J Glob Optim
DOI 10.1007/s10898-016-0452-2

Algorithms for art gallery illumination

Maximilian Ernestus[1] · Stephan Friedrichs[2,3] · Michael Hemmer[1] · Jan Kokemüller[1] · Alexander Kröller[1] · Mahdi Moeini[4] · Christiane Schmidt[5]

Received: 18 March 2015 / Accepted: 10 June 2016

Abstract The art gallery problem (AGP) is one of the classical problems in computational geometry. It asks for the minimum number of guards required to achieve visibility coverage of a given polygon. The AGP is well-known to be NP-hard even in restricted cases. In this paper, we consider the AGP with fading (AGPF): A polygonal region is to be illuminated with light sources such that every point is illuminated with at least a global threshold, light intensity decreases over distance, and we seek to minimize the total energy consumption. Choosing fading exponents of zero, one, and two are equivalent to the AGP, laser scanner applications, and natural light, respectively. We present complexity results as well as a negative solvability result. Still, we propose two practical algorithms for AGPF with fixed light positions (e.g.

✉ Stephan Friedrichs
sfriedri@mpi-inf.mpg.de

Maximilian Ernestus
maximilian@ernestus.de

Michael Hemmer
mhsaar@gmail.com

Jan Kokemüller
jan.kokemueller@gmail.com

Alexander Kröller
kroeller@perror.de

Mahdi Moeini
mahdi.moeini@wiwi.uni-kl.de

Christiane Schmidt
christiane.schmidt@liu.se

[1] TU Braunschweig, IBR, Algorithms Group, Mühlenpfordtstr. 23, 38106 Braunschweig, Germany

[2] Max Planck Institute for Informatics, Saarbrücken, Germany

[3] Saarbrücken Graduate School of Computer Science, Saarbrücken, Germany

[4] Chair of Business Information Systems and Operations Research (BISOR), Technical University of Kaiserslautern, Postfach 3049, Erwin-Schrödinger-Str., 67653 Kaiserslautern, Germany

[5] Communications and Transport Systems, ITN, Linköping University, Linköping, Sweden

Published online: 12 September 2016

vertex guards) independent of the fading exponent, which we demonstrate to work well in practice. One is based on a discrete approximation, the other on non-linear programming by means of simplex-partitioning strategies. The former approach yields a fully polynomial-time approximation scheme for the AGPF with fixed light positions. The latter approach obtains better results in our experimental evaluation.

Keywords Art gallery problem · Fading · Computational geometry · Linear program · Non-linear program · Lipschitz function · Algorithm engineering

Mathematics Subject Classification 65D18 · 68W25 · 68W40 · 68U05 · 90C05 · 90C30 · 90C90

1 Introduction

The classical art gallery problem (AGP) asks for the minimum number of guards required to cover boundary and interior of a polygon. This is one of the best-known problems in computational geometry; see the excellent book by O'Rourke [23] for an introduction to the subject. In the classical AGP guards have an infinite range of visibility. We consider a problem variant where light fades over distance. The main applications are discussed below.

1.1 Laser scanning

Consider obtaining an exact, two-dimensional laser scan of some indoor environment. A typical 360° laser scanner can be placed at any point. It operates by taking a sample in one direction then turning by a configurable angle θ and repeating the process. The time t to obtain a scan is roughly $t \sim \theta^{-1}$ and can range anywhere between seconds and several hours in real-world applications. The quality of the result depends on the sample-point density q on the walls. For an object at distance d this is about $q \sim d^{-1}$. Together, we get $q \sim td^{-1}$. The actual fading of the laser-light intensity is irrelevant in practice. The problem of minimizing the time spent scanning (with one scanner) while maintaining sufficient scanning quality is an AGP generalization with linear fading.

1.2 Realistic light

Light suffers from quadratic fading over distance. From a dimmable light bulb at brightness b an object at distance d receives light in an amount of $q \sim bd^{-2}$. We assume a linear correspondence between energy consumption and brightness. The AGP generalization of illuminating a polygonal area with the minimum total energy was introduced by O'Rourke in 2005 [12]. Since then only a restricted case of this problem has been addressed by Eisenbrand et al. [14].

1.3 Our contribution

We study two versions of the art gallery problem with fading (AGPF). One is the discrete version where guards can only be placed on finitely many a-priori known locations (e.g. vertex guards). The other version allows guards that can be positioned anywhere in the input polygon (point guards). We address the question of computational complexity as well as solvability of AGPF (both the discrete and the general variant).

Two algorithms for the AGPF with fixed guard positions constitute the core of our contribution. That is, for the case that the selected guard locations have to be chosen from a given, finite candidate-location set. Our algorithms are based on infinite LP formulations of the problem. Both solve the problem of finding the darkest point in an intermediate solution, referred to as the primal separation problem (PSP), in different ways. Our algorithms work for arbitrary fading exponents; they can be applied to the laser scanning as well as to the natural light scenario.

The first algorithm approximates the PSP by replacing continuous fading with an appropriate step function. For a finite guard candidate set, such as vertex guards, we illustrate that it yields a fully polynomial-time approximation scheme (FPTAS).

Our second algorithm solves the PSP by means of a derivative-free approach based on triangulation of the polygon's visibility coverage, followed by optimizing non-linear programs by using a simplex-partitioning procedure [24].

1.4 Overview

Related work is presented in Sect. 2. Section. 3 and 4 formally introduce the problem, and discuss fading functions and complexity results. In Sect. 5, we turn our attention to the case of fixed, finite sets of guard positions (e.g. vertex guards) and propose two algorithms, one of which yields an FPTAS. We evaluate our algorithms experimentally in Sect. 6. Section 7 gives a brief introduction to the general (point guard) case where guards can be placed anywhere in the polygon. We conclude this paper in Sect. 8.

2 Related work

Guarding problems have been studied for several decades. Chvátal [3] was the first to answer Victor Klee's question on the number of guards that are always sufficient and sometimes necessary to monitor a polygon P with n vertices; $\lfloor \frac{n}{3} \rfloor$ is the tight bound for general polygons which was soon afterwards shown by a beautiful proof of Fisk [16]. Many other polygon classes have been considered w.r.t. this kind of Art Gallery Theorems. For example, Kahn et al. [19] obtained a tight bound of $\lfloor \frac{n}{4} \rfloor$ for orthogonal polygons. Generally, the guards are allowed to be located anywhere in the polygon (point guards). Sometimes, less powerful guards have been considered, such as guards that must be placed on vertices of P (vertex guards), or guards that only have a certain illumination angle γ. For the latter problem, Tóth [25] showed an art gallery theorem with tight bound $\lfloor \frac{n}{3} \rfloor$ for $\gamma = 180^\circ$. While the above results give bounds on the number of guards in certain classes of polygons, the AGP asks for the minimum number of guards that cover a given polygon P. Several versions of this problem were shown to be NP-hard, for example the problem restricted to guards placed on vertices of simple polygons, as shown by Lee and Lin [22]. See [11,23] for surveys on the AGP.

The AGPF has only been considered for the case of fixed light-source positions in the plane and the illumination of a line (the stage). Eisenbrand et al. [14] aim at minimizing the total amount of power assigned to the given light sources such that the entire stage is lit. The authors give a convex programming formulation of the problem that leads to a polynomial-time solution. They present a constant-factor approximation and a PTAS.

Over the last few years, efficient implementations for the classical AGP emerged [6,8,15,20,21,26]. Currently, the state of the art is that optimal solutions for polygons with several thousand of vertices can be found efficiently. See [11] for a survey on that matter.

3 Notation and preliminaries

We consider a given polygon P, possibly with holes. P has vertices V, with $|V| = n$. The diameter of P is denoted $D := \operatorname{diam} P$. For a point $p \in P$ the *visibility region* $\mathcal{V}(p)$ is the (star-shaped) set of all points of P visible from p. In the original AGP, we say a *guard set* $G \subseteq P$ *covers* P if and only if $\bigcup_{g \in G} \mathcal{V}(g) = P$. The goal is to find a covering G of minimum cardinality.

In this paper, we consider a variant of this problem: We assume the guards $g \in G$ to be light sources whose intensity (and therefore energy consumption) can be controlled. We denote the intensity of $g \in G$ by $0 \leq x_g \in \mathbb{R}$. Furthermore, light suffers from fading over distance, modeled by a fading function ϱ: While a point $w \notin \mathcal{V}(g)$ does not receive any light from g, $w \in \mathcal{V}(g)$ receives $\varrho(g, w)x_g$, i.e., ϱ accounts for the possible fading exponents motivated in Sect. 1.

For a set $W \subseteq P$ of *witnesses* (points that witness sufficient or insufficient illumination), we say a guard set $G \subseteq P$ with intensities $x \in \mathbb{R}^G_{\geq 0}$ *illuminates* W if and only if every point in W is sufficiently lit. Without loss of generality, we normalize the meaning of $w \in W$ being *sufficiently lit* to

$$\sum_{g \in G \cap \mathcal{V}(w)} \varrho(g, w)x_g \geq 1 \ . \tag{1}$$

Given a set of possible light sources G, a set W of witness points, and a fading function ϱ, our objective is to illuminate W using minimum total energy $\sum_{g \in G} x_g$. We call this problem art gallery problem with fading, AGPF $\varrho(G, W)$. When ϱ is clear from context, we occasionally write AGPF(G, W) instead. In general, we are interested in illuminating all of P. Therefore, the relevant cases are AGPF(G, P)—where possible guard positions are given, e.g. AGPF(V, P) is the common discretization with vertex guards—and AGPF(P, P), the *continuous/point guard* variant where guards can be placed anywhere in P. AGPF $\varrho(G, W)$ can be formulated as a linear program (LP):

$$\min \quad \sum_{g \in G} x_g \tag{2}$$

$$\text{s. t.} \quad \sum_{g \in G \cap \mathcal{V}(w)} \varrho(g, w)x_g \geq 1 \quad \forall w \in W \tag{3}$$

$$x_g \geq 0 \qquad \forall g \in G \ . \tag{4}$$

It should be noted that AGPF(P, P) results in an LP with an infinite number of both variables and constraints. Kröller et al. [20] were the first to show how to solve this type of LP for the original AGP with fractional guards, i.e., for $\varrho(\cdot, \cdot) = 1$.

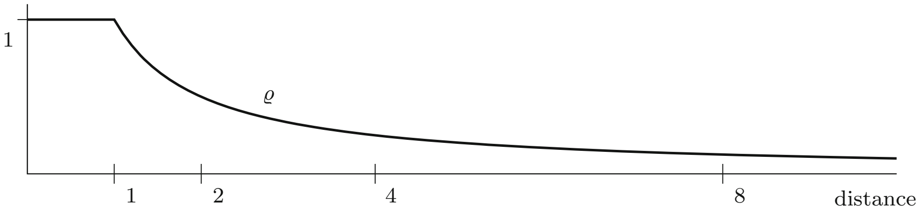

Fig. 1 Fading function ϱ

4 Fading function and complexity

4.1 Fading function

For a fading exponent $\alpha \in \mathbb{R}_{\geq 0}$, we define the fading function $\varrho(g, w)$ by

$$\varrho(g, w) := \begin{cases} \|g - w\|^{-\alpha} & \text{if } \|g - w\| \geq 1 \text{ and} \\ 1 & \text{otherwise,} \end{cases} \tag{5}$$

compare Fig. 1. Here $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^2$. As motivated above, the amount of light received by a point w from the light source g is $\varrho(g, w)x_g$ with $\alpha = 2$ for natural light, $\alpha = 1$ for laser scanners, and $\alpha = 0$ for the classical AGP.

Our motivation for capping ϱ at 1 is two-fold: (1) Without this restriction, any non-zero guard provides an infinite amount of light to itself. This results in the "optimal solution" of AGPF(P, P) to consist of every point in P glowing with an infinitesimally small brightness. This problem is not well-defined, see Lemma 1. (2) In an application, guards correspond physical objects like light bulbs or scanners. These objects are not point-shaped; they have a physical size. We are interested in obtaining coverage for the environment of the light source rather than for the light source itself. Hence, we assume that the input is scaled such that the radius 1 around a guard falls completely into the object size. That motivates capping ϱ at 1, even in the discrete version.

Lemma 1 *Let $\tilde{\varrho}(g, w) := \|g - w\|^{-\alpha}$, $\alpha > 0$, be an alternate fading function to $\varrho(g, w)$ in Eq. (5). Then* AGPF $\tilde{\varrho}(P, P)$ *has no optimal solution.*

Proof We show that for any finite $G \subset P$ any illumination $x \in \mathbb{R}^G_{\geq 0}$ either is infeasible or not optimal because it can be improved by using more guards $G' \supset G$. The dual of AGPF $\tilde{\varrho}(P, P)$ is

$$\max \quad \sum_{w \in P} y_w \tag{6}$$

$$\text{s. t.} \quad \sum_{w \in \mathcal{V}(g)} \tilde{\varrho}(g, w) y_w \leq 1 \quad \forall g \in P \tag{7}$$

$$y_w \geq 0 \qquad \forall w \in P \ , \tag{8}$$

the problem of fractionally packing bright witnesses in P such that no guard receives too much light.

Assume we are given a feasible, optimal solution x^* for AGPF $\tilde{\varrho}(P, P)$ that only uses a finite number of guards. By feasibility, there exists a guard $g \in P$ with $x^*_g > 0$. Moreover, by optimality, there exists a witness $w_p \in W$ with

$$\sum_{g \in G \cap \mathcal{V}(w_p)} \varrho(g, w_p) x_g^* = 1 ,$$

otherwise we could reduce the value of some x_g^* while preserving feasibility. Since Constraint (3) holds with equality for w_p, we have $y_{w_p} > 0$ in the dual. Now consider a point $g \in P$ converging towards w_p. As $y_{w_p} > 0$ and $\tilde{\varrho}(g, w_p) \to \infty$ as $g \to w_p$, at some point

$$\sum_{w \in W \cap \mathcal{V}(g)} \tilde{\varrho}(g, w) y_w \geq \tilde{\varrho}(g, w_p) y_{w_p} > 1$$

must hold. A point g fulfilling this inequality indicates a violated Constraint (7). Hence, y is dually infeasible, so x^* is not optimal, contradicting our assumptions. □

This shows that whenever G is finite, we can find an additional guard that improves the solution. Consequently, AGPF $\tilde{\varrho}(P, P)$ only possesses "optimal solutions" with infinitely many guards. In fact, it is easy to see that the "optimal" guard positions G are topologically dense in P, using the above ε-neighborhood argument. Then, the solution values can be scaled down arbitrarily since $\tilde{\varrho}(g, w)$ approaches ∞ as $w \to g$. This way, we can construct a series of solutions of strictly decreasing values, where G is infinite and the values of all guards converge to 0. Obviously, the limit of this process is $x \equiv 0$ which is not feasible.

Note that while we cap ϱ at 1, Eisenbrand et al. do not employ capping in [14]: Given a finite set of guard candidates S, they illuminate a stage L. The guard candidates usually are not located on the stage. Therefore, no point infinitesimally close to a guard needs to be covered. In addition, even if guard candidates were located on L, the set S is not dense in L. Hence, they never experience the situation described by Lemma 1.

4.2 Complexity results

We draw two conclusions regarding the complexity of the AGPF:

Lemma 2 *For a finite set $G \subset P$, finding integer solutions to* AGPF(G, P) *is NP-hard. Finding integer solutions for* AGPF(P, P) *is APX-hard.*

Proof For $\alpha = 0$, finding integer solutions corresponds to solving the original AGP. The corresponding hardness results [13,22] apply. □

It is tempting to solve the AGPF by reducing it to the AGP. Unfortunately, this approach results in an approximation factor which is exponential in α:

Lemma 3 *The AGPF cannot be approximated by a better factor than 2^α by first solving AGP and computing intensities afterwards.*

Proof Consider the polygon shown in Fig. 2 with a spike of length s. The optimal covering guard set consists of one guard (⋆). Illuminating the polygon with it requires an intensity

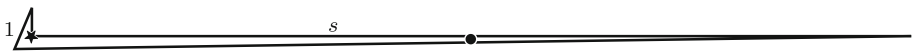

Fig. 2 An optimal guard placement for the AGP (*star symbol*) is not optimal for the AGP with fading (*star symbol* and *bullet symbol*)

of s^α. An optimal illumination uses two lights ($\star$ and $\bullet$) with intensities 1 and about $\left(\frac{1}{2}s\right)^\alpha$. Therefore, the approximation factor of the first solution is at least

$$\frac{s^\alpha}{1+\left(\frac{1}{2}s\right)^\alpha}$$

which converges to 2^α for large s. □

Note that this result holds for AGPF(V, P), as we can simply integrate a small bend in the lower edge with a reflex vertex in the height of $\bullet$. Again, the optimal solution to the AGP restricted to vertex guards would be $\star$. Two lights at $\star$ and the new reflex vertex at $\bullet$ result in the same bound on the approximation factor from Lemma 3.

5 Algorithms for fixed (vertex) guards

In this section, we turn our attention to AGPF(G, P), the AGPF with a fixed, finite set of guard candidates instead of its point guard sibling AGPF(P, P). The reasoning behind that decision is two-fold:

1. Although both, the AGP with point guards and the AGP with vertex guards, are NP-hard [22], many papers identified the vertex guard variant of the AGP to be much easier solvable in terms of practical algorithms than the point guard variant [4,7,11,20]. Since the AGPF is closely related to the AGP and our algorithms stem from our experience with the previous algorithms, see [11] for an overview, we start our investigation by tackling the discrete version.
2. The hardness in terms of tractability by practical algorithms of the AGPF lies in finding the darkest point in intermediate solutions (in this paper referred to as primal separation problem (PSP)). Generating good/optimal guard locations has emerged to be the hardest part of the AGP with point guards [11]. So we refrain from entwining the PSP and guard generation to obtain meaningful insights on the tractability of the (novel) PSP.

Staying in line with previous work and references therein, see [11] for an overview, it is safe to assume $G = V$. Our algorithms, however, only require $\bigcup_{g \in G} \mathcal{V}(g) = P$, i.e., that the instance is feasible at all.

This section proposes two algorithms for AGPF(G, P). One of them, see Sect. 5.1, replaces the continuous fading function ϱ by an appropriately chosen step function and yields an FPTAS for AGPF(G, P). The other one, in Sect. 5.2, works directly on the continuous fading function ϱ and uses simplex partitioning to identify the darkest point for a given illumination scheme.

5.1 Discrete approximation algorithm and FPTAS

In this section, we introduce DISCRETE, a discretized approximation algorithm for AGPF(G, P) that yields an FPTAS. We achieve this by replacing the continuous fading function ϱ with a step function τ, compare Fig. 3. For a fixed $\varepsilon > 0$, τ is defined as follows:

$$\tau(g, w) := \lfloor\!\lfloor \varrho(g, w) \rfloor\!\rfloor_{1+\varepsilon}, \tag{9}$$

where $\lfloor\!\lfloor \cdot \rfloor\!\rfloor_b$ denotes the hyperfloor with base b, i.e., $\lfloor\!\lfloor x \rfloor\!\rfloor_b := \max\{b^z : \ b^z \leq x, \ z \in \mathbb{Z}\}$. Therefore, τ is piecewise constant and approximates ϱ, i.e.,

$$1/(1+\varepsilon)\,\varrho < \tau \leq \varrho\ . \tag{10}$$

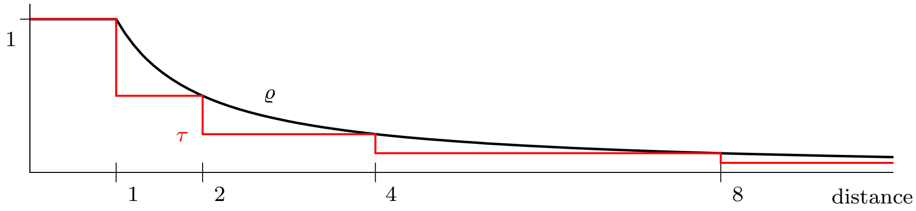

Fig. 3 Fading function ϱ and step function τ for $\varepsilon = 1$

We can use AGPF$\tau(G, W)$ to obtain a $(1 + \varepsilon)$-approximation for AGPF(G, W):

Lemma 4 *A feasible solution for* AGPF$\tau(G, W)$ *is also feasible for* AGPF $\varrho(G, W)$. *An optimal solution for* AGPF$\tau(G, W)$ *is a* $(1 + \varepsilon)$*-approximation for* AGPF $\varrho(G, W)$.

Proof The proof follows from Inequality (10):

1. Since $\varrho \geq \tau$, a solution fulfilling inequalities of type (3) with τ, does so too using ϱ. Hence every AGPF$\tau(G, W)$ solution also solves AGPF $\varrho(G, W)$.
2. Consider an optimal solution x^* of AGPF $\varrho(G, W)$ with objective value OPT. It follows from (10) that $x_\tau := (1 + \varepsilon)x^*$, i.e., scaling every guard with a factor $1 + \varepsilon$, is a feasible solution to AGPF$\tau(G, W)$. x_τ has objective value $(1 + \varepsilon)$ OPT, yielding the claimed approximation factor.

Note that our arguments also hold for $G = W = P$. □

5.1.1 An algorithm for AGPF$_\tau$

To solve AGPF$_\tau(G, P)$, we consider the illumination function

$$T_{G,x}(p) := \sum_{g \in G \cap \mathcal{V}(p)} \tau(g, p) x_g \,, \tag{11}$$

which defines the amount of light received at any point $p \in P$ w.r.t. to the fading function τ, given a solution $x \in \mathbb{R}^G_{\geq 0}$. x is feasible if and only if $T_{G,x} \geq 1$. It is easy to see that, for a single guard g, $T_{\{g\},x}$ is a piecewise constant function over P, as shown in Fig. 4. It induces an arrangement $\mathcal{A}(\{g\})$ within P with faces of constant value. P is split into two parts: One is $P \setminus \mathcal{V}(g)$, where $T_{\{g\},x}$ is constant 0. For the remaining part, note that τ is a monotonically decreasing step function taking the values $(1 + \varepsilon)^{-z}$ for $z = 0, 1, 2, \ldots$, each at distances $((1 + \varepsilon)^{z-1}, (1 + \varepsilon)^z]$ from g (exception: $[0, 1]$ for $z = 0$). This introduces $\mathrm{O}(\log_{1+\varepsilon} D)$ concentric bands of equal coverage value around g. Therefore, this part of the arrangement is the intersection of concentric circles with $\mathcal{V}(g)$.

Now consider the arrangement $\mathcal{A}(G)$ defined by overlaying the individual arrangements for all $g \in G$. Then, $T_{G,x}$ is constant over each face, edge, and vertex of this arrangement. This arrangement has a bounded complexity:

Lemma 5 *The complexity of* $\mathcal{A}(G)$ *is* $\mathrm{O}((n + \log_{1+\varepsilon} D)^2 |G|^2)$.

Proof The arrangement $\mathcal{A}(G)$ is constructed from

- n straight line segments defining P,
- $\mathrm{O}(n)$ straight line segments defining $\mathcal{V}(g)$ per $g \in G$, and

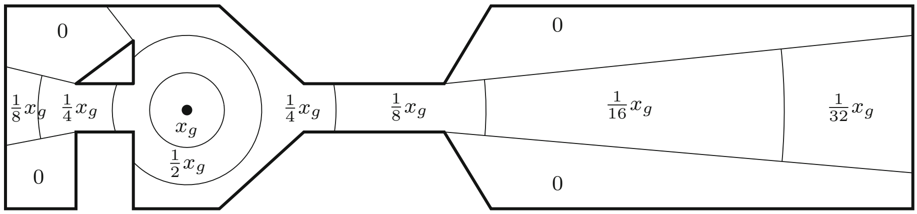

Fig. 4 The visibility arrangement $\mathcal{A}(\{g\})$ induced by $T_{\{g\},x}$ for a point g with value x_g, assuming $\alpha = 1$

- O($\log_{1+\varepsilon} D$) circles per $g \in G$,

therefore, O($(n + \log_{1+\varepsilon} D)|G|$) straight line segments and circles in total. Squaring this figure accounts for intersections in $\mathcal{A}(G)$. □

We are ready to present the overall algorithm (referred to as DISCRETE):

Step 0 (Visibility Arrangement). First compute the overlay of all guards' visibility step-functions, $\mathcal{A}(G)$. It has polynomial complexity by Lemma 5.
Step 1 (Witness Points). Place one witness w in each feature (vertex, edge, and face interior of $\mathcal{A}(G)$). Let W denote the set of these witnesses, which is polynomial in cardinality.
Step 2 (Solve LP). Solve AGPF$\tau(G, W)$, which is possible in polynomial time. The solution x is feasible and optimal for AGPF$\tau(G, P)$.

AGPF$\tau(G, P)$, our original LP, has a finite number of variables, and an infinite number of constraints since every point in P has to be illuminated. But we can exploit that $T_{G,x}$ is constant in each feature of the overlay in our LP formulation: An arbitrary witness point in an arrangement feature f is sufficiently lit if and only if all of f is sufficiently lit as well. Hence, we can solve the finite LP AGPF$\tau(G, W)$, where W contains one witness point in every feature of $\mathcal{A}(G)$.

Putting everything together, we get

Theorem 1 AGPF(G, P) *admits a fully polynomial-time approximation scheme.*

Proof Given an $\varepsilon > 0$, the arrangement $\mathcal{A}(G)$ has polynomial complexity due to Lemma 5 and the fact that $\log_{1+\varepsilon} D$ is polynomially bounded in the encoding size of ε and P. Hence the LP AGPF$\tau(G, W)$ in the discrete algorithm can be solved in polynomial time. According to Lemma 4, the result is a $(1 + \varepsilon)$-approximation to AGPF(G, P). □

5.1.2 Implementation issues and approximation of circular arcs

Although $\mathcal{A}(G)$ and W are polynomial in size, they become very large from a practical point of view. Therefore, it is more efficient to solve this with the technique of iterative primal separation (compare [9,11,20] for the classical AGP), i.e., to leave out some constraints of the LP and start with a small set W, then solve AGPF$\tau(G, W)$, and then check the $\mathcal{A}(G)$ for insufficiently lit points. If no such points exist, the solution is feasible. Otherwise, we add representative witness points for insufficiently lit features to W and re-iterate the process until the solution is feasible. We know this approach terminates because in each iteration at least one new feature of $\mathcal{A}(G)$ is covered.

It should be noted that by introducing circular arcs in the arrangement $\mathcal{A}(G)$, we have to deal with irrational numbers—to be precise, with algebraic numbers of degree 2. This is known to result in computationally expensive operations. Hence, we can replace the circular arcs by regular octagons. They still have irrational coordinates: For example, consider a regular octagon centered at the origin with its vertices on a circle of radius r. It has a vertex at $r(\cos\frac{\pi}{4}, \sin\frac{\pi}{4}) = r(1/\sqrt{2}, 1/\sqrt{2})$. However, intersection tests in the arrangement only use linear functions, which are expected to be less expensive. The octagon-based approach somewhat reduces the approximation guarantees—octagons only approximate circles—but still yields an FPTAS. However, our experimental evaluation shows that using the exact support for circular arcs of CGAL [2,10] outperforms the octagon-based approach.

5.2 Continuous optimization approach

In this section, we introduce Continuous, an alternative algorithm for AGPF(G, P). Continuous directly uses the continuous fading function ϱ instead of an approximate step function. It iteratively solves AGPF $\varrho(G, W)$, starting with $W = \emptyset$ and checks if its solution is feasible for AGPF(G, P). If a point $w \in P \setminus W$ that currently does not receive sufficient light can be found, it is added to W; the process is repeated until P is sufficiently lit.

This requires a subroutine for finding insufficiently lit points and can obviously be answered by a routine that finds the darkest point. Hence, we seek an algorithm solving the *primal separation problem (PSP),* i.e., given a current solution x, for finding:

$$\operatorname*{arg\,min}_{w\in P} T_{G,x}(w) = \operatorname*{arg\,min}_{w\in P} \sum_{g\in G\cap\mathcal{V}(w)} \varrho(g, w)x_g \,. \tag{12}$$

For the sake of presentation, we simplify $T(w) := T_{G,x}(w)$.

We solve the PSP in two steps. First, we triangulate the overlay of all visibility polygons $\mathcal{V}(g)$ for all $g \in G$ which yields a collection of triangles, each of which seen by a fixed subset of guards. In Sect. 5.2.1, we present an algorithm to solve (12) in such a triangle $\mathcal{S} \subseteq P$. The algorithm requires a function ℓ that provides a lower bound on the minimum brightness of $T_{G,x}$ in $\mathcal{S}$, i.e., $\ell(\mathcal{S}) \leq T(p)\ \forall p \in \mathcal{S}$. We discuss two such functions in Sects. 5.2.2 and 5.2.3.

5.2.1 Simplex partitioning algorithm

Continuous employs a global optimization approach based on simplex (here: triangle) partitioning [24] to solve the PSP. It iteratively keeps track of

- $\mathcal{S}_0$, the triangle to be searched for a darkest spot (the input),
- $k = 1, 2, 3, \ldots$, the iteration counter,
- $\mathcal{U}_k$, a set of triangles with $\bigcup_{\mathcal{S}\in\mathcal{U}_k} \mathcal{S} \subseteq \mathcal{S}_0$ (the remaining search region),
- $\mathcal{S}_k$, the next triangle to be partitioned,
- x_k^*, the darkest point found so far (the incumbent solution), and
- β_k, the current lower bound on the minimum brightness in $\mathcal{S}_0$.

Step 0 (Initialization). Set $\mathcal{U}_0 = \{\mathcal{S}_0\}$ as the current set of triangles. Set the current best solution to $x_0^* = \arg\min_{v\in V(\mathcal{S}_0)} T(v)$ (the darkest vertex of $\mathcal{S}_0$). Compute a lower bound $\beta_0 = \ell(\mathcal{S}_0)$ for the minimum brightness in $\mathcal{S}_0$.

In each iteration $k \in \mathbb{N}$, we do the following steps:

Step 1 (Eliminating bright triangles). Let $\mathcal{R}_k$ be the collection of triangle candidates remaining after discarding all triangles with $\ell(\mathcal{S}) > T(x_{k-1}^*)$ from $\mathcal{U}_{k-1}$, i.e., after removing the triangles that cannot contain a darkest point due to their lower bound.

Step 2 (Selecting a triangle). Consider the current collection of triangles $\mathcal{R}_k$ and select

$$\mathcal{S}_k = \arg\min_{\mathcal{S}\in\mathcal{R}_k} \ell(\mathcal{S}), \tag{13}$$

the triangle with the smallest lower bound, to be partitioned.
Step 3 (Refining the search region). Divide $\mathcal{S}_k$ into two smaller triangles by applying a bisection on its longest edge such that

$$\mathcal{S}_k = \mathcal{S}_{k_1} \cup \mathcal{S}_{k_2}. \tag{14}$$

Step 4 (Updating the incumbent). At this point, we obtain the search region

$$\mathcal{U}_k = (\mathcal{R}_k \setminus \{\mathcal{S}_k\}) \cup \{\mathcal{S}_{k_1}, \mathcal{S}_{k_2}\}. \tag{15}$$

Evaluate T at the common new vertex v_k of the triangles $\mathcal{S}_{k_1}$ and $\mathcal{S}_{k_2}$. Update

$$x_k^* = \arg\min_{v\in\{v_k, x_{k-1}^*\}} T(v), \tag{16}$$

$$\beta_k = \min_{\mathcal{S}\in\mathcal{U}_k} \ell(\mathcal{S}). \tag{17}$$

Then increase k and continue at Step 1.
Stopping criterion. Stop when there is no triangle to be partitioned. Alternatively, stop after a certain number of iterations or when a δ-estimation of the optimal solution is found, i.e., when $T(x_k^*) \leq \beta_k + \delta$.

We propose two approaches for computing the lower bounds, i.e., $\ell(\cdot)$, in Sects. 5.2.2 and 5.2.3. The first exploits geometric properties and the second the fact that T is Lipschitzian.

5.2.2 Geometric lower bound

Regarding a lower bound used to prune triangles in the simplex-partitioning algorithm in Sect. 5.2.1, observe that $\varrho(g, w)$ monotonically decreases with increasing $\|g - w\|$, i.e., light decreases with increasing distance. Fix a triangle $\mathcal{S}$, a finite set of guards G scattered in P, and $x \in \mathbb{R}^G_{\geq 0}$. According to the definition of $\varrho(g, w)$, $T_{\{g\},x}$ attains its minimum at a vertex of $\mathcal{S}$. Therefore,

$$\ell(\mathcal{S}) := \sum_{g\in G} \min_{p \text{ vertex of } \mathcal{S}} T_{\{g\},x}(p) \tag{18}$$

is a lower bound for $\min_{p\in\mathcal{S}} T_{G,x}(p)$.

5.2.3 Lipschitz lower bound

As an alternative approach to the geometric estimation of the lower bound, one may use the analytical properties of T. For example, if T is a Lipschitzian with a known Lipschitz constant L, then one can use the following result (see [24]):

Lemma 6 *Let $V(S) = \{v_0, v_1, \ldots, v_m\}$ be the vertex set of the simplex $\mathcal{S}$. Let f be a Lipschitzian on $\mathcal{S}$ with the Lipschitz constant L. Denote by z_j the function values $T(v_m)$ (for $j = 0, \ldots, m$), then we have:*

$$z^* \geq \frac{1}{m+1}\left(\sum_{j=0}^{m} z_j - L \max_{0\leq i\leq m} \sum_{j=0}^{m} \|v_i - v_j\|\right) \tag{19}$$

Proof The proof can be found in [24]; however, for the sake of completeness, we provide the complete proof. Let x be an arbitrary point of the simplex $\mathcal{S}$. Since T is a Lipschitz function (see Lemma 7)

$$|T(x) - z_j| \leq L\|x - v_j\|, \quad j = 0, 1, \ldots, m. \tag{20}$$

We know that this inequality is true for any point in $\mathcal{S}$, particularly for the optimal point x^*; consequently, we have

$$z^* = T(x^*) \geq z_j - L\|x^* - v_j\|, \quad j = 0, 1, \ldots, m.$$

By summing up these inequalities, we arrive at the following inequality

$$z^* \geq \frac{1}{m+1}\left(\sum_{j=0}^{m} z_j - L\sum_{j=0}^{m}\|x^* - v_j\|\right). \tag{21}$$

Due to the fact that x^* is unknown, we need to provide an estimation of the second sum in (21). Define the function Q as follows:

$$Q(x) := \sum_{j=0}^{m}\|x - v_j\|: \quad x \in \mathcal{S}.$$

Since $\mathcal{S}$ is a convex set and Q is a convex function, the maximum of Q is attained at one of the vertices of $\mathcal{S}$. Consequently,

$$\sum_{j=0}^{m}\|x^* - v_j\| \leq \max_{0\leq i\leq m}\sum_{j=0}^{m}\|v_i - v_j\|.$$

Using this inequality in (21) completes the proof. □

Fortunately, this bound can be used in the PSP:

Lemma 7 *T is Lipschitzian.*

Proof We provide the proof for $\alpha = 2$. The other cases can be proved in a similar way. For the sake of simplicity in the notation, let us define for any i (such that $i = 1, \ldots, m$):

- $\varrho_i(w) := \varrho(g_i, w)$.
- $(x_0, y_0) := (x^i, y^i)$ as the coordinates of the point g_i.
- (x, y) as the coordinates of the witness point w and (x_1, y_1) as the coordinates of the witness point w_x and (x_2, y_2) as the coordinates of the witness point w_y (where (through the primal-dual procedure) w_x and w_y are the potential points that will be chosen to be added into the set of the witness points).

Thus

$$\varrho_i(w) := \begin{cases} \frac{1}{\sqrt{(x-x_0)^2+(y-y_0)^2}} & \text{if } \|w - g_i\| \geq 1, \\ 1 & \text{otherwise.} \end{cases} \tag{22}$$

We want to show that $\exists L > 0$ s.t. $\forall w_x, w_y : |\varrho_i(w_x) - \varrho_i(w_y)| \leq L\|w_x - w_y\|$. We need to consider three different cases:

(i) $\|w_x - g_i\| \geq 1$ and $\|w_y - g_i\| \geq 1$,
(ii) $\|w_x - g_i\| < 1$ and $\|w_y - g_i\| < 1$, and

(iii) $\|w_x - g_i\| \geq 1$ and $\|w_y - g_i\| < 1$.

Case (i): In this case, the following definition of $\varrho_i(\cdot)$ is applied:

$$\varrho_i(w) = \frac{1}{\sqrt{(x-x_0)^2 + (y-y_0)^2}} : w \in \{w_x, w_y\}.$$

It is sufficient to show that the partial derivatives of $\varrho_i(w)$ are bounded by a positive number L. The partial derivatives of $\varrho_i(w)$ are computed as follows

$$\frac{\partial}{\partial x}\varrho_i(w) = \frac{x_0 - x}{[(x-x_0)^2 + (y-y_0)^2]^{3/2}},$$

and

$$\frac{\partial}{\partial y}\varrho_i(w) = \frac{y_0 - y}{[(x-x_0)^2 + (y-y_0)^2]^{3/2}}.$$

We will show that $|\frac{\partial}{\partial x}\varrho_i(w)| \leq L$ (where $L > 0$ is a constant). The proof of the other case, i.e., $|\frac{\partial}{\partial y}\varrho_i(w)| \leq L$, is quite similar.

Since (x, y) is located within a triangle of vertices with finite coordinates, we have $\|x\| < \infty$ and $\|y\| < \infty$. Furthermore, $\sqrt{(x-x_0)^2 + (y-y_0)^2} \geq |x - x_0| \geq 0$, and, hence, $[\sqrt{(x-x_0)^2 + (y-y_0)^2}]^3 \geq |x - x_0|^3$. This yields:

$$0 \leq \frac{1}{[\sqrt{(x-x_0)^2 + (y-y_0)^2}]^3} \leq \frac{1}{|x-x_0|^3}.$$

Consequently, we have:

$$0 \leq \frac{|x-x_0|}{[\sqrt{(x-x_0)^2 + (y-y_0)^2}]^3} \leq \frac{1}{|x-x_0|^2}.$$

The latter inequality reads as follows:

$$0 \leq |\frac{\partial}{\partial x}\varrho_i(w)| \leq \frac{1}{|x-x_0|^2}.$$

Because $|x - x_0| \geq 1$ we have $\frac{1}{|x-x_0|^2} \leq 1$. Altogether, we showed that $|\frac{\partial}{\partial x}\varrho_i(w)| \leq 1$. In a similar way, one can show that $|\frac{\partial}{\partial y}\varrho_i(w)| \leq 1$. These inequalities prove that $\varrho_i(w)$ is a Lipschitz function.

Case (ii): In this case, the function $\varrho_i(w)$ is a constant function; consequently, it is a Lipschitzian.

Case (iii): This case corresponds to

$$\|w_x - g_i\| \geq 1 \text{ and } \|w_y - g_i\| < 1$$

and we want to show that

$$\exists L > 0 \quad \text{s.t. } |\varrho_i(w_x) - \varrho_i(w_y)| \leq L\|w_x - w_y\|.$$

As $\|w_y - g_i\| < 1$, Eq. (22) yields $\varrho_i(w_y) = 1$; in addition, we have:

$$\varrho_i(w_x) = \frac{1}{\sqrt{(x_1-x_0)^2 + (y_1-y_0)^2}}.$$

The triangle inequality yields:

$$\|w_x - w_y\| = \|w_x - g_i + g_i - w_y\| \geq \|w_x - g_i\| - \|g_i - w_y\|.$$

Using $\|w_y - g_i\| < 1$ and $\|w_x - g_i\| \geq 1$, we obtain:

$$\|w_x - w_y\| > \|w_x - g_i\| - 1 \geq 0.$$

Because $\|w_x - g_i\| \geq 1$ holds, we can conclude:

$$\frac{\|w_x - g_i\| - 1}{\|w_x - g_i\|} < \frac{\|w_x - w_y\|}{\|w_x - g_i\|} \leq \|w_x - w_y\|.$$

To sum up:

$$\frac{\|w_x - g_i\| - 1}{\|w_x - g_i\|} < \|w_x - w_y\|. \tag{23}$$

In addition, we have:

$$|\varrho_i(w_x) - \varrho_i(w_y)| = \left|\frac{1}{\|w_x - g_i\|} - 1\right| = \left|\frac{1 - \|w_x - g_i\|}{\|w_x - g_i\|}\right|.$$

Using $\|w_x - g_i\| \geq 1$, this yields:

$$|\varrho_i(w_x) - \varrho_i(w_y)| = \frac{\|w_x - g_i\| - 1}{\|w_x - g_i\|},$$

and in combination with (23), we obtain:

$$\left|\varrho_i(w_x) - \varrho_i(w_y)\right| < \|w_x - w_y\|.$$

This shows that $\varrho_i(\cdot)$ is a Lipschitz function. Since any finite sum of Lipschitz functions is a Lipschitz function too, we conclude that the objective function of the PSP is a Lipschitz function. □

6 Experiments

We evaluate the Discrete and the Continuous algorithms from Sects. 5.1 and 5.2 experimentally. The Discrete algorithm comes in two variants: DiscreteCircle and DiscreteOctagon, the former uses circular arcs and the latter uses octagons to approximate the fading function, compare Sect. 5.1.2. Regarding the Continuous algorithm, we focus on the geometric bound from Sect. 5.2.2. The reason is that computing the geometric bound is simple and efficient, whereas for the Lipschitz bound (Sect. 5.2.3), we cannot achieve comparable guarantees since we do not know a tight value for the Lipschitz constant L in Eq. (20).

In order to evaluate the efficiency and solution quality of the two algorithms from Sects. 5.1 and 5.2, we conducted a series of experiments with implementations for both. The experimental setup is described in Sect. 6.1 and the results are analyzed in Sect. 6.2.

6.1 Experimental setup

6.1.1 Instances

Our test set is formed by 50 instances from the art gallery problem instance library [5], see Fig. 5. We test five instance classes (NonSimpleOrtho, Spike, VonKoch, Simple, and NonSimple) with polygons of 50–700 vertices. As an example for a solved instance, see the VonKoch-type instance in Fig. 6a.

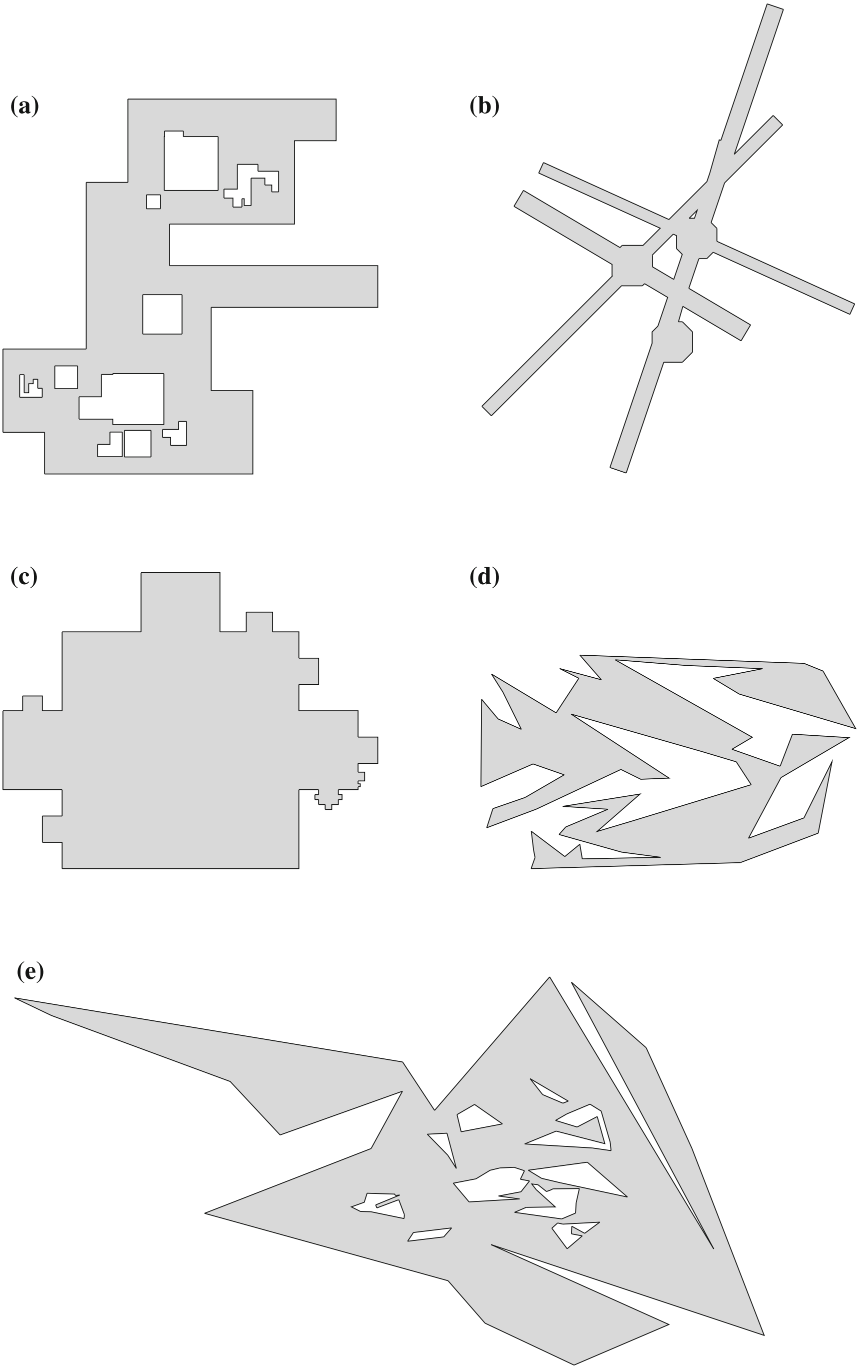

Fig. 5 Our test polygon classes. Some classes (NonSimpleOrtho, Spike, and NonSimple) have holes, others (VonKoch and Simple) do not. The number of vertices ranges from 50 to 700. **a** NonSimpleOrtho polygons are orthogonal polygons with holes. **b** Spike polygons comprise few positions that see much of the polygon. **c** VonKoch polygons are inspired by the von Koch curve. **d** Simple polygons are non-orthogonal polygons without holes. **e** NonSimple polygons comprise non-orthogonal polygons with holes

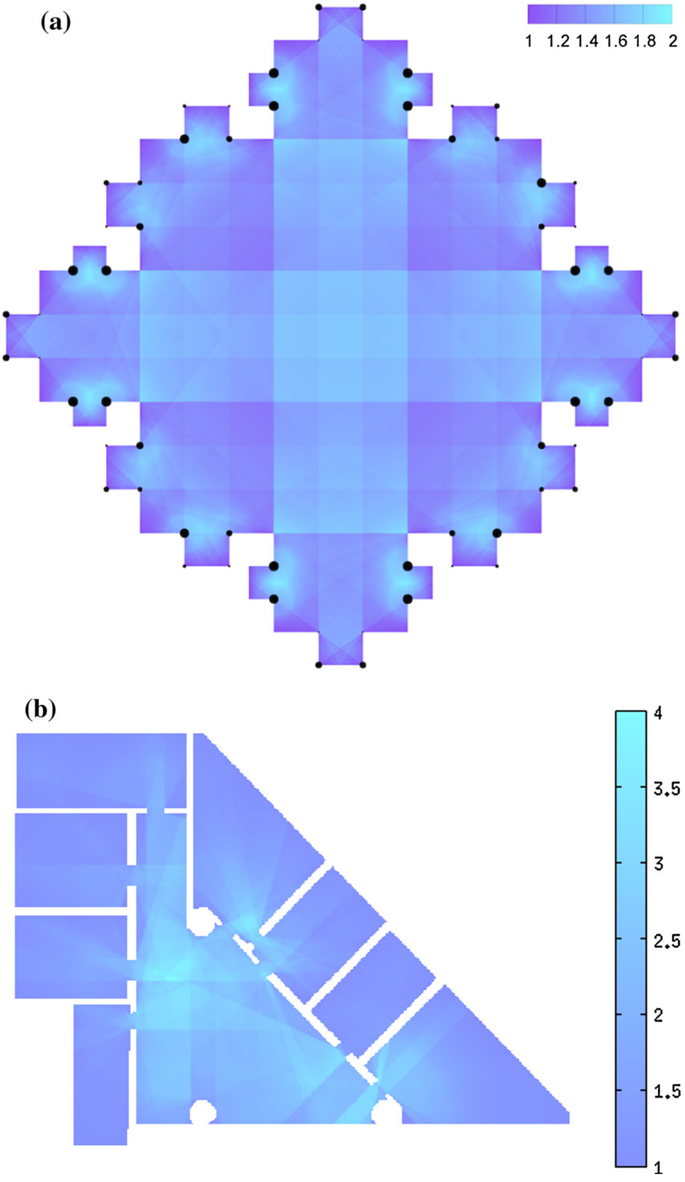

Fig. 6 AGPF solutions obtained by the Continuous approach. The *color* scales show values of coverage: *Dark blue* points are lit with exactly the necessary threshold of 1, lighter points obtain more coverage. **a** A VonKoch polygon with 100 vertices. **b** A 94-vertex office environment. (Color figure online)

Furthermore, we present a Continuous solution for a 94-vertex office environment (Fig. 6b), and a DiscreteCircle solution for a 1185-vertex cathedral (Fig. 7). These instances, however, are a demonstration and as such not included in the regular test set.

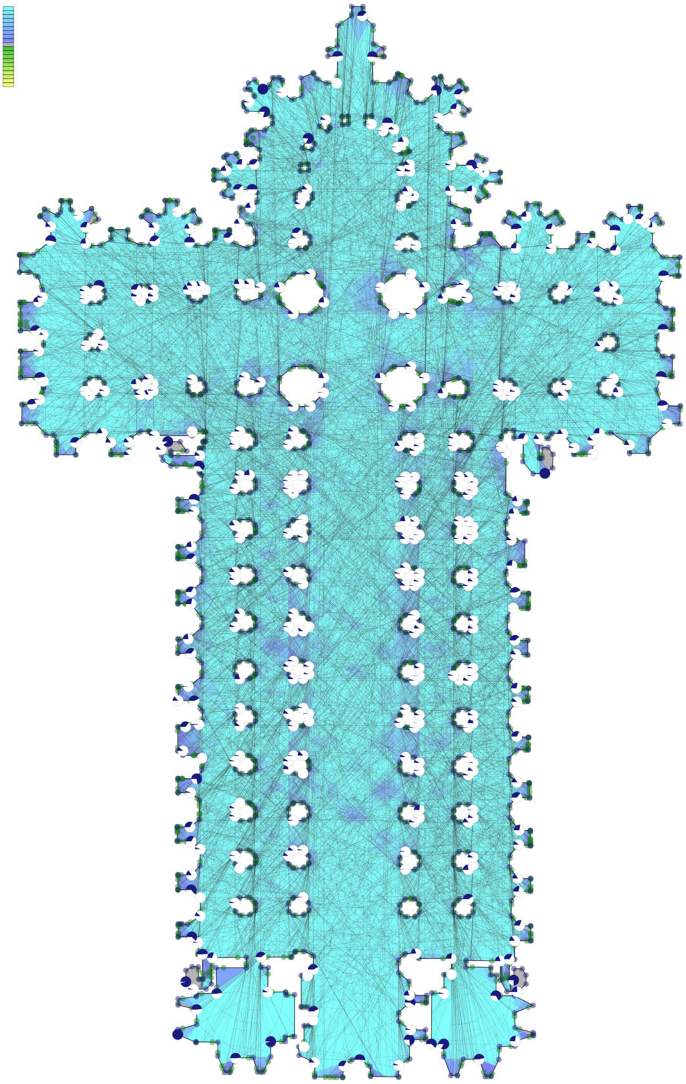

Fig. 7 An AGPF solution of a 1185-vertex cathedral polygon obtained by the DISCRETECIRCLE algorithm. The *color* scale in the *upper left* indicates the value of coverage: A *gray* point is covered with exactly the necessary value of 1, *blue* indicates that a point obtains light with a value of more than 1 (largest value for *light blue*); *yellow* and *green* points are insufficiently lit (not present in this figure). (Color figure online)

6.1.2 Polygon scaling

Solutions for the classical AGP are invariant under scaling of the input; as a result, the input complexity of algorithms solving the AGP is the number of vertices in the polygon. In the presence of fading, this is no longer the case. Especially the cutoff of ϱ at distance 1, see Eq. (5), becomes an issue: If every mutually visible pair of points in a polygon is at distance at most 1, the problem becomes equivalent to the AGP without fading. As an example, consider a convex polygon P: If P is small enough, one guard placed anywhere in P suffices, but if P is scaled up the structure of an optimal solution becomes non-obvious. The polygons in the art gallery problem instance library [5] have no natural or even consistent spatial size.

The impact of the input polygons' number of vertices on the solution time in the classical AGP, in the form of visibility calculations and arrangement complexity, has been studied extensively [1,4,6–9,11,15,20,26]. Given that fading introduces spatial size as a new source of complexity, we confine our experiments to polygons with up to 700 vertices and focus on the impact of spatial extent.

We scale the input polygons P as follows: First, we compute an estimate for the spatial complexity of P. Here, different measures could be used, for example the diameter of P, the area of P, or local feature size. Pre-experiments have shown that the average edge length of P, denoted by $\mu(P)$, provides an intuitively consistent measure for our test set. Hence, we scale P by a factor $(\Lambda\mu(P))^{-1}$, for a parameter Λ. Therefore, $\Lambda = 2$ corresponds to having the cutoff in ϱ at twice the average edge length of P. We scaled our 50 input polygons using 0.2, 0.5, 1, and 2 as possible values for Λ, yielding 200 scaled instances.

6.1.3 Parameterization

The algorithms are parameterized. Both algorithms account for the fading exponent α; we selected the two practically relevant values of 1 (laser scanner) and 2 (light).

The Discrete algorithm takes the desired approximation factor $1 + \varepsilon$ as an input, we selected 1.2, 1.6, and 2. Furthermore, we test the Discrete algorithm using circles as well as octagons, DiscreteCircle and DiscreteOctagon, to approximate the fading function; this is discussed in Sect. 5.1.2.

The Continuous algorithm is parameterized in its stopping criterion: A triangle with a difference of δ between lower and upper bound is no longer partitioned. We pick 0.01, 0.001, and 0.0001 as values for δ.

In total, this yields nine configurations, six Discrete and three Continuous, each with two choices for α. Together with the 200 scaled inputs, we run 3600 configuration-instance pairs.

6.1.4 Hardware and implementation

Our implementation of the algorithms from Sects. 5.1 and 5.2 is based on our implementation for solving the classical AGP [15,20]. We used CPLEX-12.6 [18] to solve LPs and CGAL-4.4 [2,10,17] for geometric subroutines. All experiments were carried out on a Linux-3.18.27 server with two Intel Xeon E5-2680 CPUs, i.e., 16 cores and 32 threads, and 258 GiB of RAM. We ran 15 instances in parallel, each limited to 20 minutes of CPU time and 16 GB of RAM, aborting every run which did not finish within these limits. Our implementation is not parallelized.

6.2 Results

Tables 1–3 depict the results of our experiments. For each algorithm configuration, Table 1 addresses its success rate, Table 2 its objective values, and Table 3 the required CPU time. We discuss these aspects in Sects. 6.2.1, 6.2.2, and 6.2.3, respectively. Note that each table cell corresponds to 50 test runs, i.e., one per instance.

Table 1 Success rates in percent

Falloff α		1				2			
Scaling Λ		2	1	0.5	0.2	2	1	0.5	0.2
Continuous	0.01	98	98	94	92	98	98	90	88
Continuous	0.001	98	96	88	86	96	88	84	66
Continuous	0.0001	98	88	84	68	96	86	72	42
DiscreteCircle	2	100	100	98	94	100	98	88	80
DiscreteCircle	1.6	100	100	98	84	100	98	84	60
DiscreteCircle	1.2	100	98	88	60	98	92	58	40
DiscreteOctagon	2	100	100	96	82	100	98	90	72
DiscreteOctagon	1.6	100	100	96	76	100	98	84	60
DiscreteOctagon	1.2	98	96	78	48	84	32	10	4

Table 2 Median of relative objective values

Falloff α		1				2			
Scaling Λ		2	1	0.5	0.2	2	1	0.5	0.2
Lower bound		0.85	0.88	0.90	0.93	0.85	0.88	0.91	0.87
Continuous	0.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Continuous	0.001	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Continuous	0.0001	1.00	1.00	1.00	1.00	1.00	1.00	1.00	∞
DiscreteCircle	2	1.10	1.22	1.37	1.52	1.10	1.31	1.52	1.59
DiscreteCircle	1.6	1.07	1.14	1.24	1.34	1.06	1.19	1.33	1.41
DiscreteCircle	1.2	1.02	1.05	1.09	1.13	1.02	1.06	1.14	∞
DiscreteOctagon	2	1.11	1.23	1.38	1.53	1.09	1.30	1.49	1.60
DiscreteOctagon	1.6	1.07	1.15	1.25	1.35	1.07	1.17	1.32	1.40
DiscreteOctagon	1.2	1.02	1.06	1.09	∞	1.02	∞	∞	∞

6.2.1 Success rate

We report the success rate of each algorithm configuration in Table 1 w.r.t. the imposed CPU time and memory limits described in Sect. 6.1.4. Out of the 3600 test runs, 84.9 % succeeded, 12.3 % did not meet the CPU-time limit of 20 minutes, and only 2.9 % exceeded the 16 GB memory limit (compare Sect. 6.1.4).

In the following, we account for the times and objective values of failed runs with infinity and report median values.

6.2.2 Objective value

Table 2 compares the objective values determined by the different algorithm configurations. Per instance, we consider the objective value relative to the best known solution (w.r.t. the same choice for α and Λ) and report, per cell, their median. For example, the 1.02 in the lower left corner encodes that, with $\alpha = 1$ and $\Lambda = 2$, the DiscreteOctagon algorithm

Table 3 Median of relative CPU times

Falloff α		1				2			
Scaling Λ		2	1	0.5	0.2	2	1	0.5	0.2
Continuous	0.01	1.89	2.45	4.53	8.87	2.45	3.36	8.55	19.49
Continuous	0.001	1.98	3.20	7.91	19.16	2.37	5.17	18.10	173.97
Continuous	0.0001	1.94	4.63	14.02	110.47	2.68	7.90	74.81	∞
DiscreteCircle	2	1.00	1.91	3.70	6.97	1.15	2.82	7.58	17.48
DiscreteCircle	1.6	1.05	2.07	4.62	9.88	1.23	3.44	11.23	36.96
DiscreteCircle	1.2	1.22	3.12	9.70	39.80	1.67	7.33	72.96	∞
DiscreteOctagon	2	1.13	1.91	3.45	6.59	1.38	3.08	8.26	22.27
DiscreteOctagon	1.6	1.14	2.19	4.36	11.59	1.62	4.70	16.62	65.52
DiscreteOctagon	1.2	1.80	4.76	20.69	∞	13.62	∞	∞	∞

with $1+\varepsilon = 1.2$ produced solutions at 1.02 times the objective value of the best algorithm (Continuous with $\delta = 0.01$).

Table 2 indicates that, where applicable, the median solution quality of the Continuous algorithm does not vary w.r.t. the choice of δ. Observe, however, that this does not mean that the solutions are optimal: The Continuous algorithm does not guarantee an approximation ratio.

For guaranteed bounds we include the row "Lower Bound." Each Discrete algorithm, configured as $(1+\varepsilon)$-approximation, terminating with an objective value of z yields a lower bound of $(1+\varepsilon)^{-1}z$. We present the median of the largest lower bound per instance (if applicable). Even with these conservative bounds, it is clear that the Continuous algorithm consistently produces good solutions.

The Discrete algorithm (DiscreteOctagon as well as DiscreteCircle) usually produces more expensive solutions than Continuous, but comes with the benefit of a proven approximation factor. Assuming the Continuous solutions to be near-optimal, it can be seen that the actual optimality gap (i.e., the gap between Discrete and Continuous) improves with larger values for Λ, i.e., when scaling down polygons. This might be due to the fact that more space is covered with 1-disks around guards, where ϱ and τ are equal.

6.2.3 Efficiency

We compare the CPU time of the different algorithm configurations in Table 3. Per instance, we normalize the run time to that of DiscreteCircle with $1+\varepsilon = 2$ for $\alpha = 1$ and $\Lambda = 2$, the fastest configuration. In each cell we report the median relative run time over all instances. For example, the 1.80 in the lower left corner encodes that, with $\alpha = 1$ and $\Lambda = 2$, the DiscreteOctagon algorithm with $1+\varepsilon = 1.2$ took 1.8 times as long as the DiscreteCircle configuration specified above.

For small Λ, i.e., scaled-up polygons, it can be seen that the choice of δ has a substantial impact on the CPU time of Continuous. It has, however, no measurable impact on the solution quality as discussed in Sect. 6.2.2.

Increasing Λ narrows the speed gap between Discrete and Continuous, as the former profits from a large Λ by using fewer steps in τ—and hence fewer arcs in the arrangements—whereas it makes little difference to Continuous.

Furthermore, in most configurations, DiscreteCircle is faster than DiscreteOctagon, especially for $\alpha = 2$. This means that the issue discussed in Sect. 5.1.2 is settled: Using octagons instead of circles in order to exclude circular arcs from the arrangements does not pay off, especially since it takes more octagons than circles to achieve the same approximation factor w.r.t. the continuous fading function ϱ.

6.3 Choosing an algorithm

Clearly, DiscreteCircle should be preferred over DiscreteOctagon as it has higher success rates, smaller objective values, and, in most configurations, lower run times.

As for Continuous, observe that in our experiments the choice of δ does not influence the objective value. Hence, we compare DiscreteCircle to Continuous with $\delta = 0.01$. The only configuration of DiscreteCircle yielding comparable objective values uses $1 + \varepsilon = 1.2$, but even in that case Continuous usually produces better solutions and has higher success rates.

However, Continuous does not guarantee any bounds on the approximation quality. Hence, if an approximation guarantee is required, DiscreteCircle must be preferred. Furthermore, DiscreteCircle is faster for large Λ, i.e., on polygons with a small spatial extent.

7 Future work: AGPF with point guards

Sections 5 and 6 demonstrate that AGPF(G, P) can be solved efficiently with the proposed algorithms. Hence, solving AGPF(P, P) is a natural candidate for future work.

In previous work [20], we successfully used the doubly-infinite LP formulation for AGP(P, P), together with separation routines for both the primal and the dual LP, in a practically efficient algorithm. The same could be applied here. As can be easily confirmed, the dual separation problem asks for a point $g \in P$ with

$$\sum_{w \in \mathcal{V}(g)} \varrho(g, w) y_w > 1, \tag{24}$$

given a dual solution y. Primal and dual separation problems are very similar: Given a solution x (y, respectively), they ask for the darkest (respectively brightest) point in the corresponding arrangement.

If those two problems can be solved efficiently, a possible strategy is to switch between primal and dual separation. This strategy is not guaranteed to converge, but if it does it converges to an optimal solution—and it has been shown to perform well in practice [20]. Thus, the interesting question is whether this holds true for AGPF(P, P). Both approaches from this paper allow for such a strategy.

Another possibility is to use the approach by Tozoni et al. [26] for the AGP with point guards, which uses inclusion-maximal features of a witness visibility-arrangement for guard placement. This generalizes to AGPF$_\tau$: A feature of the witness overlay dominates another if it is seen by at least the same set of guards and their coefficients $\varrho(g, w)$ are not smaller.

8 Conclusions

We introduce the art gallery problem with fading (AGPF), which is a generalization of both, the well-established art gallery problem [11,23] and the Stage Illumination Problem by Eisenbrand et al. [14]. For the AGPF, we present two efficient algorithms for the case with fixed guard positions stemming from an infinite LP formulation. While one of them is faster in practice and generally provides a good solution quality, the other provides a fully polynomial-time approximation scheme, i.e., guaranteed bounds on the solution quality. We evaluate both approaches experimentally.

Acknowledgments We would like to thank the anonymous reviewers for their suggestions that helped improve our presentation. This work was partially supported by the Deutsche Forschungsgemeinschaft (DFG) under contract number KR 3133/1-1 (Kunst!). Stephan Friedrichs, Mahdi Moeini (through Kunst!), and Christiane Schmidt were affiliated with the Technical University of Braunschweig during part of the research. Mahdi Moeini was supported by both CNRS and OSEO within the ISI project "Pajero" (France). Currently, Mahdi Moeini is affiliated to the Technical University of Kaiserslautern (Germany). Christiane Schmidt is supported by Grant 2014-03476 from Sweden's innovation agency VINNOVA, and was supported by the Israeli Centers of Research Excellence (I-CORE) program (Center No. 4/11) during the development of the paper. Open access funding provided by Max Planck Society.

References

1. Bungiu, F., Hemmer, M., Hershberger, J., Huang, K., Kröller, A.: Efficient computation of visibility polygons. CoRR, arXiv:1403.3905 (2014)
2. CGAL (Computational Geometry Algorithms Library). http://www.cgal.org/
3. Chvátal, V.: A combinatorial theorem in plane geometry. J. Comb. Theory Ser. B **18**(1), 39–41 (1975)
4. Couto, M.C., de Rezende, P.J., de Souza, C.C.: An exact algorithm for an art gallery problem. Technical Report IC-09-46, Institute of Computing, University of Campinas (2009)
5. Couto, M.C., de Rezende, P.J., de Souza, C.C.: Instances for the art gallery problem. http://www.ic.unicamp.br/~cid/Problem-instances/Art-Gallery (2009)
6. Couto, M.C., de Rezende, P.J., de Souza, C.C.: An IP solution to the art gallery problem. In: Proceedings of the 25th ACM Symposium on Computational Geometry (SoCG), pp. 88–89 (2009)
7. Couto, M.C., de Rezende, P.J., de Souza, C.C.: An exact algorithm for minimizing vertex guards on art galleries. Int. Trans. Oper. Res. **18**, 425–448 (2011)
8. Couto, M.C., de Souza, C.C., de Rezende, P.J. (2007) An exact and efficient algorithm for the orthogonal art gallery problem. In: Proceedings of the XX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI), pp. 87–94
9. Couto, M.C., de Souza, C.C., de Rezende, P.J.: Experimental evaluation of an exact algorithm for the orthogonal art gallery problem. In: Proceedings of the 7th International Workshop on Experimental Algorithms (WEA), pp. 101–113 (2008)
10. de Castro, P.M.M., Pion, S., Teillaud, M.: 2D circular geometry kernel. In: CGAL User and Reference Manual. CGAL Editorial Board, 4.5.2 edition (2015)
11. de Rezende, P.J., de Souza, C.C., Friedrichs, S., Hemmer, M., Kröller, A., Tozoni, D.C.: Engineering art galleries. CoRR, abs/1410.8720, 2014. To appear
12. Demaine, E.D., O'Rourke, J.: Open problems: open problems from CCCG 2005. In: Proceedings of the Canadian Conference on Computational Geometry, (CCCG) (2006)
13. Eidenbenz, S., Stamm, C., Widmayer, P.: Inapproximability results for guarding polygons and terrains. Algorithmica **31**(1), 79–113 (2001)
14. Eisenbrand, F., Funke, S., Karrenbauer, A., Matijevic, D.: Energy-aware stage illumination. Int. J. Comput. Geom. Appl. **18**(1/2), 107–129 (2008)
15. Fekete, S.P., Friedrichs, S., Kröller, A., Schmidt, C.: Facets for art gallery problems. Algorithmica **73**(2), 411–440 (2015)
16. Fisk, S.: A short proof of Chvátal's watchman theorem. J. Comb. Theory Ser. B **24**(3), 374 (1978)
17. Hemmer, M., Huang, K., Bungiu, F., Xu, N.: 2D visibility computation. In: CGAL User and Reference Manual. CGAL Editorial Board, 4.7 edition (2015)

18. IBM ILOG CPLEX Optimization Studio. http://www.ibm.com/software/integration/optimization/cplex-optimizer/
19. Kahn, J., Klawe, M., Kleitman, D.: Traditional art galleries require fewer watchmen. SIAM J. Algebr. Discrete Methods **4**(2), 194–206 (1983)
20. Kröller, A., Baumgartner, T., Fekete, S.P., Schmidt, C.: Exact solutions and bounds for general art gallery problems. ACM J. Exp. Algorithmics **17**(1), 2.3:2.1–2.3:2.23 (2012)
21. Kröller, A., Moeini, M., Schmidt, C.: A novel efficient approach for solving the art gallery problem. In: 7th International Workshop on Algorithms and Computation (WALCOM), pp. 5–16 (2013)
22. Lee, D., Lin, A.K.: Computational complexity of art gallery problems. IEEE Trans. Inf. Theory **32**(2), 276–282 (1986)
23. O'Rourke, J.: Art Gallery Theorems and Algorithms. International Series of Monographs on Computer Science. Oxford University Press, New York (1987)
24. Pinter, J.D.: Global Optimization in Action. Kluwer, Dordrecht (1996)
25. Tóth, C.D.: Art gallery problem with guards whose range of vision is 180°. Comput. Geom. **17**(3–4), 121–134 (2000)
26. Tozoni, D.C., de Rezende, P.J., de Souza, C.C.: A practical iterative algorithm for the art gallery problem using integer linear programming. Optimization Online (2013). http://www.optimization-online.org/DB_HTML/2013/11/4106.html