Adversarial Path Planning for Optimal Camera Positioning

Gaia Carenini and Alexandre Duplessis*

Abstract— **The use of visual sensors is flourishing, driven among others by the several applications in detection and prevention of crimes or dangerous events. While the problem of optimal camera placement for total coverage has been solved for a decade or so, that of the arrangement of cameras maximizing the recognition of objects "in-transit" is still open. The objective of this paper is to attack this problem by providing an adversarial method of proven optimality based on the resolution of Hamilton-Jacobi equations. The problem is attacked by first assuming the perspective of an adversary, i.e. computing explicitly the path minimizing the probability of detection and the quality of reconstruction. Building on this result, we introduce an optimality measure for camera configurations and perform a simulated annealing algorithm to find the optimal camera placement.**

I. Introduction

Networks of cameras, or generally of visual sensors, are widely used throughout industrial processes, detection and prevention of crimes or dangerous events, military purposes and so forth. The vast availability of different types of cameras, the decreasing cost of associated hardware, together with an increasing need for such systems are among the reasons intriguing more and more researchers to focus in this field.

A crucial concern raised by the design of a camera network is positioning optimally individual visual sensor under a set of default constraints. In fact, resulting visual measurements can be made significantly more accurate by selecting a suitable configuration established through a proper mathematical model. It's clear that different visual tasks have fairly distinct requirements, e.g., a multi-view reconstruction task stands in need of a minimum number of video sensors with predetermined ranges of angular separation, instead, some aggregate video sensor network must be fault-tolerant to camera drop out and still layouts of sensors in video sensor networks should assure a minimum level of image quality in order to get a sufficient resolution, depth of field, etc. Independently from the specific task, resolution is always a fundamental and primary information bottlenecks for vision applications.

The problem of automating the camera network design process for attaining highly accurate measurements has received comparatively little attention given its practical importance.

Our goal is to address the problem of camera placement to optimize the aggregate observability. One possible application of this research is the development of a design tool for surveillance camera placement in areas of high traffic, where each subject may take a different path through the area. This work assumes the cameras are statically mounted to view an area. Optimizing the observability of such a system means jointly maximizing the power of observation of the cameras for the area of interest.

Related work→ Suitable camera placement for the purpose of optimizing the sensors ability to capture information about a desired environment or task has been studied extensively. In [2], O'Rourke provides an in-depth theoretical analysis of the problem of maximizing camera coverage of an area, where the camera fields of view do not overlap (the so- called "art gallery" problem). Several other results have been published in this direction, e.g., [1], [3].

In recent years, research has sought to extend the framework to include limited field of view cameras and to incorporate resolution metrics into the formulation (see [22], [5], [27]). More specifically, in [34], the art gallery framework is refined by introducing a resolution quality metric. Moreover in [35], the formulation of the minimum guard coverage art gallery problem was extended in order to incorporate minimum-set cover. In the same work, reduced upper bounds were derived for two cases of exterior visibility for two- and three-dimensions.

Contribution→ Our method differs from the art gallery framework in various salient aspects. First, we study the best attacks to visibility conditioned to a given camera configuration and destination, i.e., we find out the paths minimizing the observability of the object "in-transit" from a couple of selected positions. This is done by modelling this task as an optimal motion planning problem that can be solved by applying the technique presented in [36]. We then derive an optimality measure that can be used to assess a configuration. We finally use a simulated annealing based algorithm to find a near-optimal camera placement according to this measure.

II. Preliminaries

We start by reducing the design of "attacks to observability" to a problem of optimal motion planning in a space presenting an anisotropic field of velocities; there the goal is to reach a final position (x_f, y_f) from a start position (x_0, y_0) in a minimum time, while avoiding obstacles and minimizing the risk of being recognized.

For sake of simplicity, we assume that all the cameras are punctiform and we consider a finite 2D environment defined as $\mathcal{R} \subset [a,b] \times [c,d]$ with $a, b, c, d \in \mathbb{R}$. We define an obstacle, *obs*, as a proper subset of $\mathcal{R}$ and we consider the set of the obstacles $\mathcal{O}$. We define identify each camera as a

[1] Gaia Carenini and Alexandre Duplessis are Master Students of the Department of Computer Science at ENS-PSL Research University, Paris, France. Their emails are: name.surname@ens.psl.eu

tuple $C_i = (p_i, \alpha_i, r_i)$ in which p_i is of the form (a_i, b_i, β_i) where (a_i, b_i) are the coordinates of the camera C_i, β_i is the angle that the first ray of the vision field of C_i defines with the vertical according to the standard reference system and r_i is a function describing the resolution that the camera has of an object when the distance from it changes. The vision field is the space swept by any vector moving from the position β_i to the position $\beta_i + \alpha_i$, we call it $\mathcal{F}_i$. For the model proposed below, we will assume that the recognition is jointly inversely proportional to the distance from the camera and directly proportional to the time spent in the field of view of the camera. Other assumptions could be added in a straightforward manner in the model presented that is kept simple for sake of clarity.

Fix the start at $(x_0, y_0) \in \mathcal{R}$, the destination point in $(x_f, y_f) \in \mathcal{R}$ and N cameras $\mathcal{C} = \{C_i\}_{i=1}^N$, we model cameras visual field as an anisotropic speed field $\overrightarrow{W}$ that depends on (x_f, y_f) as follows: $\forall (x,y) \in \mathcal{R}$ $\overrightarrow{w}(x,y)$ is either 0 if $(x,y) \notin \bigcup_{i\in[1,N]} \mathcal{F}_i$ or as the vector that has direction the opposite of the conjunction of (x,y) and (x_f, y_f) and has module $1/dist((a_i,b_i),(x,y))^2$.

Under this assumption, the speed of the motion that we consider is defined by the equations:

$$\begin{cases} \dot{x}(t) = (V_c + w(x,y))\sin(\theta(t)) \\ \dot{y}(t) = (V_c + w(x,y))\cos(\theta(t)) \end{cases} \tag{1}$$

where (x,y) is the mobile position, θ is the heading angle relative to north direction and V_c is a constant finite speed of the object ”in transit” that has direction corresponding to the conjunctive from the start position and the destination and constant module c.

With the formalism introduced above, we can define the resolving optimization problem that can be written as:

$$\begin{cases} \min & (t_f - t_0) \\ s.t. & \dot{x}(t) = (V_c + w(x,y))\sin(\theta(t)) \\ & \dot{y}(t) = (V_c + w(x,y))\cos(\theta(t)) \\ & (x(t_0), y(t_0)) = (x_0, y_0) \\ & (x(t_f), y(t_f)) = (x_f, y_f) \end{cases} \tag{2}$$

We observe that the control parameter of (2) is the the heading angle θ and that the solution of the optimization problem is therefore finding the θ that over time minimize the total travel time.

For sake of simplicity, we consider as control variable the unit vector naturally associated with it, i.e. $\overrightarrow{a}(t) = (sin(\theta(t)), cos(\theta(t)))$. In this case, the problem can be restated simply as:

$$\begin{cases} \min_{\overrightarrow{a}} \quad (t_f - t_0) \\ s.t \quad \dot{X} = f(X(t), \overrightarrow{a}(t)) \\ \qquad X(t_0) = X_0 \wedge X(t_f) = X_f \end{cases} \tag{3}$$

where X is the position of the object ”in-transit” and $f(X(t), \overrightarrow{a}(t))$ is the real speed of the mobile at time t.

The optimal control problem introduced (3) is classical and we can observe how the corresponding Hamilton-Jacobi equation is given by:

$$\max_{\overrightarrow{a}\in A}\{-\nabla u(X), f(X, \overrightarrow{a})\} = 1 \tag{4}$$

where $u(X)$ represents the minimum time to reach the destination starting from the point X.

III. Motion planning

In this section, we discuss the resolution of (4). The same issue has been solved in [36] in the context of optimal motion planning in presence of wind. For seek of completeness, we outline below the method.

The idea behind the resolution is to decompose recursively the problem into linked sub-problems as it happens in dynamic programming. More specifically, the resolution can be seen as a front expansion problem, where the wavefront represents the minimum time to reach the arrival point. The computation is based on the classical Huygen’s principle, which that states ”every point reached by a wavefront becomes a source of a spherical wavefront”. The evolution of the wavefront is given by:

$$||\nabla u(X)|| F\left(X, \frac{\nabla u(x)}{||\nabla u(X)||}\right) = 1 \tag{5}$$

where $F(X, \overrightarrow{n})$ is the front speed in the direction $\overrightarrow{n}$ of the outward unit vector normal to the front at point X.

Through some algebraic manipulation (see [36]), the optimal path problem can therefore be written as a front expansion problem where the speed of the wavefront is given by:

$$F(X, \overrightarrow{n}) = \max_{\overrightarrow{a}}\{-\overrightarrow{n}.f(X, \overrightarrow{a})\} \tag{6}$$

To design the optimal path between the departure point and the final point, it is sufficient to exploit the characteristics of the Hamilton-Jacobi PDE.

Several methods exist for finding an approximation of the solution of these Hamilton-Jacobi equations, in this context, we apply the so-called *ordered upwind algorithm*. Presented in [38], this technique was proven to converge to a weak solution of the PDE, in particular to the *viscosity* one. Its basic principle is to avoid useless iterations, common in Dijkstra-like methods, thanks to a careful use of the information about the characteristic directions of the PDE.

The first step consists in computing the value function, u, considering a 2D nonregular triangular mesh. The method applied is described extensively in [36]. Other methods exist, e.g. semi-Lagrangian and Eulerian discretization [38], however both of them have disadvantages requiring multiple local minimization and finding the roots of a non-linear equation respectively.

In our case, the speed of the wavefront F has a closed form, and the value can be computed using a finite-differences upwind formula of the Hamilton-Jacobi equation.

By modelling the object ”in-transit” speed as $f(\mathrm{X}, a) = V_a a + W$, the speed of the wavefront is equal to:

$$F(\mathrm{X}, n) = V_a - \langle n, W \rangle \tag{7}$$

For more details, read [37]. Fixed this definition of the wavefront speed, to the problem is applied an upwind finite-difference discretization on the simplex $(\mathrm{X}, \mathrm{X}_j, \mathrm{X}_k)$. The associated Hamilton-Jacobi equation becomes:

$$\left\|P^{-1}w(\mathrm{X})\right\|^2 V_a^2 = \left(1 + \left\langle P^{-1}w(\mathrm{X}), W\right\rangle\right)^2 \tag{8}$$

where the vector $P^{-1}w(\mathrm{X})$ is the discretization of $\nabla u(\mathrm{X})$ from the directional derivatives of u in the directions defined by the edges of the simplex $(\mathbf{x}, \mathbf{x}_j, \mathbf{x}_k)$. This equation has the property of being quadratic and has the following form:

$$Av^2_{\mathbf{x}_j\mathbf{x}_k}(\mathrm{X}) + Bv_{\mathbf{x}_j\mathbf{x}_k}(\mathrm{X}) + C = 0 \tag{9}$$

where the coefficients are given by:

$$\begin{aligned}
A &= V_a^2 \left\langle P^{-1}\alpha, P^{-1}\alpha\right\rangle - \left\langle P^{-1}\alpha, W\right\rangle^2 \\
B &= 2V_a^2 \left\langle P^{-1}\alpha, P^{-1}\beta\right\rangle - 2\left\langle P^{-1}\alpha, W\right\rangle \left(\left\langle P^{-1}\beta, W\right\rangle + 1\right) \\
C &= V_a^2 \left\langle P^{-1}\beta, P^{-1}\beta\right\rangle - \left[\left\langle P^{-1}\beta, W\right\rangle + 1\right]^2
\end{aligned}$$

The value $v_{\mathbf{x}_j\mathbf{x}_k}$ is then computed by the classical resolution formula of quadratic equation. To ensure that $v_{\mathbf{x}_j\mathbf{x}_k}$ is a good approximation of the value function, u, at the point X, the characteristic direction for the mesh point X needs to lie inside the simplex $(\mathrm{X}, \mathrm{X}_j, \mathrm{X}_k)$. The optimal trajectory is built by moving from the initial point to the destination point along the characteristic direction determined by:

$$\frac{d\mathrm{X}}{dt} = -V_a \frac{\nabla u(\mathrm{X})}{\|\nabla u(\mathrm{X})\|} + W(\mathrm{X}) \tag{10}$$

The computational complexity of this algorithm is $O(\gamma N \log N)$, where N is the number of mesh points and γ the anisotropy ratio (see [38]).

OBSTACLE AVOIDANCE→ We propose to decrease the speed of propagation of the wavefront in the parts of the environment corresponding to obstacles. In this way, we register a growth of the value function u that penalizes the passage through these areas. We define a map of values ξ as a function of the obstacles, $\mathcal{O}$. The values ξ are between 0 and an upper bound ξ_{max}. The scaled values ξ are then exploited in order to slow down the wavefront speed as follows: $(1-\epsilon)F(X, n)$. The maximum value ξ_{max} needs to be less than 1 to keep the wavefront speed strictly positive.

DL APPROACH→ The formalization of the problem proposed leads to the resolution of a Hamilton-Jacobi equation. Problems involving these kind of equations have been studied in several areas of mathematics, numerical computing and more recently in deep learning and several networks have been designed including [25], [24]. In this work, we do not pursue this direction, having decided to privilege convergence guarantees rather then speed.

IV. OPTIMAL CAMERA PLACEMENT

In this section, fixed a camera configuration $C = \{C_i\}_{i=1}^N$, we provide a measure for assessing how much effective this configuration is when it comes to preventing an object "in-transit" to go unnoticed. For sake of simplicity, we start assuming that the adversary has fixed starting position (x_0, y_0) and final destination (x_f, y_f). A fairly immediate generalization consists in averaging the optimality measure, defined below, on every pair of positions $((x_0, y_0), (x_f, y_f))$ that an adversary could take (or more realistically on a sampling of them). We define our objective so that it takes in account for the integral of the "portion" of the path, returned by applying the ordered upwind algorithm (from now on, called algorithm $\mathcal{A}$), that intersects at least one among the fields of views of the cameras. A detail to notice is that we normalize this quantity over the distance between the initial position and the end position, since this distance should not influence the complexity of an adversarial path. Therefore, for any parametrization $\gamma : [0, 1] \to \mathcal{R}$ of a valid path (i.e. a path avoiding obstacles), we define:

$$\mathcal{L}_{(p_i)_{1\le i\le N}}(\gamma) = \int_0^1 \mathbf{1}\left(\gamma(t) \in \bigcup_{1\le i\le N} \text{scope}(C_i)\right) \mathrm{d}t \tag{11}$$

where $\text{scope}(C_n) \subset \mathcal{R}$ is the set of points of R that a camera C_n has in its field of view; more formally, it is defined as:

$$\begin{aligned}
\text{scope}(C_i) = \Bigg\{ &(x, y) \in \mathcal{R} \text{ s.t.} \\
&\beta_i \le \arctan\left(\frac{y - b_i}{x - a_i}\right) \le \beta_i + \alpha_i \qquad (12) \\
&\text{and }](a_i, b_i), (x, y)[\cap \mathcal{O} = \emptyset \Bigg\}
\end{aligned}$$

where we have used the natural extension of $\arctan$ to $\overline{\mathbb{R}}$.

SIMULATED ANNEALING→ Given (11), the initial problem can be restated as follows:

$$\min_{(x_i, y_i, \theta_i)_{1\le i\le N}} \mathcal{L}(\mathcal{A}(\{(x_i, y_i, \theta_i)_i\})) \tag{13}$$

for N a given number of cameras. The optimization required by (13) cannot be performed using classical methods, e.g., gradient descent, because it's fairly complex to estimate gradients and several local minima are likely to appear in the resolution [23]. For this reason, we decided to use simulated annealing (SA) [6], an effective method for approximating global optima. The principle behind this technique was inspired by annealing in metallurgy and consists in proposing a new potentially optimizing candidate at each step and accepting always it if it scores better, with some probability p otherwise. In our framework, at each step, SA randomly selects one of the cameras n and sets its parameters randomly. The new configuration obtained C^{t+1} is then evaluated thanks to our optimality measure. If the new configuration is better than the previous one C^t, we accept the new configuration. If not, we still accept the new configuration with some probability. This is equivalent to accepting the new configuration with probability:

$$p(\text{accept}|C^t, C^{t+1}, T) = \min\{1, \exp(\frac{\mathcal{L}(\mathcal{A}(C^{t+1})) - \mathcal{L}(\mathcal{A}(C^t))}{T})\} \tag{14}$$

where T is called the annealing temperature, and is typically chosen high at the beginning and then decreased gradually. In this work, we adopt a linear schedule for T. The complete pseudocode for the algorithm is below.

```
procedure SA($T_0$)
    $C \leftarrow$ RANDOM()
    for $k = 0$ to $K$ do
        $T \leftarrow \max(0, T_0(1 - \frac{k+1}{K}))$
        $C_{\text{prop}} \leftarrow$ RANDOM()
        if $p(\text{accept}|C, C_{prop}, T) > \text{random}(0, 1)$ then
            $C \leftarrow C_{\text{prop}}$
        end if
    end for
    return $C$
end procedure
```

Concerning the optimality, SA is guaranteed to converge to the global optimum in a finite time if the candidates and the temperature satisfy some well-known weak conditions [39]. On a practical point of view, since the search-space is infinite, convergence to the global minimum may be slow, but results are still very satisfying and may be further improved by using local optimization methods.

V. Implementation

We provide an implementation of the method in Python that is freely available at `https://github.com/alexandreduplessis/Adversarial_Camera_Placement`. This includes both a tool allowing for 2D modeling of an environment with obstacles and cameras and the implementation of the placement algorithm proposed.

VI. Results

After a sequence of tests, the model developed seems to achieve fairly good performance.

A. Path planning

The implemented path planning method is the one described in III. We show the results in a simplified case, where the vector field w norm does not depend on the distance to the camera position, in order to show more easily the influence of weighing more the path length or the camera avoidance.

The results can be interpreted very intuitively, since the more we put weight on avoiding cameras (compared to minimizing the path length), the more the agent tries to spend less time in the camera scope, and thus goes closer to the camera position (see figures VI-A and VI-A).

Remark: Figures VI-A and VI-A can seem disturbing since the number of points seen by the camera does not seem to be an increasing function of the distance to the camera, but this is only due to the little size of the grid shown (10×10), and this affects neither the general behavior of the algorithm, nor the following camera placement.

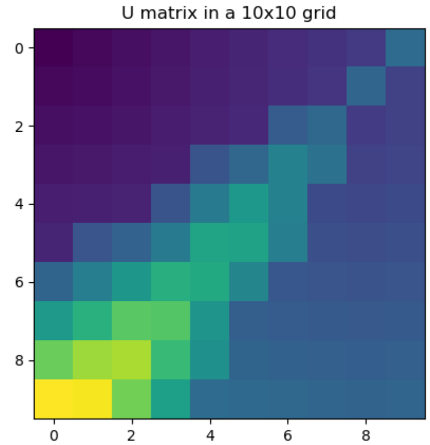

Fig. 1. U array for weak path length constraint

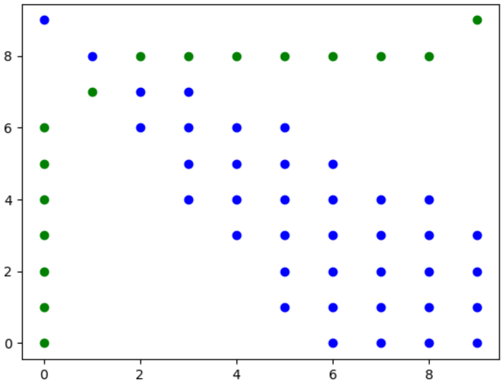

Fig. 2. Grid path visualization for weak path length constraint

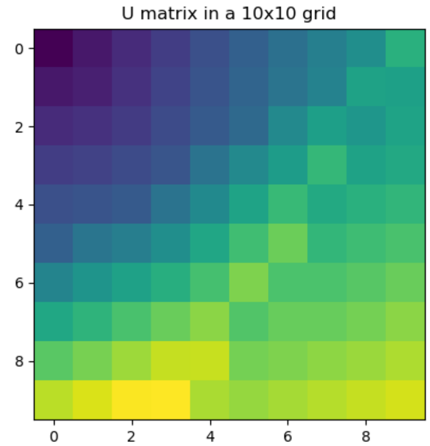

Fig. 3. U array for strong path length constraint

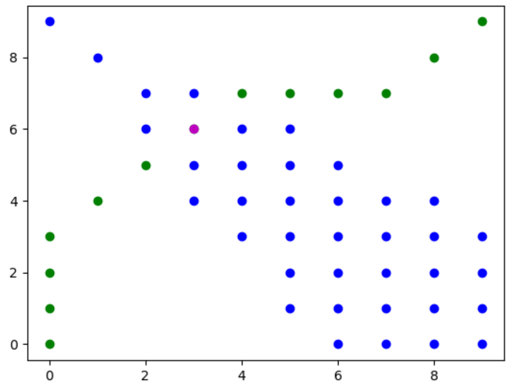

Fig. 4. Grid path visualization for strong path length constraint

Figure VI-A is a nicer visualization of the output of our algorithm for a 20 × 20 grid.

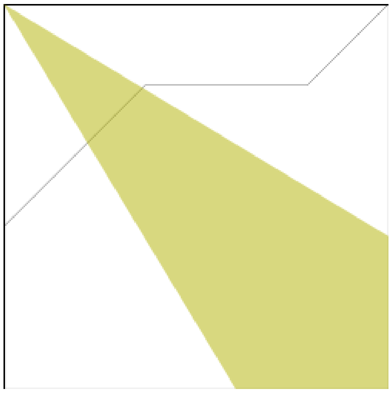

Fig. 5. Visualization of optimal path in a 20 × 20 grid

Remark An obvious remark (see VI-A) is that because of the fact that we only allow four moves from each position in the grid, so that straight lines in the continuous space that are not parallel to an axis or a bisector of them is achieved thanks to two straight lines, giving a longer path. A good improvement to the current implementation would thus be to post-replace the path by trying to directly join points not separated by a camera scope (if they do not intersect an obstacle). But it actually does not influence the camera placement algorithm for a fixed number of cameras.

B. Camera placement

The results of our simulated-annealing camera placement algorithm seem convincing. Despite slow computation, we get easily interpretable results.

Figure VI-B shows the optimal placement of one camera to prevent an agent going from bottom left corner to top right corner from avoiding the camera. Positioning the camera on the diagonal of the grid as found is in fact intuitively optimal (there are actually two global optima).

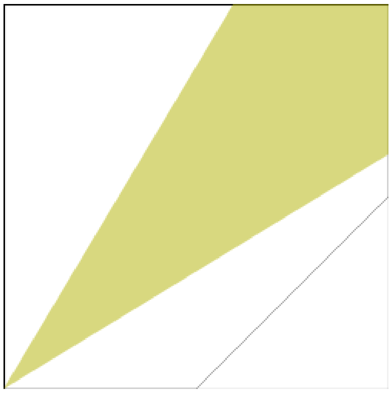

Fig. 6. Optimal placement of 1 camera and associated path

Figure VI-B shows the optimal placement of 2 cameras, and similarly gives intuitive results.

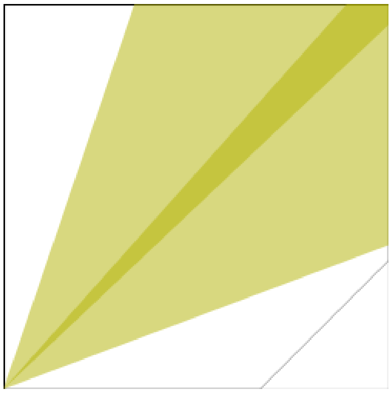

Fig. 7. Optimal placement of 2 cameras and associated path

Remark: Note that the results are dependent on the choice of the vector field w.

Finally note that obstacles are not currently used in experiments, but this should be managed soon.

VII. Conclusion

This work proposes a new multi-camera placement modeling set-up to support the design of network of visual sensor. The model allows for taking in account important constraints that are involved in the computer-vision applications operated on the cameras' recordings. In fact, the anisotropic speed field is easily customized by changing the definition of w (in equation 1), so that it explicitly accounts for resolution or any other desired property. The formalism reduce the camera placement problem to a coupled optimization that builds partially on prior work in the field of motion planning.

Limitations and Further Work→ This work presents a few limitations that we list below (in decreasing order of importance) to be addressed in future work. First of all, we haven't taken into account the triangulation constraint. This is often a requirement and can be easily obtained thanks to a modification of the SA algorithm. The second aspects that we haven't consider in our formalization was the possibility of adding cameras or modifying their parameters; in particular, it is pretty straightforward to introduce a joint optimization of the cameras parameters and their properties through a transdimentional SA [40]. The last limitation of our work is related to the speed of computation that is modest. However for this specific application, once fixed the environment computations are limited and results are satisfying.

References

[1] Chesnokov, Nicole. ”The Art Gallery Problem: An Overview and Extension to Chromatic Coloring and Mobile Guards.”(2018).

[2] O'Rourke, Joseph. ”Art gallery theorems and algorithms.” Oxford University Press, Inc., USA. (1987).

[3] Aigner, Martin and Ziegler, Gnter M. ”Proofs from THE BOOK.” Springer Publishing Company, Incorporated. (2009).

[4] Sreedevi, Indu & Mittal, Nikhil & Chaudhury, Santanu & Bhattacharyya, Asok. ”Camera Placement for Surveillance Applications”. (2011).

[5] Watras, Alex J., Jae-Jun Kim, Hewei Liu, Yu Hen Hu and Hongrui Jiang. ”Optimal Camera Pose and Placement Configuration for Maximum Field-of-View Video Stitching.” Sensors. (2018).

[6] Y. K. Hwang and N. Ahuja. ”A potential field approach to path planning.”. IEEE Transactions on Robotics and Automation, vol. 8, no. 1, pp. 23-32, doi: 10.1109/70.127236. (1992).

[7] Karaman, Sertac & Frazzoli, Emilio. Sampling-based Algorithms for Optimal Motion Planning. International Journal of Robotic Research - IJRR. 30. 846-894. 10.1177/0278364911406761. (2011).

[8] H. Nichols, M. Jimenez, Z. Goddard, M. Sparapany, B. Boots and A. Mazumdar. ”Adversarial Sampling-Based Motion Planning.” IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 4267-4274, doi: 10.1109/LRA.2022.3148464. (2022).

[9] Kularatne, Dhanushka & Bhattacharya, Subhrajit & Hsieh, M.. ”Optimal Path Planning in Time-Varying Flows Using Adaptive Discretization”. IEEE Robotics and Automation Letters. PP. 1-1. 10.1109/LRA.2017.2761939. (2017).

[10] B. Girardet, L. Lapasset, D. Delahaye and C. Rabut, ”Wind-optimal path planning: Application to aircraft trajectories,” 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV), Singapore, pp. 1403-1408, doi: 10.1109/ICARCV.2014.7064521. (2014).

[11] M. Otte, W. Silva and E. Frew. ”Any-time path-planning: Time-varying wind field + moving obstacles.”. IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016, pp. 2575-2582, doi: 10.1109/ICRA.2016.7487414. (2016).

[12] R. Bodor, P. Schrater and N. Papanikolopoulos, ”Multi-camera positioning to optimize task observability,” IEEE Conference on Advanced Video and Signal Based Surveillance, Como, Italy, 2005, pp. 552-557, doi: 10.1109/AVSS.2005.1577328. (2005).

[13] Konda, Krishna reddy & Conci, Nicola & Natale, Federica. ”Global Coverage Maximization in PTZ-Camera Networks Based on Visual Quality Assessment”. IEEE Sensors Journal. 16. 1-1. 10.1109/JSEN.2016.2584179. (2016).

[14] Yang, Yajue & Pan, Jia & Wan, Weiwei. A Survey of Optimal Motion Planning. IET Cyber-Systems and Robotics. 1. 10.1049/iet-csr.2018.0003. (2019).

[15] Olague, Gustavo & Mohr, Roger. ”Optimal Camera Placement for Accurate Reconstruction”. Pattern Recognition. 35. 927-944. 10.1016/S0031-3203(01)00076-0. (2002).

[16] Zhao, Jian & Haws, David & Yoshida, Ruriko & Cheung, Senching. ”Approximate Techniques in Solving Optimal Camera Placement Problems”. International Journal of Distributed Sensor Networks. 2013. 1705-1712. 10.1109/ICCVW.2011.6130455. (2011).

[17] J. Zhao, S. C. Cheung and T. Nguyen. ”Optimal Camera Network Configurations for Visual Tagging.”. IEEE Journal of Selected Topics in Signal Processing, vol. 2, no. 4, pp. 464-479, doi: 10.1109/JSTSP.2008.2001430. (2008).

[18] Bylow, Erik & Sturm, Jürgen & Kerl, Christian & Kahl, Fredrik & Cremers, Daniel. ”Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions.”. 10.15607/RSS.2013.IX.035. (2013).

[19] Gao, Mingfei & Yu, Ruichi & Li, Ang & Morariu, Vlad & Davis, Larry. ”Dynamic Zoom-in Network for Fast Object Detection in Large Images”. (2017).

[20] M. Melvasalo and V. Koivunen. ”Using Spectrum Maps for Surveillance Avoiding Path Planning.”.IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), pp. 1-5, doi: 10.1109/SPAWC.2018.8446033. (2018).

[21] Erdem, Ugur Murat and Stan Sclaroff. “Optimal Placement of Cameras in Floorplans to Satisfy Task Requirements and Cost Constraints.” (2004).

[22] J. -J. Gonzalez-Barbosa, T. Garcia-Ramirez, J. Salas, J. -B. Hurtado-Ramos and J. -d. -J. Rico-Jimenez, ”Optimal camera placement for total coverage,” 2009 IEEE International Conference on Robotics and Automation, Kobe, pp. 844-848, doi: 10.1109/ROBOT.2009.5152761. (2009).

[23] Rahimian, Pooya & Kearney, Joseph. ”Optimal camera placement for motion capture systems in the presence of dynamic occlusion”. 129-138. 10.1145/2821592.2821596. (2015).

[24] Kim, Jeongho & Shin, Jaeuk & Yang, Insoon. Hamilton-Jacobi Deep Q-Learning for Deterministic Continuous-Time Systems with Lipschitz Continuous Controls. (2020).

[25] Nakamura-Zimmerer, Tenavi & Gong, Qi & Kang, Wei. ”Adaptive Deep Learning for High-Dimensional Hamilton–Jacobi–Bellman Equations. SIAM Journal on Scientific Computing”. 43. A1221-A1247. 10.1137/19M1288802. (2021).

[26] A. A. Altahir et al..”Modeling Multicamera Coverage for Placement Optimization.”. IEEE Sensors Letters, vol. 1, no. 6, pp. 1-4, Art no. 5500604, doi: 10.1109/LSENS.2017.2758371. (2017).

[27] Li, Chang & Chen, Xi & Chai, Li. Coverage Optimization of Camera Network for Continuous Deformable Object. (2022).

[28] Erdem, Ugur Murat and Stan Sclaroff. “Optimal Placement of Cameras in Floorplans to Satisfy Task Requirements and Cost Constraints.” (2004).

[29] R. Bodor, P. Schrater and N. Papanikolopoulos, ”Multi-camera positioning to optimize task observability.” IEEE Conference on Advanced Video and Signal Based Surveillance, pp. 552-557, doi: 10.1109/AVSS.2005.1577328. (2005).

[30] M. S. S. Suresh, A. Narayanan and V. Menon, ”Maximizing Camera Coverage in Multicamera Surveillance Networks,” in IEEE Sensors Journal, vol. 20, no. 17, pp. 10170-10178, doi: 10.1109/JSEN.2020.2992076. (2020).

[31] S. Aghajanzadeh et al., ”Camera Placement Meeting Restrictions of Computer Vision,” 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 2020, pp. 3254-3258, doi: 10.1109/ICIP40778.2020.9190851.

[32] J. Gui, Z. Sun, Y. Wen, D. Tao and J. Ye, ”A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications.”. IEEE Transactions on Knowledge and Data Engineering, doi: 10.1109/TKDE.2021.3130191. (2021).

[33] Creswell, Antonia & White, Tom & Dumoulin, Vincent & Arulkumaran, Kai & Sengupta, Biswa & Bharath, Anil. Generative Adversarial Networks: An Overview. IEEE Signal Processing Magazine. 35. 10.1109/MSP.2017.2765202. (2017).

[34] S. Fleishman, D. Cohen-Or and D. Lischinski, ”Automatic camera placement for image-based modeling,” Proceedings. Seventh Pacific Conference on Computer Graphics and Applications (Cat. No.PR00293), Seoul, Korea (South), 1999, pp. 12-20, doi: 10.1109/PCCGA.1999.803344.

[35] Isler, Volkan & Kannan, Sampath & Daniilidis, Kostas & Valtr, Pavel. ”VC-Dimension of Exterior Visibility”. IEEE transactions on pattern analysis and machine intelligence. 26. 667-71. 10.1109/TPAMI.2004.1273987. (2004).

[36] B. Girardet, L. Lapasset, D. Delahaye and C. Rabut, ”Wind-optimal path planning: Application to aircraft trajectories,” 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV), pp. 1403-1408, doi: 10.1109/ICARCV.2014.7064521. (2014).

[37] Girardet, Brunilde & Lapasset, Laurent & Delahaye, Daniel & Rabut, Christophe & Brenier, Yohann. ”Generating optimal aircraft trajectories with respect to weather conditions”. (2013).

[38] Sethian, J.A., & Vladimirsky, A. (2003). Ordered Upwind Methods for Static Hamilton-Jacobi Equations: Theory and Algorithms. Proceedings of the National Academy of Sciences of the United States of America, 98 20, 11069-74.

[39] V. Granville, M. Krivanek and J. . -P. Rasson, ”Simulated annealing: a proof of convergence.”. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 6, pp. 652-656, doi: 10.1109/34.295910. (1994).

[40] Liu, Junbin & Sridharan, Sridha & Fookes, Clinton & Wark, Tim. ”Optimal Camera Planning Under Versatile User Constraints in Multi-Camera Image Processing Systems”. IEEE transactions on image processing : a publication of the IEEE Signal Processing Society. 23. 10.1109/TIP.2013.2287606. (2013).