The Collection of Physical Knowledge and its Application in Intelligent Systems

Benjamin Johnston

Smarter Living Studio, School of Software
Faculty of Engineering and Information Technology
University of Technology, Sydney

Abstract. Intelligence is a multidimensional problem of which physical reasoning and physical knowledge are important dimensions. However, there are few resources of physical knowledge that can be used in data-driven approaches to Artificial Intelligence. *Comirit Objects* is a project intended to encourage the general public to contribute to research in Artificial Intelligence by building simple 3D models of everyday objects via an interactive web-site. This paper describes the simplified representation and web-interface used by Comirit Objects and a preliminary investigation into the potential applications of the collected models.

Keywords: Crowd-sourcing, Physical Reasoning, Commonsense Reasoning, 3D Modeling

1 Introduction

The recent success of IBM's Watson computer on 'Jeopardy: The IBM Challenge' is an event with important implications and that offers insight into Artificial Intelligence and Artificial General Intelligence. For some time now, web search engines have demonstrated an almost uncanny ability to find relevant answers to even poorly posed queries. Watson and Google are demonstrating that in many problems of Artificial Intelligence, large quantities of data are increasingly trumping deep algorithms and deep understanding. Douglas Lenat, the founder of the Cyc project [10], is famously reported to have claimed that "Intelligence is 10 million rules" [9]. In light of the successes of data-driven Artificial Intelligence, it may be more appropriate to revise such claims for a larger, shallower data-source; thus, "Intelligence a trillion records".

A data-driven approach to Artificial Intelligence depends, obviously, on the availability of large sources of data. Wikipedia and the web are well known resources for huge quantities of unstructured textual data. There are also growing numbers of high-quality, large-scale, semi-structured and structured resources becoming freely available on the web. ConceptNet [5], WordNet [4] and FreeBase [1] are examples that are closely aligned to Artificial Intelligence research. Other resources include government public information datasets (such as data.gov) and data feeds from social networks (such as Twitter and Foursquare). While these datasets encompass a broad range of social, geographical, economic and political matters, their content is principally textual (and partly geographic and photographic).

Our world is a physical place. Knowledge of the physical world is unquestionably crucial for robots but it also has important implications for information-bots and text-based Artificial Intelligence. When interacting with others (even through written text), much of our everyday discourse concerns physical objects and the relationships between those objects. Furthermore, even when we are not discussing concrete, physical matters, regular use is made of physical analogy. The lack of large-scale datasets that describe the physical form of everyday physical objects is therefore a significant limitation in the quest to create Artificial Intelligence with general capabilities across conceptual, linguistic and physical intelligence.

The objective of this paper is twofold. First, I describe a tool for creating large-scale repositories of physical knowledge. The tool is a simple 3D modeling tool that can be used by the general public without any training. It is designed so that physical knowledge may be 'crowd-sourced' from user contributions, similar to the way that textual and relational knowledge is gathered by tools such as FACTory [3], ConceptNet [5], FreeBase [1], ISI Learner [2] and Games with a Purpose [14].

In the second half of this paper, I offer preliminary insights into how the collected data may be integrated into a general-purpose reasoning system. The models can be used for deep-reasoning by inspecting the structure of objects or instantiating the objects inside simulations. The data may also be used in more shallow applications, such as systems that retrieve content based on association and similarity, by computing similarity measures such as the earth-moving distance between pairs of objects.

2 Background

At previous AGI conferences [8, 6], I have described an architecture called *Comirit* for hybrid reasoning. The architecture is designed in view of the importance of physical and spatial reasoning in general intelligence. It adapts the method of analytic tableaux to combine formal, logical deduction with generic simulations that perform physical reasoning. The method also supports the ability for a robot to explore an environment and autonomously learn about objects within the world [6, 7]. The architecture is designed to dramatically reduce the effort involved in formalizing and engineering knowledge for an intelligent system and its use of simulations allows for vastly more efficient physical reasoning (polynomial in size, connections and time) than is possible using theorem provers (undecidable).

The simulations in Comirit are implemented as annotated multi-graphs. The graphs serve as a kind of virtual 'putty' that can be manipulated into any shape, texture and appearance to allow a machine to 'visualize' any object from rigid gears to smooth liquids. A simulation can be created in arbitrary shapes and can incorporate collisions, physical, chemical and thermal dynamics. Figure 1

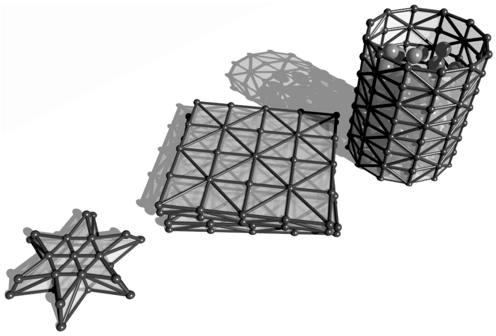


Fig. 1. Modeling richly detailed objects in Comirit: a cookie, sandwich and cup of coffee.

Fig. 2. A chair shaped like a hand. (Photo by Scott Partee)

illustrates how the representation can be used to model a lunch comprising a sandwich, coffee and biscuit.

While the simulations of Comirit are easy to work with and are vastly more efficient than formal methods, they remain too slow for reasoning over massive datasets of millions of objects. Furthermore, ignoring the computational costs, the richness of the representation is such that it is too complex to imagine populating large-scale datasets in the short-term future.

These limitations can be addressed by the combination of two approaches. First, I identify a dramatically simpler representation scheme that is still able to capture useful information about the physical world. Second, I design a web-based interface to allow the public to easily contribute physical knowledge about the world. These two approaches are explained in the following section.

This work is largely inspired by projects such as Open Mind Common Sense (OMCS) [11], ISI Learner [2] and Games with a Purpose [14]. These projects use a simple web-based interface (often in the form of games) and invite the general public to add knowledge. For example, in its earliest days, OMCS invited people to simply type any sentence of commonsense that came to mind. Today, OMCS uses a more structured collection scheme, asking visitors to verify whether or not sentences are true. For example, “Is it true that: a dog can take a walk?” (Yes? No? Sort of?). The knowledge collected by OMCS is incorporated into ConceptNet that now contains over 700,000 symbolic assertions for 150,000 concepts [5].

3 Representing Physical Knowledge

ConceptNet contains over 150,000 concepts [5] generated by volunteer contributions. The rich 3D models that I have previously used for Comirit can take several hours to produce. The creation of 150,000 objects would therefore require on the order of 40 person-years of development (plus training in 3D modeling tools). Even with the human resources to create 150,000 objects, the associated geometric calculations can be computationally challenging so that manipulating and querying this many objects in real-time can be difficult without ‘unusually clever’ indexing schemes (thus raising doubts over the ability to draw on existing physical models used in architecture, animation and CAD).

It is therefore essential that the large-scale modeling be performed with a simplified representation. Ideally, such a representation would meet the following criteria:

1. It should be computationally efficient and memory efficient to store, retrieve, manipulate, query, analyze and visualize.
2. It should be conceptually clear so that users and developers do not require extensive training to contribute models or to apply the resource in Artificial Intelligence projects.
3. It should be possible to use the unmodified representation in the same simulation and physics engines that have achieved widespread use and popularity among game developers (*e.g.*, ODE and PhysX).
4. It must be capable of describing useful properties of a wide range of objects.

I argue that voxel-based representations satisfy these criteria. A voxel is the 3-dimensional equivalent of a pixel: it refers to a small cubic region in 3D space. Instead of representing an object by a complex polygonal structure, its shape can be approximated by a set of voxels that fill a similar space. For example, four low-resolution voxel-based models of everyday objects appear in Figure 4 (of course, higher resolution models are also possible) — these figures will be used again later in the paper.

More formally, an object is represented as follows:

1. The shape of the object is described using a set of voxels. That is, by a set, V, of integer triples, v, where $v \in \mathbb{J} \times \mathbb{J} \times \mathbb{J}$.
2. The appearance of the object is described using a function that maps voxels to colors (RGB values). That is, by a function, $color$, such that for each voxel, $v \in V$, it holds that $color(v) \in (0..255) \times (0..255) \times (0..255)$.
3. Meta-data about the shape and structure of the object is described by a function mapping voxels to sets of annotation labels. That is, by a function $label$, such that $label(v) \in \mathbb{P}l$, where l is the set of possible labels and $\mathbb{P}$ is the powerset operator (on a computer system, $label(v)$ would return a list of character strings).

This simple representation scheme is ideal for creating large knowledge-bases of everyday objects:

1. Voxels are the 3D equivalent of bit-mapped or raster images. They can be stored as a simple list of coordinates, as elements in a 3D spatial array or in a spatial index. Each voxel has identical size and shape, thus ensuring that computation is fast and efficient. For example, volume can be computed by counting the number of voxels and then multiplying by the size of a single voxel (computing the volume of a polygon mesh is, in contrast, a non-trivial problem).

2. Voxels offer a simple and clear mental-model. Modeling with voxels is analogous to creating structures out of LEGO: most people can easily decompose an object into a set of cubic blocks. The representation is easy to visualize so that both users and developers can quickly understand the capabilities and limits of the representation.
3. A single voxel is a cube — a platonic solid — and may be readily mapped into the primitives of physics and game engines. For example, the ODE physics library [12] includes cube primitives (and connecting joints) so voxels have an immediate translation to ODE.
4. Even though voxels are simple, they, like LEGO blocks, can be used to define the shape of a wide number of objects. Voxels will not be able to describe the workings of intricate machinery but they can capture the approximate shape of such machinery. If necessary, dynamics of machinery can be captured as a collection of models that show the sequence of actions.

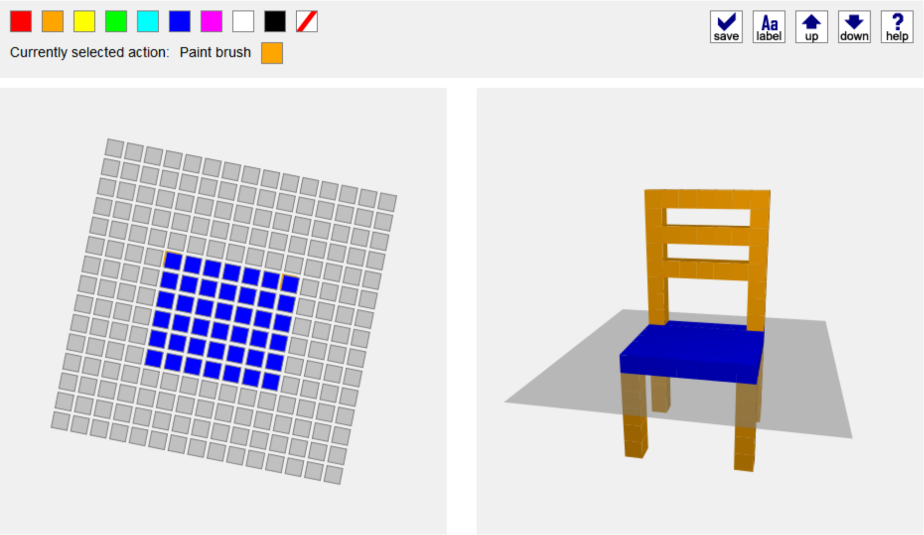
4 Collecting Physical Knowledge

The simplicity of a voxel-based representation is ideal for crowd-sourcing. Voxels require little explanation and they can be created by a simple point-and-click interface. This means that there is a low barrier to entry, allowing the public to contribute models with no training. While volunteer contributions will inevitably be less detailed and of lower quality than the models of trained specialist, having a very large number of moderate-quality models is likely to be of greater benefit in practical reasoning problems than having a small number of high-quality models. Furthermore, accepting volunteer contributions reduces the development time and costs, while dramatically increasing the potential scope, coverage and creativity of the knowledge-base.

The potential benefits of such crowd-sourcing were the inspiration for creating a system called *Comirit Objects*. The system is an interactive website designed to encourage users to participate in the creation of Artificial Intelligence through online 3D modeling.

Conceptually, Comirit Objects is a voxel-editing tool. It allows users to create 3D objects using a paintbrush metaphor generalized from the infamous Microsoft Paint (and other raster drawing software). The project relies exclusively on the goodwill and enthusiasm of unknown volunteers; as such, usability and joyfulness are of paramount importance.

In early prototypes of Comirit Objects I eliminated all user-interface with the intention of creating a simple 'discoverable' interface. The arrow-keys, along with the page-up and page-down keys were used to move a cursor around a 3D space, while the insert and delete keys could be used for sculpting in that space. The simplified interface enabled users to quickly learn the application without explanations. However, informal usability tests revealed that while the interface was easy to learn, few people had the patience to build more than one model. Furthermore, the application was designed as a browser-plugin and many users questioned the security of the plugin.



Currently selected action: Paint brush

Fig. 3. The user interface for Comirit Objects

The current version of Comirit Objects is designed to encourage the creation of multiple objects and more complex structures. Unfortunately, some simplicity is sacrificed in this objective and a short, graphical tutorial became necessary to illuminate the tool's operation. A screenshot of the tool appears in Figure 3. The left-hand side is the editing surface and the right-hand side is the visualization. The tool uses a 'slice' metaphor: a horizontal 2D slice cuts through the visualized object (the dark plane). When colors are painted in the editing surface of the left hand side, matching voxels are rendered in 3D on the 'slice' on the right hand side. Users can therefore perceive 3D modeling as somewhat analogous to brick-laying. A user mentally decomposes an object into its horizontal slices, which are then drawn (or laid) slice-by-slice. The object is created from a small palette of colors (though, the underlying representation allows for a larger spectrum of color) and textual labels can be attached to individual parts of the objects using the labeling tool.

When an object is saved, it is associated with a set of metadata. Each model is given a name or noun to describe the class of objects it belongs to (*e.g.*, is it a model of a *tree*? a *car*? a *cat*?). Associated with this noun are a set of adjectives to describe the particular instance that was created (*e.g.*, is it a model of a *small* tree? a *convertible, expensive* car? a *black* cat?). In addition, a user may offer an approximation of the object's real-world size and mass. All users are required to agree to release their model to the public domain upon submitting their contribution. The saved file can then be shared with friends, thus offering a form of encouragement to entice users to create multiple files and develop communities around model building.

Comirit Objects is available online at `http://www.comirit.com/objects/`. The interface is implemented using HTML5 Canvas 2D. An implementation based upon WebGL would allow higher performance but is not well supported in the current generation of web-browsers (at the time of writing). Comirit Objects is tested and compatible with current versions of Firefox, Chrome and Safari in addition to Internet Explorer 9. The frontend and 3D engine are implemented using JavaScript. 3D models are transmitted to and from the server using asynchronous HTTP requests (AJAX) using the JavaScript Object Notation format (JSON). The backend is implemented in Python and the web.py framework [13] using a simple file-system database.

To date, submissions have been of a high quality so quality control, aside from 'spam' detection has not been required. However, should quality prove to be a problem, it may be possible to incorporate elements of game-play to entice users to rate, validate and improve upon each other's contributions. Game-based crowd-sourcing has been used to great success with Games with a Purpose [14] and similar forms of game play can be applied directly to 3D models. For example, an adaptation of the ESP Game [15] could invite pairs of users to position labels on 3D objects, earning points for labels placed on the same part of the object.

5 Applying Physical Knowledge

Comirit Objects is in its early days. Only a small number of models have been collected so far. Nevertheless, it is possible to consider its potential applications and their implementation.

Obviously, large collections of 3D models have clear applications to robotics:

1. An observed object may be identified and named by searching a 3D knowledge-base for known objects with a similar size, structure and appearance.
2. A 3D model matched to an observed object can be used to assist with handling, manipulation and object affordances. For example, a label 'handle' on a model can be used to guide a gripper to the correct position for handling the real world-object and a label such as 'top' can be used to indicate the correct orientation to hold an object.
3. A 3D model can reveal hidden or unobservable properties of an object. For example, a model can be used to retrieve the mass of an observed object, identify whether an object is hollow or solid or even provide indications of the value of the object.
4. A knowledge-base of 3D objects can help describe an unknown object. For example, the chair in Figure 2 might be referred to as a "hand-shaped chair" by noting the object's similarity to both hands and chairs.

The crucial challenge in these applications is identifying similarity measures to compare observations and 3D models. Objects would be identified by finding those models in the knowledge-base that are most similar (or "similar enough") according to context-appropriate similarity measures. Once similarity has been computed, affordances and unobservable properties may be retrieved directly from the stored models through knowledge-base retrieval/lookup.

The benefits of a knowledge-base of everyday objects are not exclusive to robotics. Indeed, some of the more interesting applications stem from natural language understanding where spatial knowledge and knowledge of objects can be used for disambiguation and reasoning. In such domains, 3D models of objects can serve multiple roles:

- 3D models can be used to retrieve factual knowledge, relationships and appearances of everyday objects in the world. For example, a 3D model of a car and a horse could be used to understand why a car is driven from the inside but a horse is *not* ridden from its inside.
- 3D models can be used to reason about the dynamic properties and affordances of objects. For example, the sentences "I put the robot on the table and it fell over" and "I put the bag on the hat-stand and it fell over", have identical structure, yet *it* refers to different words in the sentence (4th word and 7th word respectively). The meaning of *it* can be resolved by instantiating robots, tables, bags and hat-stands in simulations and observing the stability of the simulations (*i.e.*, what could fall over?).
- 3D models can be used to determine the similarity between objects. For example, recognizing that "the popular team sport played with an olive-shaped ball" might refer to either American or Australian football, requires an ability to compare the overall shape of an olive to the shapes of balls used in a variety of sports.

Here, there are three reasoning capabilities involved: (1) recalling stored models and the properties of those models (*e.g.* via knowledge-base retrieval/lookup); (2) reasoning about the models through simulation; and (3) computing the similarity of objects. Note that capabilities 3 and 1 overlap with capabilities that are useful for robotics.

Recalling stored models (Capability 1) is a relatively straightforward task. Given the terms "car" and "horse", corresponding models can be retrieved from a large knowledge-based of 3D objects purely by the textual keys (recall that each object has meta-data including the named type of the object). Identifying the location the seat or saddle is a simple matter of searching the representation for appropriate annotations. Some additional spatial reasoning may be required to interpret the labels. For example, computing whether additional voxels occur above a label to determine if the label is on the top of the object). However, such additional computation is straightforward.

Reasoning about models through simulation (Capability 2) is an interesting and complex problem. The techniques described in earlier work on Comirit [7] can be applied directly in this domain: a voxel is transformed into a Comirit simulation and used directly for reasoning. Alternately, the 3D models can be transformed into the primitives of a physics-engine (*i.e.*, cubes, in the case of ODE). A physics engine can then be used to experimentally determine properties such as dynamic stability and center of gravity by instantiating simulations and monitoring their behaviors.

Similarity measures (Capability 3) represent the most interesting aspect of reasoning with 3D models. Many similarity measures are possible and no single

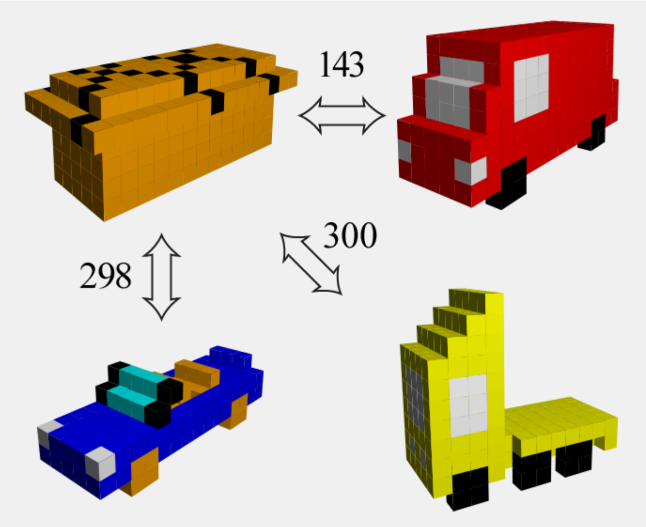


Fig. 4. Computed similarity measure between a loaf of bread and (clockwise) a minivan, a truck and a convertible. The minivan is most similar to the loaf of bread (smallest value).

similarity measure will be universal. For example, it may be useful to compare the collected 3D objects on the basis of shape, color, size, overlap, subsetting and labels or any combination of these depending on the context. In general, however, similarity may be considered as a measure of the "effort" involved in transforming one object in the knowledge-base to another object. The differing forms of similarity may then be defined by placing a 'cost' on each kind of transformation. For example, a purely shape-based similarity measure would place no cost on changing colors, labels, size and whole-of-object translation but would place a premium on transformations that involve adding, removing or moving voxels. Such a similarity measure might be implemented as the minimum translated Earth Mover's Distance.

An illustrative example of how similarity measures can be used in useful computation comes from a recent exchange in our research laboratory. A Chinese-speaking student, not knowing the correct English term, referred to a minivan as a "bread-loaf car" (this being a literal translation of the Chinese word). An intelligent system might resolve this novel term by comparing bread-loaves to cars under context-appropriate similarity measures (and also considering a car that is used to deliver bread-loaves). A simplified similarity measure that computes the minimal number of mismatching voxels, ignoring color and labels, of translations of each object is used in Figure 4 to compare a loaf of bread with three kinds of vehicles. In this illustration it is clear that the minivan is most similar to a loaf of bread. The intelligent system could then use this evidence in its inference process and resolve the ambiguity associated with the previuosly unknown term "bread-loaf car".

6 Conclusion

Physical knowledge is an important dimension of intelligence that is under-represented in existing knowledge-bases and databases. Comirit Objects is a user-friendly web-based interface that will help remedy this deficiency by crowd-sourcing models of physical objects. Comirit Objects is based on a simple voxel representation that allows untrained volunteers to contribute annotated, colored 3D models on a massive scale.

There are many potential future directions for this work. One such possibility is to extend the voxel representation to allow articulated objects, actuators and agency by, for example, allowing jointed and force-generating voxels. Of course, there also remains the challenge of applying the knowledge-base to practical reasoning problems of Artificial Intelligence, Natural Language Understanding, Commonsense Reasoning and Artificial General Intelligence.

References

1. Bollacker, K. D., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a Collaboratively Created Graph Database for Structuring Human Knowledge. In SIGMOD Conference (2008)
2. Chklovski, T.: LEARNER: A System for Acquiring Commonsense Knowledge by Analogy. Proceedings of the 2nd International Conference on Knowledge Capture (2003)
3. Cycorp Inc.: FACTory. `http://game.cyc.com/` (2010)
4. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press (1999)
5. Havasi, C., Speer, R., Alonso, J.: ConceptNet 3: a Flexible, Multilingual Semantic Network for Common Sense Knowledge. Proceedings of Recent Advances in Natural Language Processing (2007)
6. Johnston, B.: The Toy Box Problem (and a Preliminary Solution). International Conference on Artificial General Intelligence (2010)
7. Johnston, B.: Practical Artificial Commonsense. University of Technology, Sydney PhD thesis (2009)
8. Johnston, B., Williams, M-A.: Comirit: Commonsense Reasoning by Integrating Simulation and Logic. Proceedings of the First International Conference on Artificial General Intelligence (2008)
9. Kaku, M.: Visions: How Science Will Revolutionize the Twenty-First Century. Oxford University Press (1999)
10. Lenat, D.B., Guha, R.V. Building Large Knowledge Based Systems. Addison Wesley (1990)
11. Lieberman, H., Smith, D., Teeters, A.: Common Consensus: A Web-Based Game or Collecting Commonsense Goals. Proceedings of the IUI 2007 Workshop Common Sense for Intelligent Interfaces (2007)
12. Smith, R.: Open Dynamics Engine. `http://www.ode.org/` (2000–2006)
13. Swartz, A.: Web.py. `http://webpy.org/` (2011).
14. von Ahn, L.: Games with a Purpose. Computer, vol. 29, no. 6 (2006)
15. von Ahn, L., Dabbish, L.: Labeling Images with a Computer Game. In CHI-04: Proceedings of the SIGCHI conference on Human factors in computing systems (2004)