WHAT IS AN OBJECT FILE?

E.J. Green
NYU/MIT, Philosophy

Jake Quilty-Dunn
The Graduate Center, CUNY, Philosophy

TABLE OF CONTENTS

§1. Introduction

A fundamental capacity of minds like ours is the ability to perceive and keep track of objects. The nature of this capacity has often been a fault line in philosophical debates about innateness, the perception–cognition border, and reference grounding. Descartes famously argued that the ability to perceive an individual object such as a piece of wax is a matter of post-sensory judgment that relies on innate concepts, while Berkeley and Hume maintained that object representation is built up out of sensory representations of low-level features. Philosophers of mind and language in the 20th century were often preoccupied with conditions for representation of objects, citing, for example, complex linguistic abilities (e.g., Sellars 1956; Quine 1960) and abilities to re-identify objects over time (Strawson 1963). Philosophers of language have proposed that object perception plays a crucial role in grounding demonstrative reference (Kaplan 1989) and chains of reference-borrowing for proper names (e.g., Kripke 1972; Devitt & Sterelny 1999).

These classic philosophical discussions typically proceeded independently of detailed scientifically informed theories of object representation. The past couple of decades, however, have witnessed a surge of research on object representation within both philosophy and psychology. The notion of an *object file* has played an important role in many areas of this literature (e.g., Kahneman et al. 1992; Pylyshyn 2007; Carey 2009; Jeshion 2010; Jordan et al. 2010; Recanati 2012; O'Callaghan 2014; Murez & Recanati 2016; Echeverri 2016). An object file is generally characterized as a representation that (i) sustains reference to an external object over time, and (ii) stores and updates information concerning the properties of that object. Beyond this, however, there is surprisingly little agreement about what object files *are*. In this paper, we offer a theory of both the *format* and *architecture* of object files. We will defend two central claims. First, we argue that object files represent information in a non-iconic, propositional format. Second, we argue that object files store information concerning different feature categories in separate memory stores. We call the resulting picture the *multiple-slots view* of object files.

The structure of the paper is as follows. In section 2, we'll discuss some of the primary empirical reasons for positing object files. This motivation derives from work on multiple-object tracking, object reviewing, and visual short-term memory. In section 3, we'll consider the issue of representational format. We'll argue that an iconic view of object files cannot accommodate facts about object files such as the explicit encoding of individuals and abstract features, and the independent storage and forgetting of low-level perceptible features. In section 4, we'll turn to the issue of architecture. We'll distinguish two views of object file architecture: (i) a *single-slot view*, on which various features of an object are all entered into a single memory store within an object file, and (ii) a *multiple-slots view*, on which features of different categories are entered into separate memory stores within an object file. We'll argue that evidence clearly favors the latter position. In section 5, we'll enrich the multiple-slots view with an indirect-addressing model of memory storage and retrieval.

§2. Empirical Support for Object Files

Many strands of experimental work have been taken to support the existence of object files. For present purposes, we'll focus on two kinds of experiments. Experiments of the first kind examine our ability to maintain and update representations of objects as they move, while those of the second examine our ability to maintain representations of objects after they have disappeared from view.

2.1—Object reviewing and multiple-object tracking

In the object-reviewing task, a participant is shown a pair of objects on screen, and preview features (usually letters or numerals) briefly appear on those objects. After the preview features vanish, the objects move to new locations. Finally, a test feature appears, and the participant is asked to report whether it is the same as either of the preview features. A number of studies have shown that under these conditions, participants' reaction times are faster when the test feature matches one of the preview features (a case of general priming) and are even faster when it appears on the same object on which that preview feature initially appeared (Kahneman et al. 1992; Noles et al. 2005). This latter effect is known as the *object-specific preview benefit* (OSPB).

Kahneman, Treisman, and Gibbs (1992) proposed that the OSPB could be explained by appeal to object files. The idea is that when the preview feature appears on an object, it is automatically encoded in a stable representation—a file—associated with that object in visual short-term memory. Subsequent responses to a test feature are facilitated when it matches information already stored in the file for the object on which it appears. Further work within the object-reviewing paradigm has found that object files "move" with objects primarily on the basis of spatiotemporal continuity (Mitroff & Alvarez 2007; Kimchi & Pirkner 2014; although see Hollingworth & Franconeri 2009).

In the multiple-object tracking (MOT) task, a participant is presented with a set of objects, and some of these objects are flashed on and off in order to mark their status as targets. After this, all of the objects begin to move randomly about the screen for some period of time. At the end of the trial, the participant is typically asked, of a single object, whether that object was a target, or she is asked to report all of the targets. Most MOT studies have suggested that perceivers can reliably track up to about 4 objects, after which performance declines rapidly (Pylyshyn & Storm 1988; Scholl & Pylyshyn 1999; vanMarle & Scholl 2003; although see Alvarez & Franconeri 2007; Franconeri et al. 2013).[1]

Work within both the object-reviewing and MOT paradigms indicates that we have a limited capacity mechanism for maintaining representations of a small number of objects over time. To account for this, Pylyshyn (2003; 2007) has introduced the idea of a *visual index*. A visual index is a demonstrative-like symbol that picks out an object and continues to pick it out across changes in its location or surface features. Pylyshyn (2007, 38-39) and Kahneman et al. (1992, 215-216) both suggest that visual indexes may be fruitfully integrated with the object file theory (see also Carey 2009, 72ff). The general idea, which we will adopt here, is that object files are complex mental particulars consisting of indexes and short-term memory stores in which features of the indexed object are encoded. The index is what "links" the information in the store to a particular object. Such information is accurate or inaccurate by virtue of being about a particular object, and the index determines the object against which it is assessed for accuracy. Unfortunately, however, the nature of the connection between indexes and these short-term memory stores has not been clearly specified. We'll offer an account of this connection in section 5.

2.2—Visual short-term memory

The object-reviewing and MOT paradigms concern situations where an object is perceived as persisting through visible movement (or sometimes through brief occlusion). A different class of experiments has instead investigated object representations in the context of visual short-term memory (VSTM), also known as visual working memory. VSTM is distinct from earlier, higher capacity short-term memory stores such as iconic memory (Sperling 1960) and fragile visual short-term memory (Sligte et al. 2008).

In VSTM studies, participants are briefly presented with a display of objects called a *sample array*, and asked to remember one or more features of those objects. The sample array then vanishes, followed by a blank screen or mask for a brief period of about 1 second, called the *retention interval*. Finally, during the testing period, the participant makes a response that indicates how accurately or precisely she encoded features of the objects in the sample array. In both paradigms, subjects are often instructed to engage in *articulatory suppression* (e.g.,

[1] Alvarez and Franconeri (2007) found that when participants were allowed to adjust the speed of the objects to a level that they felt comfortable with, they were able to track up to 8 objects in parallel. It is thus possible that there is no strict set-size limitation in MOT (*pace* Pylyshyn 2007). The ability to track objects in parallel may be primarily limited by either the speed at which the objects travel (Holcombe & Chen 2012) or the spacing between targets and distractors (Franconeri et al. 2010).

repeating the word 'the' several times a second for the duration of the trial). The point of articulatory suppression is to place a load on verbal working memory, ensuring that information is encoded in VSTM rather than verbal working memory.

Experiments differ with respect to the task performed during the testing period. In *change-detection* tasks, participants are asked to make a simple "same-different" judgment. For instance, they might be presented with a test array the same size as the sample array and asked to indicate whether any of the objects in the test array differ from their counterparts in the same location of the sample array (Luck & Vogel 1997). In *continuous-report* tasks, participants are asked to manually adjust a probe stimulus so that it matches some object from the sample array (Wilken & Ma 2004; Zhang & Luck 2008; Fougnie et al. 2010). For example, participants may be presented with an empty square at test, and asked to select a position on a color wheel that matches the color of the corresponding item in the sample array (i.e., the item initially presented at that location).

An important study due to Luck and Vogel (1997) has been taken to support the view that VSTM stores object representations. Luck and Vogel asked participants to memorize either the colors, the orientations, or both the colors and orientations of a sample array of line segments. They found that change-detection accuracy in each of these conditions was essentially the same. In each case, participants could reliably recall about 3-4 objects, after which performance sharply declined. Luck and Vogel took this to suggest that VSTM is limited in the number of object files that can be simultaneously stored, but that there is no cost to encoding multiple features in the same object file. According to Luck and Vogel, once a file has been opened for an object, we can store *both* the color *and* the orientation of the object just as easily as we can store its color alone. As we'll see in Section 4, this picture is oversimplified in several respects. However, we'll argue there that the overall body of evidence supports the view that VSTM stores object files.

Consistent with the view that object representations in VSTM are continuous with those deployed in on-line perception, there is evidence that representations held in VSTM are used in guiding saccades to targets, and in correcting errant saccades (Hollingworth et al. 2008). Furthermore, there is evidence that representations in VSTM can interfere with MOT (Fougnie & Marois 2006, 2009; although see Zhang et al. 2010).[2] Finally, there is

[2] The situation here is complicated for two reasons. First, although Fougnie and Marois have documented dual-task interference between VSTM and MOT, it seems that an extra tracked object results, on average, in a 0.5-object reduction in the number of objects that can be stored in VSTM. Second, Zhang et al. (2010) found that VSTM interferes with MOT only when subjects are required to remember spatial information about the stored objects.

considerable neural overlap between VSTM and MOT tasks, both of which heavily recruit the intraparietal sulcus (Drew et al. 2011).

§3. The format of object files

3.1. Iconic format.

Any comprehensive theory of some domain of mental representation must characterize its representational *format*. In this section, we consider the format of object files. Formats are structural features of representational vehicles that play a role in individuating general types of representations. More succinctly, Kosslyn et al. write, "A format is a type of code" (2006, 8).

The sentence

(1) This is a large, yawning Bengal tiger.

is structured differently from a picture of a Bengal tiger yawning (Fig. 1).

Figure 1. Yawning tiger

There are, of course, differences in the contents of these representations. For example, Figure 1 represents specific and low-level properties of the tiger, such as the shape of its tongue or orientations of individual strands of fur, on which (1) is silent. (1) also explicitly represents the tiger as being a Bengal, while Figure 1 does not. But there are also differences in the structural features of the representations themselves, which are not simply differences in content. For example, pointing at a part of the picture that corresponds to part of the tiger's ear and moving your finger to the right will result in you pointing at a part closer to the part corresponding to the tiger's eye. No similar rule applies to any part of (1). This latter sort of difference arises purely from representational format.

A standard distinction between formats used in cognitive science is the distinction between *iconic* (also called depictive or pictorial) and *propositional* representations. Iconic representations figure in prominent theories of mental imagery (e.g., Kosslyn 1980; Kosslyn et al. 2006), mental-models theories of deductive inference (e.g., Johnson-Laird 2006), high-capacity short-term memory stores in early perceptual systems (e.g., Sperling 1960; Neisser 1968; see also Fodor 2007), and core cognition (e.g., Carey 2009).[3] Propositional representations are most often invoked to explain logical inference (e.g., Braine and O'Brien 1998) and the relation between thought and language (e.g., Fodor 1975), but also figure in some theories of diverse mental phenomena including early perception and imagery (e.g., Pylyshyn 1984).

Figure 1 is a prototypical case of iconic representation. Critically, the figure obeys two principles, which we'll call ICONICITY and HOLISM:

> ICONICITY: Every part of the representation represents some part of the scene represented by the whole representation.
>
> HOLISM: Each part of the representation represents multiple properties at once, so that the representation does not have separate vehicles corresponding to separate properties and individuals.

Point at any part of Figure 1, and the selected part of the image will represent some part of the scene. Furthermore, that part will represent multiple properties at once. For example, the part of the image that represents the pinkness of the tiger's tongue is the very same part that represents its texture and spatial orientation.[4] There is also no separate part of the image that represents the individual tiger, over and above the parts of the image that represent its perceptible features.

In what follows, we'll take ICONICITY and HOLISM to be the signature markers of iconic format. Variations on ICONICITY are employed by most of the theorists cited earlier (e.g., Kosslyn 1980, 33; Kosslyn et al. 2006, 11–12; Carey 2009, 459; Johnson-Laird 2006,

[3] These high-capacity perceptual memory stores include iconic memory (Sperling 1960) and fragile visual short-term memory (Sligte et al. 2008), which may utilize iconic representations. As will become clear below, we do *not* think that this is plausible for object files in VSTM.

[4] Philosophers who hold that perceptual representation is iconic have often endorsed some version of HOLISM. For instance, Burge (2014b) writes: "Just as one cannot draw a line without drawing its length, shape, and orientation, one cannot visually represent an environmental edge as such without representing its length, shape, and orientation, as such" (493).

25; Fodor 2007, 108). The "parts" of represented scenes are spatiotemporal parts, while the "parts" of the representations themselves are functional entities that are analogous to spatiotemporal parts. There need not be a literal image in the brain to implement an iconic format; all that matters is that the mental representation plays the right functional role.

ICONICITY leads naturally to HOLISM (e.g., Kosslyn et al. 2006, 11), particularly when the representation needs to encode multiple properties of things in the represented scene at once. If a representation encodes multiple properties of each part of a scene, and each part of the representation corresponds uniquely to a part of that scene, then every part of the representation must represent multiple properties concurrently.[5] Furthermore, if a representation satisfies HOLISM, then it does not contain any part that stands for an individual separately from its properties. For if a representation *R* is holistic, then each part of *R* encodes multiple properties. Therefore, no part picks out an individual alone, independent of any properties. As such, icons lack a syntactic separation between representations of individuals and representations of features. Following Kosslyn and others, we will take ICONICITY+HOLISM to characterize iconic mental representations.

Some notions of iconicity do not invoke ICONICITY+HOLISM. For example, these principles may not play an important role in some philosophical accounts of the nature of pictorial representations such as paintings (e.g., Hopkins 1998; Greenberg 2013; Briscoe forthcoming; but see Kulvicki 2015; Quilty-Dunn ms). However, our present interest is in the notion of iconic mental representation that is operative within cognitive science. It is thus not crucial that the notion of iconicity at work here fits perfectly with philosophical accounts of pictorial representations outside the mind.

Propositional representations satisfy neither ICONICITY nor HOLISM. For example, some parts of (1) do not represent parts of the scene (e.g., the word 'is'), and some parts do not represent at all (e.g., 'This…yawning'). Furthermore, distinct vehicles like 'large' and 'tiger' represent distinct properties. Instead, propositional representations are *discursive*: they comprise discrete constituents that compose in often highly constrained ways. Propositional and other discursive representations thus have "canonical decompositions" (Fodor 2007, 108) into constituents. Icons, on the other hand, do not prioritize any particular segmentation into constituents; no matter where you draw a line through Figure 1, the parts of the image on both sides of the line represent certain parts of the scene (Kosslyn 1994, 5).

Carey (2009) hypothesizes that object files are iconic. As aforementioned, she accepts something like the ICONICITY principle (2009, 452). She also seems committed to HOLISM,

[5] If every part of a representation *R* "corresponds uniquely" to some part of a scene *S*, then no two distinct parts of *R* represent the same part of *S*.

writing that object files have an iconic format "with size imagistically represented, as well as shape, color, and other perceptual properties bound to the symbols iconically" (2009, 459; see also her Figure 4.9, 147). Since Carey seems to endorse the predominant notion of iconicity in cognitive science (ICONICITY+HOLISM), we will assume this notion of iconicity in what follows. We will argue that the evidence suggests that the format of object files is not iconic in this sense, and that a propositional model easily explains all the available data.

3.2. Object files and iconic format.

As elaborated above, object files involve explicit indexes, akin to demonstratives. There is strong reason to believe that these indexes are syntactically separate from any feature representations used to attribute features to the object.[6] For example, indexes are plausibly maintained across changes in the feature representations held in an object file. Subjects can reliably track objects in MOT despite significant changes in color, shape, and size during a trial (vanMarle & Scholl 2003; Zhou et al. 2010). Thus, we contend that, at minimum, there is a syntactic separation between indexical constituents and feature representations in object files. As discussed above, this is inconsistent with HOLISM. A purely iconic model of object files is not feasible.

It is nonetheless possible that feature representations might be iconic even though the indexical constituents of object files are not.[7] Quilty-Dunn (2016) argues, however, that several object-reviewing studies indicate that object files encode abstract features in a fashion that tells against iconic format. For example, Gordon and Irwin (2000) showed an OSPB for categories like *fish*, even when the preview stimulus was the word 'fish' and the test stimulus was a picture of a fish. Results like this (see also Gordon & Irwin 1996; Jordan et al. 2010) suggest that object files explicitly represent abstract features. For the OSPB to show up for a novel picture of a fish, for example, when the preview was only the word 'fish', the information stored in an object file that can be directly used for the experimental task must be in a format that is not tied to any low-level features of the word, or indeed even the fact

[6] Note, however, that accepting the *syntactic* division between indexical constituents and feature representations does not require accepting that the former achieves reference wholly independently of the contents of the latter (see Green forthcoming).

[7] Burge (2010; 2014a) argues that perceptual representation is iconic, but also that its contents can be modeled on complex linguistic demonstratives, such as 'That F', where F is some perceptible attribute. Perhaps for Burge the attributive elements of perception are iconic and the singular elements are non-iconic. We are hesitant to ascribe any such view to Burge, however, given his apparent skepticism of the existence of syntactic properties of perceptual or cognitive representations (2010, 95–96). We are unsure how to construe claims about representational format without appeal to syntactic properties of representations, since doing so would preclude characterizing the structures of representations themselves apart from structures of their contents.

that it is a word.[8] HOLISM requires that icons lack separate constituents for separate features. It is not clear how an icon could plausibly represent a feature like *fish* in a way that is completely separable from low-level features without positing a discrete constituent that represents *fish*—and thereby violating HOLISM. The presence of a discrete constituent that represents *fish*, on the other hand, could explain why the OSPB persists despite the preview and test features being related solely in virtue of their connection to the category *fish* and otherwise lacking any relevant similarity in their low-level visible features. While an icon may represent a fish, explicit representation of the category *fish* that abstracts completely from any low-level features may be best explained in terms of a non-iconic format.[9]

If this argument succeeds, then at least some of the features stored in object files are plausibly represented by means of discrete symbols. Nonetheless, it is possible that object files contain non-iconic representations of high-level features while low-level features such as color, shape, and orientation are represented iconically. Call this the *mixed-features model*. In what follows, we'll argue that the iconic approach fails even in the case of low-level features, and thus that the mixed-features model is incorrect.

Since icons are holistic and therefore lack separate symbols for separate features, an iconic representation of the low-level features of an object should not allow independent encoding of separate low-level features. A prediction of the mixed-features model, then, is that object files should not represent, say, the color of an object without also representing its shape and orientation. There is some positive evidence in favor of this prediction. As discussed above, Luck and Vogel (1997) tested the ability of subjects to recall either the color, the orientation, or both the color and the orientation of objects. They found no decline for storage of the conjunction of features as opposed to individual features, indicating that

[8] One might suggest that these findings are due to mere associations between iconic representations of word-forms and iconic representations of visual appearances. However, in a different experiment, Gordon and Irwin (1996) showed that OSPBs were *not* observed when the preview and test features were semantically associated words, such as 'doctor' and 'nurse', demonstrating that even strongly associated information does not get stored in an object file. The effect is thus category or kind-specific, and cannot be explained by simple association.

[9] Kosslyn et al. (2006) argue that iconic representations cannot explicitly represent general categories, such as *fish* or *ball*. Icons, they suggest, are limited to representing particular exemplars of such categories. Thus: "Depictions are not abstract: they cannot directly refer to nonpicturable concepts (…); they represent individual instances (not classes), and they are specific to a particular sensory modality" (2006, 11). Burge claims both that perception is iconic and that it may represent higher-level attributes, but insists that "one can represent something as a pushing, as an instance of agency, as a pine tree, as a piano, or as one's favorite movie villain—only by visually attributing low-level attributes" (2014a, 575). Since Burge's iconic model of perception requires that perceptual representation of higher-level attributes is tied to representation of lower-level attributes (including "generic" ones, such as canonical shapes [2014a, 576]), the evidence cited here seems to target his view as well as Carey's. The relevant object-reviewing experiments do not involve the persistence of *any*, even highly generic, low-level features from preview to test phases.

VSTM stores "integrated object percepts rather than individual features" (1997, 280). The mixed-features model can explain this result. An icon does not syntactically decompose into distinct feature representations, but rather represents low-level features together. Once you store an iconic representation of the color of a line, therefore, information about its orientation comes for free.

Other results, however, cast doubt on the mixed-features model. In particular, the phenomenon of *independent forgetting* raises serious difficulties for the view. Note that if a representation encodes two features holistically, then deletion or degradation of one feature would be expected, other things being equal, to disrupt encoding of the other feature as well. Accordingly, because the iconic model holds that features are bound together holistically, it also predicts that features should be maintained or forgotten together, rather than independently. Note that we are not assuming here that it is impossible for an iconic representation to be *noncommittal* about certain low-level features such as color or size. To assume this would be to commit what Block (1983) calls the *photographic fallacy*. Rather, the point is that when a holistic representation *is* committal about two such features, then, because of HOLISM, degradation of one feature would be expected to disrupt the other feature as well.

There is strong evidence in favor of independent forgetting. Fougnie and Alvarez (2011) used a continuous-report task in which participants first viewed an array of five triangles of various colors and orientations (Fig. 2).[10] Then, in the test period, participants performed a color response followed by an orientation response, or vice versa. An array of squares appeared where the triangles were, four of which were outlines and one of which was completely white. The location of the white square indicated which of the previewed triangles was to be reported on. Participants used a mouse to click locations on a color wheel (for the color response) and a black wheel centered around the test item (for the orientation response) in which location on the wheel indicated the direction that the triangle was pointing.

[10] Half of participants were instructed to engage in articulatory suppression by repeating 'the' three times per second. There was no significant difference in task performance between those participants and the ones who were not instructed to use articulatory suppression, suggesting that the task engaged VSTM rather than verbal working memory.

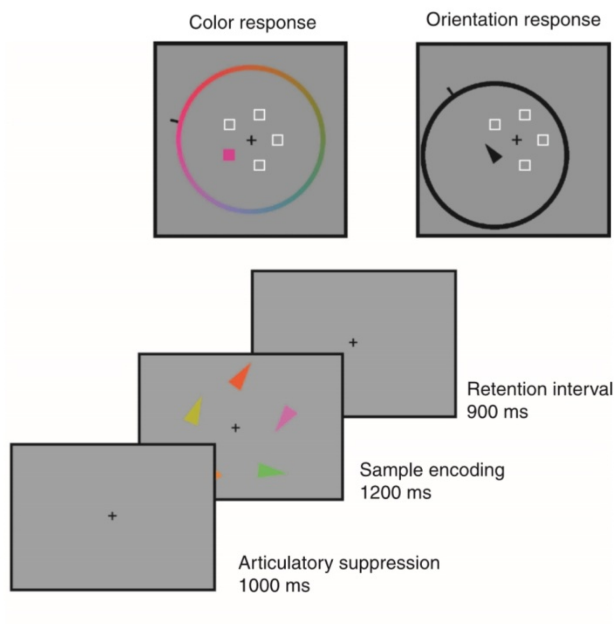

Figure 2. Experimental paradigm from Fougnie and Alvarez (2011)

Fougnie and Alvarez examined "guess trials," in which the subject's degree of error in indicating an object's color or orientation was more than three standard deviations away from the target value. The mixed-features model would predict that participants should either remember both features together or forget both features together. However, Fougnie and Alvarez found that color guess trials and orientation guess trials were only weakly correlated. Specifically, subjects retained information about color in more than 40% of orientation-guess trials, and retained information about orientation in more than 30% of color-guess trials (see also Bays et al. 2011).[11]

These results show that participants often fail to store information about the orientation of an object in VSTM while storing information about its color, and vice versa. Representations of low-level perceptible features must therefore be able to come apart and be stored separately, exactly as the mixed-features model denies. Object files must have discrete representations for distinct low-level perceptible features.

[11] Fougnie et al. (2013) found a similar result while also demonstrating a same-object benefit. One condition involved five triangles with different colors and orientations while another involved ten objects: five colored circles (i.e., with no orientation) and five black triangles with different orientations. Participants showed better performance for both color responses and orientation responses in the five-object condition than the ten-object condition, suggesting that they were forming object files that encoded features of each object and facilitated a same-object benefit (we will consider the phenomenon of same-object benefits in more detail below). However, participants also showed independent forgetting, since again color guess trials and orientation guess trials were weakly correlated. The observation of independent forgetting and same-object benefits in the same experiment suggests that even though features were independently forgotten, they were nevertheless encoded in object files.

Note also that these results make a stronger case against the iconic view than familiar demonstrations of illusory conjunctions (e.g., Treisman & Schmidt 1982; Vul & Rich 2010). The latter work indicates that individual features from separate objects can be misbound in perception, but does not settle whether the resulting representations—the *outputs* of binding—encode features holistically within a single symbol (an icon), or instead utilize discrete symbols for each feature. On the other hand, the independent forgetting evidence shows that representations of individual features can be peeled away from representations of the other features *after* perceptual binding is complete.

The proponent of the mixed-features model may make a final retreat and claim that each individual feature is represented iconically. However, it is unclear how the color of a triangle might be represented iconically without specifying its shape and orientation. Indeed, Kosslyn et al. stress that the contents of icons must be in some sense "picturable" (2006, 11), and there is no remotely intuitive sense in which one can picture the color of a triangle separately from its spatial features.

However, it is perhaps more plausible that an icon may represent spatial features without surface color (e.g., Kosslyn et al. 2006, 41). (Think, for instance, of a simple line drawing.) As such, one might claim that object files represent spatial properties via an icon, but use distinct representations for color. Consistent with this view, spatial properties do sometimes appear to bundle together. Fougnie and Alvarez (2011) did not find the same sort of independent forgetting for height and width of rectangles. Nonetheless, this result may simply be due to the fact that specifying an object's shape requires specifying values along both of these dimensions (similar to how specifying an object's color requires specifying values along dimensions of hue and saturation), and does not demand explanation in terms of iconic format.

A mixed view of this sort, of course, faces the puzzle of spelling out how representations of color and spatial features compose with indexes, given that they are in completely different formats. Moreover, it is clear that standard icons such as Figure 1 represent shape and color together, and that they do not do so by combining separate vehicles together. As such, an iconic model clearly does not *predict* independent encoding of color and spatial features, even if a view can be tailored to accommodate this fact while retaining certain iconic elements.

There is, in any case, suggestive evidence that even spatial properties are stored independently of one another in VSTM. Hardman and Cowan (2015) showed subjects arrays of rectangular bars, and compared change detection performance for four kinds of features: color (red or green), length (long or short), orientation (vertical or horizontal), and the presence or absence of a black "gap" in the middle of a bar.[12] Note that even an achromatic, purely spatial icon should arguably encode the three latter features, since they are characterized solely by an object's visible contours.[13] Hardman and Cohen, however, found that when cued to encode just a single feature during a sample array of six objects, subjects could remember an average of 4.0 colors, 3.0 gaps, 1.7 lengths, and 2.3 orientations. Thus, it is plausible that there are differences in storage capacity even within the class of spatial features, indicating that an object file may encode one spatial feature (e.g., gap presence or orientation) without encoding another (e.g., length).

3.3. Object files and propositional format.

The empirical evidence seems to indicate that object files do not represent in an iconic format. We think the evidence supports the hypothesis that object files represent in a propositional format. Like propositional representations, object files consist of distinct representations for individuals and their separate properties. Furthermore, like propositional representations, object files arrange those constituents into a larger, accuracy-evaluable structure. Camp notes that the signature property of propositional format is that "some sort of functional relation among syntactic constituents maps onto some sort of logical or metaphysical relation among the semantic values of those constituents" (2007, 157). In that case, object files seem canonically propositional.[14]

There may be more fine-grained distinctions between formats according to which object files are not propositional in the same sense as propositional attitudes (though see Quilty-Dunn ms). We view the claim that object files are propositional as a working hypothesis. Nonetheless, it is no small argument for this hypothesis is that it predicts *all* of the data surveyed above, while the competing approaches that we have considered do not. A propositional model allows individuals and features to be represented via separate representations. This separability of constituents enables propositional representations to peel apart separate features in encoding and storage, to peel feature-representations apart from representations of individuals, and to encode and store discrete symbols that stand for abstract categories such as *fish* independently of low-level features. The experimental

[12] These were the same features examined in Luck and Vogel's (1997) seminal study.

[13] Note that even if a spatial icon can be noncommittal about the precise *color* of the gap in a rectangle, it should nevertheless signal the *presence* of such a gap.

[14] In criticizing the claim that object files are iconic, we leave open the possibility that some of the constituents of propositional object files represent via syntactic magnitudes (sometimes described as *analog*). We do not think the empirical evidence demands this view, but we have not ruled it out here (cf. Carey 2009, 143–147).

literature on object files seems to implicate precisely this sort of apparatus. Moreover, in Section 5 we will outline a computational theory of how object files store propositional representations in the same way that propositional beliefs may be stored in long-term memory.

There may, for all we know, be non-propositional formats that accommodate the data as well or even better than a propositional one. We view this question as entirely empirical. Given the empirical success of the propositional view, however, we will assume it in what follows.

§4. The Architecture of Object Files: A Multiple-Slots Model

4.1. Independent memory stores

The *architecture* of a psychological system consists, roughly, in the stable functional organization of that system. More precisely, a system's architecture consists in those aspects of its functional organization that remain fixed despite changes in the information that the system processes and stores (e.g., Pylyshyn 1984). For example, if there is a genuine boundary between two psychological subsystems that prevents one subsystem from accessing information stored by another (regardless of the specific information contained in either system), then this a feature of architecture (Fodor 1983; Firestone & Scholl forthcoming).

Object files are memory mechanisms. They are representations in VSTM that store information about the properties of objects and carry that information forward in time (Gallistel & King 2009). A central question about the architecture of any memory mechanism concerns whether it is structured into independent information stores, and if so, what differentiates those stores from one another (Baddeley 2012).

Memory stores can differ most obviously in terms of either their capacity or their duration of information retention. Some memory stores can hold more information than others, and some can hold information for longer than others. Differences in capacity or retention for two kinds of information can thus make it plausible that the two kinds of information are associated with separate memory stores (think, for instance, of the classic distinction between working memory and long-term memory). However, there are other ways besides these to motivate independent memory stores.

Critically for present purposes, one way to motivate independent memory stores is to examine how two kinds of information *compete* for storage (e.g., Klauer & Zhao 2004). For example, suppose that a person is able to remember at most 4 items of type X, and at most 4 items of type Y. If she is *also* able to remember 4 X-items and 4 Y-items concurrently, then this provides strong evidence that there are separate memory stores dedicated to X and Y. In this case, there is substantial *within-category* competition, but no *across-category* competition. While each category happens to exhibit a capacity limit of 4 items, these are really independent limits that constrain different memory stores.

In what follows, we consider the question of how object files organize information from separate feature categories (e.g., color, shape, orientation, and size).[15] We have already argued that separate features (e.g., color and shape) are encoded by distinct *symbols*. However, this does not yet settle the question of *storage*. Are multiple features bound together in an object file, and if so, how are they bound?

We can distinguish three views about the manner in which VSTM stores collections of features. According to what we will call the *single-slot view*, all the features of an object, regardless of category, are entered into a single memory store (see Fig. 3). On this proposal, the capacity limit on parallel feature storage—if there is one—applies to an object file as a whole. Of course, there are a number of ways that one might understand such capacity limits. Perhaps, for example, there is a limit on the raw number of feature values that can be simultaneously entered into the slot. Alternatively, there may be a limit on the amount of *information* that can be simultaneously entered. On this view, more complex features may take up more file space than simpler features, due to their higher information load. Similarly, more determinate features may impose a higher information load than less determinate features (e.g., red_{37} versus *red*).

According to what we will call the *multiple-slots view*, on the other hand, features from different categories are entered into their own category-specific slots *within* a file (see Fig. 4). In addition to countenancing a limit on the number of object files that can be concurrently stored, this view allows that separate feature categories may have their own object-specific capacity limits. For example, we may have a limit on the number of colors or texture features that can be simultaneously attributed to a single object. Again, such limits could be understood either in terms of a raw number of feature values or in terms of information load.

The single-slot view is arguably neutral on representational format, since it neither requires nor precludes holistic feature encoding. The multiple-slots view, however, requires separate symbols for separate features, since these features must be stored in distinct memory

[15] These categories are sometimes referred to in the VSTM literature as separate feature "dimensions." However, we avoid that label here, since most are in fact multidimensional (and, in the case of shape, highly multidimensional—see Pizlo 2008).

slots. The multiple-slots view therefore suggests a non-iconic, propositional account of object-file format.

We can distinguish both of these approaches from a *pure feature-based view* (e.g., Bays et al. 2011). The pure feature-based view agrees with the multiple-slots view in holding that separate feature categories have separate memory stores in VSTM, but also holds that these memory stores are *not* consolidated into object files (see Fig. 5). On this view, we have separate capacity limits on the number of colors, shapes, sizes, and orientations that we can simultaneously store at a time, but these limits are insensitive to whether pairs of features belong to the same object or to different objects.

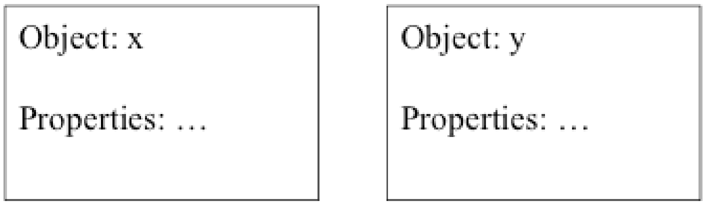

Figure 3. VSTM architecture on the single-slot view. "x" and "y" are directly referential visual indexes. (After Pylyshyn (2007: 38).)

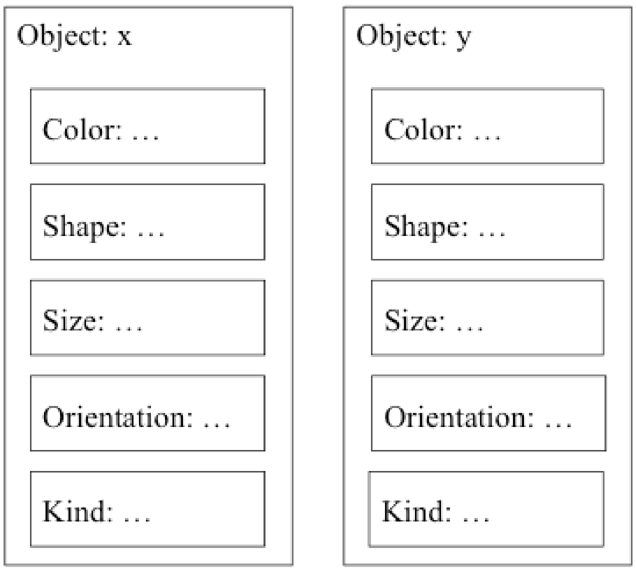

Figure 4. VSTM architecture on the multiple-slots view

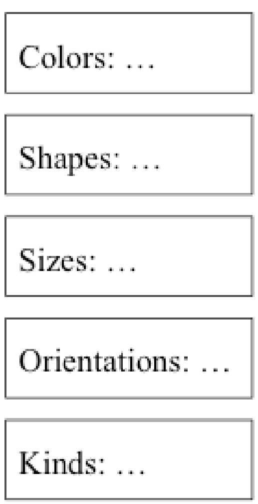

Figure 5. VSTM architecture on the pure feature-based view

The pure feature-based view can accept the evidence discussed in section 2 that object files are formed in on-line perception, but denies that they are stored in VSTM. Furthermore, the pure feature-based view is consistent with many of the central results from Luck and Vogel (1997). Recall, for instance, that Luck and Vogel found that subjects could recall 4 colors and 4 orientations simultaneously just as accurately as they could store 4 colors alone. While these results can be explained in terms of object files, a pure feature-based theorist could argue that VSTM maintains parallel memory stores for color and orientation, and that is why the two features do not compete for storage.

Nonetheless, there is compelling evidence against the pure feature-based model, and in favor of the view that VSTM stores information in an object-based fashion. Most critically, it is significantly easier to store a pair of features when they are bound into the same object than when they are distributed across separate objects. For example, Olson and Jiang (2002) found that participants were significantly more accurate in recalling 4 colors and 4 orientations when the features were bound together into 4 objects (4 oriented rectangles) than when they were distributed across 8 objects (4 colored squares and 4 oriented line segments). Similarly, in a continuous-report paradigm, Fougnie et al. (2010) found a significantly higher rate of random guess responses when participants were asked to remember color and orientation features distributed across six objects than when the features were bound together into three objects. *Prima facie*, these results fit poorly with the pure feature-based view, because the model fails to predict any advantage of object-based organization. As such, we will put the pure feature-based view aside in what follows.

In what follows we will frequently use the language of "slots" to characterize VSTM. However, a prominent dispute in the recent VSTM literature concerns whether VSTM capacity is better characterized by a fixed number of discrete slots or, instead, by the flexible distribution of resources (for discussion, see Fukuda et al. 2010; Ma et al. 2014; Gross & Flombaum forthcoming). A pure version of the slot model would hold that perceivers can store information about at most 4 objects at a time, and that this number of slots is the sole limitation on VSTM performance. Once an object is in VSTM, one can store multiple features of the object without any loss of precision or accuracy. A pure version of the flexible-resource model, on the other hand, would hold that perceivers can store information about any number of objects in parallel, but that the precision with which each feature is represented drops as more objects and features are stored.[16] Certain tenets of this view have

[16] It is admittedly unclear how best to understand memory "resources," but there are some sensible options. For instance, suppose that the VSTM representation of a feature results from averaging the noisy estimates of a number of separate neurons or neural populations. In this case, as the number of estimates increases, the precision of the resulting VSTM representation will also increase (assuming that sources of noise are

compelling empirical support. For example, Fougnie et al. (2010) have found that memory precision for an individual feature is reduced when multiple features of an object must be stored. Likewise, Bays and Husain (2008) found that memory precision for both location and orientation decreased with larger sample arrays.

While we cannot address the issue in detail here, we believe that the most plausible position will combine elements of both the slot-based approach and the resource-based approach (see Alvarez & Cavanagh 2004; Barton et al. 2009; Suchow et al. 2014). More specifically, we believe that VSTM organizes information in an object-based manner, but that there are limits on the precision with which information about an object can be represented, and these limits are characterized well by flexible-resource models. Thus, while VSTM architecture contains discrete files for separate objects, the amount of "space" within a file (i.e., the amount of information the file can hold) is determined by the proportion of resources allocated to it. As a result, memory precision will plausibly be reduced when multiple features of an object are stored, and reduced further when features of multiple objects are stored. Moreover, the nature of resource division across files may vary depending on the demands imposed by the current context (Bays & Husain 2008). For example, certain objects may be stored with greater precision than others when they are more task-relevant (or perhaps due to random variability in resource distribution—see van den Berg et al. 2012), and the same will likely be true for different features within an object.

Finally, while slot-based models typically claim that there is a fixed upper limit of 3-4 object files, we do not commit ourselves to this claim in what follows. We are concerned here with the *functional architecture* of object files, not with how many object files there are. Thus, although we call our position the "multiple-slots" view, we do not accept a pure slot-based characterization of VSTM capacity limits. Rather, we believe that the view defended here can be fruitfully supplemented with insights from the flexible-resources approach.

4.2. Within-category versus across-category conjunctions in VSTM

If the single-slot view is correct, then the VSTM capacity limit for a particular object should be sensitive simply to the total load imposed by the features to be stored for that object. As such, if we assume that the information load of each feature value is roughly the same, the view fails to predict any difference between storing a pair of colors for an object and storing a color and an orientation for that object. (Below we'll consider the possibility that features of

---

independent across circuits). On this conception, then, we might think of the "resources" dedicated to a feature in terms of the number of separate estimates (neurons or populations) available for determining the VSTM representation of that feature (see also Bays et al. 2009, 8).

different categories impose different information loads.) The multiple-slots view, on the other hand, does predict such a difference. For if two features of an object belong to the same category, then they should both draw on the capacity of the same feature slot, while features from different categories should not. For similar reasons, the pure feature-based view would also predict greater difficulty for within-category than across-category conjunctions.

Recall that Luck and Vogel (1997) proposed that VSTM capacity is limited *only* by the number of objects stored in parallel, allowing unlimited storage of features for each object without loss of precision or accuracy. This is an extreme view that is not widely held in the current literature, and the authors themselves no longer endorse it (Zhang & Luck 2008; Fukuda et al. 2010). However, Luck and Vogel performed an additional experiment to test whether there is a difference between within-category and across-category conjunctions in VSTM. The objects in this experiment were squares composed of a center and surround portion differing in color, and subjects were asked to remember both colors for each object. Critically, Luck and Vogel failed to find a significant difference between change detection performance (as a function of set size) in this "color-color" conjunction condition and the other conditions they examined. This result supports a single-slot model, since it apparently suggests that there is no cost to encoding multiple features of the same category.

Note, however, that Luck and Vogel reported a null effect. They *failed to reject* the hypothesis that there is no difference between capacity for within-category conjunctions and across-category conjunctions (e.g., color-orientation or color-shape). While null effects can be informative, especially when they are found across multiple experiments, this particular null effect has not been replicated. In fact, to the best of our knowledge, every subsequent experiment that has examined within-category conjunctions has failed to reproduce Luck and Vogel's results (Olson & Jiang 2002; Wheeler & Treisman 2002; Xu 2002; Delvenne & Bruyer 2004; Parra et al. 2009; Luria & Vogel 2011).

Wheeler & Treisman (2002) found that, in terms of sample array size, change detection performance for bicolored items was approximately half that for unicolored items. That is, we can remember the colors of 3 bicolored objects about as accurately as we can remember the colors of 6 unicolored objects. This was true for a variety of different stimulus types (see Fig. 6 below). Olson and Jiang (2002) reached a similar conclusion, although they found that with high-saturation stimuli, performance was slightly better for 3 bicolored items than for 6 unicolored items (though still significantly impaired relative to 3 unicolored items). Luria and Vogel (2011) also reported a small advantage for 2 bicolored items relative to 4 unicolored items. On the other hand, one of Delvenne and Bruyer's (2004) experiments actually revealed *worse* performance for 2 bicolored items relative to 4 unicolored items. In any case, despite some subtle differences, these studies provide converging evidence that within-category conjunctions are costly.

In contrast, several subsequent change detection experiments have confirmed Luck and Vogel's finding that across-category conjunctions can be encoded at little cost beyond encoding the individual features of the conjunction. For instance, Olson and Jiang (2002) and Fougnie et al. (2010) replicated this finding for color-orientation conjunctions, while Riggs et al. (2011) found comparable results with 7- and 10-year old children. Moreover, Delvenne and Bruyer (2004) found the same pattern for shape-texture conjunctions.[17]

On balance, then, the evidence indicates that features from the same category compete with one another to a much greater degree than features from different categories, even when the features are integrated into the same object. This finding is hard to explain on the single-slot view, but is predicted by the multiple-slots view, since the latter holds that separate feature categories have separate slots within an object file.

However, there are some responses available to the defender of a single-slot position. A first response (anticipated above) would be to claim that a pair of colors is more difficult to encode in an object file than, say, a color and a shape because colors have a higher information load than other features. However, we know of no evidence to suggest that this is the case. If anything, recent work suggests that colors are *easier* to store than other kinds of features. Recall that Hardman and Cowan (2015) found that VSTM capacity was significantly higher for colors than for other features (orientation, length, and gap presence). Similarly, Cowan et al. (2013) found that VSTM capacity was significantly higher for colors than shapes. Thus, it is unlikely that the selective deficits observed for color-color conjunction conditions are due to a systematically higher information load for color.

A second response to the studies cited above in favor of within-category competition would be to claim that early Gestalt processes parsed the center-surround stimuli into

[17] Two qualifications are in order. First, some studies have found that participants are less accurate when asked to memorize across-category conjunctions than when asked to memorize single features (e.g., Oberauer & Eichenberger 2013; Hardman & Cowan 2015). These studies have primarily tested across-category conjunctions of more than two features. We do not quarrel with the results of these experiments, however, because our claim here is only that there is a *greater* cost associated with within-category conjunctions than across-category conjunctions. To the best of our knowledge, every experiment (with the exception of Luck and Vogel (1997)) that has tested this hypothesis has confirmed it. Second, while Fougnie et al. (2010) replicated Luck and Vogel's results in the change detection paradigm, they also found results using a continuous-report paradigm that indicate that there is some cost to encoding across-category conjunctions. Specifically, memory *precision* is reduced in across-category conditions relative to single-feature conditions. However, our view is consistent with this result. Separate feature slots within an object file will plausibly draw on the same stock of memory resources. As such, storing multiple features of an object may result in losses of precision for each feature.

separate objects on the basis of color discontinuity, and later object file deployment was forced to respect this parsing (see Parra et al. (2009) for this interpretation). This kind of parsing would, for instance, be delivered by Palmer and Rock's (1994) principle of uniform connectedness, which states that regions that are homogeneous with respect to some surface property (e.g., color or texture) tend to be parsed as individual units. If these stimuli were indeed assigned object files in accordance with uniform connectedness, for example, then the findings could be accommodated by a single-slot view. For if separate object files needed to be assigned to the center and surround portions of the stimuli, then the display with 3 bicolored objects would require 6 separate object files. This is plausibly beyond the capacity limit of VSTM. This proposal could thus explain impairments in the color-color conjunction condition without positing separate feature slots *within* object files.

If bicolored stimuli are obligatorily parsed into two objects on the basis of uniform connectedness, then changes in how the colors of such stimuli are arranged (i.e., changes that result in more or fewer uniformly connected regions) should produce changes in the number of objects into which they are parsed, and so in the number of object files assigned to a stimulus. However, there is evidence that this is not the case. Wheeler and Treisman (2002) used seven different arrangements for their bicolored stimuli (Fig. 6). Some of these, like the stimuli in the Luck and Vogel (1997) study, involved just two uniformly colored regions. As is clear from Figure 4, however, they used various other stimuli that seem less likely to lead to a Gestalt of precisely two items. In some of these (such as conditions 5, 6, and 7) the same-colored regions of the object were spatially discontinuous, leading to more than two uniformly connected regions. The Gestalt-based interpretation predicts that the bicolored objects with spatially continuous same-colored regions should more likely be parsed into two objects, while bicolored objects with spatially discontinuous same-colored regions should more likely be parsed into more than 2 objects (e.g., four objects in conditions 5 and 6). However, Wheeler and Treisman found no significant differences across these various stimuli despite wide variation in their patterns of color discontinuity. If the Gestalt-based interpretation were correct, however, we would expect to see differences across these conditions, since the stimuli should have demanded different numbers of object files. The Gestalt-based interpretation is therefore highly doubtful in this experiment.[18]

[18] There is, furthermore, suggestive neurophysiological evidence that bicolored center-surround stimuli are treated as individual objects by VSTM. Contralateral delay activity (CDA) is a signal detectable in EEG recording that has been found to be a reliable marker of the number of objects currently held in VSTM (Vogel & Machizawa 2004; Ikkai et al. 2010). Luria and Vogel (2011) recently compared CDA while subjects stored either bicolored or unicolored objects. They found large and stable differences in CDA between memory arrays of 1 bicolored object versus 2 unicolored objects, but only a small difference in CDA (which disappeared later in the retention interval) between memory arrays of 2 bicolored objects versus 2 unicolored objects.

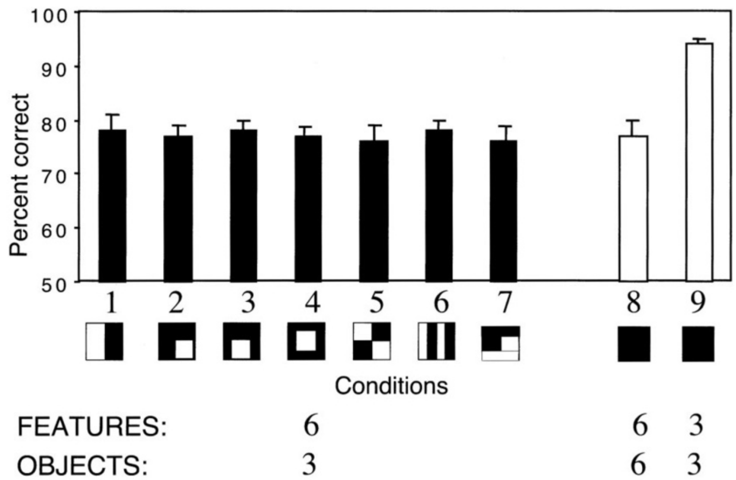

Figure 6. From Wheeler and Treisman (2002)

Nonetheless, we do not claim here that perceivers *never* assign two object files to a bicolored stimulus. We claim only that this strategy is not obligatory in all cases, and that in certain experiments there is good reason to think that it was not adopted. We suspect that perceivers are most likely to assign two object files if the differently colored regions are perceptually segmented as distinct parts of the object (e.g., Xu 2002). This is, however, perfectly compatible with the multiple-slots view.

We conclude that the single-slot model is incorrect. Features within a category compete with one another to a much greater degree than features from different categories, but the single-slot model lacks the architectural flexibility to implement this constraint. The multiple-slots view, on the other hand, offers a unified explanation of all the data discussed in this section. The view explains why features of the same category should compete with one another to a greater degree than features of different categories, and it also explains why features of different categories are more easily encoded and stored when they belong to the same object than when they belong to different objects.

§5. Multiple Slots and Indirect Addressing

In this section we show how the multiple-slots view can be supplemented with an indirect addressing model of information storage and retrieval. While we are not committed to any particular information-processing model, getting clearer about the computational details will enable us to highlight some computational virtues of the multiple-slots view. It also allows us to develop our view of the connection between visual indexes and the feature representations stored in object files. This provides a more thorough characterization of the way object files implement propositional format.

Modern computers possess a random-access memory. This means that information stored anywhere in the system's memory can be accessed using a *pointer* symbol directed to the *address* of that information—i.e., to its location in memory.[19] Addresses are to be understood functionally. The address of an item in memory is, roughly, its place in an ordered sequence to which the system's read-write operations are sensitive.[20] Adjacent addresses do not need to be implemented by physically adjacent regions of the system's hardware.

Thus, suppose a system stores information about the current date, time, and temperature. Each of these variables is associated with an address. The *content* of an address—in the ordinary, pretheoretical sense of what the address holds or contains—is a symbol structure encoding the value of the variable:

Address	**Content**
100,100	6/27/2016
100,101	2:35 p.m.
100,102	86° F

For a procedure to retrieve the current date, it probes memory using a pointer that corresponds to the relevant address. We'll symbolize this pointer with "100,100". (In what follows, whenever an address is underlined, we are denoting a pointer to the address, rather than the address itself.) It is unimportant for present purposes just how this operation works. We stress only that pointers are symbols that, when entered into read/write operations, grant access to a particular location in memory. Critically, although the pointer plays the computational role of calling information from a particular address, it can be semantically interpreted as representing the *variable* associated with that address.[21] Thus, the symbol

[19] For a detailed discussion of random-access memory architectures, including the theoretical framework that follows, see Gallistel and King (2009: ch.9; see also Gallistel 2008). Random-access memory marks an important respect in which the architecture of a modern computer differs from a Turing machine. A Turing machine accesses information *serially*. For it to retrieve information from memory, the machine head must traverse every cell between the one it is currently reading and the one that encodes the relevant information.

[20] "Read" operations retrieve information from memory in order to use it in computation, while "write" operations enter new information into memory to be retrieved at a later time.

[21] This may lead to some confusion, because pointers are sometimes described as having internal referents. On this construal, the referent of a pointer is simply the address that it "points" to. However, this is not obligatory. A pointer may represent an external variable, while playing the functional role of granting access to an internal memory location (see Gallistel 2008). This point is especially important to keep in mind in the case of visual indexes, which (we contend) represent objects in the world despite "pointing" (i.e., enabling computational access) to memory locations. Note, furthermore, that pointers in our sense are not equivalent to what Eliasmith and colleagues have called "semantic pointers" (e.g., Blouw et al. 2015). On the semantic pointer model, pointers are compressed versions of more detailed lower-level representations (analogous to JPEG files). In

100,100 represents the variable *current date*, while the symbol 6/27/2016 represents one of the many values that the variable may adopt.

This example is a case of *direct addressing*. Direct addressing occurs when a system accesses the value of a variable by using a pointer to the address of that value. However, in most cases a computer does not call the value of a variable directly via a pointer to the address of that value, but instead via pointers to addresses that themselves contain further pointers. This is known as *indirect addressing*. Indirect addressing confers substantial computational advantages, to be described below.

Suppose that our system needs to encode the dates, times, and temperatures from multiple locations at once, and that it also needs to remember the bindings between dates, times, temperatures, and locations. One way to do this is as follows (cf. Gallistel & King 2009: 161):

Address	**Content**
100,000	100,100
100,100	6/27/2016
100,101	2:35 p.m.
100,102	86° F
200,000	200,100
200,100	6/28/2016
200,101	3:35 a.m.
200,102	68° F

The system is organized in such a way that all information stored at addresses 100,100-199,999 pertains to one location (New York), while all information stored at addresses 200,100-299,999 pertains to a different location (Beijing). The way the system retrieves the value of a specific variable associated with New York (e.g., the current date) is via *pointer arithmetic*. This is an operation performed on a pointer, which returns another pointer. This works as follows. When the system probes its memory, it sends a pair of signals. One is a pointer to a particular address—call it *x*—and the other is a numeral—say, 1. Pointer arithmetic is an operation performed on the pointer stored at address *x* as a result of these signals. The operation then returns a different pointer, which in turn causes the system to access a second memory location. Thus, suppose that our system sends the following memory probe: (100,000, 1). This probe causes the content of address 100,000—namely, the pointer 100,100—to be retrieved. Pointer arithmetic is an operation performed on the

contrast, the only constraint we place on pointers is that they enable access to other representations stored in memory. They need not be compressed counterparts of the representations whose access they enable.

symbols 100,100 and 1, which (we may suppose) returns the symbol 100,101.[22] This symbol denotes the variable *current time in New York*. It in turn causes the system to retrieve the symbol stored at address 100,101, which encodes the value of that variable.

The pointers 100,000 and 200,000 (which we may assume are stored elsewhere in the system) have the computational role of enabling access to the encoded properties of New York and Beijing, respectively. We can usefully interpret these symbols as *referring* to New York and Beijing. Note that it is arbitrary in this example which pointers we store at the addresses 100,000 and 200,000 (the addresses that these pointers "point to"), since there is no natural ordering among date, time, and temperature. The important feature is simply that the memory is arranged to enable access to any of a location's characteristics by means of pointer arithmetic.

Indirect addressing is a very useful information-processing tool. Suppose, for example, that in the system described above there is an operation that can be applied to numerous locations, and in each case needs to take into account the date, time, and temperature in that location. If the system had to find each location's date, time, and temperature directly, the operation would need access to explicitly stored pointers to each of these values. With five locations, this means fifteen explicitly stored pointers. In contrast, if the system utilizes indirect addressing, then it can store just five pointers along with a general rule specifying how the information associated with each pointer is organized (e.g., date in the first slot, time in the second, and temperature in the third). For regardless of which location is probed, the operation "knows" that the time in that location can be accessed through a probe consisting of the pointer to that location along with the numeral 2. Obviously, as the memory arrays associated with each location grow larger, the advantages of indirect addressing become even more pronounced.

The indirect-addressing architecture also offers a natural implementation of the multiple-slots model of object files. Suppose that, for each object, we store a record of color, shape, size, and orientation. Here is a model of such a memory containing two object files:

Address	**Content**
100,000	100,100
100,100	is red
100,101	is square
100,102	is 3 inches

---

[22] Pointer arithmetic is thus *formally* akin to addition. However, we do not call it addition, because we take the symbol 100,101 to stand for the variable *current time in New York*, rather than a number. Strictly speaking, then, we can take pointer arithmetic to be a function mapping pairs of variables and numbers to further variables.

100,103	is 60° from vertical
200,000	200,100
200,100	is blue
200,101	is triangular
200,102	is 1 inch
200,103	is 25° from vertical

To retrieve, say, the shape of the first object, we would send the following probe: (100,000, 1). Through pointer arithmetic, the symbol 100,100 is transformed into 100,101, which then serves as a new probe to memory, returning the symbol is square. The memory is organized so that information pertaining to the first object is differentiated from information pertaining to the second object, but is also organized into separate slots on the basis of feature category.

These slots (addresses) associated with an object file are assumed to be limited in capacity. As mentioned above, this limitation could take two forms. First, it is possible that each address stores a maximum number of feature values—perhaps just one. Another possibility is that each address is limited in terms of the total information that it can hold. On the second option, it may be possible to represent an object as instantiating multiple values for a single variable (e.g., color or texture), although we know of no compelling evidence that this occurs. Thus, although we have no strong views on this issue, we are inclined toward the position that object files are limited to representing only a single feature value per category.[23]

The symbols 100,000 and 200,000 serve to enable access to the first and second object's features, respectively. On our view, these *are* visual indexes. A visual index, we propose, plays the computational role of a pointer—it allows retrieval of an object's features from memory. However, it does not represent any of those features, and in that sense it is akin to a directly referential expression in natural language. Furthermore, we propose that to occurrently attribute a feature (e.g., red) to an object denoted by an index just is to call the symbol for that feature from visual short-term memory using the index as a pointer.

To appreciate the computational advantages of the indirect addressing architecture we espouse, let's consider an implementation of the single-slot model of object files. On this

[23] One caveat is in order. It may be possible to represent an object as having multiple *compatible* values for a given feature category. For instance, it may be possible for an object file to represent an object as both square and quadrilateral (Green 2015). Unfortunately, we know of no experiments that directly address whether object files concurrently represent features at multiple levels of abstraction.

view, all of an object's features are entered into a single memory store, rather than being arranged into separate category-specific slots. The following organization accomplishes this:

Address	**Content**
100,000	is red; is square; is 3 inches; is 60° from vertical
200,000	is blue; is triangular; is 1 inch; is 25° from vertical

On this view, the memory slots associated with the two objects store a collection of feature representations. Again, this information could be retrieved via the pointers 100,000 and 200,000, which can be construed as visual indexes. However, the system now accesses the stored features of an object via *direct* addressing, rather than indirect addressing, as on the multiple-slots model.

Notice the clear drawbacks of this approach. When the system reads or writes from object file memory, it is forced to access an address that contains all of the object's features, rather than merely its color or shape. This means that whenever the system needs to retrieve information about, say, the color of the object, it is forced to also call information about all of its other properties. Furthermore, there is no clear mechanism in this architecture for the system to update *only* information about an object's color. Finally, it is hard to see how, within this framework, one could implement even relatively trivial operations across objects, such as determining whether one object is larger than another. This is because the system does not organize information in a way that makes explicit whether a given stored feature value concerns an object's size or, instead, its shape or orientation. Thus, in addition to its superior fit with the evidence, we suggest that the multiple-slots view offers a highly efficient model of how information in object files can be encoded, retrieved, and used in computational operations.

The indirect-addressing architecture also provides a more detailed account of the way in which object files instantiate a propositional format. The tokening of a (non-quantified) propositional representation minimally requires (i) a symbol representing an individual, (ii) a symbol representing a property, and (iii) a syntactic operation of concatenating those two symbols such that the property expressed by the latter is predicated of the individual picked out by the former. The symbol 100,000 constitutes (i), a symbol like is square constitutes (ii), and the pointer arithmetic performed as a result of the probe (100,000, 1) implements (iii). The result of this operation is a token propositional representation with the content <o_1 is square>, where o_1 is the referent of 100,000 (in the relevant context). The feature symbols in other slots, which are not occurrently yoked into a propositional structure with the visual index, still stand in a predicative, propositional relation to the index in virtue of their being poised to figure in the construction of a token propositional structure.

Compare the case in which one acquires the belief that Obama is the president and then stores the propositional structure Obama is the president. *Explicit* propositional storage would consist in storing a token of that very structure at some address in memory. *Implicit* propositional storage, on the other hand, would consist in maintaining a certain capacity to construct a token of that structure out of the explicitly stored concepts Obama and is the president. It is crucial to note that the *mere* capacity to token the structure, such as is facilitated by the mere co-presence of the concepts Obama and is the president in a single mind, is not sufficient for implicit storage of the propositional structure. If it were, then every possible propositional structure that can be composed out of the set of concepts in a mind would count as implicitly stored in that mind. Making sense of implicit storage of propositional structures thus requires a principled way of distinguishing it from mere co-presence of concepts in memory. We propose that this distinction can be drawn in terms of the way that information about the referent of a concept is organized and retrieved from memory.

One popular characterization of conceptual organization appeals to mental files (e.g., Fodor 2008; Recanati 2012; Murez et al. 2015). Fodor (2008, 92–100), for instance, argues that concepts should be understood as constituting the addresses of files, the access of which facilitates the access of stored information pertaining to the referent of the concept. A natural way to think about this file architecture is that the propositional structure Obama is the president is explicitly stored in a way that is accessible via the pointer Obama. Another possibility, however, is that concepts are stored without being constituents of token propositional structures. A system might store the concept is the president in a way that makes it poised to be connected with Obama into the propositional structure Obama is the president. In this case, accessing Obama can facilitate the access of is the president. Critically, this computational relationship implements predication. The propositional structure Obama is the president is tokened when is the president is accessed via a pointer that refers to Obama, and it is implicitly stored in virtue of the accessibility of the predicate concept via this pointer.

The file architecture, understood in terms of indirect addressing, enables a concept to be accessible via a memory pointer such that any concept in the file is poised to be predicatively connected with the pointer in question. A propositional structure o is F is implicitly stored by virtue of the way that is F can be retrieved from memory (viz., via a pointer that refers to o), and it is tokened when such a retrieval operation takes place. The architecture thus allows for a substantive characterization of propositional storage that does not require the explicit storage of complete propositional structures. We do not claim, however, that all propositional representations in the mind are realized in this way. We certainly do not intend to deny that there are token propositional structures explicitly stored in long-term memory. Nonetheless, the coherence of the proposal shows how predication and propositional format can arise out of addressing without explicit storage of token propositional structures.

§6. Conclusion

Many authors have recently appealed to object files in characterizing our basic capacities to perceive, think about, and re-identify objects. Nonetheless, very few have attempted to clarify the precise nature of these representations. In the current paper, we have offered accounts of the representational *format* of object files (i.e., how they are syntactically structured) and the *architecture* of object files (i.e., how they organize information). We claim (i) that object files are propositional representations consisting of discrete symbols standing for individuals and features, (ii) that feature representations are organized into separate, category-specific slots within an object file, and (iii) that representations of individuals (i.e., indexes) function computationally as pointers that enable access to these category-specific slots.

Developing the consequences of this model for philosophical views about singular thought, the perception–cognition border, and the emergence of objective representation is beyond the scope of this paper. The model provided should, however, enable a richer characterization of these topics. Object files are a key developmental and phylogenetic locus of propositional structure and objective reference. A detailed, scientifically informed model of object files is therefore crucial in achieving a deeper understanding of core features of the human mind.

References

Alvarez, G.A., & Cavanagh, P. (2004). The capacity of visual short-term memory is set both by visual information load and by number of objects. *Psychological Science* 15, 106–111.

Alvarez, G.A., & Franconeri, S. (2007). How many objects can you track? Evidence for a resource-limited attentive tracking mechanism. *Journal of Vision* 7(13), 1–10.

Baddeley, A. (2012). Working memory: Theories, models, and controversies. *Annual Review of Psychology* 63, 1–29.

Barton, B., Ester, E.F., & Awh, E. (2009). Discrete resource allocation in visual working memory. *Journal of Experimental Psychology: Human Perception and Performance* 35(5), 1359–1367.

Bays, P.M., Catalao, R.F.G., & Husain, M. (2009). The precision of visual working memory is set by allocation of a shared resource. *Journal of Vision* 9(10):7, 1-11.

Bays, P.M., & Husain, M. (2008). Dynamic shifts of limited working memory resources in human vision. *Science* 321, 851–854.

Bays, P.M., Wu, E.Y., & Husain, M. (2011). Storage and binding of object features in visual working memory. *Neuropsychologia* 49, 1622–1631.

Block, N. (1983). The photographic fallacy in the debate about mental imagery. *Nous* 17(4), 651-661.

Block, N. (2011). Perceptual consciousness overflows cognitive access. *Trends in Cognitive Sciences* 15(12), 567–575.

Blouw, P., Solodkin, E., Thagard, P., & Eliasmith, C. (2015). Concepts as semantic pointers: A framework and computational model. *Cognitive Science* 40, 1128-1162.

Briscoe, R. (Forthcoming). Depiction, pictorial experience, and vision science. *Philosophical Topics* 44(2).

Burge, T. (2010). *Origins of Objectivity*. Oxford: OUP.

Burge, T. (2014a). Reply to Block: Adaptation and the upper border of perception. *Philosophy and Phenomenological Research* 89(3), 573–583.

Burge, T. (2014b). Reply to Rescorla and Peacocke: Perceptual content in light of perceptual constancies and biological constraints. *Philosophy and Phenomenological Research* 88(2), 485-501.

Camp, E. (2007). Thinking with maps. *Philosophical Perspectives* 21, 145–182.

Carey, S. (2009). *The Origin of Concepts*. Oxford: OUP.

Cowan, N., Blume, C.L., & Saults, J.S. (2013). Attention to attributes and objects in working memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 39(3), 731–747.

Delvenne, J.F., & Bruyer, R. (2004). Does visual short-term memory store bound features? *Visual Cognition* 11, 1–27.

Devitt, M., & Sterelny, K. (1999). *Language and Reality: An Introduction to the Philosophy of Language*. Second edition. Cambridge, MA: MIT Press.

Drew, T., Horowitz, T.S., Wolfe, J.M., & Vogel, E.K. (2011). Delineating the neural signatures of tracking spatial position and working memory during attentive tracking. *The Journal of Neuroscience : The Official Journal of the Society for Neuroscience* 31(2), 659–668.

Echeverri, S. (2016). Indexing the world? Visual tracking, modularity, and the perception-cognition interface. *The British Journal for the Philosophy of Science*. 67(1), 215-245.

Firestone, C. and Scholl, B.J. (Forthcoming). Cognition does not affect perception: Evaluating the evidence for "top-down" effects. *Behavioral and Brain Sciences*.

Fodor, J.A. (1983). *The Modularity of Mind*. Cambridge, MA: MIT Press.

Fodor, J.A. (2007). The revenge of the given. In B. McLaughlin and J. Cohen (eds.), *Contemporary Debates in Philosophy of Mind* (Oxford: Blackwell), 105–116.

Fodor, J.A., & Pylyshyn, Z.W. (2015). *Minds without Meanings: An Essay on the Content of Concepts*. Cambridge, MA: MIT Press.

Fougnie, D., & Alvarez, G.A. (2011). Object features fail independently in visual working memory: Evidence for a probabilistic feature–store model. *Journal of Vision* 11(12), 1–12.

Fougnie, D., Asplund, C.L., & Marois, R. (2010). What are the units of storage in visual working memory? *Journal of Vision* 10(12), 1–11.

Fougnie, D., Cormiea, S.M., & Alvarez, G.A. (2013). Object-based benefits without object-based representations. *Journal of Experimental Psychology: General* 142(3), 621–626.

Fougnie, D., & Marois, R. (2006). Distinct capacity limits for attention and working memory evidence from attentive tracking and visual working memory paradigms. *Psychological Science* 17(6), 526–534.

Fougnie, D., & Marois, R. (2009). Attentive tracking disrupts feature binding in visual working memory. *Visual Cognition* 17, 48–66.

Franconeri, S., Alvarez, G., & Cavanagh, P. (2013). Flexible cognitive resources: Competitive content maps for attention and memory. *Trends in Cognitive Science* 17(3), 134–141.

Franconeri, S., Jonathan, S., & Scimeca, J.M. (2010). Tracking multiple objects is limited only by spatial interference, not speed, time, or capacity. *Psychological Science* 21, 920–925.

Fukuda, K., Awh, E., & Vogel, E.K. (2010). Discrete capacity limits in visual working memory. *Current Opinion in Neurobiology* 20(2), 177-182.

Gallistel, C.R. (2008). Learning and representation. In R. Menzel, & J. Byrne (eds.), *Learning and Memory: A Comprehensive Reference*. New York: Elsevier.

Gallistel, C.R., & King, A. (2009). *Memory and the Computational Brain: Why Cognitive Science Will Transform Neuroscience*. Malden, MA: Wiley-Blackwell.

Gordon, R.D. and Irwin, D.E. (1996). What's in an object file? Evidence from priming studies. *Perception and Psychophysics* 58(8), 1260–1277.

Gordon, R.D. and Irwin, D.E. (2000). The role of physical and conceptual properties in preserving object continuity. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 26(1),136–150.

Green, E.J. (2015). A layered view of shape perception. *The British Journal for the Philosophy of Science*. DOI: 10.1093/bjps/axv042.

Green, E.J. (forthcoming). Attentive visual reference. *Mind & Language*.

Greenberg, G. (2013). Beyond resemblance. *Philosophical Review* 122(2), 215–287.

Gross, S., & Flombaum, J. (forthcoming). Does perceptual consciousness overflow cognitive access? The challenge from probabilistic, hierarchical processes. *Mind & Language*.

Hardman, K., & Cowan, N. (2015). Remembering complex objects in visual working memory: Do capacity limits restrict objects or features? *Journal of Experimental Psychology: Learning, Memory, and Cognition* 41(2), 325–347.

Henderson, J. M. (1994). Two representational systems in dynamic visual identification. *Journal of Experimental Psychology: General* 123(4), 410–426.

Holcombe, A.O., & Chen, W.Y. (2012). Exhausting attentional tracking resources with a single fast-moving object. *Cognition* 123(2), 218–228.

Hollingworth, A., & Franconeri, S.L. (2009). Object correspondence across brief occlusion is established on the basis of both spatiotemporal and surface feature cues. *Cognition* 113, 150–166.

Hollingworth, A., Richard, A.M., & Luck, S.J. (2008). Understanding the function of visual short-term memory: Transsaccadic memory, object correspondence, and gaze correction. *Journal of Experimental Psychology: General* 137, 163–181.

Hopkins, R. (1998). *Picture, Image and Experience: A Philosophical Inquiry*. Cambridge: Cambridge University Press.

Ikkai, A., McCollough, A.W., & Vogel, E.K. (2010). Contralateral delay activity provides a neural measure of the number of representations in visual working memory. *Journal of Neurophysiology* 103, 1963–1968.

Jeshion, R. (2010). Singular thought: Acquaintance, semantic instrumentalism, and cognitivism. In R. Jeshion (ed.), *New Essays on Singular Thought* (Oxford: Oxford University Press), 105–141.

Johnson–Laird, P. (2006). *How We Reason*. Oxford: OUP.

Jordan, K.E., Clark, K., & Mitroff, S.R. (2010). See an object, hear an object file: Object correspondence transcends sensory modality. *Visual Cognition* 18(4), 492–503.

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

Kaplan, D. (1989). Demonstratives. In J. Almog, J. Perry, & H. Wettstein (eds.), *Themes From Kaplan*. Oxford: Oxford University Press.

Kimchi, R., & Pirkner, Y. (2014). Surface feature congruency effects in the object-reviewing paradigm are dependent on task memory demands. *Psychonomic Bulletin & Review* 21(4), 1019–1025.

Klauer, K.C., & Zhao, Z. (2004). Double dissociations in visual and spatial short-term memory. *Journal of Experimental Psychology: General* 133(3), 355–381.

Kosslyn, S. M. (1980). *Image and Mind*. Cambridge, MA: HUP.

Kosslyn, S.M., Thompson, W.L., and Ganis, G. (2006). *The Case for Mental Imagery*. Oxford: OUP.

Kripke, S. (1972). *Naming and Necessity*. Cambridge, MA: Harvard University Press.

Kulvicki, J. (2015). Analog representation and the parts principle. *Review of Philosophy and Psychology* 6, 165–180.

Luck, S.J., & Vogel, E.K. (1997). The capacity of visual working memory for features and conjunctions. *Nature* 309, 279–281.

Luria, R., & Vogel, E.K. (2011). Shape and color conjunction stimuli are represented as bound objects in visual working memory. *Neuropsychologia* 49(6), 1632–1639.

Ma, W.J., Husain, M., & Bays, P.M. (2014). Changing concepts of working memory. *Nature Neuroscience* 17(3), 347–356.

Mitroff, S.R., & Alvarez, G.A. (2007). Space and time, not surface features, guide object persistence. *Psychonomic Bulletin & Review* 14, 1199–1204.

Murez, M., & Recanati, F. (2016). Mental files: An introduction. *Review of Philosophy and Psychology* 7(2), 265–281.

Murez, M. Smortchkova, J., & Strickland, B. (2015). The mental files theory of singular thought: A psychological perspective. <ijn_01228067>

Neisser, U. (1967). *Cognitive Psychology*. Englewood Cliffs, NJ: Prentice-Hall.

Noles, N.S., Scholl, B.J., & Mitroff, S.R. (2005). The persistence of object file representations. *Perception & Psychophysics* 67(2), 324–334.

Oberauer, K., & Eichenberger, S. (2013). Visual working memory declines when more features must be remembered for each object. *Memory & Cognition* 41, 1212–1227.

O'Callaghan, C. (2014). Intermodal binding awareness. In D. Bennett, & C. Hill (eds.), *Sensory Integration and the Unity of Consciousness* (Cambridge, MA: MIT Press), 73-104.

Olson, I.R., & Jiang, Y. (2002). Is visual short-term memory object based? Rejection of the "strong–object" hypothesis. *Perception & Psychophysics* 64, 1055–1067.

Palmer, S.E., & Rock, I. (1994). Rethinking perceptual organization: The role of uniform connectedness. *Psychonomic Bulletin & Review* 1, 29–55.

Parra, M.A., Abrahams, S., Logie, R., & Della Sala, S. (2009). Age and binding within-dimension features in visual short term memory. *Neuroscience Letters* 449, 1–5.

Phillips, W.A. (1974). On the distinction between sensory storage and short-term visual memory. *Perception & Psychophysics* 16, 283–290.

Pizlo, Z. (2008). *3D Shape: Its Unique Place in Visual Perception*. Cambridge, MA: MIT Press.

Pylyshyn, Z.W. (1984). *Computation and Cognition: Toward a Foundation for Cognitive Science*. Cambridge, MA: MIT Press.

Pylyshyn, Z.W. (2003). *Seeing and Visualizing: It's Not What You Think*. Cambridge, MA: MIT Press.

Pylyshyn, Z.W. (2007). *Things and Places: How the Mind Connects with the World*. Cambridge, MA: MIT Press.

Pylyshyn, Z.W., & Storm, R. (1988). Tracking multiple independent targets: Evidence for a parallel tracking mechanism. *Spatial Vision* 3(3), 179–197.

Quilty-Dunn, J. (2016). Iconicity and the format of perception. *Journal of Consciousness Studies* 23(3-4), 255–263.

Quilty-Dunn, J. (ms). *Syntax and Semantics of Perceptual Representation*. Unpublished dissertation.

Quine, W. V. (1960). *Word and Object*. Cambridge, MA: MIT Press.

Recanati, F. (2012). *Mental Files*. Oxford: OUP.

Riggs, K.J., Simpson, A., & Potts, T. (2011). The development of visual short-term memory for multifeature items during middle childhood. *Journal of Experimental Child Psychology* 108(4), 802–809.

Scholl, B.J., and Pylyshyn, Z.W. (1999). Tracking multiple items through occlusion: Clues to visual objecthood. *Cognitive Psychology* 38, 259–290.

Sellars, W. (1956). *Empiricism and the Philosophy of Mind*. New edition, 1997. Cambridge, MA: Harvard University Press.

Sligte, I.G., Scholte, H.S., & Lamme, V.A. (2008). Are there multiple visual short-term memory stores? *PLoS One* 3(2), e1699. doi: 10.1371/journal.pone.0001699.

Sperling, G. (1960). The information available in brief visual presentations. *Psychological Monographs: General and Applied* 74(11), 1–29.

Strawson, P.F. (1963). *Individuals*. New York: Anchor Books.

Suchow, J.W., Fougnie, D., Brady, T.F., & Alvarez, G.A. (2014). Terms of the debate on the format and structure of visual memory. *Attention, Perception, and Psychophysics* 76(7), 2071-2079.

Treisman, A., & Schmidt, H. (1982). Illusory conjunctions in the perception of objects. *Cognitive Psychology* 14, 107-141.

Tye, M. (1991). *The Imagery Debate*. Cambridge, MA: MIT Press.

van den Berg, R., Shin, H., Chou, W.-C., George, R., & Ma, W.J. (2012). Variability in encoding precision accounts for visual short-term memory limitations. *PNAS* 109(22), 8780–8785.

vanMarle, K., & Scholl, B.J. (2003). Attentive tracking of objects versus substances. *Psychological Science* 14, 498–504.

Vogel, E.K., Woodman, G.F., & Luck, S.J. (2001). Storage of features, conjunctions, and objects in visual working memory. *Journal of Experimental Psychology: Human Perception and Performance* 27(1), 92–114.

Vogel, E.K., & Machizawa, M.G. (2004). Neural activity predicts individual differences in visual working memory capacity. *Nature* 428, 748–751.

Vul, E., & Rich, A.N. (2010). Independent sampling of features enables conscious perception of bound objects. *Psychological Science* 21(8), 1168-1175.

Wheeler, M.E., & Treisman, A.M. (2002). Binding in short-term visual memory. *Journal of Experimental Psychology: General* 131, 48–64.

Wilken, P., & Ma, W.J. (2004). A detection theory account of change detection. *Journal of Vision* 4, 1120–1135.

Xu, Y. (2002). Limitations of object-based feature encoding in visual short-term memory. *Journal of Experimental Psychology: Human Perception and Performance* 28, 458–468.

Zhang, W., & Luck, S.J. (2008). Discrete fixed-resolution representations in visual working memory. *Nature* 453, 233–235.

Zhang, H., Xuan, Y., Fu, X., & Pylyshyn, Z.W. (2010). Do objects in working memory compete with objects in perception? *Visual Cognition* 18(4), 617–640.

Zhou, K., Luo, H., Zhou, T., Zhuo, Y., & Chen, L. (2010). Topological change disturbs object continuity in attentive tracking. *Proceedings of the National Academy of Sciences* 107(50), 21920–21924.