Big Brother Logic: visual-epistemic reasoning in stationary multi-agent systems

Olivier Gasquet, Valentin Goranko, François Schwarzentruber

Big Brother Logic: visual-epistemic reasoning in stationary multi-agent systems

Olivier Gasquet · Valentin Goranko · François Schwarzentruber

Abstract We consider multi-agent scenarios where each agent controls a surveillance camera in the plane, with fixed position and angle of vision, but rotating freely. The agents can thus observe the surroundings and each other. They can also reason about each other's observation abilities and knowledge derived from these observations. We introduce suitable logical languages for reasoning about such scenarios which involve atomic formulae stating what agents can see, multi-agent epistemic operators for individual, distributed and common knowledge, as well as dynamic operators reflecting the ability of cameras to turn around in order to reach positions satisfying formulae in the language. We also consider effects of public announcements. We introduce several different but equivalent versions of the semantics for these languages, discuss their expressiveness and provide translations in PDL style. Using these translations we develop algorithms and obtain complexity results for model checking and satisfiability testing for the basic logic BBL that we introduce here and for some of its extensions. Notably, we show that even for the extension with common knowledge, model checking and satisfiability testing remain in PSPACE. We also discuss the sensitivity of the set of validities to the admissible angles

Olivier Gasquet
Université Paul Sabatier, IRIT, Toulouse
E-mail: olivier.gasquet@irit.fr

Valentin Goranko
Stockholm University and University of Johannesburg (visiting professorship)
E-mail: valentin.goranko@philosophy.su.se

François Schwarzentruber
ENS Rennes
E-mail: francois.schwarzentruber@ens-rennes.fr

of vision of the agents' cameras. Finally, we discuss some further extensions: adding obstacles, positioning the cameras in 3D or enabling them to change positions. Our work has potential applications to automated reasoning, formal specification and verification of observational abilities and knowledge of multi-robot systems.

Keywords visual-epistemic logical reasoning, multi-agent systems, surveillance cameras, observational abilities, knowledge, model checking, satisfiability testing

1 Introduction

Modeling and study of multi-agent systems that involve intelligent agents combining perceptual and reasoning abilities is a major field of research, with applications to AI. An important issue in that field is the analytic processing of visual perception and visual communication, where the main problem is how to represent and reason logically about the visual information that the intelligent agents receive from the environment and from each other. This affects not only agents' knowledge about the surrounding world, but also about each other's knowledge about that world, etc. and that knowledge in turn affects and guides their intelligent behavior.

In this paper we address these issues in the context of multi-agent scenarios where each agent controls a stationary surveillance camera positioned in the plane. In the basic framework presented here each camera has a fixed position and angle of view, but can rotate freely around its axis. Through the cameras, the agents can observe the surrounding world and each other. We adopt some basic assumptions about these scenarios, in order to simplify or idealize them:

▷ We identify physically the agents with their cameras. Thus, we can talk (with benign imprecision) about agents seeing objects and about cameras' knowledge.

▷ We assume that the cameras/agents are positioned at single points in the plane and that they are transparent. These assumptions can be easily relaxed and non-transparent cameras/agents of real sizes can be considered, without major modifications of the framework and the obtained results, by simply treating them both as cameras and as obstacles.

▷ We assume that the only objects of interest for any agent in our scenarios are the other cameras/agents. This assumption is not practically restrictive, because one can assume that all other objects of interest are, too, agents equipped with cameras, which cannot see anything or who are simply not interested in what they can see and know.

▷ In this study we also assume that the agents and their cameras are stationary. In reality, though, these are often mobile. Adding mobility to cameras leads to models where only the set of agents and the vision angles of their cameras are specified, but not their positions. That makes solving the

model checking and satisfiability problems apparently computationally more expensive, so we leave the case of mobile cameras to a follow-up work.

▷ In the stationary setup we assume that every agent knows the exact positions of all agents/cameras, even of those that the agent does not see currently, and that is a common knowledge amongst all agents. This assumption is certainly well justified if agents are endowed with some memory, as they can turn around to observe and memorize the positions of the others before reasoning and acting further. On the other hand, we only assume that at any given moment an agent knows the directions of vision, and therefore the scope of vision, of only those other agents that he can see at that moment.

We introduce suitable logical languages for reasoning about such scenarios which involve atomic formulae stating what agents can see, multi-agent epistemic operators for individual, distributed and common knowledge, as well as dynamic operators reflecting the ability of cameras to turn around in order to reach positions satisfying formulae in the language. We also add agents' public announcements.

Here we are interested not only in what agents can see, but also in the *knowledge* they can derive from their visual perception. The agents' knowledge that we model and study in our framework includes knowledge about other agents' observation abilities and about their direct and iterated knowledge derived from such observations. A subtle issue arises here, of how vision and knowledge are related. At first glance, it seems that agents only know about other agents' knowledge what they can immediately derive from what they can see regarding the observational abilities of those other agents. We show that this is not quite so, as there is a non-trivial deductive aspect in the analytic knowledge that agents can acquire, arising from supplementary geometric reasoning.

We introduce semantics for our logical languages on natural geometric models, as well as formal Kripke semantics for them in vision-based finite abstractions of the geometric models. We then discuss the expressiveness of our logical languages and provide translations for them in the style of Propositional Dynamic Logic PDL [12]. Using these translations we develop algorithms and obtain complexity results for model checking and satisfiability testing for the basic logic BBL that we introduce here and for some of its extensions. The key observation that enables our model checking and satisfiability testing procedures is that, while the set of geometric models over a configuration of cameras with fixed positions is infinite, the set of their vision-based abstractions, obtained by identifying models where each agent can see the same configuration of agents, is finite.

Finally, we discuss some important extensions of the present work: adding obstacles, positioning the cameras in 3D or enabling them to change positions.

Our work has potential applications to automated reasoning, formal specification and verification of observational abilities and knowledge of multi-robot systems. We are aware of very few related previous studies on the topic, e.g., [9], [6]. They have mainly considered technical, architectural and knowledge

representational aspects of agent-based systems of multi-camera surveillance and information processing, but do not involve logical reasoning. To our knowledge, the only previous study related to logical reasoning in such scenarios is the one presented in [14] and [1], which is the main precursor of the present work. Unlike the frameworks considered in the latter two papers, in this work we fix the positions of the agents. That choice has enabled us to obtain more efficient algorithms for model checking and satisfiablity testing and to determine their precise complexities, which are the main technical contributions of the paper. We have also introduced the interpretation of agents as cameras here and the turning operators that makes the language considerably more expressive and enable us to take an explicit dynamic perspective on our logics.

This work is a substantial extension and a revision of [10]. In particular, we have provided in full details the model-checking algorithms and the proofs of their complexities, and have added the algorithm for satisfiablity testing, that runs in polynomial space. We have also added the public announcements here.

The paper is organized as follows: after brief preliminaries in Section 2, we introduce the logics BBL and some extensions in Section 3. We discuss their expressiveness in Section 4 and provide translations to PDL variants of BBL in Section 5. In Section 6 we use these translations to obtain optimal complexity results and develop efficient algorithms for model checking and satisfiability testing. We then introduce and discuss briefly some further extensions in Section 7 and end with the concluding Section 8.

2 Preliminaries

2.1 Multi-agent epistemic models and logics

For the necessary background on modal logic refer e.g., to [5]. Here we only give the basics on multi-agent epistemic models and logics. For the conceptual aspects or further technical background refer e.g., to [8] or [17].

A *multi-agent epistemic structure* for a set of agents Agt is a tuple $\mathcal{S} = \langle \mathsf{Agt}, \mathsf{St}, \{\sim_{\mathsf{a}} | \ \mathsf{a} \in \mathsf{Agt}\}\rangle$, where St is the set of states (possible worlds) and $\sim_{\mathsf{a}}$ is the epistemic indistinguishability relation on St of the agent a. Given a fixed set of atomic propositions PROP, a *multi-agent epistemic model* (MAEM) extends a multi-agent epistemic structure with a *truth valuation* V that assigns to each $p \in \text{PROP}$ the set of states $V(p)$ where it is declared true.

We consider fully expressive multi-agent epistemic logics, extending classical (propositional) logic by adding the following usual epistemic operators:

- *Individual knowledge* $\mathsf{K_a}$ meaning "$\mathsf{The\ agent\ a\ knows\ that}$". We will also use its dual operator $\hat{\mathsf{K}}_{\mathsf{a}} := \neg \mathsf{K_a} \neg$, intuitively meaning "$\mathsf{The\ agent\ K_a\ considers\ it\ possible\ that}$". or "$\mathsf{It\ is\ consistent\ with\ the\ knowledge\ of\ K_a\ that}$".
- *Group knowledge* of A: $\mathsf{K_A}$, definable as $\mathsf{K_A}\varphi := \bigwedge_{\mathsf{a}\in\mathsf{A}} \mathsf{K_a}\varphi$, and meaning "$\mathsf{Every\ agent\ in\ the\ group\ A\ knows\ that}\ \varphi$". When $\mathsf{A} = \{\mathsf{a}\}$ we write $\mathsf{K_a}$ instead of $\mathsf{K}_{\{\mathsf{a}\}}$.

– *Distributed knowledge* of A: D_A, where $\mathsf{D}_\mathsf{A}\varphi$ says "It is a distributed knowledge amongst the agents in the group A that φ", intuitively meaning that the collective knowledge of all agents in the group A implies φ. For instance, if $\mathsf{K}_a\varphi$ and $\mathsf{K}_b(\varphi \to \psi)$ hold for some agents a and b, then $\mathsf{D}_{\{a,b\}}(\varphi \wedge (\varphi \to \psi))$ holds, and therefore $\mathsf{D}_{\{a,b\}}\psi$ holds, too, by the closure of knowledge under logical consequence.
– *Common knowledge* of A: C_A, where $\mathsf{C}_\mathsf{A}\varphi$, says "It is a common knowledge amongst the agents in the group A that φ", intuitively meaning that every agent in A knows φ and every agent in A knows that every agent in A knows φ, and every agent in A knows that, etc., ad infinitum.

Thus, the formulae of the language of the multi-agent epistemic logic MAEL are defined as follows:

$\varphi := p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{K}_\mathsf{a}\varphi \mid \mathsf{K}_\mathsf{A}\varphi \mid \mathrm{D}_\mathsf{A}\varphi \mid \mathrm{C}_\mathsf{A}\varphi$,

where p ranges over a fixed set of atomic propositions PROP, $\mathsf{a} \in \mathsf{Agt}$ and A is a non-empty subset of Agt.

We note that the individual knowledge K_a can also be considered definable as $\mathrm{D}_{\{\mathsf{a}\}}$, so in principle it suffices to keep in the language only the operators for distributed and common knowledge. However, we will keep the operators of individual knowledge as primitives needed to define the basic language, which will be without distributed knowledge.

In our setup there will be no abstract (i.e., uninterpreted) atomic propositions but only concretely interpreted ones of the type "a sees b" for concrete agents a and b, so epistemic structures and models will coincide.

The formal semantics of the epistemic operators at a state in a multi-agent epistemic model $\mathcal{M} = (\mathsf{Agt}, \mathsf{St}, \{\sim_\mathsf{a} \mid \mathsf{a} \in \mathsf{Agt}\}, V)$ is given by the following standard truth definitions:

(K_a) $\mathcal{M}, q \models \mathsf{K}_\mathsf{a}\varphi$ iff $\mathcal{M}, q' \models \varphi$ for all q' such that $q \sim_\mathsf{a} q'$.

(K_A) $\mathcal{M}, q \models \mathsf{K}_\mathsf{A}\varphi$ iff $\mathcal{M}, q' \models \varphi$ for all q' such that $q \sim^E_\mathsf{A} q'$, where $\sim^E_\mathsf{A} = \bigcup_{\mathsf{a}\in\mathsf{A}} \sim_\mathsf{a}$.

(C_A) $\mathcal{M}, q \models \mathsf{C}_\mathsf{A}\varphi$ iff $\mathcal{M}, q' \models \varphi$ for all q' s.t. $q \sim^C_\mathsf{A} q'$, where $\sim^C_\mathsf{A}$ is the transitive closure of $\sim^E_\mathsf{A}$.

(D_A) $\mathcal{M}, q \models \mathsf{D}_\mathsf{A}\varphi$ iff $\mathcal{M}, q' \models \varphi$ for all q' such that $q \sim^D_\mathsf{A} q'$, where $\sim^D_\mathsf{A} = \bigcap_{\mathsf{a}\in\mathsf{A}} \sim_\mathsf{a}$.

2.2 Flatland

Interaction between visual perception and knowledge has been studied in [1], in a logical framework based on the following language, where a, b are agents:

$$(\mathcal{L}_{\triangleright,\mathsf{K}}) \quad \phi ::= a \triangleright b \mid \neg\phi \mid \phi \vee \phi \mid \mathsf{K}_a\phi$$

The atomic proposition $a \triangleright b$ reads "agent a sees agent b".

The language $\mathcal{L}_{\triangleright,\mathsf{K}}$ has no abstract atomic propositions. The primary intended models are geometric models where each agent is located at a point in

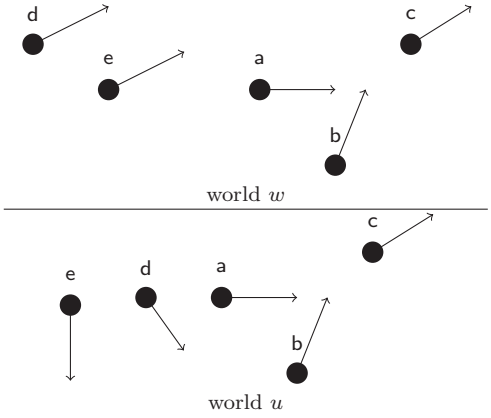


Fig. 1 Two possible 2–dimensional worlds in Flatland that are indistinguishable for agent a

the space and sees a half-space. Different agents are located at different points and every agent a who can see another agent b can also see b's direction of vision. An abstract semantics is based on Kripke models where the possible worlds are geometric models. The indistinguishability relation $\sim_{\mathsf{a}}$ for agent a in such Kripke model is defined as follows: two worlds w and u are such that $w \sim_{\mathsf{a}} u$ iff the agents seen by agent a are the same and are located in the same way in both w and u, i.e., each occupying the same position and pointing in the same direction.

Two variants of this framework have been studied in [1]: Lineland, where the space is 1-dimensional, and Flatland, where the space is 2-dimensional, i.e., a plane. Figure 1 shows two possible worlds in Flatland that are indistinguishable for agent a, because they look the same in the parts which a can see. Note that a sees both b and c in both worlds, so a can see that b sees c, and this is the case in every world which is indistinguishable from these for a, hence a *knows* that b sees c. In fact, as shown in [14], in this semantics $\mathsf{K}_a(b \triangleright c)$ is equivalent to $a \triangleright b \wedge a \triangleright c \wedge b \triangleright c$, so some occurrences of the knowledge operators can be eliminated. On the other hand, a does not see d and e in any of these worlds, so a does not know whether d sees e in any of them. In fact, d sees e in world w and d does not see e in world u.

In that framework, the model checking problem takes as an input the description of a possible world w (positions and directions of agents) and a formula ϕ in $\mathcal{L}_{\triangleright,\mathsf{K}}$ and asks whether ϕ holds in w. The satisfiability problem takes as an input a formula ϕ in $\mathcal{L}_{\triangleright,\mathsf{K}}$ and asks whether there exists a world w satisfying ϕ. It has been proved in [1] that both model checking and satisfiability testing are PSPACE-complete for Lineland and are respectively PSPACE-hard and in EXPSPACE for Flatland.

2.3 First-order Theory of the Reals

Consider the following first-order language L_F for fields:

$$\phi ::= (e > 0) \mid (e = 0) \mid \neg\phi \mid (\phi \wedge \phi) \mid \forall x\phi \mid \exists x\phi$$

where e is a polynomial expression over variables $x, y, \ldots$. L_F has a standard interpretation in the field of reals $\mathbb{R}$. The problem of checking whether a closed formula of L_F is true in $\mathbb{R}$ is in EXPSPACE (see [16] for the historically first decision procedure and [4] for the description of an algorithm that runs in EXPSPACE) The problem for the existential fragment of L_F, i.e., for closed sentences of the form

$$\exists x_1 \ldots \exists x_n \psi(x_1, \ldots, x_n)$$

where $\psi(x_1, \ldots, x_n)$ is a quantifier-free formula for which the truth-checking problem in $\mathbb{R}$ is in PSPACE (see [7]).

3 Logics for agents with stationary surveillance cameras

Here we introduce our basic logic for stationary cameras, which we dub Big Brother Logic, or BBL for short. Intuitively, we consider scenarios where many agents are positioned in the plane, each of them equipped with a camera with fixed angle of vision. The agents can freely turn their cameras but cannot change their positions. At every moment an agent sees everything that is within the current sector of vision of its camera. In particular, we assume, for technical convenience, that the cameras/agents have no dimensions and are positioned at single points in the plane, and that they are transparent, so they can be seen through. These assumptions can be easily relaxed and non-transparent cameras/agents of real sizes can be considered, by treating them both as cameras and as obstacles (see Section 7).

3.1 Languages

Hereafter we fix a set of agents $\mathsf{Agt} = \{\mathsf{a}_1, \ldots, \mathsf{a}_k\}$. The basic language of BBL involves the 'seeing' operator $\triangleright$ from Flatland, the individual knowledge epistemic operators $\{\mathsf{K}_\mathsf{a}\}_{\mathsf{a}\in\mathsf{Agt}}$, plus the *turning* (diamond) operators $\{\langle \overset{\frown}{\mathsf{a}} \rangle\}_{\mathsf{a}\in\mathsf{Agt}}$, where $\langle \overset{\frown}{\mathsf{a}} \rangle\phi$ intuitively means "a can turn around its position so that ϕ holds". We will use the standard notation from PDL: $[\overset{\frown}{\mathsf{a}}] := \neg\langle \overset{\frown}{\mathsf{a}} \rangle\neg$.

The formulae of BBL are defined as follows:

$$\phi ::= a \triangleright b \mid \neg\phi \mid \phi \vee \phi \mid \mathsf{K}_\mathsf{a}\phi \mid \langle \overset{\frown}{\mathsf{a}} \rangle\phi$$

We also introduce the following extensions of BBL:

- BBL_C: BBL plus all operators C_A for every group of agents A.

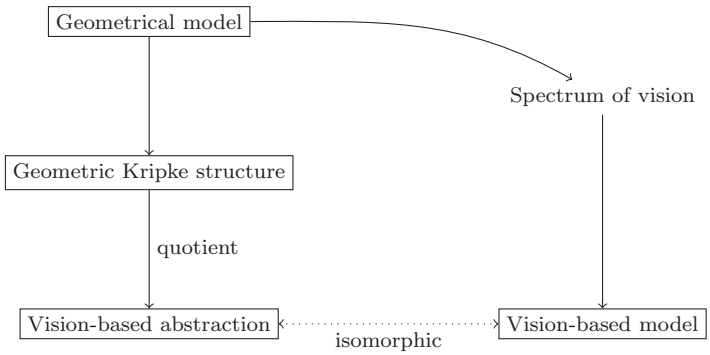


Fig. 2 An overview of the models introduced in subsection 3.2

- BBL_D: BBL plus all operators D_A for every group of agents A.
- Combinations of the above.

We will often write $\mathsf{a} \not\rhd \mathsf{b}$ as abbreviation of $\neg\, \mathsf{a} \rhd \mathsf{b}$. Some additional operators: $\mathsf{a} \bowtie \mathsf{b}$, meaning "a *sees* b *and* b *sees* a", i.e., an abbreviation of $\mathsf{a} \rhd \mathsf{b} \wedge \mathsf{b} \rhd \mathsf{a}$, and $\mathsf{B}(\mathsf{abc})$, meaning "b *is between* a *and* c"; to be discussed and defined later.

3.2 Semantics

We will introduce four different equivalent types of models, but eventually providing equivalent semantics for the logic BBL. The first type, called *geometrical models* are natural geometric representations of configurations of cameras as points in the plane and of their areas of vision as sectors between pairs of infinite rays. The second type, *geometric Kripke structures* are essentially Kripke structures with possible worlds being the different possible geometric models. The third type, called *vision-based abstractions* are again Kripke structures obtained as finite quotients of geometric Kripke structures, which will enable us to do algorithmic model checking in such structures. The last type, called *vision-based models* can be constructed directly from geometric models by identifying their *spectrum of vision* – a tuple of families of "sets of vision", one family for each agent, obtained by turning its camera in a full circle. These are essentially isomorphic to vision-based abstractions and will be used in the actual model checking algorithms. Figure 2 gives an overview of the four models introduced in this subsection.

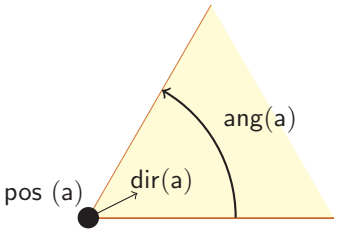
3.2.1 Geometric models

Camera configurations are formally represented by geometric models defined in the Euclidean plane $\mathbb{R}^2$ (or just a region) as follows. Every agent's camera is associated with a position (a point in the plane), an angle of vision and a unit vector representing the direction of its vision. Formally:

Definition 1 Let U be the set of unit vectors of $\mathbb{R}^2$. A *geometric model* is a tuple $(\mathsf{pos}, \mathsf{dir}, \mathsf{ang})$ where:

- $\mathsf{pos} : \mathsf{Agt} \to \mathbb{R}^2$;
- $\mathsf{dir} : \mathsf{Agt} \to U$;
- $\mathsf{ang} : \mathsf{Agt} \to [0, 2\pi)$.

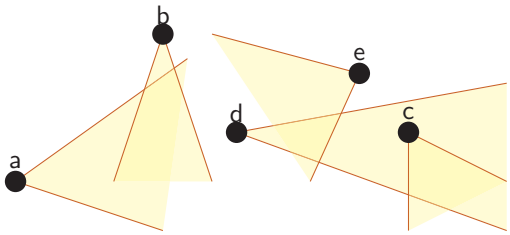
Given an agent a, the direction $\mathsf{dir}(\mathsf{a})$ is the bisector of the sector of vision of angle $\mathsf{ang}(\mathsf{a})$, depicted as follows:



We denote $C_{p,u,\alpha}$ the closed sector with vertex at the point p, angle α and bisector in direction u. For technical reasons, however, we assume that $p \notin C_{p,u,\alpha}$. The region of the plane seen by agent a is $C_{\mathsf{pos(a)},\mathsf{dir(a)},\mathsf{ang(a)}}$.

We assume that different agents have different positions, every camera can "see" everything inside and on the boundaries of its sector of vision, including the positions and the directions of vision of all agents positioned there, but can see nothing outside of it. All cameras can turn at any angle.

Example 1 (Cameras and their sectors of vision)



3.2.2 Geometic Kripke structures and semantics of BBL

Now, we will introduce the second type of models which are essentially Kripke structures in which the possible worlds are geometric models. They will involve relations corresponding to camera turning and to agents' knowledge. We will define knowledge in such models as usual in epistemic logics: an agent a knows a property ϕ iff ϕ holds in all possible worlds that are indistinguishable for him. In our case, as in the logic of Flatland [1], this means, ϕ holds in all possible worlds where agent a sees exactly what it sees in the actual world. So, those possible worlds can only differ from the actual world in the directions of vision of the agents not seen by agent a. More precisely, the semantics of all epistemic operators is given as in Section 2.1 by regarding the relation $\sim_{\mathsf{a}}$ as

follows: $\mathsf{dir} \sim_{\mathsf{a}} \mathsf{dir}'$ iff the set of agents seen by a in dir and dir' are the same and they have the same directions of view, and so does a itself. The difference with the approach described in [1] is that here the positions of the agents cannot change and the agents have common knowledge of each other's positions.

Definition 2 Given $\mathsf{pos} : \mathsf{Agt} \to \mathbb{R}^2$ and $\mathsf{ang} : \mathsf{Agt} \to [0, 2\pi)$, we define a *geometric Kripke structure* $\mathcal{M} = (\mathsf{pos}, \mathsf{ang}, \mathsf{D}, R, \sim)$ where:

- D is the set of possible functions $\mathsf{dir} : \mathsf{Agt} \to U$,
- R assigns to each agent a the following equivalence relation on D:

  $R_{\mathsf{a}} = \{(\mathsf{dir}, \mathsf{dir}') \in \mathsf{D}^2 \mid \mathsf{dir}(\mathsf{b}) = \mathsf{dir}'(\mathsf{b}) \text{ for all } \mathsf{b} \neq \mathsf{a}\}$
- $\sim$ assigns to each agent a the following equivalence relation on D:

  $\sim_{\mathsf{a}} = \{(\mathsf{dir}, \mathsf{dir}') \in \mathsf{D}^2 \mid \mathsf{dir}(\mathsf{b}) = \mathsf{dir}'(\mathsf{b}) \text{ for all } \mathsf{b} \text{ such that } \mathsf{b} = \mathsf{a} \text{ or } \mathsf{pos}(\mathsf{b}) \in C_{\mathsf{pos}(\mathsf{a}),\mathsf{dir}(\mathsf{a}),\mathsf{ang}(\mathsf{a})}\}$

  It is straightforward to show that the relations defined above are correctly defined equivalence relations.

The first two components of every geometric Kripke structure, viz. pos and ang, fix the geometric parameters of the model, i.e., positions and the angles of vision of the agents. The other three components form indeed a Kripke structure $(\mathsf{D}, R, \sim)$ where D is the set of possible worlds, while R and $\sim$ represent tuples of equivalence relations, one per agent. Note that these are, in fact, uniquely determined by pos and ang.

The truth of formulae is evaluated in such geometric Kripke structures with additionally specified direction function $\mathsf{dir} \in \mathsf{D}$, representing the current geometric model. We will call these *pointed geometric Kripke structures*. The truth conditions are defined as follows, where $\mathcal{M} = (\mathsf{pos}, \mathsf{ang}, \mathsf{D}, R, \sim)$ and $\mathsf{dir} \in \mathsf{D}$:

- $\mathcal{M}, \mathsf{dir} \models \mathsf{a} \triangleright \mathsf{b}$ iff $\mathsf{pos}(\mathsf{b}) \in C_{\mathsf{pos}(\mathsf{a}),\mathsf{dir}(\mathsf{a}),\mathsf{ang}(\mathsf{a})}$,
- $\mathcal{M}, \mathsf{dir} \models \langle \overset{\frown}{\mathsf{a}} \rangle \phi$ iff $\mathcal{M}, \mathsf{dir}' \models \phi$ for some $\mathsf{dir}' \in \mathsf{D}$ such that $\mathsf{dir}' R_{\mathsf{a}}\, \mathsf{dir}$,
- $\mathcal{M}, \mathsf{dir} \models \mathsf{K}_{\mathsf{a}} \phi$ iff $\mathcal{M}, \mathsf{dir}' \models \phi$ for all $\mathsf{dir}' \in \mathsf{D}$ such that $\mathsf{dir}' \sim_{\mathsf{a}} \mathsf{dir}$.

3.2.3 Vision-based abstractions of geometric Kripke structures

The geometric Kripke structures are infinite, but they can be partitioned into finitely many clusters such that all agents see the same sets of agents in all structures within the same cluster. We will give here an equivalent definition of the semantics of BBL, but now based on finite quotients of geometric Kripke structures w.r.t that equivalence relation, which we call *vision-based abstractions*.

More precisely, note that if all agents that are seen by a given agent (camera) are in the interior of its sector of vision, then a slight change of the direction of vision of that camera may not change what the respective agent sees. So all cameras would see the same objects if they are rotated at sufficiently small

angles. This naturally defines an equivalence relation over worlds in geometric Kripke structures where equivalent worlds satisfy the same $\rhd$-based propositions. Formally, given a geometric Kripke structure $\mathcal{M} = (\mathsf{pos}, \mathsf{ang}, \mathsf{D}, R, \sim)$ and $\mathsf{dir}, \mathsf{dir}' \in \mathsf{D}$, we define:

$$\mathsf{dir} \equiv \mathsf{dir}' \text{ iff } (\forall \mathsf{a}, \mathsf{b} \in \mathsf{Agt} : \mathcal{M}, \mathsf{dir} \models \mathsf{a} \rhd \mathsf{b} \text{ iff } \mathcal{M}, \mathsf{dir}' \models \mathsf{a} \rhd \mathsf{b}).$$

The equivalence class of dir with respect to $\equiv$ will be denoted by $\overline{\mathsf{dir}}$. Since Agt is finite, there are finitely many equivalence classes in any given geometric Kripke structure (see further Proposition 2).

Lemma 1 *Given any geometric Kripke structure $\mathcal{M} = (\mathsf{pos}, \mathsf{ang}, \mathsf{D}, R, \sim)$, for every agent a we have that $(\equiv \circ R_{\mathsf{a}}) = (R_{\mathsf{a}} \circ \equiv)$ and $(\equiv \circ \sim_{\mathsf{a}}) = (\sim_{\mathsf{a}} \circ \equiv)$, where $\circ$ is the composition of relations.*

Proof 1. Suppose $(\mathsf{dir}, \mathsf{dir}') \in (\equiv \circ R_{\mathsf{a}})$, and let dir'' be the same as dir for all agents $\mathsf{b} \neq \mathsf{a}$ but with $\mathsf{dir}''(\mathsf{a}) = \mathsf{dir}'(\mathsf{a})$. Then $(\mathsf{dir}, \mathsf{dir}'') \in R_{\mathsf{a}}$ by definition and $\mathsf{dir}'' \equiv \mathsf{dir}'$ follows from the assumption. So, $(\mathsf{dir}, \mathsf{dir}') \in (R_{\mathsf{a}} \circ \equiv)$. The other direction is similar.

2. Likewise, suppose $(\mathsf{dir}, \mathsf{dir}') \in (\equiv \circ \sim_{\mathsf{a}})$ and let dir'' be such that $\mathsf{dir}''(\mathsf{b}) = \mathsf{dir}(\mathsf{b})$ for all agents $\mathsf{b} \in \{\mathsf{a}\} \cup C_{\mathsf{pos}(\mathsf{a}),\mathsf{dir}(\mathsf{a}),\mathsf{ang}(\mathsf{a})}$ and $\mathsf{dir}''(\mathsf{b}) = \mathsf{dir}'(\mathsf{b})$ for the others. Then $(\mathsf{dir}, \mathsf{dir}'') \in \sim_{\mathsf{a}}$ by definition and $\mathsf{dir}'' \equiv \mathsf{dir}'$ follows from the assumption. So, $(\mathsf{dir}, \mathsf{dir}') \in (\sim_{\mathsf{a}} \circ \equiv)$. Similarly, the other way round.

Definition 3 Let $\mathcal{M} = (\mathsf{pos}, \mathsf{ang}, \mathsf{D}, R, \sim)$ be a geometric Kripke structure. The *vision-based abstraction of $\mathcal{M}$* is the quotient structure $\mathcal{K}_{\mathcal{M}} = (\overline{\mathsf{D}}, \overline{R}, \overline{\sim})$ defined as follows:

- $\overline{\mathsf{D}} = \{\overline{\mathsf{dir}} \mid \mathsf{dir} \in \mathsf{D}\}$,
- $\overline{R}$ maps each agent a to the equivalence relation $\overline{R}_{\mathsf{a}} = \{(\overline{\mathsf{dir}}, \overline{\mathsf{dir}'}) \mid (\mathsf{dir}, \mathsf{dir}') \in (\equiv \circ R_{\mathsf{a}})\}$,
- $\overline{\sim}$ maps each agent a to the equivalence relation $\overline{\sim}_{\mathsf{a}} = \{(\overline{\mathsf{dir}}, \overline{\mathsf{dir}'}) \mid (\mathsf{dir}, \mathsf{dir}') \in (\equiv \circ \sim_{\mathsf{a}})\}$.

The correctness of this definition follows from Lemma 1.

We can now re-define in an obvious way the semantics of BBL on vision-based abstractions of geometric Kripke structures. We are going to show that the two semantics are equivalent.

Proposition 1 *Let $\mathcal{M} = (\mathsf{pos}, \mathsf{ang}, \mathsf{D}, R, \sim)$ be a geometric Kripke structure and $\mathcal{K}_{\mathcal{M}} = (\overline{\mathsf{D}}, \overline{R}, \overline{\sim})$ be the vision-based abstraction of $\mathcal{M}$. Then the canonical mapping $h : \mathsf{D} \to \overline{\mathsf{D}}$ defined by $h(\mathsf{dir}) = \overline{\mathsf{dir}}$ is a bounded morphism from $\mathcal{M}$ (regarded as a standard Kripke structure $(\mathsf{D}, R, \sim)$) onto $\mathcal{K}_{\mathcal{M}}$, and consequently, a bisimulation.*

Proof We check the bounded morphism conditions:

Atomic harmony: $\mathcal{M}, \mathsf{dir} \models \mathsf{a} \rhd \mathsf{b}$ iff $\mathcal{K}_{\mathcal{M}}, \overline{\mathsf{dir}} \models \mathsf{a} \rhd \mathsf{b}$ for any $\mathsf{dir} \in \mathsf{D}$, by definition.

Forth: Let $\mathsf{dir}, \mathsf{dir}' \in \mathsf{D}$. If $(\mathsf{dir}, \mathsf{dir}') \in R_\mathsf{a}$ then $(h(\mathsf{dir}), h(\mathsf{dir}')) \in \overline{R}_\mathsf{a}$ by definition of $\overline{R}_\mathsf{a}$. Likewise, if $(\mathsf{dir}, \mathsf{dir}') \in \sim_\mathsf{a}$ then $(h(\mathsf{dir}), h(\mathsf{dir}')) \in \overline{\sim}_\mathsf{a}$.

Back: If $(h(\mathsf{dir}), \overline{\mathsf{dir}'}) \in \overline{R}_\mathsf{a}$ for some $\overline{\mathsf{dir}'} \in \overline{\mathsf{D}}$ then $(\mathsf{dir}, \mathsf{dir}') \in (\equiv \circ R_\mathsf{a})$. Hence, there is $\mathsf{dir}'' \in \mathsf{D}$ such that $\mathsf{dir} \equiv \mathsf{dir}''$, so $h(\mathsf{dir}) = h(\mathsf{dir}'')$, and $(\mathsf{dir}'', \mathsf{dir}') \in R_\mathsf{a}$. Likewise for $\overline{\sim}_\mathsf{a}$.

Corollary 1 *For any formula $\phi \in \mathsf{BBL}$, we have that*

$$\mathcal{M}, \mathsf{dir} \models \phi \text{ iff } \mathcal{K}_\mathcal{M}, \overline{\mathsf{dir}} \models \phi.$$

3.2.4 Vision-based models

Now, we are going to extract from vision-based abstractions of geometric Kripke structures a new, simpler and more transparent type of models for BBL, which we call a *vision-based model*. First, note that, given a vision-based abstraction $\mathcal{K}_\mathcal{M} = (\overline{\mathsf{D}}, \overline{R}, \overline{\sim})$ of a geometric Kripke structure $\mathcal{M}$, each $\overline{\mathsf{dir}} \in \overline{\mathsf{D}}$ is entirely characterized by the possible families of sets – one family for each camera – of other agents that the camera can see in a glance if suitably turned. Because of this, we can identify any possible world in $\overline{\mathsf{D}}$ with the tuple stating for each agent the set of other agents that it sees. Of course, we can precompute these tuples. Moreover, for each agent a we can pre-compute the family S_a of the sets of agents that can be seen at the same time by a, by using standard analytic geometry (we omit the routine details).

Thus, starting with a geometric Kripke structure $\mathcal{M}$ we can eventually construct a tuple of families of sets of agents $\mathcal{S} = (S_\mathsf{a})_{\mathsf{a} \in \mathsf{Agt}}$ that contains all the relevant information to provide the semantics for BBL.

The formal definition follows.

Definition 4 Let $\mathcal{M} = (\mathsf{pos}, \mathsf{ang}, \mathsf{D}, R, \sim)$ be a geometric Kripke structure and $\mathcal{K}_\mathcal{M} = (\overline{\mathsf{D}}, \overline{R}, \overline{\sim})$ be its vision-based abstraction. For any $\overline{\mathsf{dir}} \in \overline{\mathsf{D}}$ we define:

- $I_\mathsf{a}^{\overline{\mathsf{dir}}} = \{\mathsf{b} \in \mathsf{Agt} \mid \mathsf{pos}(\mathsf{b}) \in C_{\mathsf{pos}(\mathsf{a}), \mathsf{dir}(\mathsf{a}), \mathsf{ang}(\mathsf{a})}\}$ is the set of agents that a sees in (any) direction given by $\overline{\mathsf{dir}}$;
- $(I_\mathsf{a}^{\overline{\mathsf{dir}}})_{\mathsf{a} \in \mathsf{Agt}}$ is the tuple of sets of agents seen by the respective agents in directions given by $\overline{\mathsf{dir}}$;
- $S_\mathsf{a} = \{I_\mathsf{a}^{\overline{\mathsf{dir}}} \mid \overline{\mathsf{dir}} \in \overline{\mathsf{D}}\}$ is the family of all sets of agents that a can see when turning around.

Thus, we construct a tuple of families of sets of agents $\mathcal{S}(\mathcal{M}) = (S_\mathsf{a})_{\mathsf{a} \in \mathsf{Agt}}$, which we call *the spectrum of vision of* $\mathcal{M}$.

Example 2 In the geometrical model in Example 1, for the given angles we have, the families in the spectrum of vision are computed as follows:

- $S_\mathsf{a} = \{\emptyset, \{\mathsf{b}\}, \{\mathsf{b}, \mathsf{e}\}, \{\mathsf{b}, \mathsf{d}, \mathsf{e}\}, \{\mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}\}, \{\mathsf{c}, \mathsf{d}, \mathsf{e}\}, \{\mathsf{c}, \mathsf{d}\}, \{\mathsf{c}\}\})$
- $S_\mathsf{b} = \{\emptyset, \{\mathsf{a}\}, \{\mathsf{d}\}, \{\mathsf{c}, \mathsf{d}\}, \{\mathsf{c}\}, \{\mathsf{c}, \mathsf{e}\}, \{\mathsf{e}\}\}$;
- $S_\mathsf{c} = \{\emptyset, \{\mathsf{a}\}, \{\mathsf{a}, \mathsf{d}\}, \{\mathsf{a}, \mathsf{d}, \mathsf{b}\}, \{\mathsf{b}, \mathsf{d}\}, \{\mathsf{b}, \mathsf{e}\}, \{\mathsf{e}\}\}$;

– $S_{\mathsf{d}} = \{\emptyset, \{\mathsf{a}\}, \{\mathsf{b}\}, \{\mathsf{e}\}, \{\mathsf{c}, \mathsf{e}\}, \{\mathsf{c}\}\}$;
– $S_{\mathsf{e}} = \{\emptyset, \{\mathsf{b}\}, \{\mathsf{a}, \mathsf{b}\}, \{\mathsf{a}, \mathsf{b}, \mathsf{d}\}, \{\mathsf{a}, \mathsf{d}\}, \{\mathsf{d}\}, \{\mathsf{e}\}\}$.

Proposition 2 *Let $\mathcal{M} = (\mathsf{pos}, \mathsf{ang}, \mathsf{D}, R, \sim)$ be any geometric Kripke structure. Each set S_{a} in the spectrum of vision of $\mathcal{M}$ contains at most $2k-2$ sets of agents, where $k = \#\mathsf{Agt}$ is the number of agents. Hence, this is the maximal size of the set $\overline{R}_{\mathsf{a}}(\overline{\mathsf{dir}})$ in the vision-based abstraction of $\mathcal{M}$.*

Proof Consider an agent a and start turning its camera clockwise. The set of agents seen by a's camera only changes when one of the two boundary rays of the sector of vision of the camera passes through an agent. In one full circle each of these rays passes through $k-1$ agents.

Note that not every tuple of families of sets of agents $(S_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}}$ is a spectrum of vision of some geometric Kripke structure, even if it satisfies the constraints in Proposition 2. Nevertheless, we can regards every such tuple as if it is an actual spectrum of vision, so we will call every such tuple an *abstract spectrum of vision*.

We will show that on every abstract spectrum of vision a Kripke model for BBL can be built and will give semantics of BBL on such models, as follows.

Definition 5 Given an abstract spectrum of vision $\mathcal{S} = (S_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}}$, the *vision-based model over* $\mathcal{S}$ is a Kripke structure $\mathcal{V}(\mathcal{S}) = (\mathbf{\Gamma}, \mathcal{T}, \mathcal{E})$ where:

– $\mathbf{\Gamma} = \{(\Gamma_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}} \mid \Gamma_{\mathsf{a}} \in S_{\mathsf{a}} \text{ for all } \mathsf{a} \in \mathsf{Agt}\}$,
– $\mathcal{T}$ maps each agent b to the equivalence relation
$\mathcal{T}_{\mathsf{b}} = \{((\Gamma_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}}, (\Gamma'_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}}) \mid \Gamma_{\mathsf{a}} = \Gamma'_{\mathsf{a}} \text{ for all } \mathsf{a} \neq \mathsf{b}\}$,
– $\mathcal{E}$ maps each agent b to the equivalence relation
$\mathcal{E}_{\mathsf{b}} = \{((\Gamma_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}}, (\Gamma'_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}}) \mid \Gamma_{\mathsf{a}} = \Gamma'_{\mathsf{a}} \text{ for all } \mathsf{a} \in \{\mathsf{b}\} \cup \Gamma_{\mathsf{b}}\}$.
It is straightforward to check that the definition above is correct.

Thus, in summary, every geometric model $G = (\mathsf{pos}, \mathsf{dir}, \mathsf{ang})$ generates a geometric Kripke structure $\mathcal{M} = \mathcal{M}(G)$ and eventually also generates a vision-based model $\mathcal{V}(\mathcal{S}(\mathcal{M}))$ based on the spectrum of vision $\mathcal{S}(\mathcal{M})$ constructed as in Definition 4, with an "initial world" $\Gamma_{\mathsf{G}} = (\Gamma_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}}$, where $\Gamma_{\mathsf{a}} = \{\mathsf{b} \in \mathsf{Agt} \mid \mathsf{pos}(\mathsf{b}) \in C_{\mathsf{pos}(\mathsf{a}),\mathsf{dir}(\mathsf{a}),\mathsf{ang}(\mathsf{a})}\}$ for every agent a.

Example 3 In Example 2 we computed the spectrum of vision $\mathcal{S} = (S_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}}$ generated by the geometrical model G in Example 1, which also defines the corresponding initial world $\Gamma_{\mathsf{G}} = (\Gamma_a)_{a\in\mathsf{Agt}}$, where:

– $\Gamma_a = \{c, d, e\} \in S_a$;
– $\Gamma_b = \emptyset \in S_b$;
– $\Gamma_c = \emptyset \in S_c$;
– $\Gamma_d = \{c\} \in S_d$;
– $\Gamma_e = \{a, b, d\} \in S_e$.

An example of another world, $\Gamma' = (\Gamma'_a)_{a\in\mathsf{Agt}}$, such that $\Gamma_{\mathsf{G}} \mathcal{T}_b \Gamma'$, is:

– $\Gamma_a = \{c, d, e\} \in S_a$;

– $\Gamma_b = \{c, d\} \in S_b$;
– $\Gamma_c = \emptyset \in S_c$;
– $\Gamma_d = \{c\} \in S_d$;
– $\Gamma_e = \{a, b, d\} \in S_e$.

We also have $\Gamma_{\mathsf{G}} \mathcal{E}_a \Gamma'$, $\Gamma_{\mathsf{G}} \mathcal{E}_c \Gamma'$ and $\Gamma_{\mathsf{G}} \mathcal{E}_d \Gamma'$, but neither $\Gamma_{\mathsf{G}} \mathcal{E}_b \Gamma'$ nor $\Gamma_{\mathsf{G}} \mathcal{E}_e \Gamma'$.

Now, we can readily give semantics of BBL in vision-based models, via the following clauses:

– $\mathcal{V}(\mathcal{S}), (\Gamma_{\mathsf{a}})_{\mathsf{a} \in \mathsf{Agt}} \models \mathsf{b} \triangleright \mathsf{c}$ iff $\mathsf{c} \in \Gamma_{\mathsf{b}}$,
– $\mathcal{V}(\mathcal{S}), (\Gamma_{\mathsf{a}})_{\mathsf{a} \in \mathsf{Agt}} \models \langle \overset{\curvearrowright}{\mathsf{a}} \rangle \phi$ iff $\mathcal{V}(\mathcal{S}), (\Gamma'_{\mathsf{a}})_{\mathsf{a} \in \mathsf{Agt}} \models \phi$ for some $(\Gamma'_{\mathsf{a}})_{\mathsf{a} \in \mathsf{Agt}} \in \mathbf{\Gamma}$ such that $((\Gamma'_{\mathsf{a}})_{\mathsf{a} \in \mathsf{Agt}}, (\Gamma_{\mathsf{a}})_{\mathsf{a} \in \mathsf{Agt}}) \in \mathcal{T}_{\mathsf{a}}$,
– $\mathcal{V}(\mathcal{S}), (\Gamma_{\mathsf{a}})_{\mathsf{a} \in \mathsf{Agt}} \models \mathsf{K}_{\mathsf{a}} \phi$ iff $\mathcal{V}(\mathcal{S}), (\Gamma'_{\mathsf{a}})_{\mathsf{a} \in \mathsf{Agt}} \models \phi$ for all $(\Gamma'_{\mathsf{a}})_{\mathsf{a} \in \mathsf{Agt}} \in \mathbf{\Gamma}$ such that $((\Gamma'_{\mathsf{a}})_{\mathsf{a} \in \mathsf{Agt}}, (\Gamma_{\mathsf{a}})_{\mathsf{a} \in \mathsf{Agt}}) \in \mathcal{E}_{\mathsf{a}}$.

In particular, the semantics above applies to vision-based models generated by spectra of vision of geometric Kripke structures, thus providing an alternative semantics in such structures. However, both semantics are easily shown to be equivalent. Indeed, a straightforward inspection shows that, for every geometric Kripke structure $\mathcal{M}$ its vision-based abstraction $\mathcal{K}_{\mathcal{M}}$ is isomorphic to the vision-based model over its spectrum of vision, constructed in Definition 5:

Proposition 3 *For every vision-based abstraction $\mathcal{K}_{\mathcal{M}} = (\overline{\mathsf{D}}, \overline{R}, \overline{\approx})$ the mapping $h : \overline{\mathsf{D}} \to \mathbf{\Gamma}$ defined by $h(\overline{\mathsf{dir}}) = (\Gamma_{\mathsf{a}}^{\overline{\mathsf{dir}}})_{\mathsf{a} \in \mathsf{A}}$ is an isomorphism between $\mathcal{K}_{\mathcal{M}}$ and $\mathcal{V}(\mathcal{S}(\mathcal{M}))$.*

Corollary 2 *For any formula $\phi \in \mathsf{BBL}$, we have that*

$$\mathcal{K}_{\mathcal{M}}, \overline{\mathsf{dir}} \models \phi \quad \textit{iff} \quad \mathcal{V}(\mathcal{S}(\mathcal{M})), (\Gamma_{\mathsf{a}}^{\overline{\mathsf{dir}}})_{a \in \mathsf{A}} \models \phi.$$

We will henceforth identify the vision-based abstraction $h : \mathcal{K}_{\mathcal{M}}$ with the vision-based model $\mathcal{V}(\mathcal{S}(\mathcal{M}))$ and, when $\mathcal{M}$ is fixed by the context, will denote possible worlds in $\mathcal{V}(\mathcal{S}(\mathcal{M}))$ by $(\Gamma_{\mathsf{a}})_{\mathsf{a} \in \mathsf{Agt}}$, dropping the superscript $\overline{\mathsf{dir}}$.

Thus, in the long run, we have shown that all semantics defined above are equivalent, so we can work hereafter with any of them.

Definition 6 A BBL formula is *satisfiable* iff it is true in some geometric model (hence, in some geometric Kripke structure or in some vision-based model of such structure); it is *valid* iff it is true in every geometric model, respectively every geometric Kripke structure or every vision-based model of such structure.

4 Expressiveness and extensions

4.1 Expressing properties and specifications

The language of BBL and its extensions can express various natural specifications regarding the visual abilities and knowledge of the agents. Here are some examples in BBL and in $\mathsf{BBL}_{\mathsf{CD}}$:

- $\mathsf{a} \rhd \mathsf{b} \to \mathsf{K}_a[\overset{\curvearrowright}{\mathsf{b}}](\mathsf{a} \rhd \mathsf{b})$: "If a can see b, then a knows that whichever way b turns around, this will remain true."

- $(\mathsf{b} \rhd \mathsf{a} \wedge \mathsf{b} \not\rhd \mathsf{c}) \wedge \mathsf{K}_b\langle \overset{\curvearrowright}{\mathsf{a}} \rangle((\mathsf{a} \rhd \mathsf{b} \wedge \mathsf{a} \rhd \mathsf{c}) \to \hat{\mathsf{K}}_b\mathsf{K}_a(\mathsf{b} \rhd \mathsf{a} \wedge \mathsf{b} \not\rhd \mathsf{c})$:
"If b sees a but not c and knows that a can turn around so as to see both b and c then b consider it possible that a knows that b sees a but not c."

- $(\mathsf{a} \bowtie \mathsf{b}) \to \mathsf{C}_{\mathsf{a},\mathsf{b}}(\mathsf{a} \bowtie \mathsf{b})$:
"If a and b see each other, then this is a common knowledge amongst them."

- $\mathsf{K}_{\mathsf{a},\mathsf{b}}(\langle \overset{\curvearrowright}{\mathsf{a}} \rangle(\mathsf{a} \rhd \mathsf{b} \wedge \mathsf{a} \rhd \mathsf{c}) \wedge \langle \overset{\curvearrowright}{\mathsf{b}} \rangle(\mathsf{b} \rhd \mathsf{a} \wedge \mathsf{b} \rhd \mathsf{c}))$
$\to \mathsf{C}_{\mathsf{a},\mathsf{b}}(\langle \overset{\curvearrowright}{\mathsf{a}} \rangle\langle \overset{\curvearrowright}{\mathsf{b}} \rangle(\mathsf{a} \bowtie \mathsf{b} \wedge \mathsf{a} \rhd \mathsf{c} \wedge \mathsf{b} \rhd \mathsf{c}))$.
"If a and b know that each of them can turn around to see the other and c, then it is a common knowledge amongst them that they can both turn so as to see each other and c."

- $\langle \overset{\curvearrowright}{\mathsf{a}} \rangle(\mathsf{a} \rhd \mathsf{c} \wedge \mathsf{a} \rhd \mathsf{d}) \wedge \langle \overset{\curvearrowright}{\mathsf{b}} \rangle(\mathsf{b} \rhd \mathsf{d} \wedge \mathsf{b} \rhd \mathsf{e}) \to \langle \overset{\curvearrowright}{\mathsf{a}} \rangle\langle \overset{\curvearrowright}{\mathsf{b}} \rangle(\mathsf{D}_{\mathsf{a},\mathsf{b}}\mathsf{c} \rhd \mathsf{e} \vee \mathsf{D}_{\mathsf{a},\mathsf{b}}\mathsf{c} \not\rhd \mathsf{e})$.
"If a can turn around so as to see both c and d and b can turn around so as to see both d and e then both a and b can turn around so as to make it a distributed knowledge amongst them whether c sees e."

(The reader is invited to try to check by hand whether this formula is valid, in order to appreciate the complexity of the semantics.)

4.2 Defining betweenness and collinearity

Assuming that the cameras' angles of vision are strictly less that 2π, the relations of betweenness and collinearity can be expressed in the language of BBL as follows:

- The formula $[\overset{\curvearrowright}{\mathsf{a}}](\mathsf{a} \rhd \mathsf{b} \to \mathsf{a} \rhd \mathsf{c})$ is true in a geometric model precisely when c lies on the ray starting at a and passing through b, i.e., when b is between a and c or c is between a and b.
- Therefore, the formula $[\overset{\curvearrowright}{\mathsf{a}}](\mathsf{a} \rhd \mathsf{b} \to \mathsf{a} \rhd \mathsf{c}) \wedge [\overset{\curvearrowright}{\mathsf{b}}](\mathsf{b} \rhd \mathsf{a} \to \mathsf{b} \rhd \mathsf{c})$ is true in a geometric model precisely when c lies strictly between a and b. Thus, we can define

$$\mathsf{B}\,(\mathsf{acb}) := [\overset{\curvearrowright}{\mathsf{a}}](\mathsf{a} \rhd \mathsf{b} \to \mathsf{a} \rhd \mathsf{c}) \wedge [\overset{\curvearrowright}{\mathsf{b}}](\mathsf{b} \rhd \mathsf{a} \to \mathsf{b} \rhd \mathsf{c})$$

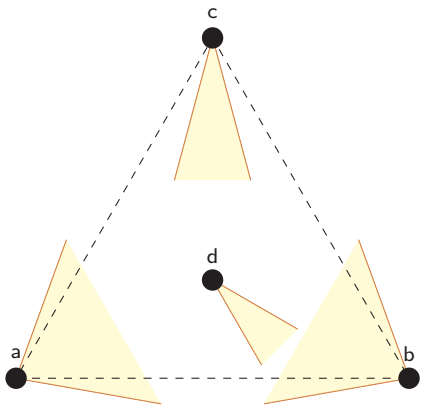


Fig. 3 Expressing the claim that d is in the triangle $\triangle(\mathsf{a},\mathsf{b},\mathsf{c})$ when the angles of vision a and b are wide enough: whenever a sees both b and c it sees d, too, and whenever b sees both a and c it sees d, too.

One can express likewise that d is in the triangle $\triangle(\mathsf{a},\mathsf{b},\mathsf{c})$ but only when at least 2 of the angles of vision of a,b,c are wide enough so each of the respective agents can turn around to see the other two. For instance, suppose that the angles of vision a and b are wide enough as depicted in figure 3. Then we express that d is in the triangle $\triangle(\mathsf{a},\mathsf{b},\mathsf{c})$ by the following formula:

$$[\overset{\curvearrowright}{\mathsf{a}}]((\mathsf{a}\rhd\mathsf{b}\wedge\mathsf{a}\rhd\mathsf{c})\to\mathsf{a}\rhd\mathsf{d})\wedge[\overset{\curvearrowright}{\mathsf{b}}]((\mathsf{b}\rhd\mathsf{a}\wedge\mathsf{b}\rhd\mathsf{c})\to\mathsf{b}\rhd\mathsf{d}).$$

Note that the modal language for plane geometry with betweenness is quite expressive (see [2]), but BBL covers only a small fragment of it because it cannot refer directly to lines, points and incidence. Nevertheless, if cameras can change their positions in the plane the expressiveness of BBL increases substantially and its full power is still subject to further investigation.

4.3 Relations between vision and knowledge

Clearly, seeing and visual knowledge are closely related: an agent knows that he sees whatever he sees: $\mathsf{a}\rhd\mathsf{b}\to\mathsf{K}_\mathsf{a}\mathsf{a}\rhd\mathsf{b}$ and knows that he does not see whatever he does not see: $\mathsf{a}\not\rhd\mathsf{b}\to\mathsf{K}_\mathsf{a}\mathsf{a}\not\rhd\mathsf{b}$. On the other hand, there is more in the agent's visual knowledge than what he can see directly. For instance, the following is valid for any agents $\mathsf{a},\mathsf{b},\mathsf{c}_1,\mathsf{c}_2,\mathsf{e}$, even if a does not see b:

$$\mathsf{a}\rhd\mathsf{c}_1\wedge\mathsf{a}\rhd\mathsf{c}_2\wedge\mathsf{B}\,(\mathsf{c}_1\mathsf{e}\mathsf{c}_2)\to\mathsf{K}_\mathsf{a}(\mathsf{b}\rhd\mathsf{c}_1\wedge\mathsf{b}\rhd\mathsf{c}_2\to\mathsf{b}\rhd\mathsf{e})$$

Thus, a has non-trivial (i.e., not logically valid) knowledge about b's visual abilities.

For more on the relations between vision and knowledge see [1].

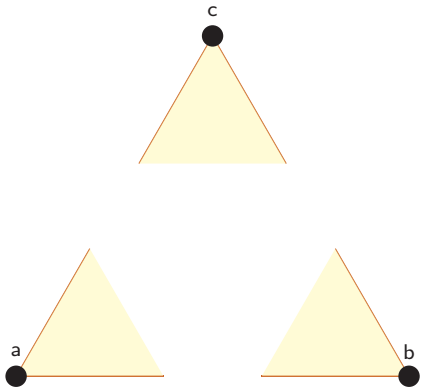


Fig. 4 See(a, b, c) is $\frac{\pi}{3}$-realizable

4.4 Axioms for cameras with different angles of vision

We will not discuss axiomatic systems for BBL-logics here, but will only demonstrate that the set of valid BBL-formulae is very sensitive to the admissible angles of vision of the agents' cameras.

For simplicity, we will assume here that all cameras have the same angle of vision α. Presumably, a given BBL-formula can be valid/satisfiable for one α and not for another, whence the following definition.

Definition 7 Given an angle α such that $0 \leq \alpha \leq \pi$, a formula $\phi \in \mathsf{BBL}$ is *α-realizable* if there is a geometric Kripke structure $\mathcal{M} = (\mathsf{pos}, \mathsf{ang}, \mathsf{D}, R, \sim)$ where $\mathsf{ang}(\mathsf{a}) = \alpha$ for each $\mathsf{a} \in \mathsf{Agt}$, which satisfies ϕ.

Consider for instance the formula

$$\mathrm{See}(\mathsf{a}, \mathsf{b}, \mathsf{c}) = \mathsf{a} \rhd \mathsf{b} \wedge \mathsf{a} \rhd \mathsf{c} \wedge \mathsf{b} \rhd \mathsf{a} \wedge \mathsf{b} \rhd \mathsf{c} \wedge \mathsf{c} \rhd \mathsf{a} \wedge \mathsf{c} \rhd \mathsf{b},$$

saying that each of these 3 agents sees the other two. Clearly, this can only be satisfied in the Euclidean plane if the sum of their cameras' angles of vision is at least π because each of them must see the entire triangle with vertices where the cameras are positioned. Thus, assuming all angles of vision equal to α, a necessary and sufficient condition for See(a, b, c) to be α-realizable is $\alpha \geq \pi/3$. Figure 4 depicts a geometrical model that shows that See(a, b, c) is $\frac{\pi}{3}$-realizable.

To generalise, consider the formula

$$\mathrm{See}(\mathsf{a}_1, \ldots, \mathsf{a}_n) = \bigwedge_{k=1}^{n} \bigwedge_{i=1, i \neq k}^{n} \mathsf{a}_k \rhd \mathsf{a}_i$$

saying that every agent in the set $\{\mathsf{a}_1, \ldots, \mathsf{a}_n\}$ can see all others. For that formula to be satisfied in the Euclidean plane, all agents in that set must

be arranged in the vertices of a convex polygon and see the entire polygon, hence the sum of their cameras' angles of vision must be at least $(n-2)\pi$. In particular, $\mathrm{See}(\mathsf{a}_1,\ldots,\mathsf{a}_n)$ is α-realizable if and only if $\alpha \geq (n-2)\pi/n$.

Consequently, for each $n \in \mathbb{N}$, the BBL-formula $\neg\mathrm{See}(\mathsf{a}_1,\ldots,\mathsf{a}_n)$ is valid in all geometric Kripke structures for BBL where all cameras have the same angle of vision α, if $\alpha < (n-2)\pi/n$, but fails in suitable geometric Kripke structures where all cameras have the same angle of vision which is at least $(n-2)\pi/n$.

Thus, there are infinitely many different axiomatic systems for BBL-logics, depending on the admissible range of angles of vision of the cameras. Moreover, modifications of the formula $\mathrm{See}(\mathsf{a}_1,\ldots,\mathsf{a}_n)$ involving both positive and negative atoms would require more involved conditions for α-realizability and the general problem of computing the realizability conditions for any given such formula is currently open. Some recent progress is reported in [11]. We only conjecture here that every interval $[a\pi, b\pi]$, for rational a, b such that $0 \leq a \leq b \leq 1$, determines a unique set of BBL-formulae that are valid in all geometric Kripke structures for BBL with angles of vision of all cameras in the interval $[a\pi, b\pi]$.

Even the case of $\alpha = \pi$ involves many non-trivial validities, expressing special instances of the Pigeonhole principle. Indeed, note that n lines in the plane, delineating the half-planes of vision of n cameras, partition the plane in at most $R(n) = 1 + 1 + 2 + \ldots + n = n(n+1)/2 + 1$ regions (when in general position, i.e., no parallels and no three concurrent lines). So, there can be at most $R(n)$ many different subsets of these n agents seeing a given point/agent. Therefore, the following formula scheme is valid, saying that for any set of $R(n)+1$ agents, two of them must be seen by the same subset of given n cameras.

$$\eta_k = \bigvee_{i=1}^{R(n)+1} \bigvee_{j=1, j\neq i}^{R(n)+1} \bigwedge_{k=1}^{n} \mathsf{a}_k \rhd \mathsf{c}_i \leftrightarrow \mathsf{a}_k \rhd \mathsf{c}_j$$

We claim that the infinite set of axioms $\{\eta_k \mid k \in \mathbb{N}\}$ cannot be replaced by any finite subset and that, consequently, the logic BBL for infinitely many cameras with angles of vision π is not finitely axiomatizable. We leave the proof of this claim and the further study of axiomatic systems and completeness results for BBL logics to future work.

4.5 Public announcements

In our framework, agents get knowledge from what they perceive. But they may also need additionnal information to achieve their surveillance mission. For example, if they are surveying several targets, they may need to inform each other of where they are actually seeing. For example, with positions as in example 1, suppose that c, d and e are targets to be watched by a and b. When b is looking towards any of the targets, it cannot be aware at the same time

that a is currently seeing both c and e (and b by the way). But it may if it is told so more or less directly, for instance: after b is informed that $\mathsf{a} \not\rhd \mathsf{b} \wedge \mathsf{a} \rhd \mathsf{e}$ then $\mathsf{K_b}(\mathsf{a} \rhd \mathsf{c} \wedge \mathsf{a} \rhd \mathsf{d} \wedge \mathsf{a} \rhd \mathsf{e})$ will be true.

Let us provide the precise details of a basic framework for such communication called *public announcements* [13] where true new facts are publicly announced, and thus learnt by each of the agents. The underlying modeling assumption for public announcements is that there is common knowledge that the communication channel is reliable and that communication is instantaneous.

We add public announcement operators, that is, we add constructions of the form $[\phi!]\psi$ reading 'if the formula ϕ is true, then after the announcement of ϕ, the formula ψ holds'. In this framework an announcement action $[\phi!]$ acts as a model modifyer which deletes the worlds in the model that do not satisfy ϕ. The semantics is based on submodels M of a vision-based model and is defined as in [13]. We extend the truth condition relation $M, (\Gamma_{\mathsf{a}})_{\mathsf{a} \in \mathsf{Agt}} \models \phi$ with the following additional clause:

- $M, (\Gamma_{\mathsf{a}})_{\mathsf{a} \in \mathsf{Agt}} \models [\phi!]\psi$ iff $M, (\Gamma_{\mathsf{a}})_{\mathsf{a} \in \mathsf{Agt}} \models \phi$ implies $M^{\phi}, (\Gamma_{\mathsf{a}})_{\mathsf{a} \in \mathsf{Agt}} \models \psi$ where M^{ϕ} is the restriction of M to ϕ-worlds (i.e. worlds where ϕ is true).

Example 4 Let us consider the geometric model of the example 1. After announcing $\mathsf{a} \rhd \mathsf{e} \wedge \mathsf{a} \rhd \mathsf{c}$, it is common knowledge that $\mathsf{a} \rhd \mathsf{d}$.

5 PDL-like form of BBL

Here we will adopt a different, PDL-like perspective on the logic BBL. For background on the propositional dynamic logic PDL refer e.g., to [12]. We can consider every sequence of (non-deterministic) turns of cameras as a program, where turning the camera of an agent a at an unspecified angle is an atomic program, denoted by $\overset{\curvearrowright}{\mathsf{a}}$. Besides, we can consider tests as atomic programs and build complex programs by applying compositions, unions and iterations to these, like in PDL. The technical advantage is that, as we will show, all knowledge operators can be represented in terms of suitable programs, and eventually the whole language of BBL and all of its extensions considered here can be translated to versions of PDL.

Besides programs from PDL, we also add public announcement operator in the language in order to capture public announcement in BBL. More precisely, let us first introduce a two-sorted language, denoted BBL_{PDL^*}, with sorts for formulae ϕ and for programs γ, as follows, where $\mathsf{a}, \mathsf{b} \in \mathsf{Agt}$.

$$\phi ::= \mathsf{a} \rhd \mathsf{b} \mid \neg\phi \mid \phi \vee \phi \mid \mathsf{K_a}\phi \mid [\gamma]\phi \mid [\phi!]\phi; \quad \gamma ::= \overset{\curvearrowright}{\mathsf{a}} \mid \phi? \mid \gamma;\gamma \mid \gamma \cup \gamma \mid \gamma^*$$

The language without the iteration * is called BBL_{PDL}. The intuitive meaning of the program operations is as in PDL, e.g., $\phi?$ is the program that tests whether ϕ is true or not, $\gamma_1;\gamma_2$ is the sequential composition of γ_1 and γ_2, while $\gamma_1 \cup \gamma_2$ represents the non-deterministic choice between γ_1 and γ_2. The

formula $[\gamma]\phi$ is read as "after all executions of γ, the property ϕ holds". We also define the dual operator $\langle\gamma\rangle$ by $\langle\gamma\rangle\phi := \neg[\gamma]\neg\phi$, meaning "there exists an execution of γ such that ϕ holds afterwards".

Example 5 For instance, the program "if a does not see b then a turns" is represented by $\gamma = (\neg \mathsf{a} \rhd \mathsf{b})?; \overset{\curvearrowright}{\mathsf{a}}$.

Let $\mathcal{S} = (S_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}}$ be an abstract spectrum of vision. The formal semantics is given in restrictions of the vision-based model over $\mathcal{S}$. First let us extend $\mathcal{T}$ in order to interpret programs:

- $\mathcal{T}_{\mathsf{b}}$ is defined as in definition 5;
- $\mathcal{T}_{\gamma_1;\gamma_2} = \mathcal{T}_{\gamma_1} \circ \mathcal{T}_{\gamma_2}$;
- $\mathcal{T}_{\gamma_1\cup\gamma_2} = \mathcal{T}_{\gamma_1} \cup \mathcal{T}_{\gamma_2}$;
- $\mathcal{T}_{\gamma^*} = \bigcup_{n\in\mathbb{N}} \mathcal{T}_{\gamma}^{n}$.

Now, we generalize the relation $\mathcal{V}(\mathcal{S}), (\Gamma_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}} \models \phi$ to restrictions of vision-based abstraction models as follows. If $\mathcal{V}(\mathcal{S}) = (\mathbf{\Gamma}, \mathcal{T}, \mathcal{E})$ is the vision-based abstraction model over $\mathcal{S}$, we write $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}', (\Gamma_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}} \models \phi$ to say that formula ϕ holds in world $(\Gamma_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}}$ when we restrict to the worlds in $\mathbf{\Gamma}'$ such that $(\Gamma_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}} \in \mathbf{\Gamma}'$ and $\mathbf{\Gamma}' \subseteq \mathbf{\Gamma}$.

Definition 8 Formally:

- $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}', (\Gamma_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}} \models [\gamma]\phi$ iff for all $(\Gamma'_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}} \in \mathcal{T}_{\gamma}((\Gamma_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}})$, if $(\Gamma'_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}} \in \mathbf{\Gamma}'$ then $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}', (\Gamma'_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}} \models \phi$;
- $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}', (\Gamma_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}} \models \mathsf{K}_{\mathsf{a}}\phi$ iff for all $(\Gamma'_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}} \in \mathcal{E}_{\mathsf{a}}((\Gamma_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}})$, if $(\Gamma'_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}} \in \mathbf{\Gamma}'$ then $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}', (\Gamma'_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}} \models \phi$;
- $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}', (\Gamma_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}} \models [\phi!]\psi$ iff $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}', (\Gamma_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}} \models \phi$ implies $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}' \cap \{(\Gamma'_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}} \mid \mathcal{V}(\mathcal{S}), \mathbf{\Gamma}', (\Gamma'_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}} \models \phi\}, (\Gamma_{\mathsf{a}})_{\mathsf{a}\in\mathsf{Agt}} \models \psi$.

5.1 Expressing the knowledge operators

Now, we will show how to simulate knowledge operators in terms of programs in BBL_{PDL}.

5.1.1 Expressing the individual knowledge operators

The (individual) knowledge operator K_a can be simulated with the following program, where $\{\mathsf{b}_1, \ldots, \mathsf{b}_n\} = \mathsf{Agt} \setminus \{\mathsf{a}\}$:

$$\gamma_{\mathsf{a}}^{K} = \left(\mathsf{a} \rhd \mathsf{b}_1? \cup (\mathsf{a} \not\rhd \mathsf{b}_1?; \overset{\curvearrowright}{\mathsf{b}_1})\right); \ldots; \left(\mathsf{a} \rhd \mathsf{b}_n? \cup (\mathsf{a} \not\rhd \mathsf{b}_n?; \overset{\curvearrowright}{\mathsf{b}_n})\right)$$

Proposition 4 *In every geometric Kripke structure $\mathcal{M}$, for every agent* a *the interpretations of γ_{a}^{K} and $\sim_{\mathsf{a}}$ in $\mathcal{M}$ are equal.*

Proof Consider a geometric Kripke structure $\mathcal{M} = (\mathsf{pos}, \mathsf{ang}, \mathsf{D}, R, \sim)$. Recall that for any $(\mathsf{dir}, \mathsf{dir}') \in \mathsf{D}^2$:

$\mathsf{dir} R_{\mathsf{a}} \mathsf{dir}'$ iff $\mathsf{dir}(\mathsf{b}) = \mathsf{dir}'(\mathsf{b})$ for all $\mathsf{b} \neq \mathsf{a}$, and

$\mathsf{dir} \sim_{\mathsf{a}} \mathsf{dir}'$ iff $\mathsf{dir}(\mathsf{a}) = \mathsf{dir}'(\mathsf{a})$ and $\mathsf{dir}(\mathsf{b}) = \mathsf{dir}'(\mathsf{b})$ for all b such that $\mathsf{a} \rhd \mathsf{b}$ holds, i.e., $\mathsf{pos}(\mathsf{b}) \in C_{\mathsf{pos}(\mathsf{a}),\mathsf{dir}(\mathsf{a}),\mathsf{ang}(\mathsf{a})}$.

Now, let $\mathsf{dir} \sim_{\mathsf{a}} \mathsf{dir}'$. Then we can obtain dir' from dir as follows: consecutively, for every agent $\mathsf{b} \in \{\mathsf{b}_1, \ldots, \mathsf{b}_n\} = \mathsf{Agt} \setminus \{\mathsf{a}\}$, do the following: if $\mathsf{a} \rhd \mathsf{b}$ do nothing, else turn around the camera of b so that $\mathsf{dir}(\mathsf{b})$ becomes equal to $\mathsf{dir}'(\mathsf{b})$. This action is clearly included in the interpretation of $\mathsf{a} \rhd \mathsf{b}? \cup (\mathsf{a} \not\rhd \mathsf{b}?; \overset{\curvearrowright}{\mathsf{b}})$, hence we obtain that $(\mathsf{dir}, \mathsf{dir}') \in R_{\gamma_{\mathsf{a}}^K}$. Thus, $\sim_{\mathsf{a}}$ is included in $R_{\gamma_{\mathsf{a}}^K}$.

Conversely, note that for each $\mathsf{b} \in \mathsf{Agt} \setminus \{\mathsf{a}\}$, the application of the program $\mathsf{a} \rhd \mathsf{b}? \cup (\mathsf{a} \not\rhd \mathsf{b}?; \overset{\curvearrowright}{\mathsf{b}_i})$ to any $\mathsf{dir} \in \mathsf{D}$ can change the direction of view of b if and only if a does not see b. Moreover, the direction of view of a remains unaffected. Thus, the program γ_{a}^K may only change (arbitrarily) the directions of view of all agents, other than a, that are not seen by a. Therefore, when applied to dir it always produces a result dir' such that $\mathsf{dir} \sim_{\mathsf{a}} \mathsf{dir}'$, whence the inclusion of $R_{\gamma_{\mathsf{a}}^K}$ in $\sim_{\mathsf{a}}$.

Group knowledge operator K_{A} is then simulated by the program: $\gamma_{\mathsf{A}}^K = \bigcup_{\mathsf{a} \in \mathsf{A}} \gamma_{\mathsf{a}}^K$.

5.1.2 Expressing distributed knowledge

The distributed knowledge operator D_{A} is simulated with the following program, where $\{\mathsf{b}_1, \ldots, \mathsf{b}_n\} = \mathsf{Agt} \setminus \mathsf{A}$:

$$\gamma_{\mathsf{A}}^D = \Big(\bigvee_{\mathsf{a} \in \mathsf{A}} \mathsf{a} \rhd \mathsf{b}_1? \bigcup \Big(\bigwedge_{\mathsf{a} \in \mathsf{A}} \mathsf{a} \not\rhd \mathsf{b}_1? \Big); \overset{\curvearrowright}{\mathsf{b}_1} \Big); \ldots; \Big(\bigvee_{\mathsf{a} \in \mathsf{A}} \mathsf{a} \rhd \mathsf{b}_n? \bigcup \Big(\bigwedge_{\mathsf{a} \in \mathsf{A}} \mathsf{a} \not\rhd \mathsf{b}_n? \Big); \overset{\curvearrowright}{\mathsf{b}_n} \Big)$$

The construction $\Big(\bigvee_{\mathsf{a} \in \mathsf{A}} \mathsf{a} \rhd \mathsf{b}_i? \bigcup \Big(\bigwedge_{\mathsf{a} \in \mathsf{A}} \mathsf{a} \not\rhd \mathsf{b}_i? \Big); \overset{\curvearrowright}{\mathsf{b}_i} \Big)$, as in the translation of individual knowledge operator, is a program that turns agent b_i if and only if no agent in A sees it. This reflects precisely the semantics of D_{A}.

5.1.3 Expressing common knowledge

In order to simulate the common knowledge operator C_{A}, we need BBL_{PDL^*}. Common knowledge among a group $\mathsf{A} = \{\mathsf{a}_1, \ldots, \mathsf{a}_k\}$ can be simulated by: $\gamma_{\mathsf{A}}^C = \Big(\gamma_{\mathsf{a}_1}^K; \ldots; \gamma_{\mathsf{a}_k}^K \Big)^*$. As the program $\gamma_{\mathsf{a}_i}^K$ represents the epistemic relation for agent a_i, the program γ_{A}^C corresponds to taking an arbitrary finite path along the epistemic relations for all agents in A. Note that this simulation is correct because the programs $\gamma_{\mathsf{a}_i}^K$ are reflexive and commute with each other.

5.2 Translation into BBL_{PDL} and BBL_{PDL^*}

Let τ be the translation from BBL into BBL_{PDL} and BBL_{PDL^*} defined by:
$\tau(\mathsf{a}\rhd\mathsf{b}) = \mathsf{a}\rhd\mathsf{b}$,
$\tau(\phi_1 \vee \phi_2) = \tau(\phi_1) \vee \tau(\phi_2)$,
$\tau(\neg\phi) = \neg\tau(\phi)$,
$\tau([\phi!]\psi) = [\tau(\phi)!]\tau(\psi)$
$\tau(X_\alpha\phi) = [\gamma^X_\alpha]\tau(\phi)$, for X is K, D or C.

The next claim follows immediately from the constructions of the knowledge simulating programs.

Proposition 5 *Given an abstract spectrum of vision $\mathcal{S} = (S_\mathsf{a})_{\mathsf{a}\in\mathsf{Agt}}$, let $\mathcal{V}(\mathcal{S}) = (\mathbf{\Gamma}, \mathcal{T}, \mathcal{E})$ be the vision-based model over $\mathcal{S}$ (def. 5). Let $\mathbf{\Gamma}' \subseteq \mathbf{\Gamma}$ and $(\Gamma_\mathsf{a})_{\mathsf{a}\in\mathsf{Agt}} \in \mathbf{\Gamma}'$. Then:*

$$\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}', (\Gamma_\mathsf{a})_{\mathsf{a}\in\mathsf{Agt}} \models \phi \text{ iff } \mathcal{V}(\mathcal{S}), \mathbf{\Gamma}', (\Gamma_\mathsf{a})_{\mathsf{a}\in\mathsf{Agt}} \models \tau(\phi).$$

6 Decision procedures

In this section, we address the model checking problem and the satisfiability problem for BBL and its extensions. All the decision problems involve formulas that may contain the public announcement operator. For any of these languages $\mathcal{L}$, the model checking problem $\mathrm{MC}(\mathcal{L})$, is defined as follows:

Input: a geometric model $\mathsf{G} = (\mathsf{pos}, \mathsf{dir}, \mathsf{ang})$ and a formula ϕ of $\mathcal{L}$.

Output: 'yes' iff $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}, \Gamma_\mathsf{G} \models \phi$ where $\mathcal{V}(\mathcal{S})$ is the vision-based model of G, $\mathbf{\Gamma}$ is the set of all worlds in $\mathcal{V}(\mathcal{S})$ and Γ_G is the initial world in $\mathcal{V}(\mathcal{S})$ that corresponds to G.

The satisfiability problem $\mathrm{SAT}(\mathcal{L})$ is defined as follows.

Input: a formula ϕ in $\mathcal{L}$;

Output: 'yes' iff there exists a geometric model $\mathsf{G} = (\mathsf{pos}, \mathsf{dir}, \mathsf{ang})$ such that $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}, \Gamma_\mathsf{G} \models \phi$ where $\mathcal{V}(\mathcal{S})$ is the vision-based model of G, $\mathbf{\Gamma}$ is the set of all worlds in $\mathcal{V}(\mathcal{S})$ and Γ_G is the initial world in $\mathcal{V}(\mathcal{S})$ that corresponds to G.

The satisfiability problem ANGLE-FIXED-SAT($\mathcal{L}$) is defined as follows.

Input: a formula ϕ in $\mathcal{L}$ and a function $\mathsf{ang}_0 : \mathsf{Agt} \to [0, 2\pi)$;

Output: 'yes' iff there exists a geometric model $\mathsf{G} = (\mathsf{pos}, \mathsf{dir}, \mathsf{ang})$ such that $\mathsf{ang} = \mathsf{ang}_0$ and $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}, \Gamma_\mathsf{G} \models \phi$ where $\mathcal{V}(\mathcal{S})$ is the vision-based model of G, $\mathbf{\Gamma}$ is the set of all worlds in $\mathcal{V}(\mathcal{S})$ and Γ_G is the initial world in $\mathcal{V}(\mathcal{S})$ that corresponds to G.

```
function ComputeVisionSets(G, a)
  T := circular array obtained by sorting the agents b ≠ a by increasing angle between an
  arbitrary unit vector (1, 0) and the ray from pos(a) to pos(b);
  if T is empty then return {∅};
  S_a := ∅

  i_begin := 0;
  i_end := 0;
  while (T[i_end + 1].angle − T[i_begin].angle)[2π] ≤ ang(a) do
    | i_end := i_end + 1;
  endWhile
  S_a.add(agents(i_begin, i_end)); (*)
  for i := 1 to 2 ∗ T.size − 1 do
    | if (T[i_end + 1].angle − T[i_begin].angle)[2π] > ang(a)) then
    |   | i_begin := i_begin + 1;
    | else
    |   | i_end := i_end + 1;
    | endIf
    | S_a.add(agents(i_begin, i_end));
  endFor
  return S_a;
endFunction
```

Fig. 5 An algorithm for computing S_a

6.1 Upper-bound for MC(BBL_{PDL^*})

The procedure for the model checking problems above consists in two stages: first compute sets S_a in the vision-based model (Def. 4) from the geometric model (Def. 1) and then do model checking on worlds of the vision-based model.

6.1.1 Computing the vision sets

Let us consider a geometrical model G and an agent a. For computing the set S_a, we consider a polar coordinate system centered in $\mathsf{pos}(\mathsf{a})$ and we consider all agents $b \neq a$ as a list sorted by angles. We perform a scan of all agents $b \neq a$ and we compute the possible sets of agents whose positions are in a sector centered in $\mathsf{pos}(\mathsf{a})$ and of angle $\mathsf{ang}(\mathsf{a})$. The set of those sets is exactly S_a. The idea is to identify the positions of the sector of vision of agent a where one of the boundary rays passes through some of the other agents, and determine the sets of agents seen by a in those positions. Let us give details of the algorithm that computes the set S_a from a geometrical model G and an agent a.

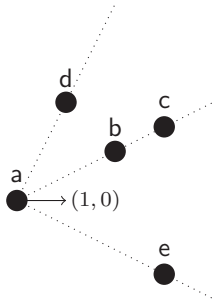
Figure 5 presents the function $ComputeVisionSets(\mathsf{G}, \mathsf{a})$ that takes as an input a geometrical model G and an agent a, and returns the set S_a.

The objective of the first line in the algorithm is to compute a polar representation, centered at $\mathsf{pos}(\mathsf{a})$, of the position of agents The computed array T is a circular array with indexes computed modulo the size of the array, that is, if we note $T.size$ the size of T, $T[0]$ is the first cell, $T[1]$ is the second cell,

etc. $T[T.size - 1]$ is the last cell, $T[T.size] = T[0]$, $T[T.size + 1] = T[1]$, etc. For all i, $T[i].angle$ is an angle in $[0, 2\pi)$, $T[i].agents$ is the set of agents that are in direction determined by $T[i].angle$ from agent a. Usually $T[i].agents$ is a singleton except when several agents are aligned with a. Furthermore, T is sorted by angles.

The integers i_{begin} and i_{end} represent respectively the first and the last index in T and they represent an interval of agents that may be seen by a. The corresponding set of agents is $agents(i_{begin}, i_{end}) = \bigcup_{i \in [i_{begin}, i_{end}]} T[i].agents$. The notation $(x - y)[2\pi]$ denotes the angle in $[0, 2\pi)$ congruent with $x - y$.

Example 6 For instance, if we consider the following geometrical model,



then the first line of the algorithm computes the following circular array T of size 3:

	T[0]	T[1]	T[2]
$T[i].angle$	$\pi/3$	$2\pi/3$	$11\pi/3$
$T[i].agents$	$\{b, c\}$	$\{d\}$	$\{e\}$

Note that index i is not for an agent but for an angle. For instance, $i = 0$ corresponds to both agents b and c.

To compute such a polar representation could be implemented as an algorithm in three steps:

```
if there are no other agents than a then exit

 1. Report all the agents different from a
T := an array of size the number of agents
i := 0
for b ∈ Agt with b ≠ a do
  | T[i].angle :=angle between (1,0) and (pos(a), pos(b))→
  | T[i].agent := b
  | i := i + 1
endFor

 2. Sort T by angles
Sort the array T by increasing value of T[i].angle

 3. Merge agents with the same angle
j := 0
for i := 1 to T.size − 1 do
  | if T[i].angle = T[j].angle then
  |   | T[j].agents := T[j].agents ∪ T[i].agents;
  | else
  |   | j := j + 1
  |   | T[j] := T[i]
  | endIf
endFor
T.size := j
```

All steps are linear in the number of agents except step 2 which can be implemented in $O(k \log k)$, where $k = \#\mathsf{Agt}$. So the first line of the algorithm of Figure 5 is in time $O(k \log k)$.

The rest of the algorithm of Figure 5 is a scan of agent a, that is, we simulate the rotation of agent a and we compute all the vision sets.

At $(*)$, we have computed the leftmost possible interval of agents that contains $T[0].agents$. Then we modify the current set of agents by adding the next agent of the list or by removing the last one if the next is too far.

Example 7 Let us consider again the array T of example 6. Suppose that $\mathsf{ang}(\mathsf{a}) = \frac{\pi}{2}$. Figure 6 depicts the step of the scan of agent a. Step 1 corresponds to $(*)$. Steps 2-6 corresponds to the loop **for** in figure 5. Transitions from step 1 to step 2 and from step 2 to step 3 correspond to the fact we can not extend the interval by keeping i_{begin} so that agent a sees all the agents in $agents(i_{begin}, i_{end})$ (we perform $i_{begin} := i_{begin} + 1$). Transitions from step 3 to step 4, from step 4 to step 5 and from step 5 to step 6 correspond to the situation where we can still extend the interval by keeping i_{begin} so that agent a still sees all the agents in $agents(i_{begin}, i_{end})$ (we perform $i_{end} := i_{end} + 1$).

The scan of agent a is linear in the number of agents. Therefore, the complexity of the computation of S_{a} is dominated by $O(k \log k)$ where $k = \#\mathsf{Agt}$.

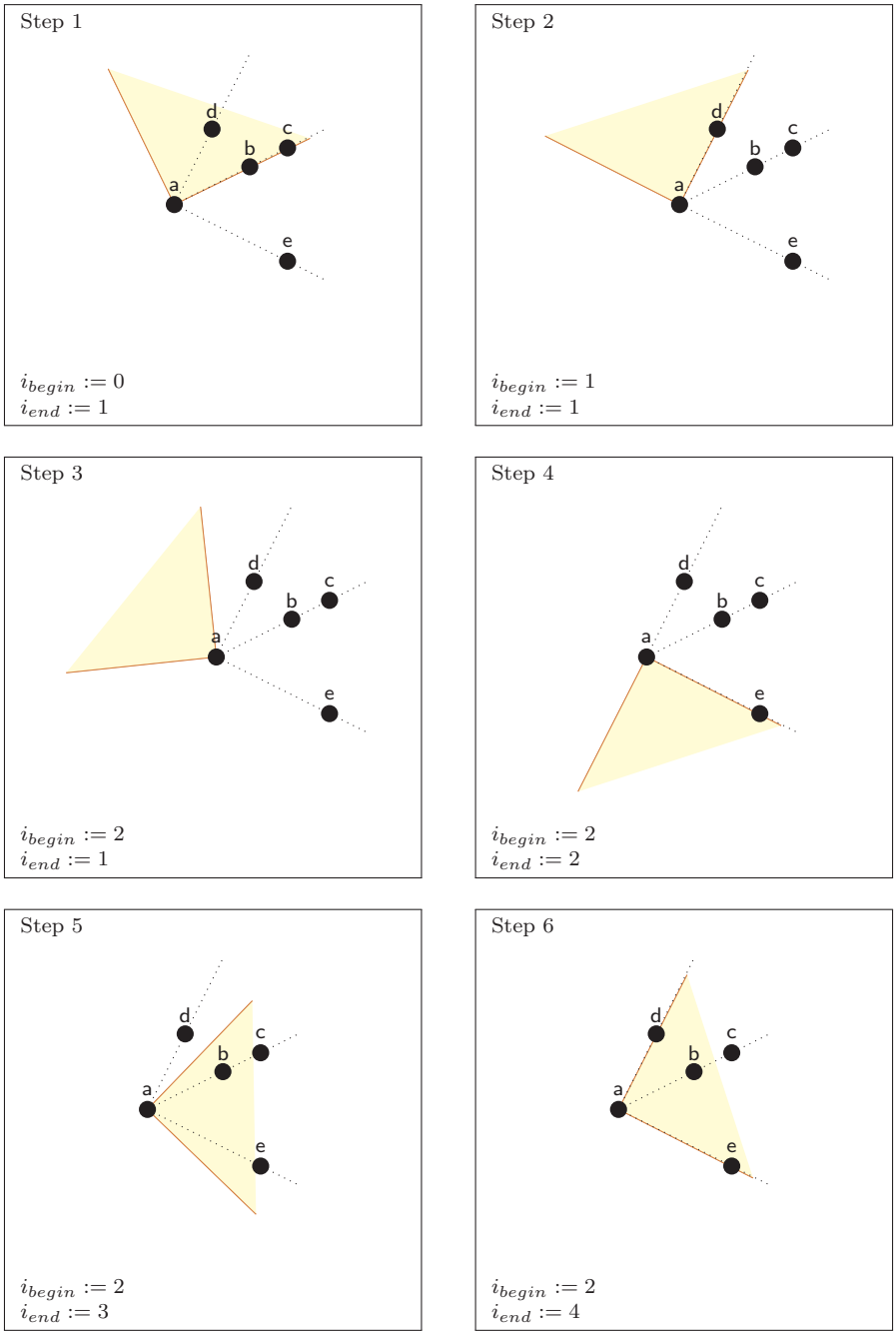


Fig. 6 Steps of a scan of agent a

6.1.2 Model checking in the vision-based model

Now, we design a procedure mc for model checking depicted in Figure 7. The specification of $mc(\Gamma, L, \phi)$ is to return true iff $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}_L, \Gamma \models \phi$ where $\mathbf{\Gamma}_L$ is the set of worlds that survive after the list of announcements in L, and Γ is a world in $\mathbf{\Gamma}_L$. If Γ is not a world in $\mathbf{\Gamma}_L$ and ϕ is not $\top$, then the value of the function is unspecified. The routine mc is inspired by standard ideas. It decomposes recursively the formula ϕ.

We use $\phi = \top$ to check that a given world is in $\mathbf{\Gamma}_L$. Note that $mc(\Gamma, L, \top)$ returns true if, and only if, the valuation Γ is still in $\mathbf{\Gamma}_L$. If $L = \epsilon$ $mc(\Gamma, \epsilon, \top)$ returns true, as Γ is always in $\mathbf{\Gamma} = \mathbf{\Gamma}_\epsilon$. Otherwise, we decompose the list of announcements L in $L' :: \psi$ and we ask whether Γ is in $\mathbf{\Gamma}_{L'}$ and that $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}_{L'}, \Gamma \models \psi$.

The treatment of announcements is highly inspired from the tableau method for public announcement designed in [3]. It may be quite surprising that a model checking procedure is inspired from a tableau method usually designed for solving the satisfiability problem. This is due to the fact that our model checking is symbolic in the sense that our Kripke structure is not fully represented in the memory. For instance, in [3] there is a rule saying that if $[\phi!]\psi$ is true in a world w after the sequence L, represented by a tableau term $(L, w, [\phi!]\psi)$, then we should either add the term $(L, w, \neg\phi)$ or the term $(L :: \phi, w, \psi)$ where $L :: \phi$ is the list L with the formula ϕ appended. This is very similar to the case $[\psi_1!]\psi_2$ in the algorithm mc.

The really interesting case concerns the model checking of a formula of the form $[\gamma]\psi$ where γ is a program, given by a regular expression. We check that ψ holds in all γ-successors Γ' of Γ by means of the subroutine $path?$. In order to check that a world Γ' is still in $\mathbf{\Gamma}_L$, we call $mc(\Gamma, L, \top)$. We also use a subroutine $path?$. The specification of the subroutine $path?$, given in Figure 8 is: $path?(\Gamma, \Gamma', \gamma)$ returns true iff there is a γ-path from Γ to Γ', i.e. a path in the graph (W, R) whose labels match the regular expression γ. For instance, if $\Gamma(R_{\mathsf{a}} \circ R_{\mathsf{a}} \circ R_{\mathsf{b}})\Gamma'$, then there is a $(\circlearrowright\mathsf{a}\,;(\circlearrowright\mathsf{a}\,;\circlearrowright\mathsf{b})^*)$-path from Γ to Γ' and therefore $path?\big(\Gamma, \Gamma', (\circlearrowright\mathsf{a}\,;(\circlearrowright\mathsf{a}\,;\circlearrowright\mathsf{b})^*)\big)$ is true.

The algorithm is recursive and decomposes the program γ. For instance, if $\gamma = \gamma_1;\gamma_2$, we ask for the existence of a γ_1-path from Γ to an intermediate world Γ'' (that is not required to be in $\mathbf{\Gamma}_L$, because this world is not the final world of a full program describing an epistemic relation) and a γ_2-path from Γ'' to Γ'.

In tests, i.e., programs of the from $\gamma = \phi?$, we check ϕ with respect to *no* announcements. Indeed, in practise, formulas ϕ are Boolean therefore announcements are not relevant for them. Furthermore, the current world Γ is an intermediate world in the execution of a program and the final world is reached by a full program describing an epistemic relation.

We want to prove that this model checking procedure works in polynomial space and the interesting case to consider is $\gamma = \gamma_1^*$. In order to treat this case, we introduce a new program construction: $\gamma = \gamma_1^{\mathbf{m}}$ where $\mathbf{m}$ is an integer

written in binary. The procedure *path*? uses the following simple observation: If there is a γ_1^*-path from Γ to Γ' then there is a $\gamma_1^{\mathbf{m}}$-path, where $0 \leq \mathbf{m} \leq 2^{k^2}$ (recall that $k = \#\mathsf{Agt}$, hence there are at most $(2^k)^k = 2^{k^2}$ distinct worlds, this provides an upper bound on the length of any elementary path). Thus, for the case of γ_1^*, we look for the existence of a $\gamma_1^{\mathbf{m}}$-path from Γ to Γ' for some $\mathbf{m} \in \{0, \ldots, 2^{k^2}\}$.

Now, for the case $\gamma_1^{\mathbf{m}}$ when $\mathbf{m} > 1$ we use the *Divide and Conquer* method and browse all possible 'intermediate' worlds Γ'' in a $\gamma_1^{\mathbf{m}}$-path from Γ to Γ', that is, there is a $\gamma_1^{\lfloor \frac{\mathbf{m}}{2} \rfloor}$-path from Γ to Γ'' and a $\gamma_1^{\lceil \frac{\mathbf{m}}{2} \rceil}$-path from Γ'' to Γ'.

Now, we prove that the number of nested recursive call of *mc* and *path*? is bounded by a polynomial in the size of the input. We define the following rank:

- $|a \rhd b| = 1$;
- $|\top| = 1$;
- $|\neg\psi| = 1 + |\psi|$;
- $|\psi_1 \wedge \psi_2| = 1 + |\psi_1| + |\psi_2|$;
- $|[\gamma]\psi| = 1 + |\gamma| + |\psi|$;
- $|[\psi_1!]\psi_2| = 1 + |\psi_1| + |\psi_2|$;
- $|\overset{\frown}{a}| = 1$;
- $|\gamma_1; \gamma_2| = 1 + |\gamma_1| + |\gamma_2|$;
- $|\gamma_1 \cup \gamma_2| = 1 + |\gamma_1| + |\gamma_2|$;
- $|\phi?| = 1 + |\phi|$;
- $|\gamma^*| = 1 + |\gamma| + \#\mathsf{Agt}^2 + 1$;
- $|\gamma^{\mathbf{m}}| = 1 + |\gamma| + \log_2 \mathbf{m}$.

Note that in *path*?, $\mathbf{m}$ is written in binary in the expression $\gamma_1^{\mathbf{m}}$ that is why we measure the size of $\gamma^{\mathbf{m}}$ by using logarithms.

We also define $|L| = \sum_{\psi \in L} |\psi|$.

With every call $mc(\Gamma, L, \phi)$ we associate the rank $|L| + |\phi|$. With every call $path?(\Gamma, \Gamma', \gamma)$ we associate the rank $|\gamma|$. The rank associated with a call is strictly decreasing during the recursive calls starting from $mc(\Gamma, \epsilon, \phi)$. For instance:

- the rank associated with $mc(\Gamma, L, [\psi_1!]\psi_2)$ is $|L| + 1 + |\psi_1| + |\psi_2|$ and the ranks associated with the direct calls, namely $mc(\Gamma, L, \psi_1)$ and $mc(\Gamma, L :: \psi_1, \psi_2)$ are respectively $|L| + |\psi_1|$ and $|L| + |\psi_1| + |\psi_2|$ and are strictly smaller than $|L| + 1 + |\psi_1| + |\psi_2|$;
- the rank associated with $mc(\Gamma, L, [\gamma]\psi)$ is $|L| + 1 + |\gamma| + |\psi|$. Direct calls are $mc(\Gamma', L, \top)$, $path?(\Gamma, \Gamma', \gamma)$ and $mc(\Gamma', L, \psi)$. The corresponding ranks are respectively $|L|$, $|\gamma|$ and $|L| + |\psi|$. They are all strictly smaller than $|L| + 1 + |\gamma| + |\psi|$.

So the number of nested recursive calls from $mc(\Gamma, \epsilon, \phi)$ is bounded by $|\epsilon| + |\phi| = |\phi|$. Note that $|\phi|$ is polynomial in the size of the instance of the model checking problem. The memory needed for local variables at each call is

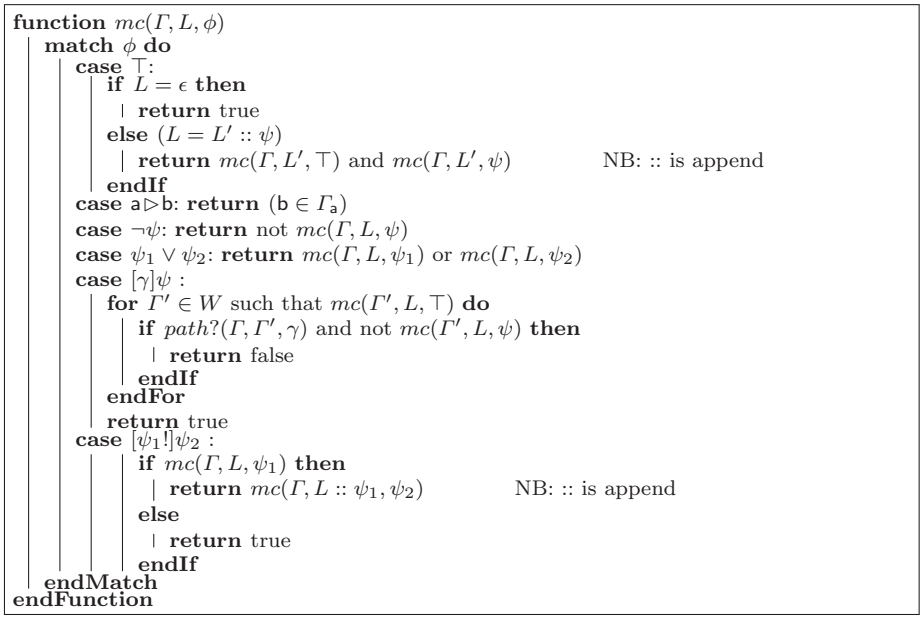


Fig. 7 Algorithm for model checking for BBL_{PDL^*}

also polynomial. So the resulting model checking procedure runs in polynomial space.

Proposition 6 *MC(BBL_{PDL^*}) is in PSPACE.*

Example 8 Let us consider the geometrical model of Example 1. Suppose we want to check that the property $[(\mathsf{a} \rhd \mathsf{e} \wedge \mathsf{a} \rhd \mathsf{c})!]K_{\mathsf{b}}\mathsf{a} \rhd \mathsf{d}$ holds. First, we apply the translation τ given in subsection 5.2. We obtain formula $\phi := [(\mathsf{a} \rhd \mathsf{e} \wedge \mathsf{a} \rhd \mathsf{c})!][\gamma_{\mathsf{b}}^K]\mathsf{a} \rhd \mathsf{d}$. Secondly, we compute the vision sets given in Example 2 with the algorithm of Figure 5. Next, we call the model checking procedure given in Figure 7. More precisely we call $mc(\Gamma, \epsilon, \phi)$ where Γ is defined by $\Gamma_{\mathsf{a}} = \{\mathsf{e}, \mathsf{d}, \mathsf{c}\}$, $\Gamma_{\mathsf{b}} = \emptyset$, $\Gamma_{\mathsf{c}} = \emptyset$, $\Gamma_{\mathsf{d}} = \{\mathsf{c}\}$ and $\Gamma_{\mathsf{e}} = \{\mathsf{b}, \mathsf{d}, \mathsf{a}\}$ and ϵ is the empty list of announcements. Formula ϕ is of the form $[\psi_1!]\psi_2$. We call $mc(\Gamma, \epsilon, \mathsf{a} \rhd \mathsf{e} \wedge \mathsf{a} \rhd \mathsf{c})$ which returns true. Therefore, we call $mc(\Gamma, [\mathsf{a} \rhd \mathsf{e} \wedge \mathsf{a} \rhd \mathsf{c}], [\gamma_{\mathsf{b}}^K]\mathsf{a} \rhd \mathsf{d})$. Now, the current formula to check is of the form $[\gamma]\psi$. We now consider all $\Gamma' \in W$ such that $mc(\Gamma', [\mathsf{a} \rhd \mathsf{e} \wedge \mathsf{a} \rhd \mathsf{c}], \top)$ returns true. They corresponds to all Γ' where $\mathsf{a} \rhd \mathsf{e} \wedge \mathsf{a} \rhd \mathsf{c}$ holds. Let us consider such a Γ'. We test whether $path?(\Gamma, \Gamma', \gamma_{\mathsf{b}}^K)$. It returns true whenever $\Gamma_{\mathsf{b}} = \Gamma'_{\mathsf{b}}$, $\Gamma_{\mathsf{c}} = \Gamma'_{\mathsf{c}}$, $\Gamma_{\mathsf{d}} = \Gamma'_{\mathsf{d}}$, $\Gamma_{\mathsf{e}} = \Gamma'_{\mathsf{e}}$. We then call $mc(\Gamma', [\mathsf{a} \rhd \mathsf{e} \wedge \mathsf{a} \rhd \mathsf{c}], \mathsf{a} \rhd \mathsf{d})$. But it turns out that, as Γ' is such that $\mathsf{a} \rhd \mathsf{e} \wedge \mathsf{a} \rhd \mathsf{c}$ holds, the only solution is to have $\Gamma'_{\mathsf{a}} = \{\mathsf{e}, \mathsf{d}, \mathsf{c}\}$. Therfore, $mc(\Gamma', [\mathsf{a} \rhd \mathsf{e} \wedge \mathsf{a} \rhd \mathsf{c}], \mathsf{a} \rhd \mathsf{d})$ returns true. Hence, $mc(\Gamma, \epsilon, \phi)$ returns true.

```
function path?(Γ, Γ′, γ)
  match γ do
    case ↷a : return (Γ′_b = Γ_b for all b ≠ a)
    case γ1;γ2 :
      for Γ″ ∈ W do
        if path?(Γ, Γ″, γ1) and path?(Γ″, Γ′, γ2)
            then return true
      endFor
      return false
    case γ1 ∪ γ2: return path?(Γ, Γ′, γ1)or path?(Γ, Γ′, γ2)
    case φ?: return (Γ = Γ′ and mc(Γ, ϵ, φ));
    case γ1* :
      if Γ = Γ′ then return true;
      for m := 1 to 2^(#Agt)^2 do
        if path?(Γ, Γ′, γ1^m) then return true;
      endFor
      return false;
    case γ1^m with m = 1: return path?(Γ, Γ′, γ1);
    case γ1^m with m > 1 :
      for Γ″ ∈ W do
        if path?(Γ, Γ″, γ1^⌊m/2⌋) and path?(Γ″, Γ′, γ1^⌈m/2⌉)
            then return true
      endFor
      return false
  endMatch
endFunction
```

Fig. 8 Algorithm for checking the existence of a γ-path from Γ to Γ'

6.2 Lower-bound results for model checking

Unsurprisingly, the lower-bound is reached even *without* the star operator in the language:

Proposition 7 *MC(*BBL_{PDL}*) is PSPACE-hard.*

Proof We will give a polynomial reduction of the satisfiability problem SAT-QBF for quantified Boolean formulas (QBF), which is PSPACE-complete [15], to MC(BBL_{PDL}). Consider a QBF $\exists p_1 \forall p_2 \dots Q_n p_n \psi(p_1, \dots, p_n)$ where Q_k is $\exists$ if k is odd and $\forall$ if k is even and where $\psi(p_1, \dots, p_n)$ is Boolean formula over the atomic propositions $p_1, \dots, p_n$. Let $\mathsf{Agt} = \{\mathsf{a}_0, \mathsf{a}_1, \dots, \mathsf{a}_m\}$. We build an instance (G, ϕ), where $\mathsf{G} = (\mathsf{pos}, \mathsf{dir}, \mathsf{ang})$, for MC($\mathsf{BBL}_{PDL}$) as follows:

- $\mathsf{pos}(\mathsf{a}_i) = (i, 0)$; $\mathsf{dir}(\mathsf{a}_i) = (-1, 0)$; $\mathsf{ang}(\mathsf{a}_i) = \frac{\pi}{4}$;
- $\phi = \langle \overset{\curvearrowright}{\mathsf{a}_1} \rangle [\overset{\curvearrowright}{\mathsf{a}_2}] \dots O_i \psi'$ where O_i is $\langle \overset{\curvearrowright}{a_i} \rangle$ if i is odd and $[\overset{\curvearrowright}{a_i}]$ if i is even and ψ' is the formula ψ in which we have replaced all occurrences of p_i by $\mathsf{a}_i \rhd \mathsf{a}_0$.

Since each of the cameras positioned as above can independently be turned so as to see a_0, and it can also be turned so as not to see a_0, the formula $\exists p_1 \forall p_2 \dots Q_n p_n \psi(p_1, \dots, p_n)$ is true in the quantified Boolean logic QBL iff $\mathcal{K}_{\mathcal{M}}, w_{\mathsf{G}} \models \phi$ where $\mathcal{K}_{\mathcal{M}}$ is the vision-based model of G.

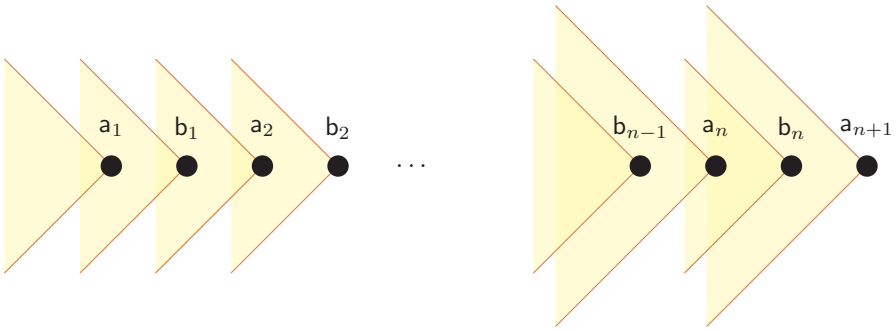
Theorem 1 *Let* BBL^- *be the fragment of* BBL *without turning operators but with the knowledge operators. Then MC*(BBL^-) *is PSPACE-hard.*

Proof The proof is similar to the proof of Proposition 7 (and to the proof for the MC problem in Lineland, in [1]). We provide a polynomial reduction of the SAT-QBF problem to MC(BBL^-) as follows. Consider a QBF formula $\exists p_1 \forall p_2 \dots Q_n p_n \psi(p_1, \dots, p_n)$ where $Q_k = \exists$ if k is odd and $Q_k = \forall$ if k is even and where $\psi(p_1, \dots, p_n)$ is Boolean formula over the atomic propositions $p_1, \dots, p_n$. We build an instance $\mathsf{G} = (\mathsf{pos}, \mathsf{dir}, \mathsf{ang}), \phi$ for MC(BBL^-) as follows:

- $\mathsf{pos}(\mathsf{a}_i) = (2i, 0)$; $\mathsf{pos}(\mathsf{b}_i) = (2i+1, 0)$;
- $\mathsf{dir}(\mathsf{a}_i) = (-1, 0)$; $\mathsf{dir}(\mathsf{b}_i) = (-1, 0)$;
- $\mathsf{ang}(\mathsf{a}_i) = \frac{\pi}{2}$; $\mathsf{ang}(\mathsf{b}_i) = \frac{\pi}{2}$;
- for ϕ we first define a sequence of formulas $(\phi_k)_{k=1\dots n+1}$ by backward induction on k:
  - $\phi_{n+1} := \psi'$; where ψ' is the formula ψ in which we have replaced all occurrences of p_i by $\mathsf{b}_i \rhd \mathsf{a}_1$
  - $\phi_k := \hat{\mathsf{K}}_{\mathsf{a}_k}(\mathsf{a}_{k+1} \rhd \mathsf{a}_k \wedge \phi_{k+1})$ if k is odd;
  - $\phi_k := \mathsf{K}_{\mathsf{a}_k}(\mathsf{a}_{k+1} \rhd \mathsf{a}_k \to \phi_{k+1})$ if k is even.

  and we set $\phi = \phi_1 = \hat{\mathsf{K}}_{\mathsf{a}_1}(\mathsf{a}_2 \rhd \mathsf{a}_1 \wedge \mathsf{K}_{\mathsf{a}_2}(\mathsf{a}_3 \rhd \mathsf{a}_1 \to \dots ...\psi') \dots)$.

  The geometrical model G is depicted as follows:



The instance (G, ϕ) for MC(BBL^-) is computable in polynomial time from the instance $\exists p_1 \forall p_2 \dots Q_n p_n \psi(p_1, \dots, p_n)$ for QBF.

The idea encoded in that formula is that the knowledge operators $\mathsf{K}_{\mathsf{a}_i}$ and $\hat{\mathsf{K}}_{\mathsf{a}_i}$ serve respectively as universal and existential quantifiers over the truth values of the boolean variable p_i, while the guards $\mathsf{a}_j \rhd \mathsf{a}_1$ ensure that the agents' cameras are positioned so as to enable each of these truth values.

Claim: The formula $\exists p_1 \forall p_2 \dots Q_n p_n \psi(p_1, \dots, p_n)$ is true in quantified Boolean logic iff $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}, \Gamma_{\mathsf{G}} \models \phi$ where $\mathcal{V}(\mathcal{S})$ is the vision-based model of the geometrical model G.

Here is a proof of the claim. The proof is done by induction on the construction of $\exists p_1 \forall p_2 \dots Q_n p_n \psi(p_1, \dots, p_n)$ (the base case is $\psi(p_1, \dots, p_n)$).

Note that formula ϕ is ϕ_1.
Consider the following property $\mathcal{P}(k)$:

'Given a partial valuation ν over $p_1, \ldots, p_{k-1}$ and a world Γ such that $\nu \models p_i$ iff b_i sees a_1, $\nu \models Q_k \ldots Q_n \psi(p_1, \ldots, p_n)$ iff $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}, \Gamma \models \phi_k$.'

We will prove $\mathcal{P}(k)$ for all $k = 1, ..., n+1$ by backward induction on k.

$\boxed{\mathcal{P}(n+1)}$ Let ν a valuation and Γ stated as in the property. We have $\nu \models \psi(p_1, \ldots, p_n)$ iff $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}, \Gamma \models \psi(\mathsf{b}_1 \rhd \mathsf{a}_1, \ldots, \mathsf{b}_n \rhd \mathsf{a}_1)$.

$\boxed{\mathcal{P}(k+1) \Rightarrow \mathcal{P}(k)}$ Let $k \in \{1, \ldots, n\}$. Suppose that $\mathcal{P}(k+1)$ and let us prove that $\mathcal{P}(k)$. Without loss of generality, we treat the case when k is odd, while the case when k is even is left to the reader. Let ν be a partial valuation over $p_1, \ldots, p_{k-1}$ and Γ a corresponding world as stated in the property. Suppose that $\nu \models \exists_k \ldots Q_n \psi(p_1, \ldots, p_n)$. This is equivalent to the fact that there is a value $v \in \{0, 1\}$ such that $\nu[p_k := v] \models Q_{k+1} \ldots Q_n \psi(p_1, \ldots, p_n)$. Without loss of generality, suppose that $v = 1$. Now, let us consider a world Γ', similar to Γ except that b_k sees a_1 and a_{k+1} sees a_k. Then Γ' is a possible world for agent a_k in Γ. Furthermore, by $\mathcal{P}(k+1)$, we have $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}, \Gamma' \models \phi_{k+1}$. It implies that $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}, \Gamma \models \phi_k$. Conversely, if $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}, \Gamma \models \phi_k$, it means that there exists a world Γ' in which a_{k+1} sees a_k and in which ϕ_{k+1} holds. Let ν' the valuation corresponding to Γ' in the following sense: p_i is true in ν' iff b_i sees a_1 in Γ'. By the induction hypothesis, $\mathcal{V}(\mathcal{S}), \mathbf{\Gamma}, \Gamma' \models \phi_{k+1}$ implies that $\nu' \models Q_{k+1}, \ldots, Q_n \psi$. Since a_{k+1} sees a_k in Γ', ν and ν' give the same truth values to each p_i when $i \leq k$. So, $\nu \models \exists_k \ldots Q_n \models \psi$. This completes the induction.

The claim follows from the truth of property $\mathcal{P}(1)$.

Corollary 3 *The model checking problem for* $\mathsf{BBL_D}$ *is PSPACE-complete.*

6.3 PSPACE-completeness of SAT(BBL_{PDL^*})

Now we address the problem of existence of a placement of cameras satisfying a given formula ϕ of BBL_{PDL^*}. We consider both the general case, SAT(BBL_{PDL^*}), and the special case where the sizes of the angles of vision are specified, ANGLE-FIXED-SAT(BBL_{PDL^*}). We will construct a polynomial translation of SAT(BBL_{PDL^*}) into the existential fragment of the first order theory of real numbers, whose satisfiability problem is in PSPACE [7], thus proving the following.

Theorem 2 *Both SAT*(BBL_{PDL^*}) *and ANGLE-FIXED-SAT*(BBL_{PDL^*}) *are in PSPACE.*

Proof It suffices to prove that SAT(BBL_{PDL^*}) runs in polynomial space, as the case of ANGLE-FIXED-SAT(BBL_{PDL^*}) is easier. The core part of the proof is in the guessing of an abstract spectrum of vision, i.e., a tuple of families of vision sets $(S_\mathsf{a})_{\mathsf{a} \in \mathsf{Agt}}$, that generates a geometrically realisable vision-based

model satisfying the input formula, if one exists. Once such a spectrum of vision is guessed, we only have to apply the model checking procedure from Section 6.1.2 to the generated vision-based model. Here is an outline of the procedure:

1. The crucial observation is that there are finitely many candidate families of vision sets per camera, and hence finitely many candidate spectra of vision, each of size polynomial in the length of the input formula. So, we start by guessing one of them, that is, we select an abstract spectrum of vision $\mathcal{S}$, which generates an (abstract) vision-based model $\mathcal{V}(\mathcal{S})$ which is so far only a *provisional* possibly satisfying model.
2. Once such a spectrum of vision $\mathcal{S}$ is provisionally guessed, we need to check for its geometric consistency, i.e., for existence of a geometric model G that generates it. We will show that the existence of such a geometric model can be expressed in $\exists\mathrm{FO}(\mathbb{R})$, hence can be verified in polynomial space.
   To begin with, the positions and the vision angles of the cameras are specified by existential quantified variables ranging over the real coordinates of the positions and the unit directions vectors of the initial and terminal rays. Now, we need to write an open FO formula claiming that for each camera its actual family of vision sets, generated in the so specified geometric model G, is precisely the one in the initially guessed spectrum of vision $\mathcal{S}$. That can be done by computing the actual family of vision sets for each camera in the specified geometric model G and checking whether it coincides with the respective family in $\mathcal{S}$.
   Further, for computing that family in G for the given cameras, we only need to explore the $2(k-1)$ directions that correspond to its 'border positions', when one of the two boundary rays of its sector of vision passes through another camera. These directions are identified by simple analytic geometry and for each of them computing the vision set is again done by a routine analytic geometry. In addition, we will have to check whether the empty set belongs to the family of vision sets for each camera.
   Once computed, the resulting $\exists\mathrm{FO}(\mathbb{R})$-formula, which is of size polynomial in the size of the input BBL_{PDL^*}-formula, is verified in polynomial space using the *mc* algorithm.

   Here are the mathematical details on the above outline. Suppose, the provisionally guessed spectrum of vision $\mathcal{S}$ for the formula ϕ we want to satisfy involves 4 cameras. Let the specified (by existentially quantified variables) positions of the cameras be at points $P_i = (x_i, y_i)$, for $i = 1, 2, 3, 4$. In order to compute the family of vision sets, say for camera 1 with specified vision angle $\alpha \leq \pi$, we do the following. (The reader may look at Figure 9.)

   (a) Compute the unit vectors v_{1j} in direction of $\overrightarrow{P_1P_j}$, for $j = 2, 3, 4$. We can assume they are represented by their direction angles, i.e.:
   $v_{1j} = (\cos\beta_{1j}, \sin\beta_{1j})$.
   Note that the components of these vectors are functions only of the coordinates of the cameras and can be assumed computed at a fixed unit time cost.

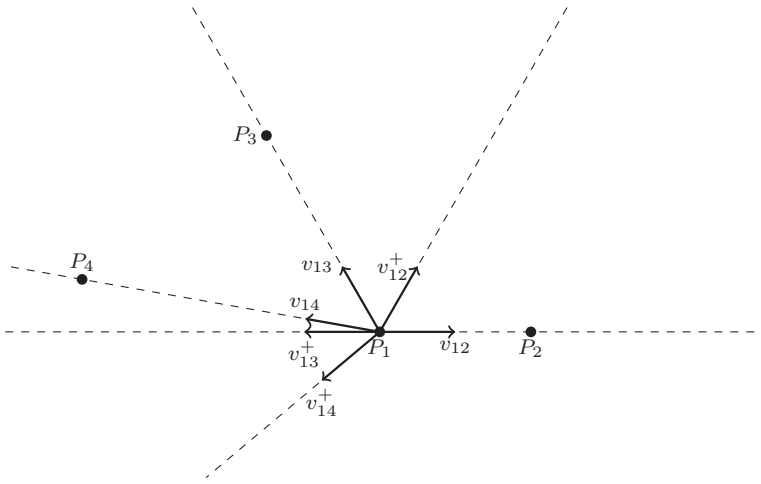


Fig. 9 Illustration of the mathematical notation in the proof of Theorem 2.

(b) Compute the unit vectors $v_{1j}^+ = (\cos(\alpha + \beta_{1j}), \sin(\alpha + \beta_{1j}))$, for $j = 2, 3, 4$, obtained by rotating v_{1j} at angle α in positive (counter-clockwise) direction. Again, each of these is computed at a fixed unit time cost.

(c) Now, we consider a ray ρ starting at P_1, regarded as the left border of the vision sector of the camera at P_1 and rotating over a full revolution angle 2π. Every position of that ray, specified by a unit direction vector $u = (\cos\gamma, \sin\gamma)$, determines uniquely the set of other cameras seen from P_1. The crucial observation now is that we only need to inspect the positions of ρ which are in direction of one of the vectors v_{1j} or v_{1j}^+, for $j = 2, 3, 4$, as these are the only 'border positions', where the current vision set changes because one of the border rays of the sector of vision passes through another camera. In addition, we need to check whether $\emptyset$ is a vision set for camera 1, see further.

(d) For a given value of the unit direction vector $u = (\cos\gamma, \sin\gamma)$ of the current position of the ray ρ, the position of the other, right border ray has a unit direction vector $u^- = (\cos(\gamma - \alpha), \sin(\gamma - \alpha))$. Now, we have to compute the vision set of the camera at P_1 for the respective sector of vision with these border rays. For each of the points P_j, for $j = 2, 3, 4$, we determine whether it is in that sector, as follows. We consider the vectors u^-, u and v_{1j}. Clearly, v_{1j} lies in the current sector of vision if and only if it coincides with one of the vectors u^- and u, or if each of the directed angles between the vectors u^-, v_{1j} and between v_{1j}, u are less than π. By standard analytic geometry, the latter holds if an only if both vectors $u^- \times v_{1j}$ and $v_{1j} \times u$ (where $\times$ is vector product) have the same direction as $u^- \times u$, if and only if the determinants

$$\begin{vmatrix} \cos(\gamma - \alpha) & \sin(\gamma - \alpha) \\ \cos\beta_{1j} & \sin\beta_{1j} \end{vmatrix}, \quad \begin{vmatrix} \cos\beta_{1j} & \sin\beta_{1j} \\ \cos\gamma & \sin\gamma \end{vmatrix}, \quad \begin{vmatrix} \cos(\gamma - \alpha) & \sin(\gamma - \alpha) \\ \cos\gamma & \sin\gamma \end{vmatrix},$$

have the same signs, which can be expressed by a simple algebraic condition over the parameters $\cos\beta_{1j}, \sin\beta_{1j}, \cos\alpha, \sin\alpha, \cos\gamma, \sin\gamma$ and eventually expressed by the truth of an open FO($\mathbb{R}$)-formula

$$INSIDE(x_1, y_1, \ldots, x_4, y_4)$$

(Recall that α is fixed and γ has one of the 2×3 border values, each of them explicitly determined by the coordinates of the camera positions).

(e) Computing the truth value of that formula for each $j = 2, 3, 4$ produces the current vision set of the camera at P_1 for the given border position of the ray ρ.

(f) Computing these vision sets for each of the 6 border positions of ρ, possibly plus the empty set $\emptyset$, produces the family of vision sets for the camera at P_1 corresponding to the geometric model G. Checking whether $\emptyset$ must be added can be done, for instance, by identifying the angles between every two consecutive (as the boundary ray ρ rotates) border directions of ρ, corresponding to the vectors v_{1j}, and comparing them with the angle of vision α of the camera. If at least one of these is greater than α, then $\emptyset$ is added to the family of vision sets for P_1, else – not. We leave the routine details to the reader.

(g) Once the family of vision sets in G for the camera at P_1 is computed, it has to be compared with the family of vision sets for that camera in the initially guessed spectrum of vision $\mathcal{S}$. (In order to avoid a combinatorial explosion in that comparison, we assume both families of vision sets lexicographically ordered.)

(h) The same procedure is applied to each camera at P_i to obtain an open formula ϕ_i of FO($\mathbb{R}$) for each $i \in \{1, \cdots, 4\}$.

The rest is a routine but tedious exercise in writing up the existential closure of the conjunctions of the above ϕ_i formulas to obtain one large but polynomially bounded in size formula of $\exists$FO(R). We leave this exercise to the interested reader.

3. Once the consistency of the guessed spectrum of vision $\mathcal{S}$ is verified, we construct the generated by it vision-based model $\mathcal{V}(\mathcal{S})$, select (guess) a world Γ in it and model check the input formula ϕ by calling $mc(\Gamma, \epsilon, \phi)$, which can be done in polynomial space, by Proposition 6.

We note that the procedure outlined above is subject to further practical optimizations. Still, it is a non-deterministic procedure taking a polynomial amount of space in the input formula ϕ. According to Savitch's theorem, NPSPACE = PSPACE [15], hence we obtain the PSPACE upper bound.

The matching PSPACE lower bound is already quite immediate.

Proposition 8 *SAT*(BBL_{PDL}) *is PSPACE-hard.*

Proof The formula $\exists p_1 \forall p_2 \ldots Q_n p_n \psi(p_1, \ldots, p_n)$ is true in QBL iff ϕ' is satisfiable where $\phi' = \langle \overset{\curvearrowright}{\mathsf{a}_1} \rangle [\overset{\curvearrowright}{\mathsf{a}_2}] \ldots O_i \psi'$ where O_i is $\langle \overset{\curvearrowright}{\mathsf{a}_i} \rangle$ if i is odd and $[\overset{\curvearrowright}{\mathsf{a}_i}]$ if i

is even and ψ' is the formula ψ in which we have replaced all occurrences of p_i by $\mathsf{a}_i \triangleright \mathsf{a}_0$. In fact, this still holds when the satisfiability problem is assumed in ANGLE-FIXED-SAT(BBL_{PDL^*}).

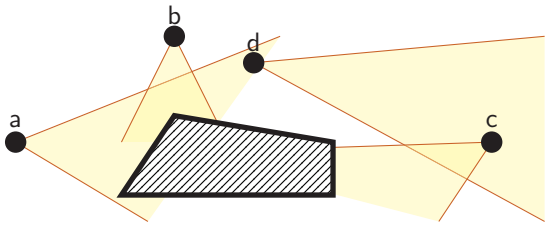
Corollary 4 *SAT*(BBL_{PDL}) *is PSPACE-complete.*

7 Further extensions of the BBL

The present work considers simplified and idealized scenarios of stationary cameras placed in the plane and without any obstacles to their vision. It is amenable to several natural and important extensions which we only briefly discuss here. The technical details and respective results will be presented in a follow-up work.

7.1 Adding obstacles

In many real life scenarios the visibility by the cameras of the objects of interest, including other agents, is impaired by stationary or moving obstacles. Here we outline how our framework can be extended by adding stationary obstacles. In geometric models, these are represented by their shapes placed in the plane, as in the figure below.



To take obstacles into account in the model checking problem, we have to modify the scan used to compute the sets S_{a}. For instance, in the example above, when computing S_{a} we note that the segment $[\mathsf{pos}(\mathsf{a}), \mathsf{pos}(\mathsf{c})]$ intersects the obstacle and thus c will not appear in S_{a}.

Suppose that each obstacle is a polygon, described by a list of its vertices. A segment $[\mathsf{pos}(\mathsf{a}), \mathsf{pos}(\mathsf{b})]$ intersects the obstacle iff the segment intersects one of the sides of the polygon. Thus, testing intersection is done in polynomial time. Eventually, the model checking procedure can be adjusted to deal with obstacles and the extended model checking problem would remain in PSPACE.

The satisfiability testing problem with obstacles ramifies into several versions, depending on whether each of the angles of vision of the cameras and the shape and position of the obstacles is assumed fixed or can vary. For instance, if each of these is fixed, the satisfiability testing problem for BBL is now modified as follows: given a set of obstacles positioned in the plane and a specification in BBL or any of its extensions, determine whether there is a

positioning of the cameras in the plane, without moving the obstacles, such that the resulting configuration satisfies the specification.

Note also that when obstacles are added, we can no longer automatically assume that the agents know each other's positions. This would still be the case if these positions were known initially, e.g. if the obstacles appeared after the agents initially scanned the environment, but cannot be assumed in general. If knowledge of the positions of the other agents, hidden behind obstacles, is not assumed then the model checking and satisfiability testing methods developed here would have to be accordingly modified.

7.2 Cameras in 3D

In reality, cameras are usually placed in the 3D space and their areas of vision are 3-dimensional. That adds essential, though not crucial, technical overhead in computing the areas of vision of the cameras, but the logical languages and the epistemic parts of their semantics remain essentially unchanged. The model checking and satisfiability testing algorithms, with or without obstacles, can be suitably modified and will be in the same complexity classes as in 2D.

7.3 Moving cameras and robots

In reality the agents and their cameras are often mobile, not stationary. The simplest way to capture that feature is to consider the same logical languages over geometric models that only involve the agents and the vision angles of their cameras, but not their positions. Thus, the model checking problem for the case of mobile cameras, asking for suitable positioning of the cameras, satisfying the input specification, reduces to satisfiability testing problem for models with stationary cameras. A PDL-like approach for capturing mobile cameras involves adding to the language an extra dynamic operator for translations in the plane along the current direction of the camera. Together with the operator for rotations (turning) these can express every movement in the plane. The respective geometric models involve again only the sets of agents and the vision angles of their cameras, but not their positions. The possible worlds (states) in geometric Kripke structures and in their vision-based abstractions remain the same. They are based on all sets of other agents that may be seen in a glance after any rotation and/or translation. In the recursive model checking procedure for the enriched languages, every occurrence of a translation operator in the formula applied to an agent a would generate a finite number of different model checking problems, arising by sliding the camera in the current direction of the agent a and only recording the changes occurring in the vision sets of the agents. Thus, eventually, the model checking problem for the case of mobile cameras can be reduced to repeated solving of finite sets of model checking problem for models with stationary cameras.

So far, mobile agents have not been considered in such framework. But, even without dynamic modalities, we have to drop the condition of having

common knowledge about the positions of agents (that is, in two possible worlds, the same agent may have different positions), we know that the model checking problem and the satisfiability problem are in EXPSPACE [1], but the optimal complexities are still under investigation. The main difficulty to obtain an algorithm in polynomial space is that the construction of vision-based abstractions (see Def. 5) seems not suitable here. Indeed, when cameras may move, it is no longer true that truths of formulas depend on what agents see [1] and some projective geometrical results, such as the theorems of Desargues and Pappus, can play essential role here.

8 Concluding remarks

We have introduced formal models for multi-agent systems, where agents control surveillance cameras, and logical languages to reason about the interaction between their vision and knowledge. The abstract scenario considered in this paper is applicable in principle to various real-life situations, ranging from systems of real-time surveillance cameras in building and public areas, through multi-robot systems, to networks of stationary satellites positioned around the globe. Still, we emphasize that the surveillance cameras idea used here is more of a paradigmatic example than a primary topic of our study. It is about visual-epistemic reasoning in a geometrically concrete world. Thus, our framework and results can have applications to automated reasoning, formal specification and verification of observational abilities and knowledge of multi-agent systems in each of these situations.

References

1. P. Balbiani, O. Gasquet, and F. Schwarzentruber. Agents that look at one another. *Logic Journal of IGPL*, 21(3):438–467, 2013.
2. P. Balbiani, V. Goranko, R. Kellerman, and D. Vakarelov. Logical theories of fragments of elementary geometry. In M. Aiello, J. van Benthem, and I. Pratt-Hartmann, editors, *Handbook of Spatial Logics*, pages 343–428. Springer, 2007.
3. P. Balbiani, H. Van Ditmarsch, A. Herzig, and T. De Lima. Tableaux for public announcement logic. *Journal of Logic and Computation*, 20(1):55–76, 2010.
4. M. Ben-Or, D. Kozen, and J. Reif. The complexity of elementary algebra and geometry. *Journal of Computer and System Sciences*, 32(2):251–264, 1986.
5. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.
6. A. L. Bustamante, J. M. Molina, and M. A. Patricio. Multi-camera and multi-modal sensor fusion, an architecture overview. In *Proc. of DCAI'2010*, pages 301–308, 2010.
7. J. Canny. Some algebraic and geometric computations in PSPACE. In *Proceedings of STOC'88*, pages 460–467. ACM, 1988.
8. R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.
9. J. García, J. Carbó, and J. M. Molina. Agent-based coordination of cameras. *Intern. Journal of Computer Science & Applications*, 2(1):33–37, 2005.
10. O. Gasquet, V. Goranko, and F. Schwarzentruber. Big brother logic: Logical modeling and reasoning about agents equipped with surveillance cameras in the plane. In *Proc. of AAMAS'2014*, pages 325–332, 2014.

11. V. Goranko, M. Merker, and C. Thomassen. Directed graphs with restricted angles of vision. Manuscript, 2014.
12. D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, 2000.
13. J. Plaza. Logics of public communications. *Synthese*, 158(2):165–179, 2007.
14. F. Schwarzentruber. Seeing, knowledge and common knowledge. In *Logic, Rationality, and Interaction*, pages 258–271. Springer, 2011.
15. M. Sipser. *Introduction to the Theory of Computation*, volume 2. Thomson Course Technology Boston, 2006.
16. A. Tarski. *A decision method for elementary algebra and geometry*. Springer, 1951.
17. H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. Springer, Dordecht, 2008.