer New York, 2010.

[18] C. Audet and J. E. Dennis, “Analysis of generalized pattern searches,” *SIAM Journal on Optimization*, vol. 13, no. 3, pp. 889–903, 2002.

[19] L. Latecki, R. Lakamper, and T. Eckhardt, “Shape descriptors for non-rigid shapes with a single closed contour,” in *IEEE CVPR*, June 2002, pp. 424–429.