ANALOGICAL ESTIMATION OF QUANTITATIVE MAGNITUDES

Jonathan Gagné
jgagne@uwaterloo.ca, Systems Design Engineering Department,
University of Waterloo, 200 University Avenue West, Waterloo, ON, N2L 3G1, Canada

Jim Davies
jim@jimdavies.org, Institute of Cognitive Science,
Carleton University, 1125 Colonel By Drive, Ottawa, ON, K1S 5B6, Canada

ABSTRACT

Visuo is an implemented Python program that models visual reasoning. It takes as input a description of a scene in words (e.g., "small dog on a sunny street") and produces estimates of the quantitative magnitudes of the qualitative input (e.g., the size of the dog and the brightness of the street). We claim that reasoners transfer quantitative knowledge to new concepts from distributions of familiar concepts in memory. We also claim that visuospatial magnitudes should be stored as distributions over fuzzy sets. We show that Visuo successfully adapts knowledge to new concepts.

INTRODUCTION

In this paper we apply analogical reasoning to an aspect of visual imagination. Human beings can easily imagine what a "big sundae" looks like. The creation of this visualization requires an enormous amount of information—much more than the information in the two-word phrase that triggers it. How do reasoners generate this information from an input containing only two symbols? We will show how a reasoner could estimate the quantitative meaning of qualitative adjectives such as "big" and qualitative prepositions such as "beside." For example, if the reasoner is asked visualize a "big orca," exactly how big should that orca be?

We implemented our theory in an operational Python computer program called Visuo. The program takes as input a set of labels, such as "long rug," or "big flag over a small lawn." In a process we call "visuospatial instantiation," the program outputs symbolic scene descriptions that include quantitative magnitudes based on input adjectives and prepositions (such as "long" or "over"). It does this through analogical reasoning, using individual experiences as well as generalized prototypes as sources.

We will describe our theory of visuospatial instantiation, Visuo (the program written to test the theory), and an evaluation.

Theory Overview. The meanings of adjectives and prepositions in natural language are relative to the objects being described. For example, a foot "over" a brake pedal is much closer than a cloud "over" a lake. We suggest that reasoners retrieve and modify appropriate memories when possible. For example, if asked to visualize a "large raven," if there are descriptions in memory of a large ravens, the large raven prototype will be retrieved and used for visualization. But when a match is not found, the reasoner infers the meaning of the adjectives and prepositions by analogy, transferring from better-known concepts in memory.

Our theory includes two phases: the training phase, when experienced visuospatial stimuli are incorporated into memory, and the visualization phase, when, given some input to visualize, information in memory is used to create a description of a new, imagined visual scene.

Reasoners experience visual stimuli in the world, and often these stimuli are labeled. For

Gagne, J. & Davies, J. (2009). Analogical estimation of quantitative magnitudes. *Proceedings of Analogy 09,* Sofia, Bulgaria.

example, a reasoner might see an object associated with a label such as “crow.” A different stimulus might be labeled as a “large crow,” or “small city.” Suppose a reasoner has experienced many crows, many of which were labeled with different qualitative size descriptors, such as “tiny,” “small,” “large,” and “huge.” Let's also suppose the reasoner has experienced many instances of ravens, which are associated with the “raven” label but *not* with qualitative size labels. When asked to imagine a large raven, since the reasoner has not experienced a raven that was labeled as “large,” it must use a meaning of “large” from some other concept in memory. According to our theory a reasoner in this situation transfers the notion of “large” from a semantically-related concept, such as “crow,” rather than a less-related concept, such as “city.” With a notion of “large” from a related concept, the reasoner can make a reasonable guess as to how large a “large raven” would be. This informs the final visualized image description.

In the training phase, Visuo takes as input a file describing visuospatial experiences. Its outputs are changes to its memory. In the visualization phase, Visuo takes as input a phrase describing what is to be visualized. The output is a propositional scene description including quantitative magnitudes.

The theory is both a model of human cognition and a contribution to artificial intelligence.

Training Phase. We focus on three aspects of encoding that are relevant to our particular task of visual imagination: a label, attributes, and quantitative values.

In the training phase each preposition and noun in the input is associated with some set of *attributes* (e.g. size, length). Each attribute (associated with each noun and preposition) is associated with a single *value*. The value represents the real-world magnitude of that attribute.

Reasoners add this information to episodic memory, and then create or modify appropriate parts of semantic memory.

Our implementation of the theory (Visuo) takes a file as input for the training phase and a text string for the visualization phase.

The training phase input file contains entries representing quantitative aspects of visual scenes. These values are measured in mental units rather than more objective metrics such as centimeters or kilograms, so a crow that is 40 centimeters long might get a “size” value of “10” in these mental units. This is based on the notion that a person need not know any particular culturally invented unit of measure to represent visuospatial magnitudes. The specific values chosen for the mental units are irrelevant. What is important is that they are proportionate to invented units up to some scale factor and that they are consistently used.

The parser reads the input file, and transforms each example into an exemplar, which contains all the characteristics of the example. Additionally, the parser identifiers each word as a noun, adjective, or a preposition.

For example, “large bat” and all exemplars created from this phrase acquire the attribute size, with an attribute value of 12 (we will describe below how these values are not stored as exact numbers, but as a distribution containing the attribute value’s degrees of membership in each fuzzy number set).

There are two methods the parser uses to create exemplars out of the input phrase (see Figure 1). If the input phrase has a preposition in it, the first method is used, which parses the phrase into two exemplars. The first exemplar is that of the entire phrase, and the second exemplar is one containing only the preposition. For example, “[[large bat] above [tree]]” would

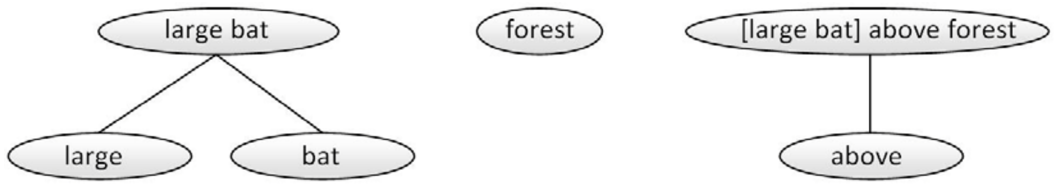

Figure 1. The result of parsing “large bat”, forest, and “[large bat] above forest”. Each node is turned into an exemplar.

be parsed into an exemplar “large bat above tree,” and also into an exemplar “above.” The first would be an exemplar of how “above” relates to a bat and a tree, and the second would be a general exemplar of “above” as a generic term.

Noun phrases use a second method. The phrase is parsed with a depth-first recursive method, and each segmentation creates a new exemplar. The very first exemplar is created from the entire input phrase. The remaining is recursively broken down into sub-phrases, where each sub-phrase is either a word, or a grouping of words as grouped by square brackets. If a sub-phrase contains only one word, then the parser does not attempt to break it down further. If the sub-phrase contains multiple words, or bracketed words, then that chunk is further broken down.

For example, the input phrase “[large bat]” would produce three exemplars. Visuo would first create an exemplar for “large bat.” Since the “large bat” contains more than one word, it would further be broken down into an exemplar for “large,” then one for “bat.” Since the previous phrase cannot be recursively broken down further, no more exemplars are created for this input phrase. The process above is the same no matter how many attributes are associated with a particular name.

Creation of Exemplars in Episodic Memory. Exemplars represent memories of objects occurring at a specific place and time (Tulving, 1984). For example, when the reasoner takes as input a crow of size “10,” that information is stored in an exemplar.

Distributions of Fuzzy Set Membership. In our theory, agents do not exactly represent perceptual magnitudes. Rather, the memory representations account for perceptual uncertainty and vagueness. Spatial receptors in the retina and the visual system of the brain have receptive regions that have varying sensitivity to different attributes, such as the different orientations of an edge (Hubel & Wiesel, 1965). In fact, detectors have been found to pick up even higher order visual concepts such as buildings and faces (Kreiman, Koch, & Fried, 2000). Also, Behavioral data show that people represent things with graded membership to categories in general (Hampton, 2007), so we conjecture that higher-level perceptual detectors in the brain also represent variable category membership as a function of the firing rates of neural populations. In our theory, fuzzy set memberships represent these differential firing rates. In fuzzy set theory, the membership of an instance in a given set is described with a fuzzy membership value ranging from 0 (clearly not in the set) to 1 (clearly a member of the set).

For an example relevant to our task, an input of “10” could be representing a magnitude in the real world of, say, 5’8”, with varying degrees of certainty and vagueness. In this example, the input number 10 becomes a *fuzzy number* (Dubois & Prade, 1987). As opposed to ‘crisp’ numbers, fuzzy numbers are numbers that have a fuzzy range of values. Each in itself is a fuzzy set. An input number is a member of all fuzzy number sets to *some* degree, represented by a number between 0 and 1. An input of “10,” for example, would have a 1.0 membership in the fuzzy number 10 and a 0.6667 membership in the fuzzy number 5.

Each distribution has a slot for all of the fifteen points on a logarithmic mental unit scale (0, 2, 5, 10, 20, 35, 65, 100, 160, 250, 400, 600, 900, 1350, 1800)[1]. It is represented logarithmically because there is evidence showing that people naturally (without educational intervention) represent distances logarithmically (Dehaene, Izard, Spelke, & Pica, 2008). The distribution itself is a list of numbers representing the membership of the crisp input value in all of the fuzzy number sets. These fuzzy numbers have overlapping ranges, just as spatial detectors do.

For example, suppose the reasoner views a crow of size 10 (in mental units.) The distribution to represent this would contain the following information:

(0.0, 0.0, .67, 1.00, .33, 0.0, ...) [crow1 size]

[1] Only approximates the logarithmic scale; Also, zero is not in the logarithmic scale.

Each number in the distribution is a membership of the crisp input number (in this case "10") for each point on the mental unit line (0, 2, 5, 10, 20, 35,...). This process of turning a crisp number into fuzzy memberships is called "fuzzification." Each of these distributions is associated with an attribute, and also with either a preposition or a noun. A particular noun phrase can have multiple exemplars, each with its own distribution—one for each attribute.

We cannot know exactly what numbers are on this scale nor how real world magnitudes translate to these mental units. Thus the specific numbers we use in the program are arbitrary, but we believe that a different set of logarithmically-organized numbers would not change the program's behavior in a way relevant to our hypotheses. The details of how Visuo executes fuzzification can be found in Gagné and Davies (under review).

Exemplars. In this theory, each exemplar contains some number of distributions, as described above. The example of [[large crow] over [thick tree]] is parsed. Reasoners will create exemplars for each of the following nodes: large crow, large, crow, large crow over thick tree, over, thick tree, thick, and tree. Each of these exemplars will have one distribution for every attribute that is associated with it in the input.

Prototypes in Semantic Memory. In our theory, semantic memory consists of prototypes. Inspired by Rosch (1973), they are memories of general concepts of things, abstracted from specific instances. They represent the family resemblance of a category. For quantitative attributes this translates to mean values. We call this part of memory "semantic" because it is representing the meaning of words. The first time Visuo experiences a crow, it creates a prototype for *crow*. For each subsequent experience of a crow, instead of creating an additional prototype of *crow*, it modifies the existing one with the new exemplar

Suppose the reasoner experiences ([large crow] size=10). The reasoner creates a distribution for size and associates it with the prototype of "large crow," containing all of the information from its exemplar, plus a count *n* of the number of examples experienced so far with this label (for the first instance of "large crow," the number of instances seen would be, of course, one). As the crisp input number is stored as a distribution over fuzzy number sets, it differs from more traditional prototypes, which store exact means.

Upon experiencing another large crow, each fuzzy membership number v_{avg} in the distribution is averaged based on

$$v_{\text{avg}} = \frac{(n \times v_{\text{old}})}{n+1} + \frac{v_{\text{new}}}{n+1}$$

where v_{old} is the previous value, v_{new} is the new value. Following the calculation of v_{avg}, the number n is increased by 1. The reasoner incorporates each new experience of the same category into the prototype. For each point in the mental unit scale, the prototype represents the mean value of the memberships all exemplars for the corresponding fuzzy number. In this way, the prototype represents an average of all experiences.

Like exemplars, prototypes also keep separate records for each attribute, containing a distribution and a count of experienced examples. This also happens for attributes that might be thought to be irrelevant to an adjective, such as "large." For example, an experienced "large crow" might have a "size" and a "brightness" (indicating how dark it is.) Though one would not think that the largeness of the crow describes the brightness, the prototype stores it. This could allow Visuo to pick up on correlations in the environment.

Note that there is also a prototype created for "large." This is a representation of largeness that is independent of what objects have actually been seen (e.g., crows, tuna sandwiches). It is the reasoner's generic representation of "large."

Similarly prototypes are created or modified for each distribution in the exemplar.

VISUALIZATION PHASE

Above we described the training phase, how a reasoner collects and represents observations conducted before the task of visualization starts. The specific task we are describing in this paper is to estimate a quantitative magnitude from a qualitative (verbal) stimulus. The kinds of input our theory endeavors to understand (in both the training and visualization phases) includes inputs such as: [raven], [bright [long [large raven]]], and [[bright [long [large raven]]] above [tall tree]].

These examples are instances of a generalized recursive grammar:

```
NP -> [N]
NP -> [Adj NP]
S  -> [NP (Prep NP)]
```

where *S* is an input string (simplified sentence), *N* is a noun, *NP* is a noun phrase, *Adj* is an adjective, and *Prep* is a preposition.

For example, if the reasoner is asked to imagine a “large crow” (the “visualization stimulus”), how can the reasoner use memories of crows to effectively guess how large this imagined raven should actually be? If asked to imagine a “large raven,” how can the reasoner use memories of *crows* to make an estimate of the raven’s actual size? This is the task of the visualization phase.

We conjecture that using *context* affects the outputs, making output more psychologically realistic. By context we mean both the context of the other things in memory (e.g., the crows and ravens you have seen before), as well as the context in the visualization stimulus (i.e., the other words in the input).

At a high level, the reasoner takes the visualization stimulus and tries to find a matching prototype in semantic memory. If it is found, then the information in that prototype is de-fuzzified and output. This is the trivial case.

The more interesting case is when there is *not* an exact match found in semantic memory. When this happens, the reasoner breaks the visualization stimulus into smaller pieces, recursively, until either matches are found, or the input cannot be further broken down.

For example, suppose the visualization stimulus is “[large raven] above tree.” The reasoner will search its semantic memory for prototypes describing the entire stimulus [[large raven] above [tree]]. Suppose the reasoner has not experienced a scene labeled this way. The reasoner parses the input and then searches for prototypes describing the first node [large raven]. If that fails, the reasoner searches for prototypes for “large” and for “raven.”

Specifically, the reasoner searches for the distribution of “large” that most closely matches “raven.” For example, suppose the only prototypes for “large” in semantic memory are for crows, cities, and the generic concept of “large.” The reasoner will use a distribution of “large” for the most semantically-related concept. In this case, it would be “large crow.” The information in the “large crow” distribution is transferred and adapted to the new “large raven” distribution. With the creation of this new distribution, the reasoner now has an exact match for “large crow,” which it has been searching for.

This is conceptually combined with “above” and “tree,” finally creating a new exemplar and prototype for [[large raven] above [tree]]. With this final prototype created, the reasoner can de-fuzzify the prototype (as in the trivial case), and the reasoner outputs a number for how large the large raven is, and how far above the tree the large raven is.

We will now describe the processes of retrieval, conceptual combination, transfer, and de-fuzzification.

Retrieval: Searching for Prototypes in Semantic Memory. The system depth-first searches the parse tree, looking for matches for each node in semantic memory. If the node represents a phrase containing a compound term (e.g., “large raven,” as opposed to a single term, such as “raven”) then Visuo will only match to prototypes in memory for which there is an exact match on the that phrase. If the compound term is not found, then the children nodes are searched. This process continues until the terms are found. At this point, Visuo combines the prototypes through conceptual combination, creating a new prototype for “large raven,” which might get further combined with other parent nodes until there is a prototype created for the visualization stimulus.

Conceptual Combination. Conceptual combination is the process of combining ideas together to create a new idea. There are two methods used in this paper, one for the merger of an adjective with a noun phrase, and one for combining two noun phrases with a preposition.

An adjective is combined with a noun phrase by disambiguating the sense of the adjective, understanding how the adjective modifies similar concepts through the creation of a concept modifier, and finally, merging the adjective and noun such that the relevant properties are transferred. Noun phrases are merged with a preposition by disambiguating the sense of the preposition.

Word Sense Disambiguation. If Visuo is searching for a sense of “large” that is relevant to a “raven”, the first step is to retrieve from memory all prototypes that contain the word large and pair them with the generic version of the prototype (i.e., the prototype that is the noun portion). For instance, if Visuo experiences three different kinds of situations when “large” is used (“large tree”, “large crow”, “large city”) then those prototypes will be returned. Although Visuo retrieves the relevant constants in linear time, we conjecture that since the brain performs many of its tasks in a highly paralleled nature, this task is being done in parallel and hence in constant time.

“Raven” is compared to each of the generic versions of the prototypes (tree, crow, city) using the Wu-Palmer similarity measure (Wu & Palmer, 1994) as implemented in NLTK version (Bird & Loper, 2004) of WordNet (Fellbaum, 1998). The prototypes identified to be most similar are selected as the specific source prototype (“large crow”) and the general source prototype (“crow”). The general target prototype is the noun prototype (raven) that is to be combined with the concept modifier to create the specific target prototype (large raven).

The process of finding an appropriate sense of a preposition is slightly different. In the first step, all of the prototypes of the preposition in memory are returned as a list. The similarity measure used is the product of the Wu-Palmer similarity between the left nouns and the Wu-Palmer similarity of the right nouns. The adapted similarity measure allows Visuo to recognize that a “bird over forest” is more similar to “bat over tree” than it is to “clouds over forest”, even though the former shares a word in common (forest) and the latter does not. In this example, product similarity is calculated by the result of the bat’s similarity to bird, multiplied by the result of forest’s similarity to tree.

Concept Modifier Creation. We can imagine that an adjective modifies a noun (or noun phrase) in the mind of a reasoner. For example, we have an idea of what a dog looks like, and when specified that the dog is “small,” it changes the concept we have in mind. This allows a reasoner to use adjectives to describe new concepts as long as the meaning of the adjective can be abstracted and transferred from a similar situation. We hold that reasoners have concept modifiers, which are created from adjective senses and used to modify nouns, creating new adjective-noun concepts. It is the instantiation of the functional capabilities of an adjective in a particular context. For example, a concept modifier created from the adjective “large” as it is used for “crow” can be applied to “raven” to create a concept for “large raven”. The details of how Visuo uses concept modifiers can be found in Gagné and Davies (under review).

Concept modifiers are implemented in Visuo as data structures that are comprised of one or more *attribute modifiers*. Attribute modifiers are data structures that represent how each attribute in a noun prototype is modified by an adjective prototype such that the noun prototype becomes an adjective-noun prototype (a concept modifier is made of some number of attribute modifiers). For example, if the prototype of a raven has two attributes, *size* and *colour_brightness*, and Visuo is performing conceptual combination on "large" and "raven", then up to two attribute modifiers will be created. More specifically, there is one attribute modifier for each attribute that is present in the general source, specific source, and general target prototypes.

Attribute modifiers contain two distributions, a modifier density and a multiplier. The *modifier density distribution* is a normalized copy of the attribute's distribution of the general source prototype (e.g., "raven"). The *multiplier* is created by a piece-wise multiplication of the general source prototype by the specific source prototype. More formally, the multiplier **m** is a vector where each element is defined as

$$\mathbf{m} = \left(\frac{G_0}{S_0}, \frac{G_1}{S_1}, \frac{G_2}{S_2}, \dots, \frac{G_{n-1}}{S_{n-1}}\right)$$

where *n* is the size of the distributions, G_i is the *i*th element in the general source concept distribution, and S_i is the *i*th element in the specific source concept.

Transfer. In our running example, since crows are smaller than ravens, simply applying the concept modifier by multiplying the attribute modifiers' multiplier distribution by the "raven" attribute distributions would result in an inappropriate matching of the numbers in the distribution. In fact, without adjusting the distributions, a "large raven" could have the distribution of what a "small raven" should have, or it could have a distribution containing nothing (i.e., all zeros). So we conjecture that reasoners associate the values of the two distributions with the *percent of the distributions that those values cover*.

Visuo creates a density distribution for the modifier values and one for the target prototype, which is a representation of how dense the data is at different parts of the distribution. For example, a density distribution might tell us that most of the data is in the low ranges, with very little in the high ranges. This density distribution is stored with the respective attribute modifier as the modifier density distribution. At this point the multiplier is ready to be combined with the target prototype "raven."

Each value in the target distribution is multiplied by some percentage of the numbers in the multiplier. This percentage is determined by matching elements of the target density distribution to sections of the multiplier density distribution. For example, the first number in the target distribution might be multiplied by the first two, or even the first 2.6 numbers in the multiplier. If this percentage matching is not done, then large portions of distributions end up being unjustifiably multiplied by zero. To be more specific, the new attributes are created by the Density Distribution Product (Gagné & Davies, under review).

The final concept to create is the root concept, which consists of three concepts: "large raven," "above," and "tree." All that is missing for the merger is the concept of "above" as it applies to this context. This is found once again by the WordNet's implementation of Wu-Palmer similarity measure, as described in the section on retrieval. The attribute values of "[large bat] above forest" are used to create an exemplar and prototype of "above" for the concept [[large raven] above tree]. The distribution is copied directly, rather than being created by changing a target with a multiplier, since there is no target to modify.

The creation of the complete novel concept is now complete.

De-fuzzification. Once Visuo has a prototype that matches the input, the final step is de-fuzzification, which is the process of transforming a fuzzy qualitative distribution (such as an attribute's distribution) into a quantitative crisp number. The crisp number *N* is computed by taking a weighted average of the distribution as defined by:

$$N = \frac{\sum_i \left(u_i \times \mathrm{val}(S_i)\right)}{\sum_i u_i}$$

where, u_i is the membership value in the distribution and $\text{val}(S_i)$ is the value of the fuzzy number (e.g., val(35F) = 35). The denominator $\Sigma(u_i)$ is used to normalize the result.

Multiple Attributes. As noted above, multiple attributes can be transferred for a single specific concept. For example, if one is asked to imagine a "large dog," there is more to our imagined dog than simply a size. The dog has a colour, a way it holds itself, and several other features. Up to this point, this paper has primarily only illustrated the generation of the estimate for a single attribute, in this case, "size." Note, however, that Visuo will use the process described above for every attribute in the final prototype. Depending on the information in the memory, the imagined "large crow" might come with quantitative estimates of wingspan, size, brightness, elevation, orientation, weight, etc., just as a person would.

EVALUATION

To evaluate our theory we tested the Visuo implementation. For input data we used those collected from the online game Peekaboom (von Ahn, Liu, & Blum, 2006), which contains 57,797 images from the web with clouds of coordinates attached to labels. For example, an image with a cat in it will have points in the image associated with the label "cat."

We analyzed the spatial information of the images from the Peekaboom database that contained at one of the following labels: cat, crow, dog, person, raven, skyscraper, and tower. We used the first 100 instances of each concept except for raven and crows, since the database only contained 12 and 57 of these labels, respectively. We created input for Visuo from this data by measuring the width-to-height ratio (which we will call simply "width ratio") of each instance of the labels, where width ratio is relative to the height. Width ratio r is calculated by

$$r = 100 \times \left(\frac{\max(x) - \min(x) + 15}{\max(y) - \min(y) + 15}\right)$$

where *max(x)* is the x-coordinate that is most to the right of the image, *min(x)* is the x-coordinate that is most to the left of the image, *max(y)* is the y-coordinate that is most to the bottom of the image, and *min(y)* is the y-coordinate that is closest to the top of the image. The ratio is used, since the spatial information in the database is only known up to a scale factor. By dividing the width by the height, the effects of the unknown scale factor are removed, and hence, true spatial information can be known.

In Peekaboom, when one player clicks a part of the image to reveal it to his or her partner, it reveals that pixel plus all pixels in a 20 pixel radius. Therefore the farthest revealed point will be 20 pixels away from the click, although at times, this reveals pixels that are not part of the object. To account for this, we used an estimated value of 15, which we found to be a good compromise. For demonstration purposes, we multiplied each pixel count by 100 to translate it into mental units.

Of the instances used, the thinnest 30% are labeled as thin (e.g. thin tower), the thickest 30% are labeled as thick crows, and the 40% in between are labeled as medium. The only instances that were not labeled were the ones used in the visualization phrase. For example, if visuo is asked to visualize a [thin raven], then all the concepts besides the raven would be labeled. Furthermore, Visuo should select "thin crow" as a source analog because of its relative semantic closeness. If the program works as our theory predicts, then it should guess that a thin raven's width (width to height ratio) is more similar to crows than to the other concepts, and even more importantly, the estimates should be close (< 20%) and preferably very close (< 10%).

To evaluate the results, the estimated values were compared to the actual values and the error ε was determined by

$$\varepsilon = \left|2 \times \frac{(\hat{r} - r)}{(\hat{r} + r)}\right|$$

where $\hat{r}$ is the estimated ratio and r actual ratio.

Table 1. Comparison between Visuo's estimated width ratios and the actual width ratios.

Target Specific Prototype	Source Specific Prototype	Estimated Ratio	Actual Ratio	% Error
thin cat	thin dog	77.74	80.58	3.59
medium cat	medium dog	112.54	113.41	0.77
thick cat	thick dog	168.10	164.10	2.41
				Avg. 2.26
thin crow	thin raven	78.49	78.52	0.03
medium crow	medium raven	161.23	136.55	16.57
thick crow	thick raven	252.37	284.37	11.93
				Avg. 9.51
thin dog	thin cat	72.69	64.27	12.29
medium dog	medium cat	103.42	102.33	1.05
thick dog	thick cat	153.59	163.46	6.22
				Avg. 6.52
thin person	thin dog	54.67	54.27	0.74
medium person	medium dog	83.65	83.17	0.58
thick person	thick dog	129.42	130.47	0.80
				Avg. 0.71
thin raven	thin crow	80.75	78.19	3.22
medium raven	medium crow	135.03	147.30	8.69
thick raven	thick crow	226.14	211.27	6.80
				Avg. 6.24
thin skyscraper	thin tower	44.60	41.75	6.60
medium skyscraper	medium tower	72.83	73.71	1.20
thick skyscraper	thick tower	135.70	137.70	1.46
				Avg. 3.09
thin tower	thin building	32.50	30.85	5.19
medium tower	medium building	56.64	53.84	5.10
thick tower	thick building	122.69	127.66	3.97
				Avg. 4.75
Overall Average				**4.73 %**

As shown in Table 1, the estimates are extremely close to the actual value. The worst estimates were of "medium crow" (16.57%) and "thick crow" (11.93%), which used "medium raven" and "thick raven" as estimates. These estimates are quite reasonable considering the Peekaboom database only contained 6 instances of "medium ravens" and 3 instances of "thin ravens" to train from. Increasing the number of instances dramatically improves results, which can be seen with the width ratio prediction for a person. For all three person predictions, the error was 0.8% or less, which indicates extremely accurate predictions. Overall, the average estimate error is 4.73%, indicating that the cognitive model Visuo can accurately estimate the quantities of unknown spatial attributes based on the quantities of semantically similar concepts.

CONCLUSION

There have been numerous systems built that implement analogical visuospatial reasoning (e.g., Ferguson & Forbus, 2000), but the research on generating visual representations has been limited, coming primarily out of computer graphics (e.g., "image synthesis" Johnson et al., 2006). Though such systems often use images from a database, the research is not couched in terms of analogy nor case-based reasoning.

Our main claims are as follows. First, reasoners store quantitative perceptions as membership distributions across a logarithmic set of fuzzy numbers. Second, these perceptions can be labeled with linguistic phrases. Third, when visualizing, reasoners will retrieve an appropriate prototype from memory, and determine the crisp output based on de-fuzzification of that prototype. Fourth, when retrieval is impossible, reasoners transfer meaning of these descriptors and relations from semantically related concepts in the prototype case base.

REFERENCES

Bird, S. & Loper, E. (2004) NLTK: The Natural Language Toolkit. *Proceedings of the ACL demonstration session*, Barcelona: Association for Computational Linguistics, pp 214-217.

Collins, A. & Quillian M. R. (1972). Experiments on Semantic Memory and Language Comprehension. In *Cognition in Learning and Memory*. Wiley, New York.

Dehaene, S., Izard, V., Spelke, E., & Pica, P. (2008). Log or Linear? Distinct Intuitions of the Number Scale in Western and Amazonian Indigenous Cultures. *Science* **320**(5880) 1217-1220.

Dubois, D. & Prade, H. (1987). Fuzzy numbers: an overview. In: *Analysis of Fuzzy Information Vol. I: Mathematics and Logic* (J.C. Bezdek, ed.), Boca Raton, Florida: CRC Press.

Fellbaum (Ed.) (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

Ferguson, R. W., & Forbus, K. D. (2000). GeoRep: A flexible tool for spatial representation of line drawings, *Proceedings of the 18th National Conference on Artificial Intelligence*. Austin, Texas: AAAI Press.

Gagné, J. & Davies, J. (under review) Visuo: A model of visuospatial instantiation of-quantitative magnitudes. Submitted to *Knowledge Engineering Review. Special Issue on Visual Reasoning.*

Johnson. M, Brostow. G.J., Shotton J., Arandjelovic. O., Kwatra, V., & Cipolla, R. (2006). Semantic Photo Synthesis, *Computer Graphics Forum* (Proc. Eurographics), 25(3), 407-413.

Hampton, J. A. 2007 Typicality, Graded Membership, and Vagueness. *Cognitive Science*, **31** (3), pp 355-384.

Hubel, D.A. & Wiesel, T.N. (1965). Receptive fields and functional architecture in two non-striate visual areas (18 and 19) of the cat. *Journal of Neurophysiology*, **28**, pp 229-289.

Kreiman, G., Koch, C., & Fried, I. (2000). Category-specific visual responses of single neurons in the human medial temporal lobe. *Nature Neuroscience, 3,* 946-953

Rosch, E.H. (1973). Natural categories. *Cognitive Psychology*, **4**, pp 328-350.

Tulving, E. (1984). Precis of Elements of Episodic Memory. *Behavioral and Brain Sciences*, **7**, pp 223–268.

Von Ahn, L., Liu, R. & Blum, M. (2006). Peekaboom: A Game for Locating Objects In Images. Computer-Human Interaction Conference, (2006). Montréal, Québec, Canada, 55--64.

Wu, Z. & Palmer, M. (1994). Verb semantics and lexical selection. In: *32nd Annual Meeting of the Association for Computational Linguistics*, pp 133-138.