Rapid Unsupervised Encoding of Object Files for Visual Reasoning

Rachel Flood Heaton (rmflood2@illinois.edu)
Department of Psychology, University of Illinois, 603 E. Daniel St.
Champaign, IL 61820

John E. Hummel (jehummel@illinois.edu)
Department of Psychology, University of Illinois, 603 E. Daniel St.
Champaign, IL 61820

Abstract

Visual thinking plays a central role in human cognition, yet we know little about the algorithmic operations that make it possible. Starting with outputs of a JIM-like model of shape perception, we present a model that generates object file-like representations that can be stored in memory for future recognition, and can be used by a LISA-like inference engine to reason about those objects. The model encodes structural representations of objects on the fly, stores them in long term memory, and simultaneously compares them to previously stored representations in order to identify candidate source analogs for inference. Preliminary simulation results suggest that the representations afford the flexibility necessary for visual thinking. The model provides a starting point for simulating not only object recognition, but also reasoning about the form and function of objects.

Keywords: visual reasoning; shape perception; object files; structural description; type-token problem

Introduction

Visual thinking plays a central role in human cognition. From deciding whether a quantity of soup will fit into a storage container, to interpreting graphical representations of data, or reading a circuit schematic, people routinely engage visual reasoning in the service of understanding the world and solving problems. Visual thinking figures prominently in our most creative and uniquely human acts, including mathematics, engineering, art and design. But in spite of its centrality, comparatively little is known about the algorithmic basis for visual and visually-assisted reasoning (but see Hummel & Holyoak, 2001, Johnson-Laird, 1983, Lovett and Forbus, 2017, for progress in this direction). Instead, most computational work in high-level vision has been and continues to be addressed to the problem of object recognition, the tacit assumption often being that object recognition is the final stage of ventral visual processing, as though once an object has been visually recognized, there is nothing left to be done. Most models in this tradition, including modern deep nets for object recognition, represent objects as holistic templates of various kinds, which is a representational format that does not lend itself to any kind of explicit visual reasoning (Hummel, 2000; see Hummel, 2013, for a review).

The problem of visual thinking places strong constraints on the kinds of representations—for example of object shape or scene layout—the visual system must deliver to the rest of the cognitive architecture. It places equally important constraints on the kind of cognitive architecture that operates on those representations (Hummel, 2000; Lovett & Forbus, 2017). In particular, that architecture must be prepared to reason and generalize extremely flexibly—specifically, with the flexibility of an explicitly relational (i.e., symbolic) system (Hummel & Holyoak, 1997, 2001, 2003a; Lovett & Forbus, 2017). And for that purpose, the visual system must be equipped to represent the visual world in terms of arrangements of objects and object parts in terms of their spatial relations (as opposed to, e.g., their literal locations in the retinal image; Hummel, 2000).

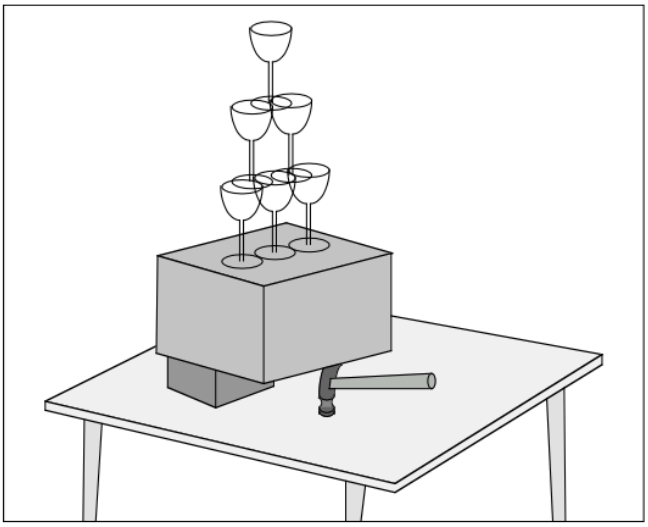

Figure 1: An example of visual reasoning in a novel context (Green & Hummel, 2004). Even if one has never seen this image before, it is obvious that moving the hammer is ill-advised.

Consider, for example, the arrangement depicted in Figure 1 (from Green & Hummel, 2004), and imagine oneself in need of the hammer. Upon a glance at the figure, it is clear that one should not simply pick up the hammer, as doing do would cause the wine glasses to fall and break. We can easily understand this property of Figure 1 in spite of the fact that, for most people, the arrangement in the figure is completely unfamiliar. To put the power of this inference into perspective, note that an associative response to Figure 1

(e.g., of the kind that would be learned by a deep net) might specify that there was something fragile in the scene, and it might even specify that a hammer as an object capable of breaking things, but would it would be incapable of even representing (much less inferring) a complex relational thought such as "moving the hammer is ill-advised because it would result in the wine glasses falling." Although the natural associative relation between hammers and breaking is to think of hammers as objects that break things, in the case of Figure 1, the hammer is *preventing* the glasses from being broken.

Making the appropriate inference about the arrangement in Figure 1 requires us to perceive the spatial relations between the hammer, the boxes and the wine glasses, and to infer from those relations what kinds of actions will and will not result in the glasses falling (Green & Hummel, 2004). Crucially, this inference depends much more on the relations between the objects than on the features or identities of the objects themselves: If we were to replace the wine glasses with a baby, the same relations would be in place, and the same inference would follow; the same is true if we replace the hammer with any other object of an appropriate size to support the box.

Similarly, even recognizing and reasoning about a novel instance of a known object class (for example, a new kind of coffeemaker), requires this kind of representational flexibility: the carafe of a coffeemaker may not always be perfectly cylindrical, especially if its designer was feeling creative, but it will always reside below the filter basket. The coffeemaker may even contain extra parts (e.g. thrown in for flourish) or have parts removed for a minimalist aesthetic, but barring extreme artistic license, it will still be recognizable as a coffeemaker.

In other words, visual inference, and even object recognition, depend on our ability to represent relations independently of the object/parts serving as arguments of the relations, and to simultaneously bind the objects/parts to their relational roles (Hummel & Biederman, 1992; Hummel & Holyoak, 1997, 2003a).

In summary, what is needed is a visual system capable of delivering relational (i.e., symbolic) representations of objects or scenes in terms of their constituent parts and the relations among them, and a cognitive architecture that is capable of using those representations in order to make flexible relational inferences.

Perceiving Relations with JIM and Reasoning About Them with LISA

Models of high-level vision that generate explicitly relational representations are comparatively rare. The examples with which we are familiar are Winston (1975), Lovett and Forbus (2017), and Hummel and Biederman's (1992; Hummel, 2001; Hummel & Stankiewicz, 1996, 1998) JIM. We will focus on JIM, a neural network that was originally developed as a model of object recognition, and in that context has accounted for, and successfully predicted, a very large number of findings in the literature on shape perception and object recognition (for a review, see Thoma & Davidoff, 2007). As such, JIM provides a psychologically and neurally plausible theory of the shape representations that can be derived from line drawings of objects. As elaborated shortly, the model is also useful as a basis for visual reasoning because it generates visual representations that are both explicitly relational and in a format that is directly usable by the LISA model of relational reasoning (Hummel & Holyoak, 1997, 2003a).

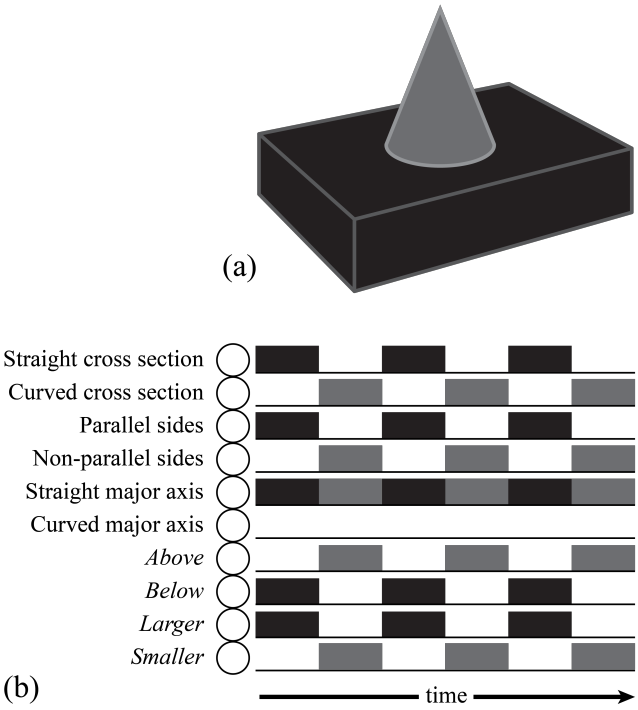

Figure 2: (a) A cone on top of a brick. (b) The JIM representation of a cone on top of a brick. Circles are units representing shape attributes. Bars indicate the activity of corresponding units over time, with black bars corresponding to the brick and gray to the cone.

JIM (Hummel & Biederman, 1992) represents objects as configurations of geons (basic volumetric shapes; Biederman, 1987) in specific spatial relations to one another. For example, the simple object in Figure 2a would be represented as a cone on-top-of, smaller-than and orthogonal-to a brick. The cone and brick are represented in JIM, not as atomic primitives, but as patterns of activation distributed over neuron-like units representing their shape attributes (Figure 2a). For example, the cone would be represented by units specifying that it has a curved cross section, a straight major axis, non-parallel sides, and a slightly elongated aspect ratio; the brick would be represented as having a straight cross section, a straight major axis, parallel sides, and a slightly elongated aspect ratio. The units representing the cone are bound to units representing its relational roles (here, *smaller* and *above*), and the units for the brick are bound to its roles (*larger* and *below*) by synchrony of firing: Units

representing the cone and its roles fire in synchrony with one another, and out of synchrony with the units representing the brick and its roles (Figure 2b). (These synchrony relations are established in the model's V1- and V2-like first layers, by lateral interactions between local units representing the geons' edges and the vertices where they coterminate; see Hummel & Biederman, 1992.)

The resulting representations (Figure 2b) are then matched to stored representations in JIM's long-term memory (LTM) for the purposes of object recognition. This representational format also happens to be identical to the format LISA (Hummel & Holyoak, 1997, 2003a) uses to represent role-argument bindings for the purposes of relational reasoning. In LISA, relational roles and their arguments are represented as patterns of activation over units representing their semantic content, and bound into complete propositions by synchrony of firing: Within a proposition, such as *on-top-of* (cone, brick) or *loves* (John, Mary), units for a relational role (e.g., *above*, *below*, *lover*, or *beloved*) fire in synchrony with the units representing the arguments to which they are bound (with *above* firing with *cone*, or *lover* firing with *John*) and out of synchrony with the units coding the proposition's other role bindings (*below+brick* or *beloved+Mary*).

Armed with these representations, LISA accounts for roughly 100 major empirical phenomena in relational reasoning, including its development (e.g., Doumas et al., 2008) and its decline with brain damage, normal aging, and frontotemporal dementia (for reviews, see Hummel & Holyoak, 2003b; Knowlton et al., 2012). As such, we take JIM and LISA as empirically well-grounded starting points for developing a model of visual thinking.

Although the kinds of representations JIM generates provide a natural basis for reasoning by LISA, the problem remains of adapting JIM-like representations for a LISA-like inference engine. That problem is the focus of the current modeling effort.

Object Files as a Basis for Visual Reasoning

Figure 2b illustrates the kind of distributed representation LISA uses to represent the semantic content of propositions in working memory (WM). To encode these representations into LTM, LISA uses a hierarchy of progressively more localist representations (Figure 3). At the bottom of the hierarchy, semantic units represent relational roles and their arguments in a distributed fashion (as in Figure 2b). *Argument* and *role* units (Figure 3) code arguments and relational roles in a localist fashion and share bidirectional excitatory connections with the corresponding semantic units. *Sub-proposition* (SP) units locally code role-filler bindings, such as *above+cone* and *below+brick*, and *proposition* units bind multiple role-filler bindings into complete propositions, such as *on-top-of* (cone, brick). Collections of related propositions are linked together with *group* (for our current purposes, *object file*) units. The resulting hierarchy of units serves both to represent propositions in LTM and as the basis for analogical mapping and the other functions LISA performs.

This hierarchy serves as a natural way to represent structural descriptions of objects and scenes (a very similar hierarchy encodes objects into LTM in JIM; Hummel & Biederman, 1992). For example, in order to represent an object, propositions would represent the spatial relations among the object's parts, and collections of such propositions would constitute a description of the complete object. Moreover, these descriptions can be nested hierarchically (with propositions taking other propositions as arguments; Hummel & Holyoak, 1997), making it possible to represent entire scenes as hierarchical collections of objects in various relations to each other.

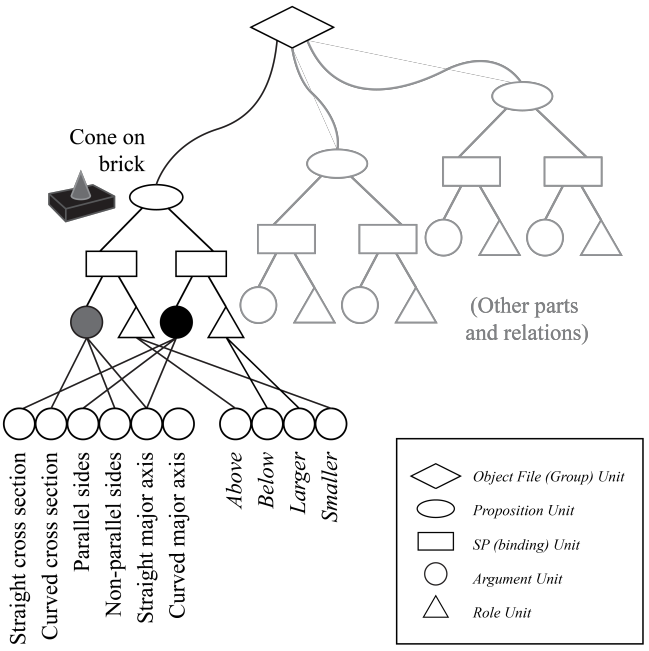

Figure 3. The LISAese representation of an object file. Left: representation of the proposition expressing the relations between the cone and the brick (roughly, *on-top-of-and-smaller-than* (cone, brick)). In light gray: other potential propositions in the object file.

Borrowing from Kahneman et al. (1992), we refer to collections of propositions encoding the properties of objects and/or scenes as *object files*. Importantly, the propositions composing an object file are assumed to encode (in the limit) everything visible about the object, including its shape, color, trajectory, and so forth. We also assume they are hierarchical, describing the properties of (and relations among) both whole objects and of individual object parts. In other words, we assume that the goal of early- and middle-vision, as well as visual attention, is to deliver a hierarchical description of the visual world. Although JIM provides an algorithmic basis for computing some of these properties and relations from object images, we assume that the visual input to the object files is much richer than any computational model is currently capable of providing. In the current effort, we therefore assume this visual input as a given to the model. Specifically,

we assume visual preprocessing that delivers descriptions of objects in terms of their properties (shape, color, etc.), spatial relations to one another, and the spatial relations among their parts. We assume that these descriptions are temporally bound into packages corresponding to bindings of relational roles to their arguments (e.g., Figure 2b; Hummel & Biederman, 1992), where the arguments can either be whole objects or object parts.

The Model

Given such a representation as a basic input, constructing an explicit object file from that input means encoding the propositions—i.e., collections of synchronized patterns of activation—into active memory (Cowan, 2001) so that they can be compared to the contents of LTM and reasoned about. The current model borrows and adapts elements of LISA's *self-supervised learning* algorithm (Hummel & Holyoak, 2003a) to accomplish this task. Like LISA and JIM, the current model uses synchrony of firing at multiple temporal scales in order to bind roles to their arguments, role-argument bindings into complete propositions, and collections of propositions into whole objects or scenes.

At the fastest temporal scale (i.e., the *phase*, which we assume to last about 25 ms; Hummel & Holyoak, 1997), units coding relational roles fire in synchrony with the units coding for the features of their arguments. At the next temporal scale (the *phase set*, corresponding to about 100 - 200 ms), mutually desynchronized role-filler bindings are grouped into complete propositions. And at the slowest temporal scale (corresponding to about 200 – 1000 ms), multiple propositions (phase sets) are grouped into complete units—either whole objects, or small groups of objects in specific relations (Green & Hummel, 2004). Each of these temporal scales corresponds to a specific kind of unit in the hierarchy in Figure 3, with the fastest corresponding to argument, role and SP units, the second slowest corresponding to proposition units, and the slowest to object file (group) units; units at each scale of the hierarchy integrate their inputs over corresponding temporal intervals (Hummel & Holyoak, 1997).

One at a time, patterns of activation corresponding to individual phases, i.e., parts or objects in specific relational roles, are presented to the model. These patterns correspond to packages being delivered by early to middle visual processing (e.g., in visual area LOC). In response to each such package, the model's task is twofold: One task is to encode new packages (phases), as they arrive into active memory, and integrate them into the representation of the emerging object file (Figure 3). This operation is performed by a simple kind of mapping-guided Hebbian learning (i.e., Hummel & Holyoak's, 2003a, self-supervised learning). At the same time, the model performs the parallel task of matching these incoming patterns to stored patterns in LTM (stored object files). That is, the model attempts to recognize each stimulus as an instance of a familiar object category at the same time as it encodes it into active memory as a new object file to be reasoned about.

By the time several phase sets have been processed, the object file will contain a collection of propositions describing (for example) the object's parts in terms of their spatial relations. If the object is familiar, the model will also have activated one or more existing object files in LTM, effectively recognizing/categorizing the object. The preceding describes the model in the language of visual cognition. In the language of analogical reasoning, the model will have encoded a new *target analog* (the object file) to be reasoned about, and it will have retrieved one or more *source analogs* (i.e., existing object files) to use in the service of reasoning about the target. Once this process is complete, the machinery of analogical reasoning (as embodied in LISA) can take over, mapping the target onto the source in order to identify corresponding elements and relations, using the source to drive inferences about the target, and inducing a more abstract schema capturing what the source and target have in common (Hummel & Holyoak, 2003a).

Token Formation

This very coarse description of the model's operation necessarily glosses over numerous implementation details, but all of these are standard to LISA's operation (see Hummel & Holyoak, 2003a, Appendix A). However, one aspect of the algorithm warrants discussion in greater detail. In LISA, argument, role, SP, proposition, and group units represent *tokens* of objects, roles, and so forth, in the context of the larger structure in which they reside. For example, the *cone* unit in Figure 3 represents a token of "cone" in the context of the specific object file depicted in the Figure; the abstract *type* "cone" is represented by the shape units (in LISA, "semantic units") to which the token is connected (Figure 2b). This type/token distinction becomes apparent in the case of scenes containing more than one instance of a given object or geon: If an image contains, say, two cones, then the resulting object file must contain separate argument units for each, even if those units are connected to otherwise identical shape units: Constructing an object file from an image requires the model to distinguish clearly between types ("a cone") and tokens of those types ("this cone").

Keeping this type/token distinction straight is complicated by the fact that a given token is likely to fire more than once in the output of visual processing: If the features of a cone fire at time *t*, and a cone also fired at time *t*-5, then how can we know whether the cone that is firing now is the same one (the same *token*) that fired 5 iterations ago? (In this respect, the object files created by the current model differ from those postulated by Kahneman et al., 1992, in that their object files were assumed to be unitary tokens for single objects. By contrast, the object files created here are hierarchical tokens that can, themselves, contain tokens for smaller parts.)

The current model solves this problem by exploiting the role of mappings in self-supervised learning (Hummel & Holyoak, 2003a). In brief, the current model, like LISA, knows when a new token is required by knowing the mappings between the tokens composing the source of an inference (here, an object file in LTM) and those composing

the target of that inference (the emerging object file): If an unmapped token fires in the source, then a new token is required in the target. The current model exploits a similar constraint by mapping each token in the emerging object file to the location of the corresponding part or object in the image: In essence, it knows whether the cone firing at time *t* is the same token as the one from time *t*-5 by knowing whether they occupy the same location. (This heuristic is admittedly too simple and will fail with, for instance, moving stimuli. In general, we assume that tokens are distinguished, not by locations in the image, but by spatiotemporal trajectories in 3-space.)

The model is still in an early stage of development—and is, itself, only a component of a much larger emerging model—but preliminary simulations provide an encouraging proof of concept.

Simulations

We ran four sets of simulations as basic tests the model's ability to rapidly encode object files from oscillatory visual inputs of the kind illustrated in Figure 2b. In each simulation, objects were presented and encoded in the model's LTM; subsequently, additional objects were presented to be encoded and categorized as one of the known objects. Objects were constructed by combining 14 parts, P1… P14, into arrangements by placing them in various two-place relations, with roles R1…R15. Each part was coded as a 10-dimensional feature vector, and each role of a relation was also coded as a 10-dimensional vector. In addition, 6 units served as location tags, L1…L6, which as discussed above, permit the model to solve the type-token problem. The full feature space was thus 26-dimensional (10 for parts, 10 for roles, and 6 for location tags). The binding of a given part, P_i, to a given relational role, R_j, was implemented as the concatenation of vector P_i with vector R_j and location vector L_k (synchrony of firing is equivalent to vector addition). We manipulated the relationships between the stored and stimulus objects by varying the parts of the objects, *P*, the relations, *R*, and the locations *L* in which they were instantiated. For clarity in what follows, we will refer to a given part in a given location as $P_{i,k}$. The assignment of features to part vectors *P*, role vectors, *R*, and location vectors *L* was randomized on every simulation.

Table 1 shows the library of objects used in all simulations. In the table, objects are denoted using the format (using object O_1 as an example):

$$[(P_{1,1}, R_1) + (P_{4,4}, R_2)], [(P_{1,2}, R_3) + (P_{4,4}, R_4)],$$

where $(P_{1,1}, R_1)$ denotes part P_1 in location L_1 bound to role R_1, and $(P_{4,4}, R_2)$ denotes P_4, in L_4, bound to R_2; and the square brackets around these expressions indicate that roles R_1 and R_2 form a single relation linking P_1 to P_4. Note that P_1 appears in two locations in O_1, L_1 and L_2, and thus instantiates two tokens of the same type in the representation of O_1.

Simulation 1 was the most basic test of the model's ability to encode and match objects. We encoded objects O_1-O_3 into the model's memory and then tested its ability recognize object O_1. Unsurprisingly, it recognized O_1 as O_1 on three of three simulation runs, in the sense that it activated the O_1 group unit more than the group units for O_2 or O_3 (roughly 0.7 versus 0.6 or less, respectively; objects O_2 and O_3 are as active as they are because there is no lateral inhibition between group [object file] units).

Simulation 2 tested the model's ability to recognize an object when it has an extra part. On three runs, the model was initially trained on objects O_1-O_3, and then tested with O_4. O_4 is the same as O_1, but with an extra part, P_3, in a new relation to P_4. In addition to encoding O_4 as a new object file, the model also recognized it as most similar to object O_1 with activation about 0.7, versus about 0.5 for O_2 and O_3. When the model was then tested with O_1 as a stimulus (after O_4 was encoded into memory), the model recognized O_1 as O_1 (about 0.7), but also activated O_4 as a close match (about 0.6 versus about 0.5 for O_2 and O_3).

Table 1: Object Library for Simulations

O_1	$[(P_{1,1}, R_1) + (P_{4,4}, R_2)], [(P_{1,2}, R_3) + (P_{4,4}, R_4)]$
O_2	$[(P_{5,5}, R_7) + (P_{11,11}, R_8)], [(P_{8,8}, R_9) + (P_{11,11}, R_{10})], [(P_{10,10}, R_{11}) + (P_{11,11}, R_{12})]$
O_3	$[(P_{13,13}, R_{14}) + (P_{4,4}, R_2)], [(P_{12,12}, R_{13}) + (P_{12,12}, R_{14})]$
O_4	$[(P_{1,1}, R_1) + (P_{4,4}, R_2)], [(P_{1,2}, R_3) + (P_{4,4}, R_4)], [(P_{3,3}, R_5) + (P_{4,4}, R_6)]$
O_5	$[(P_{5,5}, R_7) + (P_{11,11}, R_8)], [(P_{10,10}, R_{11}) + (P_{11,11}, R_{12})]$
O_6	$[(P_{6,6}, R_7) + (P_{11,11}, R_8)], [(P_{8,8}, R_9) + (P_{11,11}, R_{10})], [(P_{10,10}, R_{11}) + (P_{11,11}, R_{12})]$

Simulation 3 tested the model's ability to recognize an object with a missing part. In three runs, the model was again trained with O_1-O_3 and tested with O_5, which is like O_2, but missing part P_8. The model correctly recognized O_5 as most similar to O_2 on two out of the three runs. On the third run, the model classified O_5 as most similar to both O_2 and O_1 equally. We speculate that in this case the part and relation vectors randomly generated for O_1 happened to be similar to those of O_2, in which case this result would be an example of a neighborhood effect. However, in all simulations, when O_2 was re-presented after O_5 was encoded, the model recognized it as an instance of O_2, with O_5 as a close second (both near 0.7), preferentially activating both over O_1.

Finally, simulation 4 tested the effect of replacing one part with another. Again, in three runs, the model was trained on O_1-O_3, and then tested with O_6 (O_6 is like O_2, but with P_5 replaced by P_6). On two of three runs, the model recognized O_6 as most similar to O_2. On the third, the model slightly favored O_3. Once again, we speculate that this result is due to neighborhood effects created by the randomization of the vectors. In all runs, when O_2 was re-presented to the model, it activated O_2 (greater than 0.7), with O_6 as a close second.

Discussion

The online generation of object files from the output of middle-to-late vision is a crucial step in visual thinking. We

present a model that, starting with outputs of a JIM-like model of shape perception, generates representations that can be stored in memory for future recognition and can be used by a LISA-like inference engine to reason about those objects. Preliminary simulation results suggest that this approach provides a promising starting point for simulating both object recognition and the visual-cognitive interface.

Simulations demonstrated that the model can correctly recognize familiar objects (simulation 1) as well as new objects created by adding (simulation 2), deleting (simulation 3), and replacing (simulation 4) parts of familiar objects. All of these transformations pose problems for non-compositional (e.g. template-based) accounts of object recognition (Biederman, 1987), but they are commonplace in human interactions with objects. Parts are often deleted by occlusion or by modification of the physical object (e.g. as when a tire is removed from a car); added, as when new parts are added to objects to extend functional capabilities; or replaced (e.g., for styling reasons). These types of modifications are especially common in commercially designed objects, so our ability to recognize and reason about these objects depends on our ability to tolerate these types of modifications: The first time we see a new model of coffeemaker, we may decide that the styling is not to our liking, but we do not stare at it in confusion about what it is.

Crucially, the representations used by this model are not only useful for recognition, as shown by the simulations, but also lend themselves naturally to reasoning about the objects' function. In particular, these representations are already in "LISAese", the representational format used by the LISA model, and as such are available to the full inductive power of that inference engine. For example, given an object file describing a novel coffee maker, LISA is well-equipped to infer that the handle is where the pot should be grasped, the filter basket is where the ground coffee should be placed, and the carafe is where the brewed coffee will collect. Once the model is supplied with a JIM-like front-end, it should be in a position to start with object images and end with inferences about those objects.

Acknowledgments

This research was supported by AFOSR Grant AF-FA9550-12-1-003.

References

Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review*, *94* (2), 115-147.

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, *24*, 87–185.

Doumas. L. A. A., Hummel, J. E., & Sandhofer, C. M. (2008). A theory of the discovery and predication of relational concepts. *Psychological Review, 115*, 1 - 43.

Green, C. B., & Hummel, J. E. (2004). Relational perception and cognition: Implications for cognitive architecture and the perceptual-cognitive interface. In B. H. Ross (Ed.), *The psychology of learning and motivation*, Vol 44. (pp. 201-223). San Diego: Academic Press.

Hummel, J. E. (2000). Where view-based theories break down: The role of structure in shape perception and object recognition. In E. Dietrich & A. Markman (Eds.), *Cognitive dynamics: Conceptual change in humans and machines* (pp. 157 - 185). Mahwah, NJ: Erlbaum.

Hummel, J. E. (2001). Complementary solutions to the binding problem in vision: Implications for shape perception and object recognition. *Visual Cognition*, *8*, 489 - 517.

Hummel, J. E. (2013). Object recognition. In D. Reisberg (Ed.) *Oxford Handbook of Cognitive Psychology*, 32-46, Oxford, UK: Oxford University Press.

Hummel, J. E., & Biederman, I. (1992). Dynamic binding in a neural network for shape recognition. *Psychological Review*, *99*, 480-517.

Hummel, J. E., & Holyoak, K. J. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review*, *104*, 427-466.

Hummel, J. E., & Holyoak, K. J. (2001). A process model of human transitive inference. In M. Gattis (Ed.). *Spatial schemas in abstract thought* (pp. 279-305). Cambridge, MA: MIT Press.

Hummel, J. E., & Holyoak, K. J. (2003a). A symbolic-connectionist theory of relational inference and generalization. *Psychological Review*, *110*, 220-264.

Hummel, J. E., & Holyoak, K. J. (2003b). Relational reasoning in a neurally-plausible cognitive architecture: An overview of the LISA project. *Cognitive Studies: Bulletin of the Japanese Cognitive Science Society, 10*, 58-75.

Hummel, J. E., & Stankiewicz, B. J. (1996). An architecture for rapid, hierarchical structural description. In T. Inui & J. McClelland (Eds.). *Attention and Performance XVI: Information Integration in Perception and Communication* (pp. 93-121). Cambridge, MA: MIT Press.

Hummel, J. E., & Stankiewicz, B. J. (1998). Two roles for attention in shape perception: A structural description model of visual scrutiny. *Visual Cognition*, *5*, 49-79.

Johnson-Laird, P. N. (1983). *Mental Models*. Cambridge, MA: Cambridge University Press.

Kahneman, D. & Treisman, A., & Gibbs, B. J (1992). The reviewing of object files: Object-specific integration of information. *Cognitive Psychology*, *24*, 175-219.

Knowlton, B. J., Morrison, R. G., Hummel, J. E., & Holyoak, K. J. (2012). A neurocomputational system for relational reasoning. *Trends in Cognitive Sciences*, *17*, 373-381.

Lovett, A., & Forbus, K. (2017). Modeling visual problem solving as analogical reasoning, *Psychological Review*, 124, 60-90.

Thoma V., Davidoff J. (2007) Object Recognition: Attention and Dual Routes. In: Osaka N., Rentschler I., Biederman I. (eds) *Object Recognition, Attention, and Action*. Springer, Tokyo.

Winston, P. (1975). Learning structural descriptions from examples. In P. Winston, *The Psychology of Computer Vision* (pp. 157-209). New York: McGraw-Hill.