Approximate Databases and Query Techniques for Agents with Heterogeneous Perceptual Capabilities

Patrick Doherty
Department of Computer Science
University of Linköping
Linköping, Sweden
patdo@ida.liu.se

Witold Łukaszewicz **Andrzej Szałas**
Department of Computer Science
University of Economics and Computer Science
Olsztyn, Poland
and University of Linköping, Linköping, Sweden
witlu@ida.liu.se andsz@ida.liu.se

Abstract – *In this paper, we propose a framework that provides software and robotic agents with the ability to ask approximate questions to each other in the context of heterogeneous and contextually limited perceptual capabilities. The framework focuses on situations where agents have varying ability to perceive their environments. These limitations on perceptual capability are formalized using the idea of tolerance spaces. It is assumed that each agent has one or more approximate databases where approximate relations are represented using intuitions from rough set theory. It is shown how sensory and other limitations can be taken into account when constructing approximate databases for each respective agent. Complex relations inherit the approximativeness inherent in the sensors and primitive relations used in their definitions. Agents then query these databases and receive answers through the filters of their perceptual limitations as represented by 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

Research in cognitive robotics is concerned with endowing robots and software agents with higher level cognitive functions that enable them to reason, act and perceive in changing, incompletely known, and unpredictable environments. Research in robotics has traditionally emphasized low-level sensing, sensor processing and control tasks. One of the open challenges in cognitive robotics is to integrate techniques from both disciplines and 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 which are implied by the need for qualitative knowledge structures in high-level reasoning tasks.

Low-level sensor data is quantitative in nature, yet higher-level reasoning tasks require the use of properties and relations among individuals in specific domains of discourse and the associated inference mechanisms which use combinations of base properties and relations in reasoning processes. To add to the difficulty, sensors, by their very nature introduce uncertainty and noise in the data. In order to provide an accurate representation of a robotic environment, some of this uncertainty, or lack of knowledge, should be translated into the higher-level knowledge structures.

In other words, some of the high-level knowledge structures will be approximate in nature, having both quantitative and qualitative characteristics. Such structures are useful in bridging the gap between purely quantitative data generated by sensors and purely qualitative data used in symbolic reasoning tasks.

In addition, the perceptual limitations of a robotic agent induced by its sensor suite should be taken into account not only when the robotics agent reasons about its external and internal environments, but also when one or more robotic agents communicate with each other by asking questions about 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 software and robotic 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 question 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 order to provide the proper level of detail for the specific framework in question, the following set of abstractions will be used in the paper. Each robotic agent will have access to the following functionalities and representations:

- An abstraction called a tolerance space which is used to represent similarity of data points for both basic and complex data domains.[1]
- A set of sensors and a sensor model for each sensor. The sensor models will take into account the contextual indiscernibility of signal data by using tolerance spaces to represent that indiscernibility.

[1] Of course, similarity has been studied in many contexts. For a discussion of a similarity-based measures that can be applied in defining tolerance functions of tolerance spaces, see, e.g., [1]).

- One or more databases capable of holding relational data. These databases may contain representations of crisp relations or approximate relations. The approximate relations will be represented using intuitions from the discipline of rough set theory. 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 robotics agent to ask questions about knowledge in its own databases or in the databases of other agents. These queries will be approximate in nature due to the approximate nature of the knowledge stored in the databases. They will also be contextualized by perceptual limitations represented as tolerance spaces on more complex data domains.

The paper is structured as follows. In Section 2, representations of approximate relations and queries are introduced using intuitions from rough set theory. In Section 3, the important concept of a tolerance space is introduced. These spaces are used to represent indiscernibility, uncertainty and similarity between data. In Section 4, a generic sensor model is presented where uncertainty in sensor data may be translated into parameterized tolerance spaces. This convenient representation allows for the possibility of relating sensor information to relational definitions in a homogeneous manner since tolerance spaces will also be associated with approximate relations in databases. This is considered in Section 5 where approximate databases are introduced. At this point, the basic representational structures for approximate knowledge derived through approximate sensor data will have been introduced. The main idea of agent communication with heterogeneous perceptual capabilities is then introduced in Section 6 with examples. Section 7 concludes with a discussion.

2 Set Approximation

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., [2]). 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 precisely since the knowledge about objects is incomplete. We thus assume that any imprecise concept is replaced by a pair of precise concepts called the lower and the upper approximation of the imprecise concept, where (see also Figure 1):

- the lower approximation consists of all objects which with certainty belong to the concept (Z_+)

- the upper approximation consists of all objects for which it is possible that they belong to the concept ($Z_\oplus$)

- the complement of the upper approximation consists of all objects which with certainty do not belong to the concept (Z_-)

- the difference between the upper and the lower approximation 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 ($Z_\pm$).

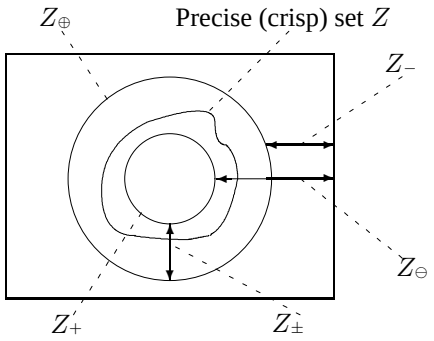

Fig. 1: Approximations of a set.

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_- we denote the complement of $Z_\oplus$. The *boundary region* of Z, defined as $(Z_\oplus - Z_+)$, is denoted by $Z_\pm$. □

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

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

By an *approximate database* we understand a tuple

$$D = \left\langle U, \left\{ r^j \mid r^j = \left\langle r^j_+, r^j_\oplus \right\rangle \text{ and } j \in J \right\} \right\rangle,$$

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 $\langle a_j \mid j \in J\rangle$, where for any $j \in J$, a_j is the number of arguments (*arity*) of r^j. □

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[2] 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 1st-order language, an approximate query can always be generated automatically.

3 Tolerance Spaces

Tolerance spaces have been introduced in [6]. Technically, they allow us to classify a universe of individuals into indiscernibility or tolerance neighborhoods based on a parameterized tolerance relation. This is a generalization on 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 this paper, they will be used for the representation of limitations on an agent's perceptual capabilities, sensor uncertainty, and approximate databases.

Definition 3.1 By a *tolerance function* on a set U we mean any function $\tau : U \times U \longrightarrow [0, 1]$ such that for all $x, y \in U$,

$$\tau(x, x) = 1 \quad \text{and} \quad \tau(x, y) = \tau(y, x). \qquad \square$$

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 $\tau^p \stackrel{\text{def}}{=} \{\langle x, y\rangle \mid \tau(x, y) \geq p\}$. The relation τ^p is also called the *parameterized tolerance relation*. □

In what follows, $\tau^p(x, y)$ is used to denote the characteristic function for the relation τ^p.

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

Definition 3.3 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 value $n^{\tau^p}(u)$.□

The concept of tolerance spaces plays a fundamental role in our approach.

Definition 3.4 A *tolerance space* is defined as the tuple $TS = \langle U, \tau, p\rangle$, consisting of

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

[2] We deal with relational databases, where queries are formulated as first-order or fixpoint formulas (for textbooks on this approach see, e.g., [3, 4, 5]).

4 Sensor Models and Tolerance Spaces

In this section, we provide a simple sensor model (see also [6]), based on a generalization of that in [7], and one method for modeling uncertainty in sensor data which integrates well with tolerance spaces.

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 the current state of the environment is dynamic and changes with time.

We denote a sensor S_i as a function of the environment E and time point t, $S_i(E, t)$. S_i is a function which returns a pair of functions,

$$S_i(E, t) = \{V_i(t), \epsilon_i(t)\}.$$

Depending on the type of sensor being modeled, $V_i(t)$ will be a function that returns the values of the physical attributes associated with the sensor. 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 will be supplied in the form of $\epsilon_i(t)$. The temporal argument is supplied since the accuracy of a sensor may vary with time. 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 unmanned aerial vehicle. Let a_{temp} be the physical attribute associated with temperature in the environment, where the actual temperature is $E(t)(a_{temp})$ and the value returned by the sensor is $V_i(t)(a_{temp})$. The following constraint holds:

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

By using tolerance spaces, accuracy bounds for a physical attribute can be represented equivalently as tolerance relations to degree p 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.

In the following, we will drop the temporal argument for ϵ and assume the accuracy bounds for attributes do not change with time. Let $TS_{S_{ik}} = \langle V_{S_{ik}}, \tau_{s_{ik}}, p_{s_{ik}}\rangle$ be a tolerance space for the kth physical attribute, a_{i_k} associated with the sensor S_i, where,

- $V_{S_{ik}} = \{x \mid lb \leq x \leq ub, x \in D\}$, where D 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, ub. We associate a distance measurement $\delta(x) = |x-y|$ with the value set $V_{S_{ik}}$, which includes all the values that can be read from the sensor S_i.

- Both the tolerance function $\tau_{s_{ik}}$, and the tolerance parameter $p_{s_{ik}}$ are defined as follows,

$$\tau_{s_{ik}}(x,y) = 1 - \frac{\delta(x,y)}{\delta(lb,ub)},$$
$$p_{s_{ik}} = 1 - \frac{\epsilon_i}{\delta(lb,ub)}.$$

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 $V_i(a_{temp})$ is the current value measured by the sensor S_i then we would know that $E(a_{temp}) \in n^{p_{s_{ik}}}(V_i(a_{temp}))$. 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 value itself.

Example 4.1 Let S_R, S_G and S_B be sensors detecting values of R, G, B color attributes.[3] 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:

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

In this case an agent using sensor data from S_R, S_G, S_B will be unable to distinguish between color values where values of τ on R, G, B attributes are not less than 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 5. □

5 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 arguments 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 will be rough 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 will be 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 will ask whether that tuples neighborhood is a member of a lower approximation of the relation or intersects with its upper approximation as in rough set theory. In this case we will 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 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 ternary tuples 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 the new relation. 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 to use rough set machine learning techniques 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 au-

[3] Relations related to color are perhaps not the best to use as examples since there are many sophisticated techniques for dealing with noise and uncertainty associated with color. On the other hand, this domain provides a simple and intuitive vehicle to present our ideas within the page constraints of the article.

tomatically generate lower and upper approximations for each relation. This will be demonstrated in Example 5.4. Under these assumptions, we would have an approximate relational database with tolerance spaces associated with some or all of the approximate relations.

Let's now assume 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.[4] 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 both rough sets and tolerance spaces. In Section 6, it will be 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.

Formal definitions and examples now follow.

Definition 5.1 Let $TS = \langle U, \tau, p\rangle$ be a tolerance space and let $S \subseteq U$. The *lower and upper approximation of S wrt TS*, denoted respectively by S_{TS^+} and $S_{TS^\oplus}$, are defined by

$$S_{TS^+} = \{u \in U : n^{\tau_p}(u) \subseteq S\}$$
$$S_{TS^\oplus} = \{u \in U : n^{\tau_p}(u) \cap S \neq \emptyset\}. \quad \square$$

In the 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 whether the neighborhood around the the tuple is a member of the relation. If so, the answer is yes, if there is an intersection, 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 learning the approximate relations directly, one can assume the tolerance spaces as given and apply them to raw data to generate an approximate database. The following definition and example show how this is done.

Definition 5.2 Let $D = \langle U, \{r^j \mid j \in J\}\rangle$ be a relational database. Then we say that a sequence of tolerance spaces $\overline{TS} = \langle TS_j \mid j \in J\rangle$ 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. $\square$

Definition 5.3 Let $D = \langle U, \{r^j \mid j \in J\}\rangle$ 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

$$D^{\overline{TS}} = \left\langle U, \left\{ \left\langle {r^j}_{TS_j^+}, {r^j}_{TS_j^\oplus} \right\rangle \mid j \in J \right\} \right\rangle.$$

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

$$D^{\overline{TS}} = \left\langle U, \left\{ \left\langle (r^j_+)_{TS_j^+}, (r^j_\oplus)_{TS_j^\oplus} \right\rangle \mid j \in J \right\} \right\rangle. \quad \square$$

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 5.4 Consider a situation where a ground operator (agent Ag_G) is communicating with a UAV[5] (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 that there no visible connection between object 3 and any other objects, 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 capabilities relative to some attributes such as color which will be modeled below.

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

Table 1: Database considered in Example 5.4 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

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	-

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 represents 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.

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 5.5

- A relation symbol r *occurs positively* (respectively *negatively* in a formula if it appears under an even (respectively odd) number of negations.[6]
- For any formula α referring to relations in D,
  - 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 $r_\oplus^j$

[5] Unmanned Aerial Vehicle.

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

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

  - by $\alpha_\oplus$ we understand the formula α in which any positive occurrence of a relation symbol, say r^j, is replaced by $r_\oplus^j$ and any negative occurrence of r^j is replaced by r_+^j. □

Example 5.6 Consider formula $r^1(\bar{x}) \wedge \neg r^2(\bar{y})$. Then:

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

The two formulas on the righthand side would represent an approximation of the original formula. □

We will 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 5.7 Let D be a (crisp or approximate) database. By a *tolerance query* we mean a tuple $\langle Q, TS_Q \rangle$, 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 tolerance query Q in database D wrt context $\overline{TS}$* is given by

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

6 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. 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 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. 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 will 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 will be 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 [6].

We begin with a general definition of a *tolerance agent* specializing that provided in [6].

Definition 6.1 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 will then be 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 agents:

- *query agent* $TA_1 = \langle Ag_1, D_1, \overline{TS}_1\rangle$
- *answer agent* $TA_2 = \langle Ag_2, D_2, \overline{TS}_2\rangle$,

will then conform 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, $\left\langle R'_{T_Q^+}, R''_{T_Q^\oplus}\right\rangle$, 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.

Remark 6.2 Observe that the context of agent Ag_1 is not present in the schema directly. One can, however, observe, that T_Q will usually strongly depend on $\overline{TS}_1$. In particular, if Q is of the form $r^j(\bar{x})$ then, in most cases, T_Q will be the j-th tolerance space in $\overline{TS}_1$. □

The definitions describing this interaction now follow.

Definition 6.3 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 meaning[7] of Q in database D_2 wrt context $\overline{TS}_2$. □

Example 6.4 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 5.4 (i.e., Ag_V does not recognize the difference between colors dr and b).

Consider the following query which the agent will ask itself internally:[8]

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

To compute the answer we first consider (cf. Definition 5.7):

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

[7] As provided by Definition 5.7.

[8] 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.

which, according to Definition 5.5, is

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

Using the approximations of V, S, E and C wrt T_V, from Tables 2 and 3, the above query evaluates to:

$$\big\langle \{\langle 1,2\rangle\}, \{\langle 1,2\rangle, \langle 2,1\rangle\} \big\rangle .$$

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 6.5 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 6.4
- $T_G = \langle U, \tau_G, p_G\rangle$ such that $\tau_G^{p_G}$ identifies equal elements and additionally dr with r.

Suppose Ag_G wants to ask Ag_V for information about colors of objects satisfying the following tolerance query:[9]

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

Using Definition 6.3, agent Ag_V will then evaluate 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

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

Thus Ag_V will return the following answer to Ag_G:

$$\Big\langle \{r\}, \{r,b,dr\} \Big\rangle$$

Ag_G will then compute the final answer by interpreting the above one relative to its tolerance space T_G using Definition 6.3 and the database approximation shown in Tables 4 and 5. The final answer is then $\Big\langle \emptyset, \{r,b,dr\} \Big\rangle$. □

7 Conclusions

The techniques described in this paper provide a basis for developing an efficient and tractable formal framework for fusing sensor data with approximate knowledge in databases and querying that knowledge while taking into account the inherent perceptual or contextual limitations associated with robotic or softbotic systems. Many techniques that were assumed but not described in detail in this paper, such as the generation of approximate relations through rough set machine learning techniques, their representation in standard relational databases, translating approximate queries into SQL queries, fusing tolerance spaces, or generating complex approximate relations in terms of primitive relations have been developed elsewhere and described in submitted or published references. These techniques are also being implemented and tested in a UAV project where physical sensors and real-time constraints must be taken into account. A book [8] describing some of this additional work is forthcoming.

[9] 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.

Table 4: Approximation (lower approximations) of the relational database given in Table 1 wrt perception capabilities of agent Ag_G as defined in Example 6.5.

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 6.5.

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

Acknowledgment

The paper has been supported in part by the WITAS project grant under the Wallenberg Foundation, Sweden.

References

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

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

[3] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley Pub. Co., 1996.

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

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

[6] P. Doherty, W. Łukaszewicz, and A Szałas. Tolerance spaces and approximative representational structures. In A. Günter, R. Kruse, and B. Neumann, editors, *Proceedings 26th German Conference on Artificial Intelligence*, volume 2821 of *LNAI*, pages 475–489. Springer-Verlag, 2003.

[7] R. R. Brooks and S.S. Iyengar. *Multi-Sensor Fusion*. Prentice-Hall, 1998.

[8] P. Doherty, W. Łukaszewicz, and A. Szałas. *Knowledge Engineering: A Rough Sets Approach*. Springer Physica-Verlag, 2004. To appear in 2004.