Available online at www.sciencedirect.com

ScienceDirect

Information Fusion 8 (2007) 56–69

INFORMATION FUSION

www.elsevier.com/locate/inffus

Communication between agents with heterogeneous perceptual capabilities ☆

Patrick Doherty [a,*], Witold Łukaszewicz [a,b], Andrzej Szałas [a,b]

[a] Department of Computer and Information Science, University of Linköping, Linköping, Sweden
[b] University of Economics and Computer Science, Olsztyn, Poland

Received 28 October 2004; received in revised form 17 May 2005; accepted 17 May 2005
Available online 14 July 2005

Abstract

In real world applications robots and software agents often have to be equipped with higher level cognitive functions that enable them to reason, act and perceive in changing, incompletely known and unpredictable environments. One of the major tasks in such circumstances is to fuse information from various data sources. There are many levels of information fusion, ranging from the fusing of low level sensor signals to the fusing of high level, complex knowledge structures. In a dynamically changing environment even a single agent may have varying abilities to perceive its environment which are dependent on particular conditions. The situation becomes even more complex when different agents have different perceptual capabilities and need to communicate with each other.

In this paper, we propose a framework that provides agents with the ability to fuse both low and high level *approximate* knowledge in the context of dynamically changing environments while taking account of heterogeneous and contextually limited perceptual capabilities.

To model limitations on an agent's perceptual capabilities we introduce the idea of partial tolerance spaces. We assume that each agent has one or more approximate databases where approximate relations are represented using lower and upper approximations on sets. Approximate relations are generalizations of rough sets.

It is shown how sensory and other limitations can be taken into account when constructing and querying approximate databases for each respective agent. Complex relations inherit the approximativeness of primitive relations used in their definitions. Agents then query these databases and receive answers through the filters of their perceptual limitations as represented by (partial) tolerance spaces and approximate queries. The techniques used are all tractable.

Keywords: Rough sets; Database and sensor fusion; Approximate reasoning; Intelligent agents; Cognitive robotics; Software agents

1. Introduction

In real world applications robots and software agents often have to be equipped with higher level cognitive functions that enable them to reason, act and perceive in changing, incompletely known and unpredictable environments. One of the major tasks in such circumstances is to fuse information from various data sources. There are many levels of information fusion, ranging from the fusing of low level sensor signals to the fusing of high level, complex knowledge structures. In a

☆ A short version of this study was presented at the Seventh International Conference on Information Fusion (FUSION 2004) in Stockholm, Sweden [1]. Supported in part by the WITAS UAV project grant under the Wallenberg Foundation, Sweden and an NFFP03 research grant (COMPAS).

* Corresponding author. Tel.: +4613282426; fax: +4613245868.
E-mail addresses: patdo@ida.liu.se (P. Doherty), witlu@ida.liu.se (W. Łukaszewicz), andsz@ida.liu.se (A. Szałas).

1566-2535/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.inffus.2005.05.006

dynamically changing environment even a single agent may have varying abilities to perceive its environment which are dependent on particular conditions. The situation becomes even more complex when different agents have different perceptual capabilities and need to communicate with each other.

Research in more traditional robotics has emphasized low-level sensing, sensor processing and control tasks. One of the open challenges in cognitive robotics is to develop architectures which seamlessly combine low-level sensing and sensor processing with the generation and maintenance of higher level knowledge structures. This implies signal-to-symbol transformations at many levels of abstraction. One particularly difficult issue involves the quantitative to qualitative transformations required to support the use of qualitative knowledge structures in high-level reasoning tasks. To add to the difficulty, sensors, by their very nature, introduce uncertainty and noise in the data. In order to provide an adequate representation of a robotic environment, some of this uncertainty, or lack of knowledge, should be reflected in the higher-level knowledge structures. In other words, some of the high-level knowledge structures should be approximate in nature, having both quantitative and qualitative characteristics.

The perceptual limitations of a robotic agent constrained by the peculiarities of its sensor suite should be taken into account not only when the robotic agent reasons about its external and internal environments, interpreting its own measurements made in different conditions, but also when one or more robotic agents communicate with each other by asking questions concerning each others' knowledge about the world or themselves. In this case, two robotic agents communicating with each other can only ever ask queries of an approximative nature and receive answers of an approximative nature as seen through their respective filters of perceptual limitation.

In this paper, we propose a technique that can provide agents with the ability to ask *approximate* questions to each other in the context of heterogeneous perceptual capabilities and approximate knowledge derived through uncertain sensor data. Even though they may have concepts in common, their ability to perceive individuals as having specific properties or relations can be distinct. The concern then is how this affects the questions that can be asked and the replies that can be generated by agents with perception functions limited to varying degrees. In particular, we address the following problems related to information fusion:

- given various information sources[1] with heterogeneous perceptual capabilities, what facts based on those sources can be accepted by an agent to be certain, what facts are unknown and what surely do not hold?
- given agents equipped with (approximate) knowledge, what knowledge common to a group of agents should be accepted as certain, unknown or surely not holding?

The methodology developed in the current paper allows one to deal with both problems in a uniform fashion.

In order to set the proper context as to how this work is specifically related to data or information fusion, it should be emphasized that in the past several years, attempts have been made to broaden the traditional definition of data fusion as state estimation via aggregation of multiple sensor streams. There is a perceived need to broaden the definition to include the many additional processes used in all aspects of data and information fusion identified in large scale distributed systems. In this case, the nodes in such systems may not only include sensors in the traditional sense, but also complex systems where data and information are fused at many different levels of abstraction to meet the diverse situation assessment needs associated with different applications. It is at this level of abstraction the techniques in this paper should be understood.

One of the more successful proposals for providing a framework and model for this broadened notion of data fusion is the JDL data fusion model [2] and its revisions [3,4]. In [3] for example, data fusion is defined as "the process of combining data or information to estimate or predict entity states" and the data fusion problem "becomes that of achieving a consistent, comprehensive estimate and prediction of some relevant portion of the world state".

We are interested in nodes in such complex systems which store relevant portions of the world state where world state information is approximate in nature and mechanisms for asking questions about nodes is in place as part of the higher-level information fusion process. In this case, some of these nodes may be viewed as containing tolerance agents which manage approximate information in a node and may communicate across nodes about such information and assist in further fusions of information content in the larger system.

In order to provide the proper level of detail for the framework considered here, the following set of abstractions is used in the article. Each agent will have access to the following functionalities and representations:

- an abstraction called a (partial) tolerance space which is used to represent similarity of data points for basic domains in addition to complex data domains;
- a set of sensors and a sensor model for each sensor. The sensor models take into account the contextual

[1] For example, those provided by actual or virtual sensors.

indiscernibility of signal data by using tolerance spaces to represent that indiscernibility;

- one or more databases capable of holding relational data. These databases may contain representations of crisp relations or approximate relations.

  Approximate relations are represented using lower and upper approximations on sets. Approximate relations are generalizations of rough sets [5]. The intention is that sensor data is used in the generation of some of these approximate relations stored in the databases. Tolerance spaces again play a role in the generation of approximate relations from specific attributes in vectors or arrays of attributes representing sensors;
- a query mechanism which permits each agent to ask questions about knowledge in its own databases or in the databases of other agents. These queries are approximate in nature due to the approximate nature of the knowledge stored in the databases. They are also contextualized by perceptual limitations represented as tolerance spaces on more complex data domains.

Fig. 1 provides a high-level schematic of the ideas involved in the proposed framework. Any agent (we refer to ours as tolerance agents) is assumed to operate in an embedding or external environment. The environment is modeled using individuals, properties of individuals and relations between individuals. The environment model an agent has access to and from which it can reason about is assumed to be limited by both inherent perceptual limitations and contextual perceptual limitations. Both types of limitations will be represented using tolerance spaces. Tolerance spaces generate neighborhoods of indiscernibility around individuals in any domain of discourse. When applied to relations or properties, they function as transformers which vary the degree of approximativeness associated with such relations and properties.

Inherent perceptual limitations are intended to represent those limitations built-in to the agent. These are assumed to be static and non-contextual, rarely subject to change. For example, color-blindness limits the ability of certain agents to discern differences in certain colors. Inherent limitations in the types of sensors used to discern the environment provide other examples. Tolerance spaces ($TS_1, \ldots, TS_i$ in Fig. 1) are used to represent such limitations and result in a model of the environment represented as approximate relations, each consisting of an upper and lower approximation derived through application of tolerance spaces to data domains or relations and properties. These base relations and properties are stored in the agent's approximate database.

If these were the only form of perceptual or other limitations on the agent's ability to discern, then the agent could query the approximate database information internally using techniques which will be introduced in the paper. In this case, the agent has access to the external environment only through the eyeglasses of the sum of its inherent perceptual limitations.

Besides inherent perceptual limitations, it is often the case that agents have other types of limitations of a contextual nature. For example, when a human agent is tired, its ability to perceive is often impaired. In the case where a human agent is hungry or angry, this can also impair judgment or an ability to discern. In other cases, the external environment may change due to weather or other conditions and an agent may experience what we call contextual perceptual limitations. These limitations are modeled in the same way as inherent perceptual limitations through the use of tolerance spaces ($TS_{i+1}, \ldots, TS_{i+j}$ in Fig. 1). Tolerance spaces may be applied iteratively to relations and properties resulting in a sequence of approximations modeled using upper and lower approximations. In this case, the agent when querying its approximate database, does this through the eyeglasses of the sum of not only its inherent perceptual limitations, but also through the sum of its contextual perceptual limitations.

Fig. 1 depicts a single agent scenario. When two or more agents with different types of perceptual limitations want to ask each other questions, the sum of perceptual limitations have to be taken into account as there is a give and take of information processing as queries pass through the filters of tolerance spaces associated with internal and external queries. This process will also be considered in the paper.

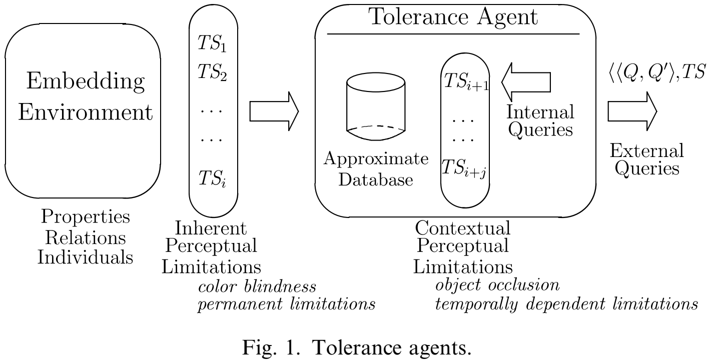

Fig. 1. Tolerance agents.

The paper is structured as follows. In Section 2, representations of approximate relations and queries are introduced. In Section 3, the important concept of a (partial) tolerance space is introduced. These spaces are used to represent indiscernibility, uncertainty and similarity between data elements or sets. In Sections 4–6 three sensor models are presented, where uncertainty in sensor data may be translated into (partial) tolerance spaces. This convenient representation allows for the possibility of relating sensor information to relational definitions in a homogeneous manner since tolerance spaces are also associated with approximate relations in databases. This is considered in Section 7 where approximate databases are introduced. At this point, basic representational structures for approximate knowledge derived through approximate sensor data are defined and discussed. The main idea of agent communication with heterogeneous perceptual capabilities is then provided in Section 8 with examples. In Section 9, a number of tractability results related to the techniques are provided. Finally, Section 10 concludes with a discussion.

This paper is an extended version of [1]. Compared to [1], a number of new ideas have been introduced in addition to many more explanatory comments about the motivations and techniques associated with this framework. Partial tolerance spaces are introduced for the first time as a generalization of the tolerance spaces used in [1]. Partial tolerance spaces are then used as a basis for integrating more traditional probabilistic sensor models with approximate relations. Complexity results for all techniques used are also provided.

2. Approximations of sets and relations

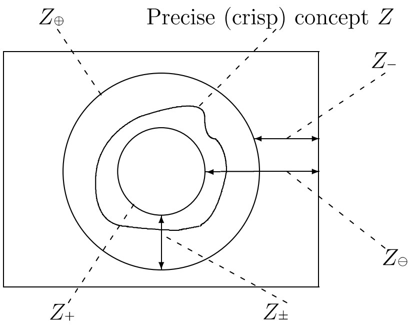

Fig. 2. Approximations of a concept.

The methodology we propose in this paper uses a generalization of a number of ideas associated with rough set theory which was introduced by Pawlak (see e.g., [5]). In many AI applications one faces the problem of representing and processing incomplete, imprecise, and approximate data. Many of these applications require the use of approximate reasoning techniques. The assumption that objects can be observed only through the information available about them (in this case, sensors with limits on discernibility) leads to the view that knowledge about objects in themselves, is insufficient for characterizing sets or relations[2] precisely since the knowledge about objects is incomplete. We thus assume that any imprecise concept, say Z, is replaced by a pair of precise concepts called the lower and the upper approximation of the imprecise concept, where (see also Fig. 2):

- the lower approximation, Z_{+}, consists of all objects which with certainty belong to the concept;
- the upper approximation, $Z_{\oplus}$, consists of all objects for which it is possible that they belong to the concept;
- the complement of the upper approximation, Z_{-}, consists of all objects which with certainty do not belong to the concept;
- the complement of the lower approximation, $Z_{\ominus}$, consists of all objects for which it is possible that they do not belong to the concept;
- the difference between the upper and the lower approximation, $Z_{\pm}$, constitutes a boundary region of an imprecise concept, i.e. the set of elements for which it is unknown whether they belong to the concept.

More precise definitions follow.

Definition 2.1. By an approximate set we shall understand a pair $\langle X, Y\rangle$, where X and Y are sets such that $X \subseteq Y$. The set X is interpreted as the *lower approximation* of a set and Y as its *upper approximation*.

By *lower and upper approximation operations*, denoted by indices $+$ and $\oplus$, we understand operations on sets such that for any crisp set Z, $Z_{+} \subseteq Z \subseteq Z_{\oplus}$.

By Z_{-} and $Z_{\ominus}$ we denote the complement of $Z_{\oplus}$ and of Z_{+}, respectively. The *boundary region* of Z, defined as $(Z_{\oplus} - Z_{+})$, is denoted by $Z_{\pm}$.

We will also need a notion of (relational) databases and approximate queries.

Definition 2.2. By a (*crisp*) *database* we understand a tuple $D = \langle U, \{r^j | j \in J\}\rangle$, where U is a finite set, called the *domain* of D, and $\{r^j | j \in J\}$ is a finite collection of relations over U.

By an *approximate database* we understand a tuple $D = \langle U, \{r^j | r^j = \langle r^j_{+}, r^j_{\oplus}\rangle\}_{j \in J}\rangle$, where r^j_{+}s and $r^j_{\oplus}$s are crisp relations of the same arity, satisfying $r^j_{+} \subseteq r^j_{\oplus}$.

By the *type* of a (crisp or approximate) database D we understand the sequence $\langle a_j | j \in J\rangle$, where for any $j \in J$, a_j is the *arity* of r^j.

[2] Sets and relations are dealt with uniformly, since a relation is simply a set of tuples.

Observe that crisp relational databases are approximate relational databases with equal lower and upper approximations of relations.

We will also require a definition of approximate queries. In essence, an approximate query provides an upper and lower approximation on an original crisp query.

Definition 2.3. By an *approximate query* we shall understand a pair $Q = \langle Q'(\bar{x}), Q''(\bar{x})\rangle$, where Q' and Q'' are formulas of a given logic, where $\bar{x}$ are all free variables (common to Q' and Q''), such that for any underlying database[3] D, $D \models Q'(\bar{x}) \rightarrow Q''(\bar{x})$. Formulas Q', Q'' are called the *lower* (respectively, *upper*) *approximation part* of Q.

By $Q'(\bar{x})_D$ (respectively, $\langle Q'(\bar{x}), Q''(\bar{x})\rangle_D$) we denote the result of evaluating the query $Q'(\bar{x})$ (respectively, the approximate query $\langle Q'(\bar{x}), Q''(\bar{x})\rangle$) in the database D.

Given a crisp query represented as a logical formula in a first-order language, an approximate query can always be generated automatically.

3. Tolerance spaces

In this paper we generalize the notion of tolerance spaces considered in [9]. Technically, tolerance spaces allow us to classify a universe of individuals into similarity or tolerance neighborhoods based on a parameterized tolerance relation. This is a generalization of the indiscernibility partitions used in rough set theory, where instead of partitions, the neighborhoods provide a covering of the universe. Conceptually, these spaces are quite versatile in terms of application. In this paper, they are used for the representation of limitations on an agent's perceptual capabilities, sensor uncertainty, and approximate databases.

In distinction to the work in [9], tolerance functions, which are part of a tolerance space are generalized to be partial. Thus, the similarity relation between individuals in a domain can be partial. For some objects it might be unknown whether they are similar or not. This is important in applications where only partial knowledge about the embedding environment is available. An example of such an application might be a sonar model (see Section 5), where objects hidden behind other objects cannot be reached by a sound wave. This results in the unavailability of information about such objects.

Definition 3.1. By a *partial tolerance function* on a set U we mean any function

$\tau : U \times U \rightarrow [0, 1] \cup \{\textsc{Unknown}\}$

such that for all $x, y \in U$, $\tau(x, x) = 1$ and $\tau(x, y) = \tau(y, x)$.

A partial tolerance function is called a *total tolerance function* if, for any $x, y \in U$, $\tau(x, y) \in [0, 1]$.

Definition 3.2. For $p \in [0, 1]$, by a *tolerance relation to a degree at least p, based on* τ, we mean the relation τ^p given by $\tau^p \stackrel{\text{def}}{=} \{\langle x, y\rangle \mid \tau(x, y) \geqslant p\}$. The relation τ^p is also called the *parameterized tolerance relation*. The notation $\tau^p(x, y)$ is used to denote the characteristic function for the relation τ^p.

Remark 3.3. Observe that $\tau(x, y) \geqslant p$ does not hold when $\tau(x, y) = \textsc{Unknown}$.

A parameterized tolerance relation is used to construct tolerance neighborhoods around individuals.

Definition 3.4. By a *neighborhood function wrt* τ^p we mean a function given by

$n^{\tau^p}(u) \stackrel{\text{def}}{=} \{u' \in U \mid \tau^p(u, u') \text{ holds}\}.$

By a *neighborhood* of u wrt τ^p we mean the approximate set $\langle n_{+}^{\tau^p}(u), n_{\oplus}^{\tau^p}(u)\rangle$, where $n_{+}^{\tau^p}(u) \stackrel{\text{def}}{=} n^{\tau^p}(u)$ and $n_{\oplus}^{\tau^p}(u) \stackrel{\text{def}}{=} (n^{\tau^p}(u) \cup \{y \mid \tau(u, y) = \textsc{Unknown}\})$ are *lower and upper approximations of the neighborhood*, respectively.

Remark 3.5

1. The lower approximation $n_{+}^{\tau^p}(u)$ consists of elements which, in the context of available knowledge, are surely similar enough to u, while the upper approximation $n_{\oplus}^{\tau^p}(u)$ consists additionally of elements that might be similar to u due to the unknown status of the tolerance function.
2. It is important to note that the above choice of the definition of $n_{+}^{\tau^p}(u)$ results in non-monotonic behavior of the induced reasoning mechanism at the symbolic level. Namely, updating information about unknown similarities can reduce the size of concept approximations and, in effect, reduce the number of derived consequences. The choice made in Definition 3.4 reflects, in a sense, the open world assumption. In more advanced applications one might want to define $n_{+}^{\tau^p}(u)$ using a more flexible formalism of local closed world assumption, as defined, e.g., in [10,11].
3. Note that in the case where τ is a total tolerance function, $n_{+}^{\tau^p}(u) = n_{\oplus}^{\tau^p}(u)$, and the neighborhood can be considered as a single crisp set rather than a pair of approximations. In this case, a neighborhood coincides with that used previously in [9].

Partial and total tolerance spaces play a fundamental role in our approach to modeling perceptual limitations of agents and approximate relations in databases.

[3] We deal with relational databases, where queries are formulated as first-order or fixpoint formulas (for textbooks on this approach see, e.g., [6–8]).

Definition 3.6. A *partial tolerance space* is defined as tuple $TS = \langle U, \tau, p \rangle$, consisting of

- a non-empty set U, called the *domain* of TS;
- a *partial tolerance function* τ;
- a *tolerance parameter* $p \in [0,1]$.

If τ is a total tolerance function, then TS is called a *total tolerance space*.

Remark 3.7. In the sequel, total tolerance functions and total tolerance spaces are often simply called *tolerance functions* and *tolerance spaces*.

We will also require a definition of approximations of a set (relation) in the context of partial tolerance spaces.

Definition 3.8. Let $TS = \langle U, \tau, p \rangle$ be a partial tolerance space and let $S \subseteq U$. The *lower and upper approximations of S wrt TS*, denoted respectively by S_{TS^+} and $S_{TS^\oplus}$, are defined by

$$S_{TS^+} \overset{\text{def}}{=} \{u \in U : n_+^{\tau_p}(u) \subseteq S\}$$

$$S_{TS^\oplus} \overset{\text{def}}{=} \{u \in U : n_\oplus^{\tau_p}(u) \cap S \neq \emptyset\}.$$

As in the case using total tolerance spaces, the approximations defined above coincide with those given in [9].

Remark 3.9. In many applications, e.g., shape recognition, one has to consider tolerance spaces on sets. In order to induce such tolerance spaces on the basis of tolerance spaces on elements, we require tolerance functions on sets. Similarity measures on sets that could be adapted to the context of tolerance spaces, have been intensively studied in the area of computer vision (see e.g., [12]) and fuzzy sets (see e.g., [13]).

4. A generic sensor model

In this section, we provide a simple sensor model (see also [9]), based on a generalization of that in [14], and a method for modeling uncertainty in sensor data which integrates well with tolerance spaces.

The point to this is an assumption that basic or primitive properties and relations in many domains will be derived through various aggregations of sensor data. Sensor data involves individual readings of values with uncertainty due to noise, etc. In essence, individual data values have neighborhoods of indiscernibility. Tolerance spaces can be used to represent such indiscernibility or similarity. Using tolerance spaces on data readings as building blocks, primitive approximate relations can be defined which inherit the approximate readings of data values. The following sections are intended to show how this process may be set up in addition to how some standard probabilistic approaches to modeling sensors may be integrated. Work in the latter sections should be considered tentative.

A sensor is used to measure one or more physical attributes in an environment E. The value sets associated with a physical attribute might be the real numbers, as in the case of measurement of the temperature or velocity of an object; Boolean values, as in the measurement of the presence or absence of an object such as a red car; integer values, as in the case of measurement of the number of vehicles in a particular intersection; or scalar values, such as the specific color of a vehicle.

An environment E can be viewed as an abstract entity containing a collection of physical attributes that are measurable. Vectors or n-dimensional arrays of attribute/value pairs could be used to represent a particular environment. One may want to add a temporal argument to E, so that the current state of the environment is dynamic and changes with time.

Any attribute a can be viewed as a function of the environment E and time point t, i.e., $a : E \times \textsc{Time} \to V_a$, where $\textsc{Time}$ is the set of considered time points and V_a is the set of possible values of a.

We consider a sensor S_i as a pair of functions of the environment E and time point t, $S_i(E,t) = \langle V_i(E,t), \epsilon_i(E,t) \rangle$. Depending on the type of sensor being modeled, $V_i(E,t)$ can be a function that returns the values of the physical attributes associated with the sensor, as sensed at time t in environment E. V_i might return a single value, as in the case of a single temperature sensor, or a vector or array of values for more complex sensors. For any physical attribute measured, explicit accuracy bounds are supplied in the form of $\epsilon_i(E,t)$. The temporal and environment arguments are supplied since the accuracy of a sensor may vary with time and change of environment. As in the case of V_i, ϵ_i might return a single accuracy bound or a vector or array of accuracy bounds.

For example, suppose S_{temp} is a sensor measuring the temperature of a PC104 box on an UAV.[4] Let a_{temp} be the physical attribute associated with temperature in the environment, where the actual temperature is $a_{temp}(E,t)$ and the value returned by the sensor is $V_{temp}(E,t)$. The following constraint holds:

$$a_{temp}(E,t) \in [V_{temp}(E,t) - \epsilon_{temp}(E,t), V_{temp}(E,t) + \epsilon_{temp}(E,t)].$$

By using tolerance spaces, accuracy bounds for a physical attribute can be represented equivalently as parameterized tolerance relations on the value set for the attribute. In this manner, we can use neighborhood functions to reason about the tolerance or accuracy neighborhoods around individual sensor readings and combine these into neighborhoods for more complex virtual sensors.

[4] Unmanned aerial vehicle.

In the following, we will drop the environment and temporal argument for ϵ_{ik} and assume the accuracy bounds for attributes do not change with environment and time.

Let $\bar{a}_i = \langle a_{i1}, \ldots, a_{ik} \rangle$ be the tuple of attributes measured by sensor S_i and let $TS_{ik} = \langle V_{ik}, \tau_{ik}, p_{ik} \rangle$ be a tolerance space for the kth physical attribute, a_{ik}, associated with the sensor S_i, where

- $V_{ik} \stackrel{\text{def}}{=} \{x \in D_{ik} | lb_{ik} \leqslant x \leqslant ub_{ik}\}$, where D_{ik} is a value domain such as the reals or integers. It is assumed that the legal values for a physical attribute have a lower and upper bound, lb_{ik}, ub_{ik}. We associate the same distance measurement $\delta(x,y) \stackrel{\text{def}}{=} |x-y|$ with all value sets V_{ik};
- both the tolerance function τ_{ik}, and the tolerance parameter p_{ik} are defined by

$$\tau_{ik}(x,y) \stackrel{\text{def}}{=} 1 - \frac{\delta(x,y)}{\delta(lb_{ik}, ub_{ik})}, \quad p_{ik} \stackrel{\text{def}}{=} 1 - \frac{\epsilon_{ik}}{\delta(lb_{ik}, ub_{ik})}.$$

The neighborhood function can be used to compute the possible actual values of a physical attribute in the environment, given a sensor reading, under the assumption that the accuracy bounds have been generated correctly for a particular sensor and the sensor remains calibrated. For example, if $V_{temp}(E,t)$ is the current value measured by the sensor S_{temp}, then we would know that $a_{temp}(E,t) \in n_{p_{temp}}(V_{temp}(E,t))$. So, the tolerance neighborhood around a sensor reading always contains the actual value of the physical attribute in the environment E and it would be correct to reason with the neighborhoods of sensor values, rather than the sensor values themselves.

Example 4.1. Let S_R, S_G and S_B be sensors detecting values of R, G, B color attributes.[5] The universe of values is restricted in those cases to integers in interval [0, 255]. Assume that all sensors have the same accuracy, say 5. Then the tolerance space for all three cases is $\langle [0,255], \tau, p \rangle$, where

$$\tau(x,y) = 1 - \frac{|x-y|}{255}, \quad p = 1 - \frac{5}{255} \approx 0.9804.$$

In this case an agent using sensor data from S_R, S_G, S_B is unable to distinguish between color values where values of τ on R, G, B attributes are greater than or equal to 0.9804. These physical attributes and their associated tolerance spaces can be used to construct more complex attributes and knowledge structures in terms of these. These new attributes and knowledge structures would inherit the accuracy (inaccuracy) of the primitive sensor data used in their construction. We consider this in Section 7.

5. Probabilistic sensor models

In many cases, a sensor model returns a (partial) probability function $p(e)$ meaning that a particular event can correctly be classified as e with probability p. As an example, we consider below a probabilistic sonar model discussed, e.g., in [15]. Sonar refers to any sensor using sound to measure range. The field of view of the sonar is projected onto an *occupancy grid* (see Fig. 3).

The region marked as **I** is probably empty, region **II** contains elements (rectangles of the occupancy grid) probably occupied, and regions **III** and **IV** contain elements for which it is unknown whether they are occupied or not.[6]

The size of grid elements (rectangles) depends on a particular application.

Probabilistic models for sonars return functions $p_{Occ}(e)$ meaning that grid element (rectangle) e is occupied with probability p (assuming that $p_{Occ}(e)$ can be UNKNOWN). Suitable formulas for calculating such probabilities are provided, e.g., in [15]. Observe that probabilities in regions **III** and **IV** are unknown, so partial tolerance spaces should be useful for modeling these kinds of situations.

In many applications we basically deal with shapes that do not have to be unions of grid elements. Such shapes can be considered as arbitrary sets of points. The first tolerance space we define, TS_{grid}, is then the one that represents grid elements as neighborhoods.

We let $TS_{grid} = \langle V_{points}, \tau_{points}, 1.0 \rangle$, where

- V_{points} is the set of points in a given occupancy grid;
- $\tau_{points}(a,b) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{when points } a, b \text{ are in the} \\ & \quad \text{same grid rectangular} \\ 0 & \text{otherwise.} \end{cases}$

Observe that TS_{grid} is total and the neighborhood of a point is the rectangular the point belongs to. Any complex 2-dimensional shape can now be represented by lower and upper approximations wrt TS_{grid}. In fact, due to the sonar resolution, its measurements do not distinguish between points inside the same rectangle.

Now assume that the probabilities supplied by the sonar are themselves approximate. This might, at first glance, seem a bit unusual. However, the calculated probabilities are usually indeed approximate. It is then reasonable to consider a range approximating the actual probability, rather than its single value.

[5] Of course, there are many techniques for dealing with noise and uncertainty associated with color, more sophisticated than R, G, B attributes. On the other hand, this domain provides a simple and intuitive vehicle to present our ideas.

[6] The elements in region III are hidden behind elements probably occupied and elements in region IV are outside of the sonar's field of view.

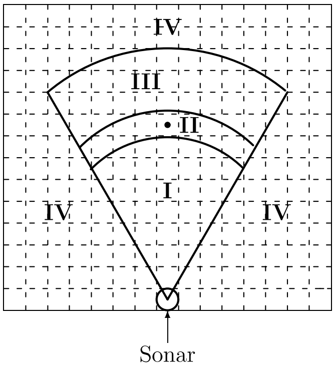

Fig. 3. Sonar's field of view projected into an occupancy grid.

To model this situation we first define a partial distance function δ_{prob} on probabilities (in $[0,1] \cup \text{Unknown}$) returned as sonar results,

$$\delta_{prob}(p_1,p_2) \overset{\text{def}}{=} \begin{cases} |p_1 - p_2| & \text{when } p_1,p_2 \neq \text{ Unknown} \\ \text{Unknown} & \text{otherwise.} \end{cases}$$

Now we define the tolerance space on probabilities, $TS_{prob} = \langle V_{prob}, \tau_{prob}, p_{prob} \rangle$, where

- $V_{prob} \overset{\text{def}}{=} [0,1] \cup \text{Unknown};$
- $\tau_{prob}(p_1,p_2) \overset{\text{def}}{=} \begin{cases} 1 - \delta_{prob}(p_1,p_2) \\ \quad \text{when } \delta_{prob}(p_1,p_2) \neq \text{ Unknown} \\ \text{Unknown} \quad \text{otherwise.} \end{cases}$
- $p_{prob} \in [0,1].$

We are interested in approximating the relation $Occ(a)$, which denotes that a point a in the occupancy grid is occupied. Since Occ is a Boolean relation, we have to transform the probabilistic measure provided by the sonar model into a two-valued one. To do this, we assume that a given grid element e is occupied when $p_{Occ}(e)$ is greater than or equal to a certain threshold.

We then first assume a probability threshold, say q, such that,[7]

$$Occ(a) \overset{\text{def}}{\equiv} [p_{Occ}(n^{TS_{grid}}(a)) \geqslant q] \equiv [p_{Occ}(n^{TS_{grid}}(a)) \in [q,1]].$$

Now the required approximations are given by:

$$Occ_{TS^{+}_{prob}}(a) \equiv [p_{Occ}(n^{TS_{grid}}(a)) \in [q,1]_{TS^{+}_{prob}}]$$
$$Occ_{TS^{\oplus}_{prob}}(a) \equiv [p_{Occ}(n^{TS_{grid}}(a)) \in [q,1]_{TS^{\oplus}_{prob}}].$$

Example 5.1. Assume that the probability threshold $q = 0.8$ and the tolerance parameter in TS_{prob} is $p_{prob} = 0.95$. Consider the situation presented in Fig. 4, where the area of interest, reflected by the occupancy grid, is restricted to the field of view of the sonar.

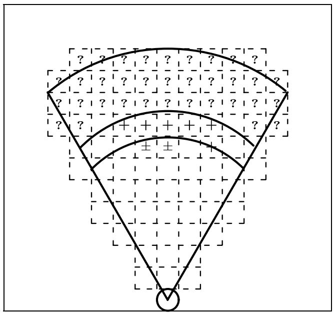

Fig. 4. Sonar's field of view considered in Example 5.1.

Assume further that grid elements marked by + are occupied with probability greater than 0.9, those marked by ± are occupied with probability 0.77 and that unmarked elements are occupied with probability less than 0.75. Probabilities associated with elements marked by '?' are unknown.

It is easily seen that $Occ_{TS^{+}_{prob}}(a)$ holds for points belonging to grid elements marked by +, whereas $Occ_{TS^{\oplus}_{prob}}(a)$ holds, in addition, for points belonging to grid elements marked by ± or by '?'

6. Dempster–Shafer theory and tolerance spaces

The Dempster–Shafer theory of evidence provides an alternative approach to sensor models (see, e.g., [15]). In contrast to pure probabilistic approaches, the theory deals with representation of partial evidence by means of belief functions and belief mass. Each sensor contributes to a belief mass $m(P)$, where P is any set of propositions. The set of all propositions in question is referred to as a *frame of discernment*. It is assumed that the total belief mass is 1.0, i.e., if the frame of discernment is Θ then $\sum_{\emptyset \neq P \subseteq \Theta} m(P) = 1.0$.

Consider the sonar model and property Occ, as described in Section 5. Here we are interested in the frame of discernment $\Theta = \{Occ, Empty\}$. In our case the belief mass, for a given point a, is distributed between $\{Occ(a)\}$,$\{Empty(a)\}$,$\{Occ(a), Empty(a)\}$. The last set represents the level of ignorance. Thus the distribution of belief mass can be represented as triples $\langle o,e,u \rangle$, meaning that $m(\{Occ(a)\}) = o$, $m(\{Empty(a)\}) = e$ and $m(\{Occ(a), Empty(a)\}) = u$.[8]

Let

$$\begin{aligned} Occ(a) \overset{\text{def}}{\equiv} [& m(\{Occ(a)\}) \\ & \geqslant m(\{Empty(a)\}) \wedge m(\{Occ(a)\}) - m(\{Empty(a)\}) \\ & \geqslant q \wedge m(\{Occ(a), Empty(a)\}) \leqslant r], \end{aligned}$$

where $q, r \in [0,1]$ are fixed parameters.

[7] Recall that $n^{TS_{grid}}(a)$ is the grid element containing point a.

[8] In the general case we deal with tuples of arity equal to the number of all non-empty subsets of Θ.

Consider a tolerance space on triples

$$TS \overset{\text{def}}{=} \langle \{ \langle o,e,u \rangle | o,e,u \in [0,1] \text{ and } o+e+u = 1.0 \}, \tau, p \rangle,$$

where, for example, $\tau(\langle o,e,u \rangle, \langle o',e',u' \rangle) \overset{\text{def}}{=} 1 - \max \{ |o-o'|, |e-e'|, |u-u'| \}$ and $p \in [0,1]$ is a fixed parameter. Then,

$$\begin{aligned} Occ_{TS^{+}}(a) \equiv {} & [\forall o,e,u \tau(\langle o,e,u \rangle, \langle m(\{Occ(a)\}), \\ & m(\{Empty(a)\}), m(\{Occ(a), Empty(a)\}) \rangle) \\ & \geqslant p \rightarrow [o > e \wedge (o-e) \geqslant q \wedge u \leqslant r]] \\ Occ_{TS^{\oplus}}(a) \equiv {} & [\exists o,e,u \tau(\langle o,e,u \rangle, \langle m(\{Occ(a)\}), \\ & m(\{Empty(a)\}), m(\{Occ(a), Empty(a)\})) \\ & \geqslant p \wedge o > e \wedge (o-e) \geqslant q \wedge u \leqslant r]. \end{aligned}$$

Although these results are tentative in nature, one can begin to see how standard probabilistic models of sensors can be integrated with approximate relations using tolerance spaces as a bridge. Assuming this is possible, such sensor-data induced relations can be stored in approximate databases and be queried by agents. This is the topic for the following section.

7. Approximate databases and sensors

Standard relational databases store relations as tables where each column represents an argument to the relation and a row represents the instantiation of each relational argument to a value in that argument's value space. Each row is a tuple of which the relation represented by the table holds. In the standard case, each argument has a specific value in its value space, but if a tolerance space is associated with an argument then it has the effect of creating neighborhoods of uncertainty, similarity, or indiscernibility around each argument value.

This should then induce a tolerance space for specific tuple domains creating neighborhoods around tuples. This should in turn affect the answers to any queries to relations in the database since the question is not whether a tuple holds for a relation, but whether the tuple through the filter of its associated tolerance space holds. In addition, the relations stored in the database are approximate relations having both a lower and upper approximation.

For instance, suppose we were to use the sensor attributes for S_R, S_G and S_B from Example 4.1 in defining a relation *reddish* or *darkish* in color where each of these relations take S_R, S_G and S_B as arguments. Each argument has a tolerance space associated with it which is determined by the specific characteristics of the sensors used to measure these attributes or even takes into account specific contexts of use. Since these tolerance spaces are parameterized, parameters can be contextualized and derived through machine learning techniques or statistical and probabilistic methods.

The next step is to integrate the approximate nature of arguments into the definitions of relations which use these arguments. The net result is a tolerance space for an approximate relation where any tuple in the relation has its own neighborhood induced by the tolerance space for the relation. Rather than asking whether a tuple is a member of a relation, we ask whether that tuple's neighborhood is included in the lower approximation of the relation or intersects with its upper approximation as in rough set theory. In this case we use tolerance spaces on tuples rather than the usual discernibility partitions for attributes.

Suppose TS_R, TS_G and TS_B are tolerance spaces for the sensor attributes, as in Example 4.1. Then, in the case of the relation *darkish* for example, we would like to generate a tolerance space $TS_D = \langle U, \tau, p \rangle$ where U is the set of triples representing RGB values and where $TS_D = f(TS_R, TS_G, TS_B)$. The function f can be defined in many ways or even machine learned. Its definition will generally be dependent on the domain and application in question.

In a similar manner, one can define additional tolerance spaces for new relations in terms of the tolerance spaces associated with the relations used to define those relations. In this manner one can recursively construct complex knowledge representations at many levels of abstraction which inherit the approximativeness of sensor output and more primitive defining relations.

We also assume that each relation in the database has a lower and upper approximation. In the case of the relation *darkish*, both $darkish_{+}$ and $darkish_{\oplus}$, the lower and upper approximations for *darkish* would be stored or implicitly represented in the database. There are a number of ways to generate approximate databases. A direct method would be, for example, to use rough set machine learning techniques (see, e.g., [16]) to automatically generate lower and upper approximations for approximate relations. An indirect method would start with a relational database and tolerance spaces for each of the relations. These tolerance spaces could then be used to automatically generate lower and upper approximations for each relation. This is demonstrated in Example 7.3. Under these assumptions, we would have an approximate relational database with tolerance spaces associated with some or all of the approximate relations.

Let us now assume that an agent wants to access information in its internal database which is in fact approximate and represents that agent's perceptual limitations as encoded through the tolerance spaces used to generate the approximate database. The database would then be queried in the following manner. Given a query to the database represented as a logical formula in a first-order language, the query is automatically transformed into an approximate query.[9] One can then generate all tuples satisfying the lower and upper approximations of the query or simply ask whether a specific tuple satisfies the query.

These techniques describe how approximate knowledge structures, which take both sensor and relational uncertainty into account, can be generated and represented as approximate databases using tolerance spaces. In Section 8, it is shown how tolerance spaces representing perceptual limitations of agents themselves can be used together with approximate queries to take these limitations into account when asking and receiving answers from other agents. In fact, one can even model the fact that an agent may have contextual perceptual limitations when asking questions of itself since additional tolerance spaces can be applied in asking questions to an approximate database as we will see.

Definition 3.8 does the job. In this definition, U might represent a primitive data set such as that used for a particular sensor, or a complex data set such as a set of tuples. For example, consider a relational database with one relation S of k-arity and with universe U consisting of all k-tuples. In this case, the relation may represent raw data about S. Suppose there is also a tolerance space $TS = \langle U, \tau, p\rangle$. TS creates neighborhoods around tuples. An agent, when asking whether a tuple $\bar{x}$ is a member of the relation is really asking what is the relationship between the neighborhood around the tuple and the relation. If the neighborhood is included in the relation, the answer is 'yes', if their intersection is non-empty, the answer is 'maybe', if the intersection is empty, the answer is 'no'.

In fact, this particular use of tolerance spaces can be generalized to relational databases with an arbitrary number of relations where the data in the database is assumed to be raw data about the relations. Using tolerance spaces, the relational database can be turned into an approximate database where each relation is viewed as having an upper and lower approximation. Rather than machine learn the approximate relations, one can assume the tolerance spaces as given and apply them to raw data to generate an approximate database. The following definitions and example show how this is done.

Definition 7.1. Let $D = \langle U, \{r^j | j \in J\}\rangle$ be a relational database. Then we say that a sequence of tolerance spaces $\overline{TS} = \langle TS_j | j \in J\rangle$ is *compatible with D*, provided that for any $j \in J$, $TS_j = \langle U_j, \tau_j, p_j\rangle$, where U_j is the set of all tuples of arity the same as the arity of r^j.

Definition 7.2. Let $D = \langle U, \{r^j | j \in J\}\rangle$ be a relational database and $\overline{TS}$ be a sequence of tolerance spaces compatible with D. If D is crisp, then by an *approximation of D wrt* $\overline{TS}$, we mean the structure $D^{\overline{TS}} = \left\langle U, \left\{\left\langle r^j_{TS^+_j}, r^j_{TS^\oplus_j}\right\rangle \mid j \in J\right\}\right\rangle$.

If D is approximate, where for $j \in J$, $r^j = \langle r^j_+, r^j_\oplus\rangle$, then the *approximation of D wrt* $\overline{TS}$ is defined as $D^{\overline{TS}} = \left\langle U, \left\{\left\langle (r^j_+)_{TS^+_j}, (r^j_\oplus)_{TS^\oplus_j}\right\rangle | j \in J\right\}\right\rangle$.

Note that in the latter case, one can still apply additional tolerance spaces to upper and lower approximations of a relation since these are also represented as relations in the database.

Example 7.3. Consider a situation where a ground operator (agent Ag_G) is communicating with a UAV (agent Ag_V), which is flying over a road segment. Assume Ag_V can provide information about the following relations, and that Ag_V has these in common with Ag_G:

- $V(x, y)$—there is a visible connection between objects x and y;
- $S(x, y)$—the distance between objects x and y is small;
- $E(x, y)$—objects x and y have equal speed;
- $C(x, z)$—object x has color z, where we consider colors b, r, dr, denoting "brown", "red" and "dark red", respectively.

Let the actual situation on a road segment be given by the (crisp) relational database shown in Table 1, where, e.g.,

- the first row means that there is a visible connection between objects 1 and 2, the distance between object 1 and objects 2, 5 is small, object 1 has equal speed with objects 2, 5 and that the color of object 1 is r;
- the third row means that there is no visible connection between object 3 and any other object, the distance between object 3 and object 2 is small, object 3 has equal speed with object 2 and that the color of object 3 is dr.

Note that our UAV agent does not have direct access to this information since it may have limited perceptual

Table 1
Database considered in Example 7.3 reflecting a situation on a road segment

Object	V	S	E	C
1	2	2, 5	2, 5	r
2	1	1, 3, 4	1, 3, 4	b
3	–	2	2	dr
4	–	2	2	r
5	–	1	1	dr

[9] Both the original and approximate query can be translated into an SQL query in a straightforward manner.

capabilities relative to some attributes such as color which is modeled below.

Consider the approximation of the relational database given in Table 1 wrt the tolerance space $T_V = \langle U, \tau_V, p_V \rangle$, where $\tau_V^{p_V}$ identifies equal elements and additionally *dr* with *b*. This tolerance space is intended to represent a perceptual limitation of the UAV agent. The resulting approximation is presented in Tables 2 and 3. Now, e.g.,

- the first row in Table 2 indicates that there surely is a visible connection between objects 1 and 2, the distance between object 1 and objects 2, 5 is surely small, object 1 has surely equal speed with objects 2, 5 and that the color of object 1 is surely *r*;
- the third row in Table 3 indicates that there cannot be any visible connection between object 3 and any other object, the distance between object 3 and object 2 might be small, object 3 might have equal speed with object 2 and that the color of object 3 might be *b* or *dr*.

Table 2
Approximation (lower approximations) of the relational database given in Table 1 wrt the perception capabilities of agent Ag_V

Object	V_+	S_+	E_+	C_+
1	2	2, 5	2, 5	*r*
2	1	1, 3, 4	1, 3, 4	–
3	–	2	2	–
4	–	2	2	*r*
5	–	1	1	–

Table 3
Approximation (upper approximations) of the relational database given in Table 1 wrt the perception capabilities of agent Ag_V

Object	$V_\oplus$	$S_\oplus$	$E_\oplus$	$C_\oplus$
1	2	2, 5	2, 5	*r*
2	1	1, 3, 4	1, 3, 4	*b*, *dr*
3	–	2	2	*b*, *dr*
4	–	2	2	*r*
5	–	1	1	*b*, *dr*

Note that several tolerance spaces could be associated with each type of data in a table if desired.

We now consider how to generate an approximate query from a crisp query represented as a logical formula.

Definition 7.4

- A relation symbol *r* *occurs positively* (respectively *negatively*) in a formula if it appears under an even (respectively odd) number of negations.[10]
- For any formula α,
  - by α_+ we understand the formula α in which any positive occurrence of a relation symbol, say r^j, is replaced by r^j_+ and any negative occurrence of r^j is replaced by $r^j_\oplus$;
  - by $\alpha_\oplus$ we understand the formula α in which any positive occurrence of a relation symbol, say r^j, is replaced by $r^j_\oplus$ and any negative occurrence of r^j is replaced by r^j_+.

Example 7.5. Consider formula $r^1(\bar{x}) \wedge \neg r^2(\bar{y})$. Then

- $[r^1(\bar{x}) \wedge \neg r^2(\bar{y})]_+ \equiv [r^1_+(\bar{x}) \wedge \neg r^2_\oplus(\bar{y})]$;
- $[r^1(\bar{x}) \wedge \neg r^2(\bar{y})]_\oplus \equiv [r^1_\oplus(\bar{x}) \wedge \neg r^2_+(\bar{y})]$.

The two formulas on the right-hand side would represent an approximation of the original formula.

We allow for the possibility of providing a tolerance space with an approximate query whose purpose is to represent a contextual perceptual limitation of the querying agent. Such queries are called tolerance queries, the definition of which follows.

Definition 7.6. Let *D* be a (crisp or approximate) database. By a *tolerance query* we mean a pair $\langle Q, TS_Q \rangle$, where

- $Q = \langle Q'(\bar{x}), Q''(\bar{x}) \rangle$ is an approximate query;
- TS_Q is a tolerance space for tuples of arity the same as the arity of $\bar{x}$.

If $\overline{TS}$ is a sequence of tolerance spaces compatible with *D*, then the *meaning of a tolerance query Q in a database D wrt a context* $\overline{TS}$ is given by

$$\left\langle [(Q'(\bar{x}))_{D^{\overline{TS}}}]_{TS^+_Q}, [(Q''(\bar{x}))_{D^{\overline{TS}}}]_{TS^\oplus_Q} \right\rangle.$$

8. Agent communication with heterogeneous perceptual capabilities

Consider a multi-agent application in a complex environment such as the Web where software agents reside, or a natural disaster in an urban area where physical robots reside. Each agent will generally have its own view of its environment due to a number of factors such as the use of different sensor suites, knowledge structures, reasoning processes, etc. (see Fig. 1). Agents may also have different understandings of the underlying concepts which are used in their respective representational structures and will measure objects and phenomena with different accuracy. How then can agents with different

[10] As usual, it is assumed here that all implications of the form $p \to q$ are substituted by $\neg p \vee q$ and all equivalences of the form $p \equiv q$ are substituted by $(\neg p \vee q) \wedge (\neg q \vee p)$.

knowledge structures and perceptual accuracies understand each other and effect meaningful communication and how can this be modeled? In this section, we propose a framework to do this using tolerance spaces as the main representational tool to model many of these types of limitations and mismatches.

The approach may be summarized as follows. It is assumed that each agent has its own database and that agents have a common ontology. The database may be crisp or approximate and generated in any number of ways, some having been demonstrated already. The idea is that some perceptual and other limitations have already been encoded in the respective databases of the agents. For any tolerance agent, we also assume an additional context consisting of a sequence of tolerance spaces. These may cover all, some or none of the relations in the database and are intended to represent additional limitations which are contextual in nature. The agent need not be aware of these limitations, but will always view its knowledge through this filter when asking questions internally and this context may be used when generating a tolerance query to be asked of another agent.

When an agent asks a question of another agent using a tolerance query, the question is interpreted by the other agent through its context and its database. Two sets of tuples are returned, representing the lower and upper approximation of the original query. The agent who asked the question, will then apply the tolerance space associated with its tolerance query to the result returned by the questioned agent. The net result is an answer which takes into account both the perceptual limitations of the questioned agent and the current limitation associated with the tolerance query. Initial work with these ideas may be found in [9].

We begin with a general definition of a *tolerance agent*, specializing that provided in [9].

Definition 8.1. By a *tolerance agent* we shall understand any tuple $\langle Ag, D, \overline{TS}\rangle$, where Ag is an agent, D is its (crisp or approximate) database and $\overline{TS}$, called the *context of agent Ag*, is a sequence of tolerance spaces compatible with D.

Here we do not define what an agent is specifically, as the framework we propose is independent of the particular details. The assumption is that the Ag part of a tolerance agent consists of common functionalities normally associated with agents such as planners, reactive and other methods, knowledge bases or structures, etc. The knowledge bases or structures are also assumed to have a relational component consisting of a relational database (D). When the agent introspects and queries its own knowledge base its limited perceptual capabilities are reflected in any answer to a query due to its context.

Suppose that two tolerance agents have different perceptual capabilities and consequently different tolerance spaces. It is then necessary to define the meaning of queries and answers relative to the two tolerance agents. As previously advocated, a tolerance agent, when asked about a relation, answers by using the approximations of the relation wrt its tolerance space. On the other hand, the agent that asked the query has to understand the answer provided by the other agent wrt to its own tolerance space.

The dialog between two tolerance agents:

- *query agent* $TA_1 = \langle Ag_1, D_1, \overline{TS_1}\rangle$,
- *answer agent* $TA_2 = \langle Ag_2, D_2, \overline{TS}_2\rangle$,

then conforms to the following schema:

1. TA_1 asks TA_2 a question using a tolerance query $Q = \langle\langle Q', Q''\rangle,\ T_Q\rangle$; in fact, it sends to TA_2 the approximate query $\langle Q', Q''\rangle$ without T_Q;
2. TA_2 computes the answer approximating its database according to its current context $\overline{TS}_2$ and returns as an answer the approximate relation $\langle Q'_{+}, Q''_{\oplus}\rangle_{D_2^{\overline{TS}_2}}$. In order to simplify notation, we denote this relation by $R = \langle R', R''\rangle$;
3. TA_1 receives R as input and interprets it according to the context T_Q indicated in the query. The resulting interpretation, $\langle R'_{T_Q^+}, R''_{T_Q^\oplus}\rangle$, provides the answer to the query, as understood by TA_1 and takes into account the perceptual limitations of both agents.

This schema will only work properly under the assumption of a common vocabulary, but see [17] for additional methods related to agent querying where agent vocabularies are mismatched.[11]

Remark 8.2. Observe that the context of agent Ag_1 is not present in the schema directly. One can, however, observe, that T_Q usually strongly depends on $\overline{TS}_1$. In particular, if Q is of the form $r^j(\bar{x})$ then, in most cases, T_Q is the jth tolerance space in $\overline{TS}_1$.

The definition describing this interaction now follows.

Definition 8.3. Let TA_1 and TA_2 be as above. Let $Q = \langle\langle Q', Q''\rangle, TS_1\rangle$ be a tolerance query, which is asked by TA_1 and answered by TA_2. Then the *result of query Q* is defined as the lower and upper approximation wrt TS_1 of the meaning[12] of Q in database D_2 wrt context $\overline{TS}_2$.

Example 8.4. Consider first a tolerance agent $\langle Ag_V, D, T_V\rangle$, where Ag_V and the tolerance space T_V are as provided in Example 7.3 (i.e., Ag_V does not recognize the difference between colors *dr* and *b*).

[11] The machinery of [17] provides in such cases answers to queries, which are approximate in the sense of Definition 2.1.

[12] As provided by Definition 7.6.

Consider the following query which the agent asks itself internally:[13]

$$\langle V(x,y) \wedge [C(x,r) \vee C(y,b)], V(x,y) \wedge [C(x,r) \vee C(x,dr) \vee C(y,b)], T_V\rangle.$$

To compute the answer we first consider (cf. Definition 7.6)

$$\langle [V(x,y) \wedge [C(x,r) \vee C(y,b)]]_{+}, [V(x,y) \wedge [C(x,r) \vee C(x,dr) \vee C(y,b)]]_{\oplus}\rangle,$$

which, according to Definition 7.4, is

$$\langle V_{+}(x,y) \wedge [C_{+}(x,r) \vee C_{+}(y,b)], V_{\oplus}(x,y) \wedge [C_{\oplus}(x,r) \vee C_{\oplus}(x,dr) \vee C_{\oplus}(y,b)]\rangle.$$

Using the approximations of V, S, E and C wrt T_V, from Tables 2 and 3, the above query evaluates to $\langle\{1,2\}\rangle$,$\langle\{1,2\}\rangle$.

One may interpret the result as stating that the tuple $\langle 1,2\rangle$ definitely satisfies the original query while the tuple $\langle 2,1\rangle$ may satisfy the original query, but more precision would be required to state this with certainty.

Example 8.5. Consider the tolerance agents $\langle Ag_V, T_V\rangle$ and $\langle Ag_G, T_G\rangle$ where

- Ag_V and T_V are as described in Example 8.4;
- $T_G = \langle U, \tau_G, p_G\rangle$ such that $\tau_G^{p_G}$ identifies equal elements and additionally dr with r. (The database approximation wrt T_G is shown in Tables 4 and 5.)

Suppose Ag_G wants to ask Ag_V for information about colors of objects satisfying the following tolerance query:[14]

$$\langle\langle \exists x,y \cdot [V(x,y) \wedge C(x,z)], \exists x,y \cdot [S(x,y) \wedge E(x,y) \wedge C(x,z)]\rangle, T_G\rangle.$$

Using Definition 8.3, agent Ag_V then evaluates this tolerance query in the context of its perception capabilities, i.e., according to the database approximation given in Tables 2 and 3. The answer returned by Ag_V is

$$\langle \exists x,y \cdot [V_{+}(x,y) \wedge C_{+}(x,z)], \exists x,y \cdot [S_{\oplus}(x,y) \wedge E_{\oplus}(x,y) \wedge C_{\oplus}(x,z)]\rangle.$$

Thus Ag_V returns to Ag_G answer $\langle\{r\},\{r,b,dr\}\rangle$. Ag_G then computes the final answer by interpreting the above one relative to its tolerance space T_G using Definition 8.3.

The final answer is then $\langle\emptyset, \{r,b,dr\}\rangle$.

Table 4
Approximation (lower approximations) of the relational database given in Table 1 wrt perception capabilities of agent Ag_G as defined in Example 8.5

Object	V_{+}	S_{+}	E_{+}	C_{+}
1	2	2, 5	2, 5	–
2	1	1, 3, 4	1, 3, 4	b
3	–	2	2	–
4	–	2	2	–
5	–	1	1	–

Table 5
Approximation (upper approximations) of the relational database given in Table 1 wrt perception capabilities of agent Ag_G as defined in Example 8.5

Object	$V_{\oplus}$	$S_{\oplus}$	$E_{\oplus}$	$C_{\oplus}$
1	2	2, 5	2, 5	r,dr
2	1	1, 3, 4	1, 3, 4	b
3	–	2	2	r,dr
4	–	2	2	r,dr
5	–	1	1	r,dr

9. Some complexity issues

Let us assume that the query language used with crisp relational databases (as defined in Definition 2.2) is tractable, i.e., all queries can be computed in deterministic polynomial time. We have already suggested (see Definition 2.3) formulating queries using classical first-order logic or fixpoint calculus. These choices are justified by the following facts (see, e.g., [6]):

- first-order queries are PTIME-computable, i.e., computable in deterministic polynomial time wrt the size of the database domain; moreover such queries are computable in deterministic logarithmic space;
- fixpoint queries are PTIME-computable (thus also in deterministic polynomial space); moreover, fixpoint queries capture the whole complexity class PTIME, provided that the database domain is linearly ordered.

Of course, use of SQL is also tractable. In fact, its expressive power is equal to the expressive power of first-order queries.

Let us consider a fixed crisp or approximate database with domain U. Under the above assumptions, it is easily seen that approximate queries, in the sense of Definition 2.3, are PTIME-computable in the size of U.

Consider a tractable partial tolerance space TS, i.e., a space with a PTIME-computable tolerance function. Then parameterized tolerance relations and neighborhood functions (see Definitions 3.2 and 3.4) are PTIME-computable. In addition, for any set $S \subseteq U$, the lower and upper approximations, $S_{TS^{+}}$ and $S_{TS^{\oplus}}$, (see Definition

[13] For simplicity, we provide one tolerance space and assume that two tuples are identified if the arguments representing color have values within the same neighborhood and arguments not representing colors have equal values.

[14] Observe that $V(x,y) \rightarrow (S(x,y) \wedge E(x,y))$ thus the lower approximation part of the query is indeed included in its upper approximation part.

3.8) are PTIME-computable. In consequence, approximations of databases, as defined in Definition 7.2, are PTIME-computable.

Similarly, since α_+ and $\alpha_\oplus$ of Definition 7.4 may increase the size of a formula by a constant factor only, tolerance queries (see Definitions 7.6 and 8.3) are PTIME-computable. Thus all tolerance-based notions are tractable so the techniques we deal with are tractable too, assuming that all necessary functions, such as probabilities, etc., are tractable too.

The only methods considered in the paper, for which tractability might be difficult to obtain, are those that use sensor models incorporating the Dempster–Shafer theory. The reason is that in this case all subsets of the frame of discernment might have to be considered. Of course, the set of all subsets of a set, say Θ, is exponential in the size of Θ. However, the complexity here is not caused by tolerance-based techniques, but by the Dempster–Shafer theory itself.

10. Conclusions

In this paper, we have proposed a framework that provides agents with the ability to fuse both low and high level *approximate* knowledge in the context of dynamically changing environments as well as heterogeneous and contextually limited perceptual capabilities. We have shown how these limitations can be taken into account when constructing and querying approximate databases for each respective agent.

In order to model the discussed phenomena, we have introduced the idea of partial tolerance spaces. This idea can further be generalized to arbitrary similarity spaces (see [18]) reflecting arbitrary similarity measures.

The techniques proposed in this paper are all tractable and are being implemented and tested in a UAV project [19–21], where physical sensors and real-time constraints must be taken into account. A book [16] describing much of additional relevant work is forthcoming.

References

[1] P. Doherty, W. Łukaszewicz, A. Szałas, Approximate databases and query techniques for agents with heterogenous perceptual capabilities, in: Proceedings of the 7th International Conference on Information Fusion, FUSION'2004, 2004, pp. 175–182.

[2] F. White, A model for data fusion, in: Proceedings of the 1st National Symposium for Sensor Fusion, vol. 2, 1988.

[3] A. Steinberg, C. Bowman, Revisions to the JDL data fusion model, in: D. Hall, J. Llinas (Eds.), Handbook of Multisensor Data Fusion, CRC Press LLC, Boca Raton, FL, 2001.

[4] J. Llinas, C. Bowman, G. Rogova, A. Steinberg, E.Waltz, F. White, Revisions and extensions to the JDL data fusion model II, in: P. Svensson, J. Schubert (Eds.), Proceedings of the 7th International Conference on Information Fusion, 2004. Available from: <URL http://www.fusion2004.foi.se/papers/IF04-1218.pdf>.

[5] Z. Pawlak, Rough Sets. Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht, 1991.

[6] S. Abiteboul, R. Hull, V. Vianu, Foundations of Databases, Addison-Wesley, Reading, MA, 1996.

[7] H.-D. Ebbinghaus, J. Flum, Finite Model Theory, Springer-Verlag, Heidelberg, 1995.

[8] N. Immerman, Descriptive Complexity, Springer-Verlag, New York, Berlin, 1998.

[9] P. Doherty, W. Łukaszewicz, A. Szałas, Tolerance spaces and approximative representational structures, in: A. Günter, R. Kruse, B. Neumann (Eds.), Proceedings 26th German Conference on Artificial Intelligence, LNAI, vol. 2821, Springer-Verlag, Berlin, 2003, pp. 475–489.

[10] P. Doherty, W. Łukaszewicz, A. Szałas, Efficient reasoning using the local closed-world assumption, in: A. Cerri, D. Dochev (Eds.), Proceedings of the 9th International Conference on AIMSA 2000, LNAI, vol. 1904, Springer-Verlag, Berlin, 2000, pp. 49–58.

[11] P. Doherty, J. Kachniarz, A. Szałas, Using contextually closed queries for local closedworld reasoning in rough knowledge databases, in: S. Pal, L. Polkowski, A. Skowron (Eds.), Rough-Neuro Computing: Techniques for Computing with Words, Cognitive Technologies, Springer-Verlag, Berlin, 2003, pp. 219–250.

[12] Y. Gdalyahu, D. Weinshall, Measures for silhouettes resemblance and representative silhouettes of curved objects, in: B.F. Buxton, R. Cipolla (Eds.), Proceedings of the 4th European Conference on Computer Vision ECCV'96, vol. 1065 of Lecture Notes 27 in Computer Science, vol. II, Springer, Berlin, 1996, pp. 365–375.

[13] X. Wang, B. De Baets, E. Kerre, A comparative study of similarity measures, Fuzzy Sets and Systems 73 (2) (1995) 259–268.

[14] R.R. Brooks, S. Iyengar, Multi-Sensor Fusion, Prentice-Hall, Englewood Cliffs, NJ, 1998.

[15] R.R. Murphy, An Introduction to AI Robotics, MIT Press, Cambridge, MA, 2000.

[16] P. Doherty, W. Łukaszewicz, A. Skowron, A. Szałas, Knowledge Engineering: A Rough Sets Approach, Springer Physica-Verlag, Berlin (to appear).

[17] P. Doherty, W. Łukaszewicz, A. Szałas, Approximative query techniques for agents using heterogenous ontologies, in: D. Dubois, C. Welty, M.-A. Williams (Eds.), Proceedings on the 9th International Conference on Principles of Knowledge Representation and Reasoning, KR'2004, AAAI Press, 2004, pp. 459–468.

[18] P. Doherty, A. Szałas, On the correspondence between approximations and similarity, in: S. Tsumoto, R. Slowinski, J. Komorowski, J. Grzymala-Busse (Eds.), Proceedings of the 4th International Conference of RSCTC'2004, LNAI, vol. 3066, Springer-Verlag, Berlin, 2004, pp. 143–152.

[19] P. Doherty, Advanced research with autonomous unmanned aerial vehicles, in: Proceedings on the 9th International Conference on Principles of Knowledge Representation and Reasoning, 2004 (extended abstract for plenary talk).

[20] P. Doherty, P. Haslum, F. Heintz, T. Merz, T. Persson, B. Wingman, A distributed architecture for intelligent unmanned aerial vehicle experimentation, in: Proceedings of the 7th International Symposium on Distributed Autonomous Robotic Systems, 2004.

[21] P. Doherty, G. Granlund, K. Kuchcinski, E. Sandewall, K. Nordberg, E. Skarman, J. Wiklund, The WITAS unmanned aerial vehicle project, in: Proceedings of the 14th European Conference on Artificial Intelligence, 2000, pp. 747–755.