Visual Imagination in Context: Retrieving a Coherent Set of Labels with Coherencer

Michael O. Vertolli (michaelvertolli@gmail.com)
Institute of Cognitive Science, 1125 Colonel By Dr.
Ottawa, ON K1S 5B6 CA

Jim Davies (jim@jimdavies.org)
Institute of Cognitive Science, 1125 Colonel By Dr.
Ottawa, ON K1S 5B6 CA

Abstract

A cognitive model of visual imagination will produce what we call "incoherent" results when it adds to an imagined scene that comes from multiple contexts (e.g., "arrow" and "violin" with "bow"). We approach this problem by exploring the co-occurrence of labels in images. We show that adding an incremental algorithm for examining networks of co-occurrence associations to the top-*n* co-occurring labels with a particular query produces greater coherence than just selecting the top-*n* labels or randomly selecting labels.

Keywords: imagination; coherence; artificial intelligence

Introduction

Humans imagine various scenes and situations in order to facilitate a broad range of cognitive abilities. Planning, problem solving, hypothetical thinking, counterfactual thinking, theory of mind, and mental time travel are all included in this list (Davies, Atance, & Martin Ordas, 2011). Though there is an extensive literature on the role of the imagination as it facilitates these abilities (see, for example, Markman, Klein & Suhr, 2012), the production of these imagined scenes remains largely unexplored. For the sake of brevity, this work exclusively focuses on the visual faculty, which is the most studied in the literature (Davies, Atance, & Martin Ordas, 2011).

When someone imagines a visual scene while reading a novel, of a woman walking a dog, for example, the scene might be constructed from visual memories from many different experiences. How these particular experiences are selected from the larger set of all experiences is not intuitively obvious. If there is more than just the woman and dog in the scene, it is unclear what makes the inclusion of certain elements (e.g., a leash, tree, path, bird, sky, or sun) more likely or appropriate than others (e.g., a rollercoaster, map of Spain, or cruise ship).

What is known is that people do not *arbitrarily* assemble what they imagine, even if it is largely fictional or involves fantastical creatures. There is an intuitive coherence imposed on imagined scenes that prevents unusual combinations.

One way that humans might make this selection is through the co-occurrence of objects in visual memory (by visual memory we mean only the memory of visual things, and not a specific subsystem like the visuo-spatial sketchpad). Thus, when one is imagining a scene given an environmental query (e.g., a novel, question, or problem), mental processes might search visual memory for other objects that often occur with that query. If one imagines a woman walking a dog, it is not surprising that leashes, trees, birds, etc. are more likely come to mind. They all often occur together in the world. In this case, "a woman walking a dog" serves as the query and the other elements are what are returned by some imagination process. One way we might explore this idea is through the use of computational models.

The Science of Imagination Laboratory Imagination Engine (SOILIE) is a computational model for the production of imagined scenes (Breault, Ouellet, Somers, & Davies, in press). In place of human 'experiences' and 'objects,' SOILIE has images from the web and their corresponding labels. In order to generate a novel scene, SOILIE must determine which labels are appropriate to select when given a particular query. And, much like the human description given above, SOILIE currently uses co-occurrence relations to make this selection. That is, how often one label is present in the same image with another label. The top *n* labels that co-occur with a particular query are then chosen to be included in the imagined scene. We call this method the "top-*n* model."

SOILIE derives these co-occurrence relations from the Peekaboom database of labelled images. With over fifty thousand of these images and ten thousand labels, the Peekaboom database is one of the largest and most diverse data sets of its kind. The data set is the combined result of two online games: the ESP Game and Peekaboom (Von Ahn, Liu, & Blum, 2006). In the former, pairs of players are shown the same image and without communicating try to enter the same words (Von Ahn & Dabbish, 2004). Words that both players enter are interpreted as being related to the image that they are viewing. In this way, common labels are applied to images collected from the internet. To prevent an overly constrained data set with only common words, labels that were repeatedly used for an image would become unusable. This generated a longer list and greater label diversity.

SOILIE's data set comes from a related game, Peekaboom, which uses ESP game data. Both games are designed to produce data that can be used in scientific research. Since these games were built with vision research

Vertolli, M.O. & Davies, J. (2013). Visual imagination in context: Retrieving a coherent set of labels with Coherencer. In R. West & T. Stewart (Eds.), Proceedings of the 12th International Conference on Cognitive Modeling, Ottawa: Carleton University (unnumbered six page online proceedings)

in mind, they seem particularly relevant for SOILIE's task and illustrative of the co-occurrence cognitive model.

SOILIE uses the Oracle of Objects (Astudillo, 2009) as its top-*n* model. The Oracle of Objects is an interface for a database of co-occurrence probabilities extracted from the Peekaboom database described above. Co-occurrence probabilities are calculated by dividing the total number of images (I) in the Peekaboom database that contain the co-occurring label (l) *and* a particular query (q) by the total number of images with *just* the query. Using set theory notation, this yields:

$$P(q,l) = \frac{|I_q \wedge I_l|}{|I_q|}$$

Where ∧ indicates set intersection and || indicates cardinality (i.e., the total number of elements in the set). The Oracle of Objects returns the ten labels that have the highest co-occurrence value given a particular query.

However, after working with this implementation of the top-*n* model, a problem became apparent. When images are selected purely on the basis of their co-occurrence with the initial query, the scenes produced are often contextually incoherent.

For example, SOILIE was queried with the word 'bow,' which is a homonym. Each sense of a homonym should return a different set of coherent co-occurrence relations. A 'bow' can be defined as a form of knot, the front of a ship, to bend at the waist, a weapon that shoots arrows, a tool used to play certain stringed instruments, etc. In the Peekaboom database, the ten highest co-occurring words with 'bow' are: arrow, woman, man, tie, hair, glasses, ribbon, hat, dog, and present. There are at least two distinct meanings of 'bow' in this output. The first makes sense in the context of the word 'arrow' and the second in the context of the word 'present.' These different contexts are invisible to SOILIE. Thus, the image that would be generated from this query of 'bow' that included both an arrow and a tie would be incoherent as it is drawing from two different contexts.

The problem of 'coherence' is not exclusive to SOILIE. Models that address context need to find a way to select coherent combinations (Hullett & Mateas, 2009). Algorithms that only consider a single factor, such as the top-*n* model, are fundamentally insensitive to context: they collapse higher-order information into a problem-space that cannot accommodate the required complexity. Context intuitively requires multiple factors. This explanation does lend itself to an alternative.

One way humans might resolve the problem of incoherence is by augmenting a process like the top-*n* model with a more detailed, associative search of their experiences. Our hypothesis is that this augmented approach will generate more coherent scenes than the top-*n* model alone. The augmentation occurs by examining the co-occurrence of each of the returned labels with each other, rather than just with the query.

Theory

The new, augmented approach, from now on described as Coherencer, operates as follows. First, a top-*n* search gathers the initial pool of four labels that co-occur with just the query. Then, an associative search checks the degree to which each label in the pool co-occurs with all the others, as well as with the query.[1] The network of co-occurrence relations that results is tested against a selection threshold. Labels with low co-occurrence in the network are swapped out and new labels that co-occur with the query are randomly swapped in until the threshold for the network as a whole is exceeded. Once the threshold is exceeded, the set that remains is returned for inclusion in the imagined scene.

Research in neuroscience suggests that visual working memory can hold approximately three to five objects of average complexity (Cowan, 2001; Edin, *et al*., 2009; Luck & Vogel, 1997; Marois & Ivanoff, 2005; Wheeler & Treisman, 2002). Thus, it is assumed that on average an imagined scene has approximately three to five elements in it at any given time. Similarly, four labels, excluding the query, are retrieved from the co-occurrence data gathered from the training set and five labels in total are selected per scene. We decided that this number, despite being in the upper part of the range, was the most useful: preliminary research suggested that larger sets of labels increased the divergence in the success of the models, five is still in the accepted range for visual working memory, and the query does not really need to be maintained in working memory to the same degree (an individual could always re-query) nor does it need to be retrieved.

The first part of Coherencer operates much like the Oracle of Objects by selecting the four labels with the highest co-occurrence probability with the query (i.e., a top-4 model). The second part takes all five labels, including the query, and produces a co-occurrence matrix. Using the 'bow' example, described above, one would get Table 1.[2]

Table 1: Co-occurrence matrix of 'bow,' showing the co-occurrence of the label in the column with the label in the row.

Labels	Bow	Arrow	Woman	Hat	Tie
Bow	-	0.206	0.235	0.118	0.118
Arrow	0.021	-	0.006	0.003	0.000
Woman	0.003	0.001	-	0.061	0.019
Hat	0.005	0.001	0.201	-	0.030
Tie	0.013	0.000	0.151	0.074	-

Each cell in the matrix holds the co-occurrence probability, with the row as the query (q) and the column as the co-

[1] The co-occurrence with the query still contributes to the co-occurrence overall, which is why it is included in the search despite previous consideration in the top-*n* model.

[2] Though the values of a label with itself should actually be 1 mathematically, it was convenient to ignore the values in the context of the implementation.

occurring label (l). Note that the co-occurrence matrices that result from Coherencer are not symmetrical. Since the total number of images that contain each label can be different, the probability that results can also be different.

In the next step, the mean for each row is calculated and then the mean of these means is calculated (i.e., the mean probability for the entire matrix). If this total mean is higher than a threshold,[3] the sample is deemed 'coherent,' the algorithm stops, and returns the set. If the total mean is below the threshold, the label row with the lowest mean probability is removed from the sample and a new label is randomly selected from all the remaining labels that co-occur (above 0.0) with the initial query. If the algorithm exhausts these co-occurring labels without finding a coherent selection, it will fail and a coherent image is deemed impossible. Thus, Coherencer yields a member (R) of the solution set (S), where S meets the following criteria in set theoretic notation:

$$S = \left\{ T \subseteq C_q \middle| q \in T \wedge |T| = 5 \wedge \left(\frac{1}{|T \times T|} \sum_{k \in T \times T} P(l_1, l_2) \right) > 0.37 \right\}$$

Here, q is the query; C_q is the set of all co-occurring labels with q; T is any subset in S; $T \times T$ is the set of all label pairs (l_1, l_2) of the elements of T with itself (also known as the Cartesian product); P means the co-occurrence probability as previously defined; $\wedge$ means 'and' as per propositional logic; $||$ means cardinality; $\subseteq$ means 'subset or equal to'; and, $\in$ means 'element of.'

To test the quality of this method we compared it to the competing top-*n* method and a random control method.

Method

There are three models that were compared: Coherencer, the top-*n* model, and a random search. The top-*n* model only retrieves the four labels with the highest co-occurrence probability with the query. The random search method randomly selects any four labels in the training set, ignoring co-occurrence.

The comparison proceeded as follows. The images in the Peekaboom database were randomly divided into equal halves: a test and training set. Each half was filtered to only include labels that were contained in both halves. After this filtration, images that had no labels were removed. The test and training sets were left with 28,496 and 28,483 images, respectively, after filtration.

One thousand labels were randomly selected, algorithmically, from the remaining 5,179 labels. This process occurred twice with possible overlap in the labels, once for each of the chi-square tests run on the results.

Each label in the first set of one thousand labels was run through the top-*n* and random search and each label in the second set of one thousand labels was run through the top-*n* and Coherencer. Each query plus four returned labels is an imagined scene.

The results for each of the algorithms were assessed with regard to the test set. If at least one image in the test set contained the five labels that were selected by a particular algorithm, including the query, the algorithm scored one point. If there were no images containing the five labels, the imagined scene did not score a point. Using the total number of points (which could range, theoretically, between 0 and 1000), we compared the random search to the top-*n* model as well as the top-*n* model to Coherencer.

Results

The results came out as expected. Coherencer had more successful matches with the test set than the top-*n* model. Similarly, the top-*n* model had more successful matches with than the random search. The statistical details are as follows.

Top-*n* performed significantly better than random search, $\chi^2(1) = 69.46$, $p < .001$ (see Table 2). This seems to represent the fact that, based on the odds ratio, the odds of success were 10.03 times higher with the top-*n* model than with the random search. Coherencer also performed significantly better than top-*n*, $\chi^2(1) = 37.62$, $p < .001$ (see Table 3). The effect, based on the odds ratio, suggests that the odds of success were 2.32 times higher for the Coherencer model than for the top-*n* model.

Table 2: χ^2 calculation between random search and top-*n*.

Model		Test		Total
		Failure	Success	
Random search	Count	990.0	10.0	1000.0
	Expected	949.0	51.0	
	Std. Residual	1.3	-5.7	
Top-*n*	Count	908.0	92.0	1000.0
	Expected	949.0	51.0	
	Std. Residual	-1.3	5.7	
Total	Count	1898.0	102.0	2000.0

Table 3: χ^2 calculation between Coherencer and top-*n*.

Model		Test		Total
		Failure	Success	
Top-*n*	Count	914.0	86.0	1000.0
	Expected	867.5	132.5	
	Std. Residual	1.6	-4.0	
Coherencer	Count	821.0	179.0	1000.0
	Expected	867.5	132.5	
	Std. Residual	-1.6	4.0	
Total	Count	1735.0	265.0	2000.0

[3] Preliminary tests showed 0.37 to be ideal. A more detailed analysis will be performed in future work.

Discussion

The results support the idea that Coherencer generates elements that create a more coherent scene than the top-*n* model, which is in turn an improvement on a random search. For Coherencer, this fact is all the more impressive, given how many challenges were stacked against it. Partially, these challenges were due to lexical confounds. For example, synonyms confine the search space (e.g., when one searches 'dog' it will not include the related associations for 'puppy'), confound the output (e.g., by returning 'puppy' when searching for 'dog'), or results in false negatives (e.g., 'puppy' is included in an image with the other labels but 'dog' is not; the match fails as a result).[4] Hyponyms and hypernyms (e.g., 'dog' to 'German Shepherd' and 'German Shepherd' to 'dog,' respectively) result in similar problems as do meronyms (e.g., 'nose' to 'face') and other lexical relations. Future research by the Science of Imagination Laboratory is focused on ameliorating many of these problems. In addition, there are also problems in the train-test design. Using only half of the available data (the training set) underrates the performance.

There has been some related research. Paul Thagard's (2000) work on coherence is worth noting in particular. Thagard explicitly outlines a similar algorithm to Coherencer as one of the possible computational approaches for resolving coherence problems. He calls this approach an "incremental" algorithm. Like Coherencer, the incremental algorithm evaluates the coherence of a single element at a time relative to the current pool of selected elements. However, there are some pronounced differences. First, the incremental algorithm builds its initial pool one element at a time using the method just described whereas Coherencer seeds its initial pool with the top-*n* algorithm (i.e., relative only to the query). Second, the space within the incremental algorithm's pool is infinite: it could literally contain the entire set of possible elements if that maximized coherence. In contrast, Coherencer has a finite limit on the size of its pool. Third, Coherencer does not maximize coherence; it makes sure that it passes a certain threshold. Fourth, once an element has been selected by the incremental algorithm, it cannot be unselected (i.e., there is no backtracking with selected elements). Coherencer maintains backtracking capabilities for selected elements; though, it cannot backtrack relative to rejected elements (another commonality). Despite these differences, we take both algorithms to be of the same class of algorithms and 'incremental' works as a label for this class: it highlights the serial approach that is a defining feature of the class of algorithms.

After defining the original, incremental algorithm, Thagard proceeds to argue that these algorithms are problematic, at least prescriptively (i.e., for indicating the ideal), since they often lead to suboptimal solutions. Largely, this is a result of the serial increments by which the algorithm makes comparisons, a property that we take to be central to the class of incremental algorithms. That is, since Coherencer and similar incremental algorithms can only examine the relations of the current element, rather than all possible elements in parallel, these algorithms are more likely to get trapped on optima that are less than ideal outside their current locale.

For Coherencer, for example, this means that it may be the best decision to remove 'arrow' (lowest total row and column co-occurrence of 0.238) given the set described previously, but 'arrow' might actually be a part of the set with the highest possible co-occurrence for the query. Rejecting 'arrow,' which is reasonable at this stage, prevents the algorithm from re-attaining it later or even finding the theoretical 'best set' of which arrow is a part. Though 'backtracking' for these rejected elements has been implemented as a fix for serial approaches, Thagard suggests that this is still worse than some alternatives.

In response to this problem, Thagard (2000) proposes a connectionist model that is not hindered by serial processes. These models examine possible solutions in parallel, which decreases the possibility of getting stuck on sub-optimal solutions. In brief, the construction of this model proceeds as follows.

For every element (e.g., label), construct a node in the network. For every positive constraint between two elements (i.e., where inclusion of one element, at minimum, increases the likelihood of inclusion of the other element; co-occurrence in the context of this work), construct an excitatory link between the corresponding nodes. For every negative constraint (i.e., the opposite of a positive constraint; unclear in the current context, but possibly a co-occurrence of 0), construct an inhibitory connection between corresponding nodes. Assign an initial activation to each node and then update all the nodes in parallel using one of the many update equations (see, for example, McClelland & Rumelhart, 1989). Continue updating the activations until the nodes in the network cease to fluctuate or fluctuate within a small margin.

Thagard has implemented a number of such networks in related domains (e.g., Thagard, 1989, 1991, 1992a, 1992b, 2000; Nowak & Thagard, 1992a, 1992b; Eliasmith & Thagard, 1997; Thagard, Holyoak, Nelson & Gochfeld, 1990). Though there are counter-arguments to his position, they are not of interest here. Instead, what is important is the relationship Thagard highlights between incremental algorithms and bounded rationality.

Bounded rationality is an idea initially proposed by Herbert Simon (1991) that focuses on the limitations of human decision making. Thagard observes that both incremental algorithms and much of human thinking results in sub-optimal solutions to problems of various types, including coherence problems. Thus, he argues, this

[4] We would argue that these challenges do not completely generalize to the top-*n* model. For example, increasing the search space would increase the interference or noise for the top-*n* model while it increases the possibility of finding a better match for Coherencer. Improvements that restrict the search space *might* generalize. Future research will examine these features.

intersection between the bounds of rationality and incremental algorithms is in need of further research.

In a similar line of inquiry, research in working memory has also described a limited, serial system—the episodic buffer—that roughly matches what Thagard is describing (Baddeley, 2000; Baddeley, 2002a; Baddeley, 2002b). The episodic buffer is believed to be the means of integration for the different sense modalities as well as the retrieval mechanism for long-term memories. Such a system would be perfectly poised to play a leading role in imaginative processes, at least in so far as it might supply the 'material' on which these processes operate. Even in such a restricted role, the limitations of the episodic buffer might result in downstream limitations, and this suggests a rather simple explanation for bounded rationality. Two interesting additional implications are that chunking may play a role in expanding the memory limitations of iterative models and that attention may have a significant role in the integrative processes of these mechanisms.

In the current research, these observations suggest interesting implications for Coherencer. With respect to human cognition, Coherencer might better model the bounds of human rationality than the alternatives, including Thagard's connectionist models. The parallels with the episodic buffer suggest that Coherencer might also provide a useful computational model for this subsystem of working memory. Both of these parallels give credence to Coherencer as a useful model of certain processes in human cognition. Future research will continue to examine these parallels.

Other noteworthy implications can be drawn from the model. First, Coherencer's augmentation of the top-*n* model suggests that humans might generally use the most common associations when imagining visual scenes, as well as other modalities. More specifically, the query and top-*n* model combined might illustrate a form of cognitive priming. This priming would act much like the initial seeding process (i.e., using the top associations for the query from the top-*n* model). This interpretation also supports the exclusion of the query from the limitations of visual working memory capacity: research suggests that implicit visual memory, like that of visual priming, might have a much higher capacity than visual working memory (see, for example, Chun, 2000; Chun & Nakayama, 2000). Parallels between Coherencer and visual or word associations and priming in human populations might be an avenue for exploring these questions.

Coherencer also suggests that humans compare pairs of associations when checking for coherence. However, there is no obvious reason why processing should stop there. Comparisons of triplets, quadruplets, and even quintuplets of associations might lend themselves to coherence problems and their associated solutions. Implications for cognition could be different for these complex networks. Future research will explore these possibilities.

Conclusion

Contextual coherence in visual imagination is a challenging problem both theoretically and computationally. The current research supports the use of Coherencer as a useful augmentation of the original top-*n* model for this problem. We have shown support for the notion that co-occurrence with the query alone is not sufficient for returning a coherent set of labels and that our solution, checking the co-occurrence between all the returned labels, can improve coherence of objects in imagined scenes.

This research contributes to the Science of Imagination Laboratory's work on SOILIE, the exploration of incremental algorithms and bounded rationality, as discussed by Thagard (2000), and to research on the episodic buffer in working memory. Future research will explore different types of thresholds, different numbers of associated terms, chunking, backtracking for rejected elements, direct comparisons with Thagard's connectionist models, comparisons with other models, the resolution of various lexical confounds, the role of attention, parallels with the episodic buffer, and parallels with working memory more generally.

References

Astudillo, C. (2009). *Co-occurrence of objects in an image*. Website. Retrieved from http://ing.utalca.cl/~castudillo/research/pkb/co_ocurrence/form.php

Baddeley, A. (2000). The episodic buffer: a new component of working memory?. *Trends in Cognitive Sciences*, *4*(11), 417-423.

Baddeley, A. (2002a). Fractionating the central executive. *Principles of frontal lobe function*, 246-260.

Baddeley, A. D. (2002b). Is working memory still working?. *European Psychologist*, *7*(2), 85-97.

Breault, V., Ouellet, S., Somers, S., & Davies, J. (in press). SOILIE: A computational model of 2D visual imagination. In N. Rußwinkel, U. Drewitz & H. van Rijn (eds.), *Proceedings of the 11th International Conference on Cognitive Modeling*, Ottawa: Carleton University.

Chun, M. M. (2000). Contextual cueing of visual attention. *Trends in cognitive sciences*, *4*(5), 170-178.

Chun, M. M., & Nakayama, K. (2000). On the functional role of implicit visual memory for the adaptive deployment of attention across scenes. *Visual Cognition*, *7*(1-3), 65-81.

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, *24*(1), 87-114.

Davies, J., Atance, C. & Martin Ordas, G. (2011). A framework and open questions on imagination in adults and children. *Imagination, Cognition, and Personality, Special Issue on Mental Imagery in Children*. 31:1-2, 143-157.

Edin, F., Klingberg, T., Johansson, P., McNab, F., Tegnér, J., & Compte, A. (2009). Mechanism for top-down control of working memory capacity. *Proceedings of the National Academy of Sciences*, *106*(16), 6802-6807.

Eliasmith, C., & Thagard, P. (1997). Waves, particles, and explanatory coherence. *British Journal for the Philosophy of Science, 48*, 1-19.

Simon, H. A. (1991). Bounded rationality and organizational learning. *Organization Science*, *2*(1), 125-134.

Hullett, K., & Mateas, M. (2009, April). Scenario generation for emergency rescue training games. In *Proceedings of the 4th International Conference on Foundations of Digital Games* (pp. 99-106). ACM.

Luck, S. J., & Vogel, E. K. (1997). The capacity of visual working memory for features and conjunctions. *Nature*, *390*(6657), 279-280.

Markman, K. D., Klein, W. M., & Suhr, J. A. (Eds.). (2012). *Handbook of imagination and mental simulation*. Psychology Press.

Marois, R., & Ivanoff, J. (2005). Capacity limits of information processing in the brain. *Trends in Cognitive Sciences*, *9*(6), 296-305.

McClelland, J. L., & Rumelhart, D. E. (1989). *Exploration in parallel distributed processing*. Cambridge, MIT Press.

Simon, H. A. (1991). Bounded rationality and organizational learning.*Organization Science*, *2*(1), 125-134.

Thagard, P. (1989). Explanatory coherence. *Behavioral and Brain Sciences*, *12*, 435-467.

Thagard, P. (1991). The dinosaur debate: explanatory coherence and the problem of competing hypotheses. In J. Pollock, and R. Cummins (eds.), *Philosophy and AI: essays at the interface* (pp. 279-300). Cambridge, MIT Press.

Thagard, P. (1992a). Adversarial problem solving: modeling an opponent using explanatory coherence. *Cognitive Science, 16*, 123-149.

Thagard, P. (1992b). *Conceptual revolutions*. Princeton, Princeton University Press.

Thagard, P. (2000). *Coherence in thought and action*. Cambridge, MIT Press.

Thagard, P., Holyoak, K., Nelson, G., and Gochfeld, D. (1990). Analog retrieval by constraint satisfaction. *Artificial Intelligence, 46*, 259-310.

Von Ahn, L., & Dabbish, L. (2004). Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 319-326). ACM.

Von Ahn, L., Liu, R., & Blum, M. (2006). Peekaboom: a game for locating objects in images. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 55-64). ACM.

Wheeler, M. E., & Treisman, A. M. (2002). Binding in short-term visual memory. *Journal of Experimental Psychology: General*, *131*(1), 48.