

Perimeter-based Calculation of Object Coverage in Multi-Camera Systems

Cristian Duran-Faundez, Daniel G. Costa and Vincent Lecuire

Abstract—For many visual monitoring scenarios, it may be desired to accurately compute the segment of an object's perimeter that is being viewed by a particular camera, e.g. supporting applications for pattern recognition systems and coverage optimization. As objects may have different formats, generic polygon-based approaches may bring inaccurate results, since some simplifications may be taken. In this context, it is proposed in this paper an algorithm to accurately calculate the objects' contours that can be viewed by a set of cameras, employing a discretization technique. The proposed algorithm computes the percentage of an object's perimeter that is being viewed, which may be used as a parameter of coverage quality. Therefore, besides accurately computing effective viewing of cameras, a visual coverage quality metric for a set of objects is derived, which can be directly exploited for optimizations in camera surveillance systems and visual sensor networks.

Index Terms—Camera coverage, objects viewing, objects contours, visual sensing.

I. INTRODUCTION

CAMERA coverage is a relevant topic that has driven many research efforts in last decades, from the perspective of optimal camera placement [1] until sensing coverage optimizations in modern cameras systems [2], [3]. For some applications, the main optimization problem may be concerned with the minimum number of cameras to cover an area of interest, which may be a room, a road, a square or a generic wide area [4]. On the other hand, for object (target) coverage, cameras may need to cover segments of objects or even objects as a whole, regardless the monitored field [5]. Camera coverage may be then a challenging task, with different particularities according to the characteristics of visual coverage applications.

In general, objects of interest may have different formats, which may be not predicted by a system. Actually, for some applications, cameras may only need to cover part of an object to detect presence, as for example in intrusion detection systems [6]. In a different way, some applications may consider different perceptions over the same object, since cameras may view it under different perspectives [7]. With new wireless sensing technologies based on visual coverage, object viewing has been considered for many applications, but visual coverage over objects has still been

Manuscript received March 01, 2016; revised May 26, 2016. This work was supported in part by the Brazilian research agency CNPq under Grant number 441459/2014-5 and by the University of the Bío-Bío, under Grants DIUBB 161610 2/R and GI 160210/EF.

Cristian Duran-Faundez is with the Department of Electrical and Electronic Engineering, DIEE, University of the Bío-Bío, Concepción, Chile. E-mail: crduran@ubiobio.cl

Daniel G. Costa is with the Department of Technology, State University of Feira de Santana, Feira de Santana, Brazil. E-mail: danielgcosta@uefs.br

Vincent Lecuire is with the Research Center for Automatic Control of Nancy, University of Lorraine, CNRS, Nancy, France. E-mail: vincent.lecuire@univ-lorraine.fr

commonly processed through simplifications, where they are usually modeled as circles or 2-D or 3-D polygons.

For objects moving over an area covered by a multi-camera system, it may be desired to accurately compute the percentage of objects' perimeters that are being viewed by each camera. By doing so, it is possible not only to compute effective viewing over objects, but also to know if critical segments of the objects' perimeters are being viewed, as for example the plates of moving cars or bar codes on packages on a conveyor belt.

The viewed segments of an object's perimeter represents a percentage of visual coverage over that object, which we define generically as the Effective Object Viewing (EOV). This metric can be used to indicate the level of visual coverage over a set of objects, which may be considered, for example, when defining coverage quality metrics [8] in visual sensing systems. Moreover, it might be exploited to dynamically adjust the orientations of cameras for more effective viewing over objects.

In this context, it is proposed herein a mathematical model to accurately calculate objects perimeters captured by cameras, computing the actual contours of objects represented as simple polygons in a 2-D space. The accurate calculating of objects' perimeters and the computing of an Adaptive EOV (A-EOV) coverage metric are the main contributions of this paper.

The remainder of this paper is organized as follows. Section II brings some related works. The proposed approach is defined in Section III. Practical uses of the A-EOV metric are discussed in Section IV. Section V presents numerical results, followed by conclusions and references.

II. RELATED WORKS

Sensing coverage is a major design issue for multi-camera systems, directly related to the quality of camera-based monitoring applications. However, many relevant challenging issues have to be properly addressed, notably in camera coverage, visual data coding and processing. In such way, in the last years, many works have proposed promising approaches for coverage enhancement and assessment for different scenarios, influencing this paper in different ways.

Generally speaking, efficient visual coverage will be deeply related to the way cameras are positioned. And cameras deployment will be typically guided by monitoring requirements. Actually, cameras may be used to view an area of interest in different perspectives, but frequently they will be concerned with area, target or barrier coverage [9]. For all these cases, cameras may also be deployed in a random or deterministic way [3]. For random deployment, cameras may be scattered over a monitored field, with unpredictable positions and orientations after deployment, which could be more

common in a wireless visual sensor network context. On the other hand, deterministic deployment requires previous knowledge about the characteristics of the monitored field and targets or areas to be monitored, but optimal coverage is more feasible in this scenario. Whatever the case, cameras positioning is a relevant issue that impacts multi-camera systems [10].

Sometimes, coverage metrics may be desired to assess the visual monitoring quality of cameras. The work in [11] proposes a metric to measure the coverage quality of wireless visual sensor networks, computing the probability of a randomly deployed network to be K-Coverage, which is achieved when every point is covered by at least K sensors. In this case, for higher values of K, more visual sensors will be viewing the same area of a monitored field, which may be beneficial for many applications. In a different way, it is proposed in [12] a metric to compute the coverage quality when visual sensors are deployed for target coverage.

When adjustable cameras are employed, the coverage area may be optimized to better address the monitoring requirements of applications [13], [14]. The work in [15] computes an optimal configuration for visual sensors with changeable orientations, where visual coverage is based on the definition of non-disjoint cover sets. The work in [16] adjusts the sensors' FoV to optimize the overall coverage, achieving maximized viewing of a monitored field. For that, sensors are reconfigured to new optimal positions.

Sensing coverage is also relevant when addressing redundancy. In general, sensing redundancy is based on overlapping of sensing areas, but the way overlapping will be processed will depend on monitoring requirements of the applications [17], [18]. For redundancy computation, target perspectives may also be considered, providing a more accurate definition of redundancy [7].

Cameras orientations will define how targets or areas will be viewed, since coverage perspectives may impact the quality of visual monitoring systems. The work in [19] addresses the problem of computing the minimum number of visual sensors for barrier coverage, taking viewing angles in the optimization process. In that work, objects are approximated to regular polygons. In [20], an object has to be viewed under different angles, since cameras may view different sides of the object. The optimization problem in [20] is then to compute the minimum set of sensors that can retrieve images that satisfy the resolution requirements of the application, assuming objects with cylindrical shapes. The work in [21] computes the minimum set of cameras to cover objects satisfying some angle constraint, optimizing the k-angle problem when the angle of viewing is a key parameter. In all these works, objects modeling is simplified for easier computing.

Actually, when performing target coverage, segments of viewed objects (targets) may have different importance for the performed tasks. Thus, knowing the viewed segments of objects' perimeters potentially allows pre-processing of retrieved data in different types of multi-camera systems, e.g. supporting decision procedures and prioritization approaches [22]. The innovative proposed mathematical computation of viewed segments, without processing image data, may then bring significant results, specially for resource-constrained systems.

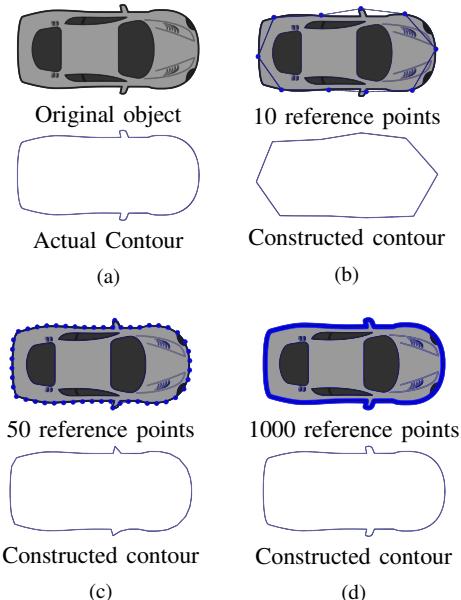


Fig. 1: Example of 2-D discretized contours.

III. PROPOSAL

In order to allow that the viewed segments of any kind of 2-D object can be properly identified and accounted, for any possible format of objects, it is proposed in this paper a novel approach that is intended for mathematical processing of visual coverage. For that, in this paper, objects and targets are used interchangeably to refer to any static or moving element that needs to be viewed.

The fundamentals of the proposed approach are stated in next subsections.

A. Definition of objects contours

In a 2-D space, a discretized contour of an object can be described as simple polygon, P , defined as a vector of P_l connected segments (edges), $P = [P(0), P(1), P(2), \dots, P(P_l - 1)]$. Each segment $P(i)$ is defined as a pair $P(i) = (P(i)_{P0}, P(i)_{P1})$, where $P(i)_{P0} = (P(i)_{P0_x}, P(i)_{P0_y})$ and $P(i)_{P1} = (P(i)_{P1_x}, P(i)_{P1_y})$ are the coordinates of the two points describing segment $P(i)$. Segments are connected through common points in the closed chain, so that $P(P_l - 1)_{P1} = P(0)_{P0}$, $P(0)_{P1} = P(1)_{P0}, \dots, P(i - 1)_{P1} = P(i)_{P0}$, $P(i)_{P1} = P(i + 1)_{P0}, \dots, P(P_l - 2)_{P1} = P(P_l - 1)_{P0}$. We consider that the discretized contour is well constructed so it faithfully follows the edge of the object in the chain, and elements in P are ideally organized so that segments do not intersect other edges (definition of a simple polygon), as exemplified in Figure 1.

The figure shows examples of different contours representing the original object at the top of Figure 1a. Figures 1b, 1c and 1d show examples of different constructed polygons describing the geometry of the original object's contour. It is clear that, depending on the morphology of the target object, greater and more uniform-size amounts of segments, in non-straight sectors, better describe the actual contour (e.g., in Figure 1, differences between the constructed contour with 1000 segments and the actual contour are almost imperceptible).

$$\Delta(P(i), s) = \begin{cases} 1 & \text{if the segment } P(i) \text{ is entirely contained in the triangle representing the field of view of sensor } s. \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$\bar{\tau}(P, i, s) = \begin{cases} 0 & \text{if, considering a triangle determined by segment } P(i) \text{ and the coordinate of } s, \text{ there is no segment } P(j), P(j) \neq P(i) \text{ that is entirely contained in such a triangle or is intersected in at least one point with that triangle's perimeter.} \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

B. Coverage metric for object viewing

While some simplifications could be taken for easier processing, assuming objects as e.g. circumferences or uniform polygons, the achieved results might not be accurate. However, as the contours of an object can be precisely calculated when employing the proposed approach, and knowing the configuration of the considered cameras, it is possible to compute the percentage of any object that is being viewed.

If there is a way to compute all segments that comprise any object, assuming a defined number of reference points, it is possible to compute the complete perimeter of the objects. And this information can be processed in two different ways: (a) it is possible to know if relevant parts of objects, as for example people's faces or cars' plates, are being viewed by at least one camera, and (b) we can account the percentage of the objects' contours that are being viewed.

Although we initially addressed (a) in [22], a coverage quality metric is still necessary for issue (b). Actually, for this particular issue, we propose the Adaptive Effective Object Viewing (A-EOV) metric as the sum of measures associated to segments in the field of view of a node camera, divided by the maximum possible value for this sum. The "adaptive" was inserted into the metric name as any format of objects may be processed by the proposed algorithm, indicating that the algorithm can inherently adapt to different coverage scenarios.

Let us define $\delta(P(i))$ as the length of segment $P(i)$, given by the Euclidean distance between its two describing points, i.e.:

$$\delta(P(i)) = \sqrt{(P(i)_{P1_x} - P(i)_{P0_x})^2 + (P(i)_{P1_y} - P(i)_{P0_y})^2} \quad (3)$$

Moreover, let us define Δ and $\bar{\tau}$ as the functions described by expressions 1 and 2, respectively.

Equation 1 is used to know which blocks are potentially seen by sensor s . Equation 2 is used to know if there are no occlusions between a given segment and sensor s . As seen, it is considered in this work that a given segment (edge) of a polygon is captured by a sensor node s if and only if it is entirely contained in the field of view of s and if it can be entirely viewed by that sensor.

Logical conditions required to know whether a segment of a polygon is entirely contained or not in a triangle can be computed as follows: Let $T_{P1} = (T_{P1_x}, T_{P1_y})$, $T_{P2} = (T_{P2_x}, T_{P2_y})$ and $T_{P3} = (T_{P3_x}, T_{P3_y})$ be the coordinates of the vertices of a triangle T , and let $P(i)_{P0} = (P(i)_{P0_x}, P(i)_{P0_y})$ and $P(i)_{P1} = (P(i)_{P1_x}, P(i)_{P1_y})$ be the two points describing a segment of line $P(i)$, then $P(i)$ is entirely contained in triangle T if and only if there exist two intersection points with coordinates $P_I0 = (P_I0_x, P_I0_y)$ and $P_I1 = (P_I1_x, P_I1_y)$ of the edges of T and the straight

line described by Equation 8, such that all of the following conditions are satisfied:

$$\min(P_I0_x, P_I1_x) \leq P(i)_{P0_x} \leq \max(P_I0_x, P_I1_x) \quad (4)$$

$$\min(P_I0_x, P_I1_x) \leq P(i)_{P1_x} \leq \max(P_I0_x, P_I1_x) \quad (5)$$

$$\min(P_I0_y, P_I1_y) \leq P(i)_{P0_y} \leq \max(P_I0_y, P_I1_y) \quad (6)$$

$$\min(P_I0_y, P_I1_y) \leq P(i)_{P1_y} \leq \max(P_I0_y, P_I1_y) \quad (7)$$

$$\frac{x - P(i)_{P0_x}}{P(i)_{P1_x} - P(i)_{P0_x}} = \frac{y - P(i)_{P0_y}}{P(i)_{P1_y} - P(i)_{P0_y}} \quad (8)$$

Additionally, information about the redundancies of the fields of view of multiple cameras capturing the object described by P can be provided, by defining $\nu = [\nu_0, \nu_1, \dots, \nu_{P_l-1}]$, where:

$$\nu_i = \sum_{s=0}^{S-1} \Delta(P, i, s) \quad (9)$$

$$\Delta(P, i, s) = \Delta(P(i), s) \cdot \bar{\tau}(P, i, s) \quad (10)$$

thus, each element of ν , ν_i , contains the number of sensors entirely capturing segment $P(i)$.

Finally, by defining operation $Q(\nu_i)$ as expressed in Equation 11,

$$Q(\nu_i) = \begin{cases} 1 & \text{if } \nu_i > 0 \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

we can compute a new coverage metric for object viewing. Actually, as we are considering that objects of any format can be processed by the proposed approach, and as we already achieved some initial results for perimeters calculation of circumference-modelled objects in [23], we reinforce the idea of the proposed metric as being "adaptive".

The A-EOV can finally be calculated as:

$$\text{A-EOV} = \frac{\sum_{i=0}^{P_l-1} \delta(P(i)).Q(\nu_i)}{\sum_{j=0}^{P_l-1} \delta(P(j))} \quad (12)$$

IV. ON EXPLOITING OBJECTS' PERIMETER CALCULATION

When employing the proposed approach, the perimeter of objects viewed by camera systems can be computed. And this information can be exploited in different ways. Next subsections discuss two potential problems that can be addressed by the proposed approach.

A. Availability of monitoring applications

In general camera networks, availability will be a function of different parameters, notably cameras failures, communication problems and bad (insufficient) visual coverage [17]. Depending on monitoring requirements, which may be concerned with targets viewing [18], unavailability may be resulted when cameras are not viewing at least the minimum expected number of targets. However, as viewing perspectives over targets may be also important for some applications, availability may be also computed as average percentage of viewed objects' perimeters, which may be assessed directly by employing the A-EOV metric.

Objects in camera systems may have different dimensions, potentially impacting target monitoring quality. While small targets may be sometimes more likely to be viewed, large targets may be not satisfactorily covered by deployed visual sensors. The A-EOV metric can then numerically reflect the impact of different targets formats and sizes on the overall monitoring quality.

B. Relevance-based prioritization

In many cases, different regions of an object contour may have different relevances. For example, in a surveillance and access control application, it may be more relevant to get an image of the car number plate than a rear port. To consider this relevant issue, it is possible to fragment an object edge into O different non-intersected fragments $F(o)$, $o \in [0, O]$, each one defined as a vector $F(o) = [F(o)_V, F(o)_0, F(o)_N]$, where $F(o)_N$ is the number segments considered in $F(o)$ starting at $F(o)_0$, and $F(o)_V$ is a S long vector where each element $F(o)_{Vs}$ gives:

$$F(o)_{Vs} = \begin{cases} 1 & \text{if } \left(\prod_{q=F(o)_0}^{F(o)_N+F(o)_0-1} \Delta(P, q \mod P_l, s) \right) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

Equation (13) states that $F(o)_{Vs}$ will give 1 (one) if and only if all segments considered in $F(o)$ are viewed by camera sensor s .

When designing priority-based optimizations, two different characteristics must be defined [24]: a) the prioritization parameter and b) the way it will be exploited. In general, prioritization may be based on any approach that produces data with different relevance for applications [25], [24]. After defining the parameter, an optimization approach may be defined, which may optimize different procedures related to sensing, coding and packet transmission (in different communications layers).

When performing visual monitoring over a set of static or moving targets, source prioritization may be defined according to the way those targets are viewed, as also defined by us in [22]. If we can know which segments of targets' perimeters are being viewed, and if a priority index is associated to defined segments, different relevances may be associated to cameras according to the viewed targets. For example, cameras that view more relevant segments of targets may transmit prioritized packets (e.g. for routing and congestion purposes) or even employ high-quality image coding for more significant data, defining then a mechanism to assure

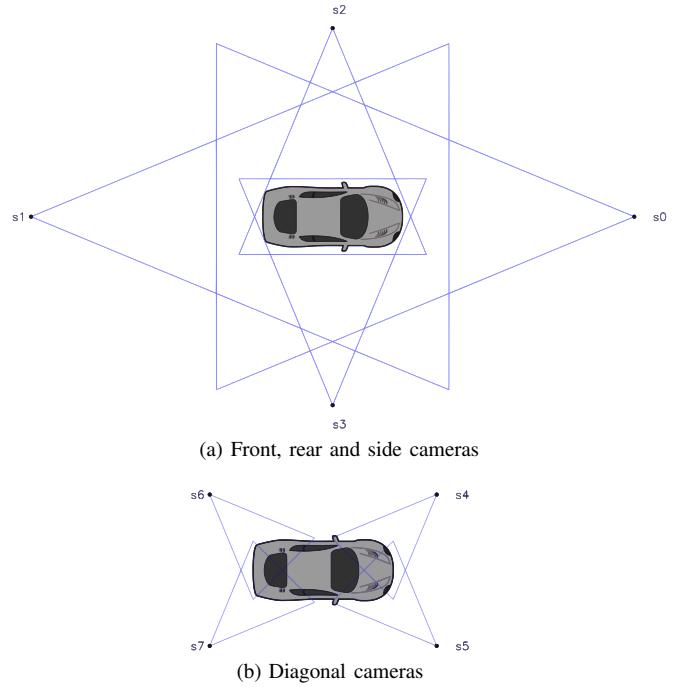


Fig. 2: Case study. Object with eight different camera sensors (s_0, \dots, s_7).

TABLE I: Scenarios for experimentation.

| Scenario | Active camera sensors |
|----------|-----------------------|
| S1 | s_0 |
| S2 | s_0, s_1 |
| S3 | s_0, s_1, s_2 |
| S4 | s_0, s_1, s_2, s_3 |
| S5 | s_4 |
| S6 | s_4, s_5 |
| S7 | s_4, s_5, s_6 |
| S8 | s_4, s_5, s_6, s_7 |

some Quality of Service (QoS) [24], [26]. The work in [22] associates viewed perimeters to priority indexes, but objects in that work are modeled only as rectangles.

V. RESULTS

In order to validate the proposed approach, concerning to computing performance and A-EOV calculation, some experiments were performed, as described in this section.

Experimental results are provided by considering a general case study in which the vehicle (object) in Figure 1a is captured by up to eight different sensor nodes. In order to consider different possible scenarios we define up to eight camera sensors with fields of view as depicted in Figure 2.

The proposed algorithm was implemented in C++ Language with OpenCV v2.4.9 in order to facilitate automatic contours detection and visualization. The experiments were executed in a server with Intel Xeon 3.1GHz, 8GB DDR4 RAM and Ubuntu 14.1 system.

A. Precision and execution time results

Without loss of generality, precision results will be provided by measuring A-EOV over two basic geometric forms approximated by a variable number of segments: (i) the

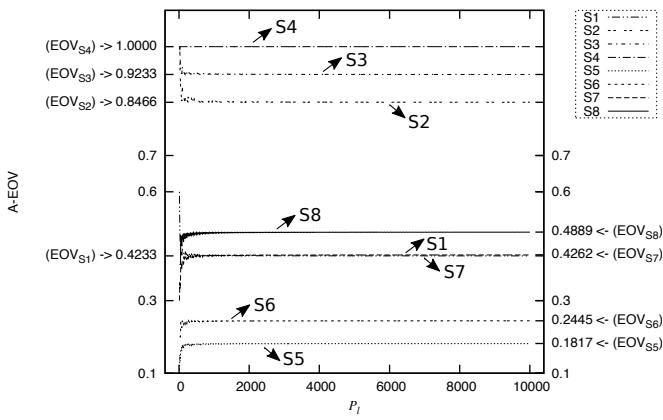


Fig. 3: A-EOV results. Containing circle case.

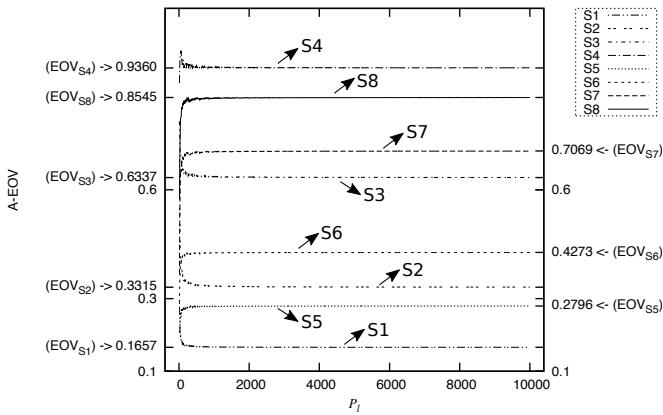


Fig. 4: A-EOV results. Containing rectangle case.

smallest circle and (ii) the smallest rectangle entirely containing the target vehicle (the smallest circle entirely containing a target object corresponds to the approach presented in [23]). This aims to provide valuable results to show how well the polygon-based approximation represents curved shapes (a circle, in this case) and general straight-line based polygons (such as a rectangle). We selected these shapes because it is easy to compute exact values based in elementary geometry and to compare them with our approximated values. For results purposes, we define the eight different situations as summarized in Table I.

The graphics of Figures 3 and 4 show A-EOV results for the different scenarios when varying the number of segments approximating both, said circle and rectangle, respectively, from 10 to 10000 edges (P_l).

We denote EOV_{Sc} the EOV calculated with simple geometry for scenario Sc (the exact reference value). In the graphics, it is possible to observe a rapid trend of A-EOV to the EOV, in all cases. For example, in case S1, the error (absolute difference) between A-EOV and EOV_{S1} is of 0.0167 when the number of segments describing the circle is $P_l = 100$. When increasing P_l , this error decreases to 0.0006967 and 0.0000033, for amounts of $P_l = 1000$ and $P_l = 10000$ segments, respectively. This means that we can have decently approximated results with some hundreds of segments, if speed of calculation is a requirement. More accuracy can be achieved by selecting greater amounts of segments (increasing P_l) but at expense of higher calcula-

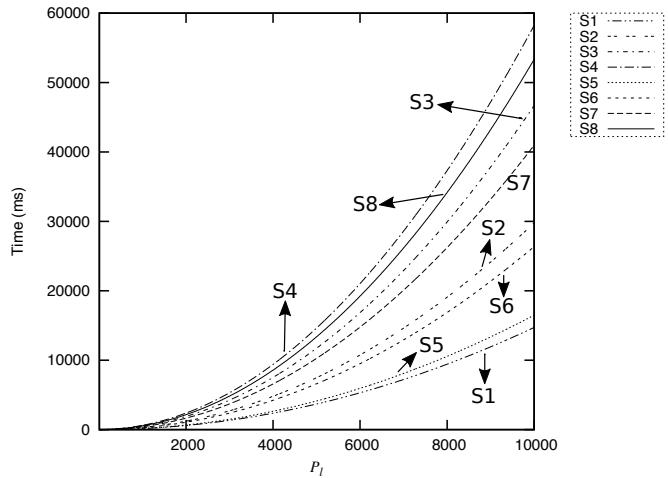


Fig. 5: Execution times. Containing circle case.

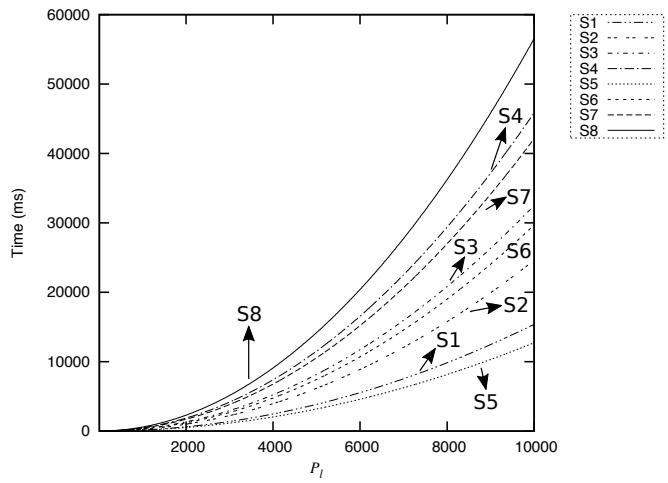


Fig. 6: Execution times. Containing rectangle case.

tion times. Also, calculation times are increased by bigger amounts of camera sensors. Figures 5 and 6 show obtained execution time of our algorithm, for same scenarios and amounts of segments.

B. Simulation of a scene using the case study

In order to illustrate potential uses of the proposed model, we summarize results from various simulations, considering the object of Figure 1a. First tests consider different cases where the target object is discretized by $P_l = 10$, $P_l = 50$, $P_l = 100$, $P_l = 200$, $P_l = 500$, $P_l = 1000$, $P_l = 2000$, $P_l = 5000$, and $P_l = 10000$ points, and distribution of sensor nodes of scenario S8. The simulation considers a vehicle monitoring scenario, in which the four camera nodes are intended to provide images of the car passing an access control, with a priority in capturing the car's plate. To do so, four non-overlapped fragments $F(0) \dots F(3)$ have been defined, being $F(1)$ and $F(3)$ the fragments describing an area in which the car plate should be located, at the rear and front of the car, respectively. $F(1)$ and $F(3)$ were defined as a portion of the perimeter equivalent to one third of the width of the container rectangle described before. Figure 7 illustrates different positions of the target vehicle (object), at different times, entering and leaving the sensors' zone.

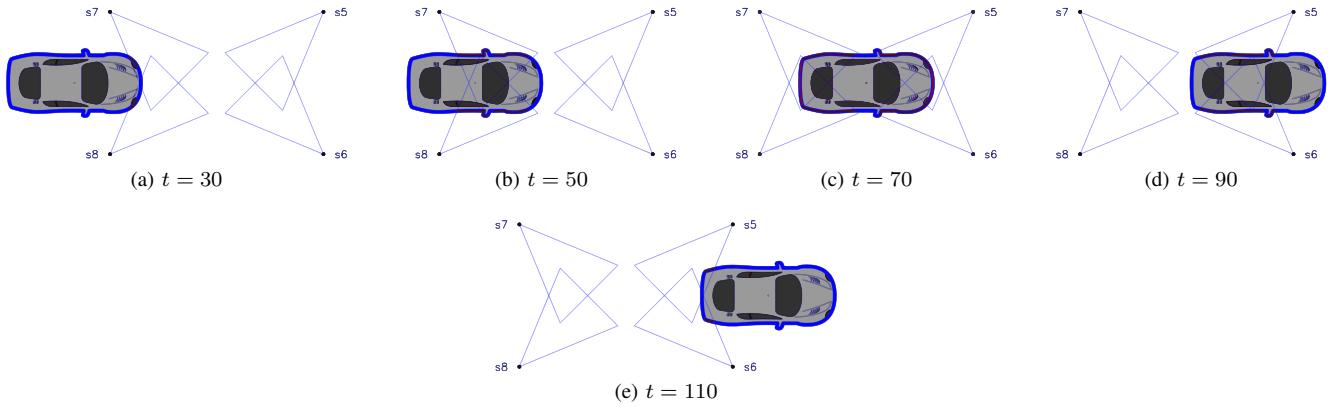


Fig. 7: Simulation of the case study at different times.

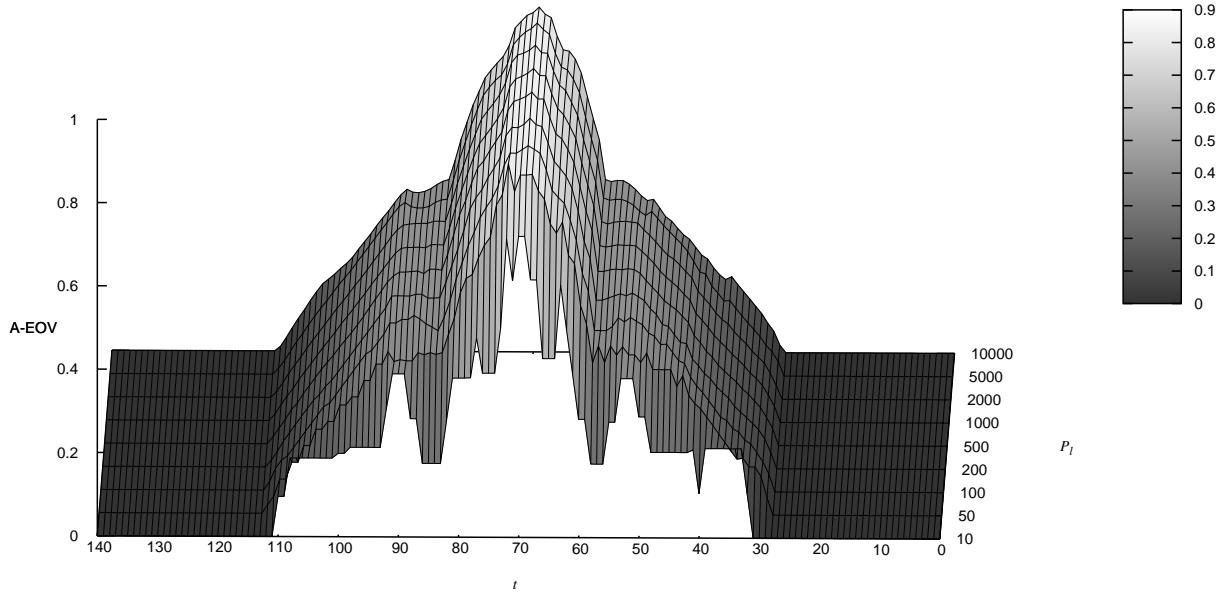


Fig. 8: Simulated A-EOV for the selected case study.

Different values of A-EOV at different times can be seen in the Figure 8, since we consider a moving object. The graphic shows the moments in which the car enters the sensor zone (near $t = 30$) and leaves (near $t = 110$). The maximum A-EOV is allowed at $t = 70$, when the car is at the center of the four cameras.

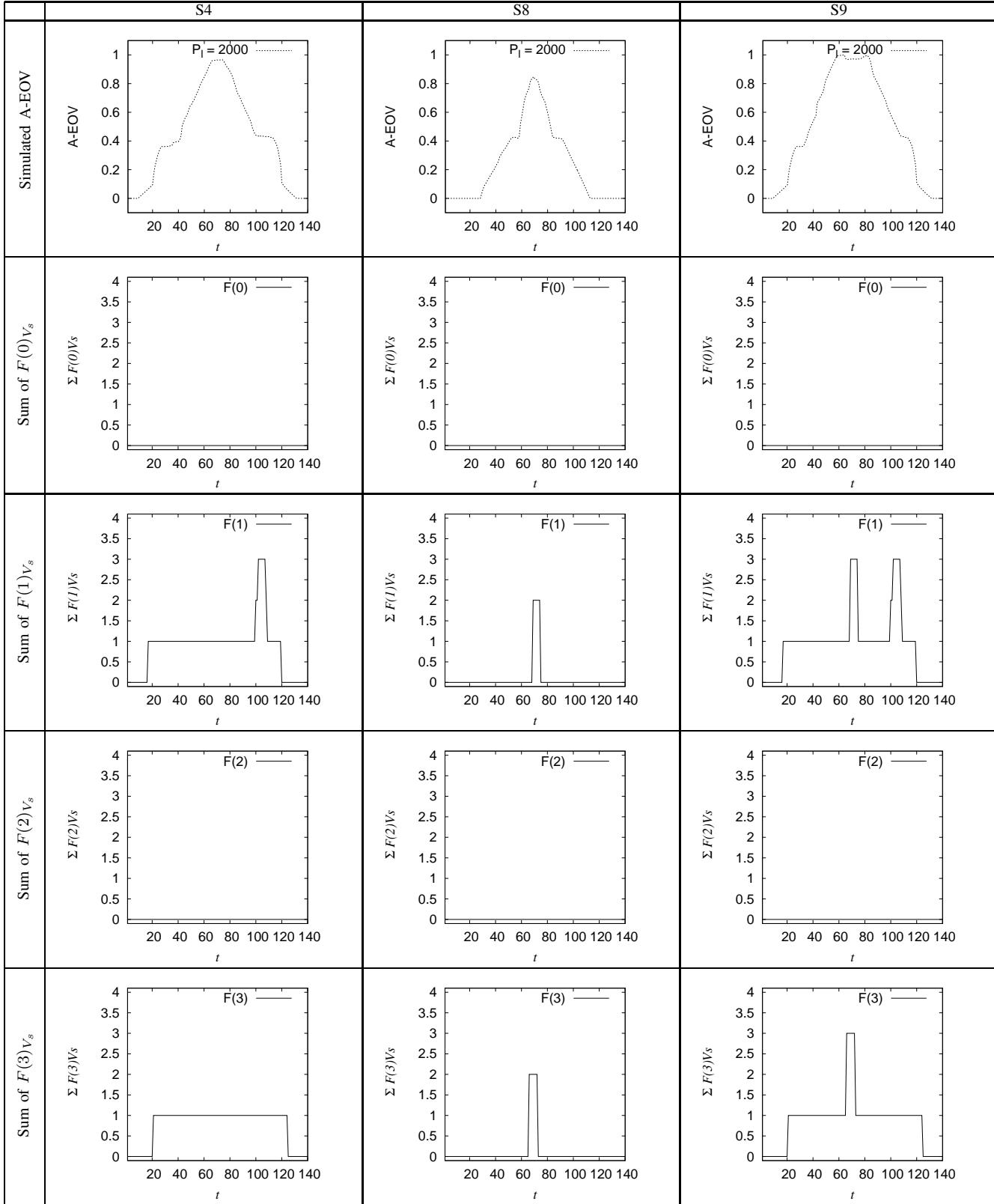
Other simulations allow us to visualize how the calculation of $F(o)$ vector can be used. For that, we executed same $P_l = 2000$ points case described before, with simulation scenarios S4, S8 and S9, being S9 a new scenario where all of the eight cameras were activated. Results of these applications are summarized in Table II, where summed of $F(o)_{V_s}$ values are graphed. This represents the number of camera sensors viewing fragments $F(o)$. And this can be useful when we need to observe particular regions of an image with different priorities.

The graphics show different quantities of cameras entirely viewing each different fragment. For example, for scenario S8, the graphics show that fragment $F(3)$ (front region of the vehicle) is entirely viewed by two cameras in a window time between $t = 66$ and $t = 72$, approximately. The fragment $F(1)$ (rear region of the vehicle) is viewed (also by two

cameras) between $t = 69$ and $t = 74$, approximately. We must note that the sums of $F(0)_{V_s}$ and $F(2)_{V_s}$ remain zero, in all cases. This means that no area fragments $F(0)$ and $F(2)$ are entirely viewed by camera at any time. This is explained by the fact that the mirrors at both sides of the car (original object) do not let to observe the little portions of the doors next to them. If this represents an impediment for the desired application results, a relaxation of the first condition of Equation 13 must be considered.

Finally, with the described method, the calculation of percentages of viewed fragments are trivial. Knowing the A-EOV along the time, for moving objects, can support decisions about the number and positions of cameras, with direct application on traffic monitoring, industrial automation, surveillance, among others. Moreover, as we can also assess the viewing of particular segments of objects, prioritization-based approaches can also be proposed [22].

Moreover, the described vectors can be also useful for visualization. Figure 9 shows an example with simulated scenario S8, adopting $P_l = 2000$ Points, and $t = 70$. In the figure, vector ν has been used to color different segments of the contouring perimeter, going from dark red colors (less

TABLE II: Simulated sums of $F(o)_{V_s}$ for the selected case study with $P_l = 2000$.


cameras viewing that area) to full red (more cameras viewing that area). Other vectors could be used to enhance visual information in a simulation environment.

VI. CONCLUSION

Many camera systems will be designed to cover a set of static or moving objects, with different particularities. For

some applications, it may be desired to compute the effective viewing over objects, which is related to the percentage of the objects' perimeters that are being viewed by cameras. The proposed approach may be employed for objects of any format, allowing a more precise modeling of object viewing.

Multi-camera systems, and more recently, wireless visual sensor networks, will be central in near future integrated

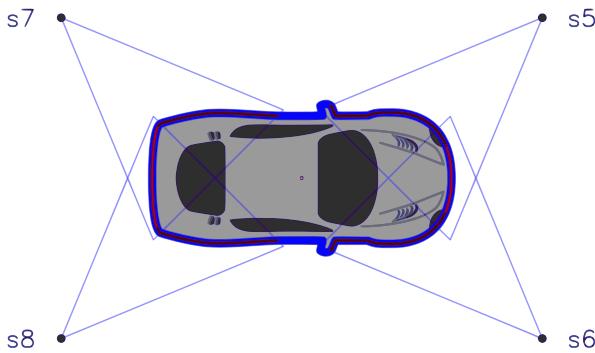


Fig. 9: Visualization of captured perimeter portion at $t = 70$.

monitoring systems, from complex smart city environments to smaller domestic networks [27]. The proposed approach and the computed A-EOV may then be used for a variety of functions, supporting efficient processing of visual coverage.

As future works, the proposed approach will be implemented in a real multi-camera system in order to allow new validation procedures. Moreover, object viewing will be modeled in 3-D space for more complete results.

REFERENCES

- [1] M. Marengoni, B. Draper, A. Handson, and R. Sitaraman, “A system to place observers on a polyhedral terrain in a polynomial time,” *Image and Vision Computing*, vol. 18, pp. 773–780, 1996.
- [2] A. Mavrinac and X. Cheng, “Modeling coverage in camera networks: A survey,” *International Journal of Computer Vision*, vol. 101, no. 1, pp. 205–226, 2013.
- [3] D. Costa and L. Guedes, “The coverage problem in video-based wireless sensor networks: a survey,” *Sensors*, vol. 10, pp. 8215–8247, 2010.
- [4] J.-J. Gonzalez-Barbosa, T. Garca-Ramirez, J. Salas, J.-B. Hurtado-Ramos, and J.-J. Rico-Jimnez, “Optimal camera placement for total coverage,” in *IEEE International Conference on Robotics and Automation*, 2009.
- [5] J. Hong, J. Kim, D. Kim, D. Li, and A. O. Tokuta, “Desperate coverage problem in mission-driven camera sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 2014, pp. Article ID 109 785, 10 pages, 2014.
- [6] C. Pham, “Large-scale intrusion detection with low-cost multi-camera wireless image sensors,” in *Proceedings of IEEE Conference on Wireless and Mobile Computing, Networking and Communications*, 2015, pp. 484–491.
- [7] D. Costa, I. Silva, L. Guedes, F. Vasques, and P. Portugal, “Optimal sensing redundancy for multiple perspectives of targets in wireless visual sensor networks,” in *IEEE International Conference on Industrial Informatics*, 2015, pp. 185–190.
- [8] V. P. Munishwar and N. B. Abu-Ghazaleh, “Coverage algorithms for visual sensor networks,” *ACM Transactions on Sensor Networks*, vol. 9, p. Article No. 45, 2013.
- [9] T. Sauter, “Energy-efficient coverage problems in wireless ad hoc sensor networks,” *Computer Communications*, vol. 29, no. 4, pp. 413–420, February 2006.
- [10] D. Pescaru, V. Gui, C. Toma, and D. Fuiorescu, “Analysis of post-deployment sensing coverage for video wireless sensor networks,” in *International Conference RoEduNet*, 2007.
- [11] L. Liu, H. Ma, and X. Zhang, “On directional k-coverage analysis of randomly deployed camera sensor networks,” in *IEEE International Conference on Communications*, 2008, pp. 2707–2711.
- [12] M. Alaei and J. M. Barcelo-Ordinas, “Node clustering based on overlapping fovs for wireless multimedia sensor networks,” in *IEEE Wireless Communications and Network Conference*, 2010, pp. 1–6.
- [13] M. Rahimi, S. Ahmadian, D. Zats, R. Laufer, and D. Estrin, “Magic numbers in networks of wireless image sensors,” in *Workshop on Distributed Smart Cameras*, 2006.
- [14] D. Devarajan, R. J. Radke, and H. Chung, “Distributed metric calibration of ad hoc camera networks,” *ACM Transactions on Sensor Networks*, vol. 2, no. 3, pp. 380–403, 2006.
- [15] Y. Cai, W. Lou, M. Li, and X.-Y. Li, “Target-oriented scheduling in directional sensor networks,” in *IEEE Infocom*, 2007, pp. 1550–1558.
- [16] D. Costa, I. Silva, L. Guedes, F. Vasques, and P. Portugal, “Enhancing redundancy in wireless visual sensor networks for target coverage,” in *Brazilian Symposium on Multimedia and the Web*, 2014, pp. 31–38.
- [17] ——, “Availability issues in wireless visual sensor networks,” *Sensors*, vol. 14, pp. 2795–2821, 2014.
- [18] ——, “Availability assessment of wireless visual sensor networks for target coverage,” in *IEEE International Conference on Emerging Technologies in Factory Automation*, 2014, pp. 1–8.
- [19] B. Xu, Y. Zhu, D. Li, D. Kim, and W. Wu, “Minimum (k, w)-angle barrier coverage in wireless camera sensor networks,” *International Journal of Sensor Networks*, vol. 19, pp. 1–10, 2015.
- [20] K.-Y. Chow, K.-S. Lui, and E. Y. Lam, “Maximizing angle coverage in visual sensor networks,” in *IEEE Conference on Communications*, 2007, pp. 3516–3521.
- [21] Y.-C. Tseng, P.-Y. Chen, and W.-T. Chen, “The k-angle object coverage problem in a wireless sensor network,” *IEEE Sensors Journal*, vol. 12, pp. 3408–3416, 2012.
- [22] C. Duran-Faundez, D. G. Costa, V. Lecuire, and F. Vasques, “A geometrical approach to compute source prioritization based on target viewing in wireless visual sensor networks,” in *IEEE World Conference on Factory Communication Systems*, 2016.
- [23] D. G. Costa and C. Duran-Faundez, “Assessing availability in wireless visual sensor networks based on targets’ perimeters coverage,” [*Submitted*].
- [24] D. G. Costa, L. A. Guedes, F. Vasques, and P. Portugal, “Research trends in wireless visual sensor networks when exploiting prioritization,” *Sensors*, vol. 15, no. 1, pp. 1760–1784, 2015.
- [25] D. G. Costa and L. A. Guedes, “Exploiting the sensing relevancies of source nodes for optimizations in visual sensor networks,” *Multimedia Tools and Applications*, vol. 64, pp. 549–579, 2013.
- [26] M. M. Qabajeh, A. H. Abdalla, O. Khalifa, and L. K. Qabajeh, “Geographical multicast quality of service routing protocol for mobile ad-hoc networks,” *Engineering Letters*, vol. 18, no. 3, pp. 212 –225, 2012.
- [27] T. Takabatake, “Power consumption on topologies for a sensor-based home network,” *IAENG International Journal of Computer Science*, vol. 39, pp. 307–311, 2012.