Replicating an Influential Experiment Investigating Visual Working Memory Capacity

Marvin Theiß
Student ID: [redacted]

Programming for Behavioral and Neurosciences
MBB–MA–REF–1 (MA–HMA–12)

August 29, 2023

Working Memory Capacity

The experiment I chose to replicate is a visual working memory capacity experiment by Vogel and Machizawa (2004). This experiment is part of #EEGManyLabs, a large-scale effort by an international collaboration of researchers "to directly test the replicability of key findings from 20 of these [most influential and continually cited] studies [in the field of EEG research] in teams of at least three independent laboratories" (Pavlov et al., 2021).

In their study, Vogel and Machizawa (2004) provide evidence for "lateralized [brain] activity in humans that reflects the encoding and maintenance of items in visual memory". Moreover, their study strongly suggests that the amplitude of this neural activity (measured by recording ERPs, see Figure 1) serves as a "strong neurophysiolog-

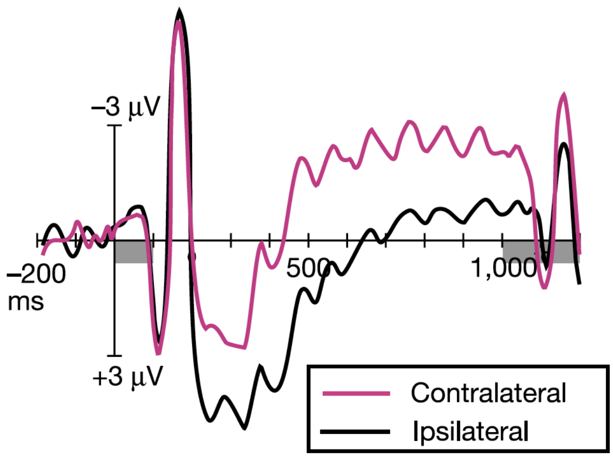

Figure 1. Neural activity in the form of ERP waveforms recorded at the lateral occipital and posterior parietal electrode sites, averaged across twelve participants. Gray boxes indicate presentation of memory and test arrays.

Note. Adapted from "Neural activity predicts individual differences in visual working memory capacity," by E. K. Vogel and M. G. Machizawa, 2004, *Nature, 428*(6984), p. 748 (https://doi.org/10.1038/nature02447). Copyright 2004 by Nature Publishing Group.

ical predictor of an individual's [visual working memory] capacity" (Vogel & Machizawa, 2004).

Experimental setup. To arrive at these results, Vogel and Machizawa (2004) had their subjects complete the following visual working memory task: On each trial, participants were briefly shown a bilateral array of colored squares. Subjects had to memorize the items from only one of the two hemifields, which was indicated with an arrow prior to the onset of the memory array. After a retention interval spanning 900 ms, "memory was tested (...) with the presentation of a test array that was either identical to the memory array or differed by one colour" (Vogel & Machizawa, 2004). To indicate whether the two arrays were identical or different, subjects pressed one of two buttons.

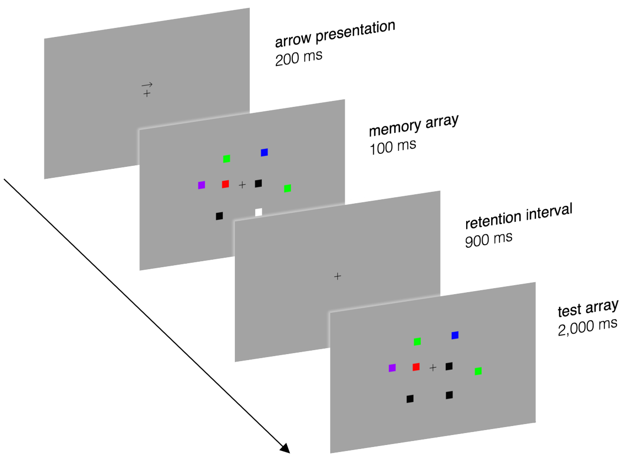

Figure 2. Schematic flow of a single trial. Size of stimuli has been increased for demonstration purposes.

Results. The experiments conducted by Vogel and Machizawa (2004) suggest that there is lateralized activity (contralateral to the memorized hemifield) reflecting "the maintenance of successful representations in visual memory". This activity comes in the form of a "large negative-going voltage (. ...) focused primarily over the posterior parietal and lateral occipital electrode sites" with a peak latency of approximately 200 ms (ERPs were time-locked to the onset of the memory array) (Vogel & Machizawa, 2004).

The mere presence of the described signal during the experiment does not imply that this contralateral delay activity reflects "the maintenance of successful representations in visual memory" since visual memory performance is driven by numerous processes (Vogel & Machizawa, 2004). How then did the authors arrive at their conclusion?

1. By repeating the experiment with different array sizes (one, two, three, and four), Vogel and Machizawa (2004) showed that the "amplitude [of the delay activity] was highly sensitive to the number of items in the memory array". Additionally, this amplitude was significantly larger for correct trials compared to incorrect trials.

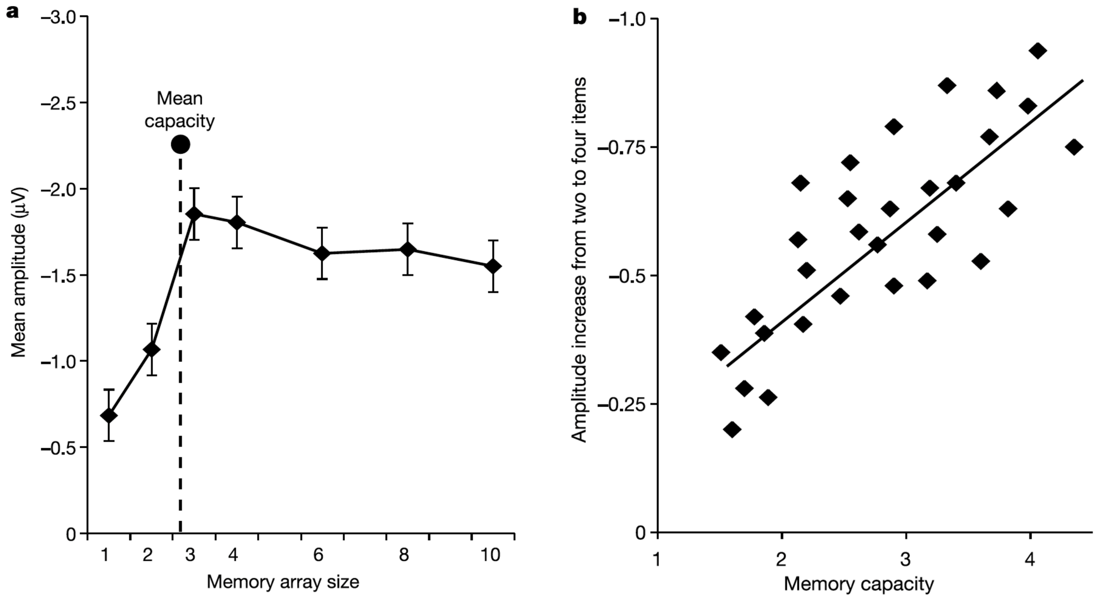

Figure 3. Results obtained by Vogel and Machizawa (2004). **a** Mean amplitude increases for larger arrays (roughly) until the participants' mean capacity is reached before leveling off. Vertical error bars indicate 95% confidence intervals. **b** An individual participant's working memory capacity strongly correlates ($r = 0.78$, $p < 0.0001$) with the amplitude increase from two to four squares per hemifield.

Note. Adapted from "Neural activity predicts individual differences in visual working memory capacity," by E. K. Vogel and M. G. Machizawa, 2004, *Nature, 428*(6984), p. 749 (https://doi.org/10.1038/nature02447). Copyright 2004 by Nature Publishing Group.

2. In a third experiment, Vogel and Machizawa (2004) had participants complete the same task again; this time with array sizes of two, four, and six squares. While there was a significant increase in amplitude going from two to four squares per array, there was no additional increase when increasing the array size further from four to six squares. Given that a typical individual's memory capacity is right around three to four items, this implies that the delay activity's amplitude only increases for larger stimulus arrays below or at an individual's capacity and does not increase any further for supra-capacity arrays.

3. This finding was further reinforced by yet another experiment in which participants completed the same task again, this time the arrays consisted of two, four, eight, and ten squares each. The finding was identical: While there was a significant increase in amplitude going from two to four squares per hemifield, no such increase was observed when moving from four to either eight or ten squares per array (Vogel & Machizawa, 2004).

4. The amplitude of the contralateral delay activity initially increases monotonically as the number of squares per array increases. However, it quickly reaches its maximum at three items per hemifield and then starts to drop off. Notably, the visual working memory capacity averaged across participants[1] was "2.8 items, which is approximately when the memory delay activity reaches a limit" suggesting that these two measures are strongly related (Vogel & Machizawa, 2004).

5. Building on this finding, further analysis of the data showed that the increase in amplitude between arrays of two and four items was "very strongly correlated" with the individual subject's memory capacity, "with low-capacity subjects producing very little amplitude increase and high-capacity subjects showing larger amplitude increases" (Vogel & Machizawa, 2004).

Implementation in MATLAB

The replication of the experiment by Vogel and Machizawa (2004) is implemented in MATLAB (The MathWorks Inc., 2023), using the Psychophysics Toolbox extensions (Brainard, 1997; Kleiner et al., 2007; Pelli, 1997). The source code consists of several helper functions and two main scripts that can be used to collect and analyze data. I will first address the latter and then briefly go over the additional MATLAB functions.

Collecting & analyzing data. The source code for data collection and analysis is split into two separate scripts. The `WorkingMemoryCapacity.m` script can be used to run the experiment by Vogel and Machizawa (2004) and collect data. Once sufficient data has been collected, the script `analzyeData.m` can be executed to analyze this data. Progressing in logical order, I will first explain the `WorkingMemoryCapacity.m` script and then move on to the `analyzeData.m` script. Since both scripts are heavily commented, I will aim to provide only a broad overview.

Data collection. To enhance readability, the `WorkingMemoryCapacity.m` script is divided into separate blocks of code. I will briefly describe each of these sections separately in chronological order.

[1] Visual working memory capacity was computed using the formula $K = S \times (H - F)$, where S indicates the number of items in the array, H is the hit rate, and F denotes the false alarm rate. This formula was developed by Cowan (2001) who had improved on an earlier version proposed by Pashler (1988).

Configuration of experiment. There are several parameters that can be changed to control the course of the experiment (e.g., number of trials, number of items in a single array, etc.). These can be found here.

Keyboard settings. This is where the different response keys are assigned to the structure array `key`.

Instruction/error messages. Instructions will be given to the participant prior to the experiment. There are also several errors that could potentially be thrown during the experiment that require custom error messages. All of these text blocks are defined here and assigned to the `Msg` structure array.

Stimuli setup & preparation. The colors of the squares and their possible positions on screen are defined and computed here. I follow the setup used by Vogel and Machizawa (2004), i.e., the squares are $0.65° \times 0.65°$ in size and "were presented within two $4° \times 7.3°$ rectangular regions that were centred 3° to the left and right of a central fixation cross on a grey background (...) with the constraint that the distance between squares within a hemifield was at least 2° (centre to centre)". I use the `visualAngleToSize` function to convert these measurements from degrees of visual angle to the number of pixels on the screen.

Positioning and size of fixation cross & arrow(s). Here, I compute the position and the size of both, the central fixation cross and the arrow indicating which array (left or right) to memorize.

Trial structure & setup. This is where the `trials` table is constructed, which contains all the information needed to run the experiment. The experiment is set up so that half of the trials require participants to remember the array in the left hemifield. In the other half of the trials, participants have to remember the array in the right hemifield. Similarly, on half of the trials one square changes color, while on the other half of the trials all squares remain the same. The colors of the squares are randomized, as is the additional color that the *odd* square changes to (if applicable), with the constraint that "a given colour could appear no more than twice within an array" (Vogel & Machizawa, 2004). Once everything is set up, the order of the trials is shuffled.

Record participant data. A dialog box is opened to gather participant data, including their ID, sex, and year of birth. Subsequently, the participant's ID is used to generate a filename to store the results.

Open PTB window. That is pretty much what happens; I open a new Psychtoolbox window and define some basic settings.

Obtain timing information. I use the `Screen('GetFlipInterval')` command to query the duration of a single frame, and I use this value to convert the presentation duration from seconds to the number of frames.

Start of experiment. At the beginning of the experiment, I present general instructions to the participant. Practice trials are also included to allow subjects to familiarize themselves with the task. The number of practice trials can be set in the *Configuration of experiment* section at the top of the script.

Loop over individual trials. Again, this is pretty much all that happens. The structure of a single trial is identical to the setup used by Vogel and Machizawa (2004), which is illustrated in Figure 2. Accurate timing is achieved by storing timestamps at every flip and setting the `when` argument of the `Screen('Flip')` command on subsequent flips.

End of experiment. I wipe the screen and present a thank-you-message. A countdown indicates when the Psychtoolbox window will close.

Save data & shut down. I save the collected data to a CSV file and call the `endExperiment` function, which turns off character listening and closes all open Psychtoolbox windows.

Error handling. If an error occurs during the experiment, I save all the data collected up to that point and then re-throw the error message.

Helper function(s). This is where the `endExperiment` function described earlier is defined.

Data analysis. The `analyzeData.m` script is straightforward. When executed, it reads all valid data files[2] (i.e., only those that contain complete sets of data) from the `data/` directory. For each file, the participant's hit rate and false alarm rate are computed. Next, these two values are used to estimate the participant's working memory capacity using Cowan's formula (2001) and all of this data is appended to a table called `analyzedData`. Finally, once all data has been processed, the `analyzedData` table is sorted by participants' IDs and the number of items prior to being displayed in the command window.

Additional functions. The source code also includes four additional functions

[2]The valid datasets are identified via the `matches` function.

that can be used to collect data via the `WorkingMemoryCapacity.m` script. Two of these functions, `drawArrow` and `drawFixationCross`, do exactly what you would expect them to do. The `configurePsych` function can be used to set up parameters for a new Psychtoolbox session. Finally, the function `visualAngleToSize` helps to convert the size of a viewed object from degrees of visual angle to *mm* based on the distance from which the object is viewed.

Replication of Results

To see if I could replicate the findings[3] by Vogel and Machizawa (2004), I ran the experiment on myself.

Setup. As in the first experiment conducted by Vogel and Machizawa (2004), I completed 240 trials in which arrays consisted of four items each. Presentation durations, appearance, and positioning of stimuli (see Fig. 2) were identical[4] to those used by Vogel and Machizawa (2004). I ran the experiment using the following setup:

- MacBook Pro (2021, M1 Pro Chip)
- 16.2 inch Liquid Retina XDR Display
- macOS Ventura 13.4.1
- MATLAB version 9.14.0 (R2023a)
- Psychtoolbox version 3.0.19

The screen brightness was set to maximum and the automatic brightness adjustment was disabled. Importantly, the screen was the only source of light in an otherwise pitch-black room. Viewing distance was approximately 55 *cm*.

Results. I did not respond in time in one of the 240 trials, resulting in 239 valid trials. I recorded a hit rate of 87.4% and a false alarm rate of 2.5%. Applying the formula by Cowan (2001), this yields an estimated working memory capacity of approximately 3.4 items, which is well within the upper range of capacities observed by Vogel and Machizawa (2004).

[3] Since I do not have access to an EEG lab, the replication attempt refers only to the obtained measures of visual working memory capacity.

[4] In theory, this is true. In practice, however, the timing was slightly off due to the setup (macOS) I was using. The Psychtoolbox extension displayed the following warning after I finished the experiment: `INFO: PTB's Screen('Flip', 10) command seems to have missed the requested stimulus presentation deadline a total of 364 times out of a total of 1528 flips during this session.`

References

Brainard, D. H. (1997). The Psychophysics Toolbox. *Spatial Vision*, *10*(4), 433–436. https://doi.org/https://doi.org/10.1163/156856897X00357

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, *24*(1), 87–114. https://doi.org/10.1017/S0140525X01003922

Kleiner, M., Brainard, D. H., & Pelli, D. G. (2007). What's new in Psychtoolbox-3? *Perception*, *36*(ECVP Abstract Supplement), 14. https://doi.org/10.1177/03010066070360S101

Pashler, H. (1988). Familiarity and visual change detection. *Perception & Psychophysics*, *44*(4), 369–378. https://doi.org/10.3758/BF03210419

Pavlov, Y. G., Adamian, N., Appelhoff, S., Arvaneh, M., Benwell, C. S., Beste, C., Bland, A. R., Bradford, D. E., Bublatzky, F., Busch, N. A., Clayson, P. E., Cruse, D., Czeszumski, A., Dreber, A., Dumas, G., Ehinger, B., Ganis, G., He, X., Hinojosa, J. A., ... Mushtaq, F. (2021). #EEGManyLabs: Investigating the replicability of influential EEG experiments. *Cortex*, *144*, 213–229. https://doi.org/10.1016/j.cortex.2021.03.013

Pelli, D. G. (1997). The VideoToolbox software for visual psychophysics: Transforming numbers into movies. *Spatial Vision*, *10*(4), 437–442. https://doi.org/https://doi.org/10.1163/156856897X00366

The MathWorks Inc. (2023). MATLAB (9.14.0) [Computer software]. The MathWorks Inc. https://www.mathworks.com

Vogel, E. K., & Machizawa, M. G. (2004). Neural activity predicts individual differences in visual working memory capacity. *Nature*, *428*(6984), 748–751. https://doi.org/10.1038/nature02447

Declaration of Authorship and AI Tools

Authorship and academic integrity. I hereby declare that I have written this report independently and have not used any other than the specified resources. The parts of the work that are taken from other works in wording or meaning are indicated by references and sources. This also applies to drawings, sketches, example scripts, and the like.

Use of AI tools. I declare that I have read the "Rules for Tools" for this course. The following list contains all AI tools that were used to create this report, the source code it relates to, or both, along with a brief summary of the purpose of their use.

- The *grammarly.com* Grammar Checker was used to check written text (including this report, source code files, and the README of the GitHub repository that this project is hosted in) for spelling and grammar mistakes.
- No further AI tools were used.

[redacted]

Marvin Theiss