Behavior Research Methods, Instruments, & Computers
1999, 31 (3), 423-428

Object Relocation: A program for setting up, running, and analyzing experiments on memory for object locations

ROY P. C. KESSELS, ALBERT POSTMA, and EDWARD H. F. DE HAAN
Utrecht University, Utrecht, The Netherlands

Object Relocation is a computer program for Windows 95, with which experiments on spatial memory for object locations can be designed, run, and analyzed. Because of its clear graphical user interface, no long and complex command syntax is needed. Basically, a stimulus consists of a frame that contains a chosen number of locations (i.e., the actual spatial layout) to which objects can be assigned. When the experiment is run, these stimuli are presented to the subject for a variable period of time. Subsequently (either with or without a delay), the objects are presented in a row above the frame and have to be relocated to the correct positions. Finally, the raw data can be analyzed efficiently, using various error scores, and an SPSS-ready output file can be produced. Object Relocation is a very flexible program: New objects and positions can easily be added, and various options for presentation and relocation are present.

The mechanisms of spatial memory—that is, memory for information about the location of objects and routes in our environment—can be assessed with the help of various techniques. Newcombe and Russell (1969), for example, described the use of a stylus maze; subjects have to learn a route between two points on a board containing 64 blocks, with the help of feedback and a few rules. Spatial memory can be tested as well by measuring performance on a tapping task, using a number of blocks in an irregular layout, with the Corsi block test (De Renzi, Faglioni, & Previdi, 1977; Milner, 1971). This test also has found wide clinical acceptance (Lezak, 1995). The Misplaced Objects test, in which subjects have to remember the locations of objects in different rooms of a house—using either a small model (Crook, Ferris, & McCarthy, 1979) or a picture on a computer screen in which a 3 × 4 grid indicates the 12 rooms (Crook, Youngjohn, & Larrabee, 1990)—can be used to test delayed recall of object locations.

An example of a more experimentally applied task is the *toy test* (sometimes referred to as the *kim game*), in which a subject has to remember the location of 16 toys placed on a table (Smith & Milner, 1981) or of pictures of objects on a sheet (Smith, Leonard, Crane, & Milner, 1995) and has to relocate them correctly. More recently, Chalfonte and Johnson (1996) displayed objects within a matrix on a computer screen that had to be remembered. PET and fMRI studies typically employ more simple spatial memory tasks—for example, recognition of the position of three briefly presented dots on a screen (Jonides et al., 1993) or recognition of the spatial layout of three different pictures of objects (Köhler, Moscovitch, Winocur, Houle, & McIntosh, 1998).

One of the major problems with the methods used for research on spatial memory is the lack of a standardized procedure for administration and scoring, even for clinical purposes. Although standard testing procedures are available for the Corsi, maze, and Misplaced Objects tests, these tasks are highly inflexible and address only a single aspect of spatial-memory functioning (either serial recall of positions in the Corsi and maze test or forced-choice recall of object locations in the Misplaced Objects test). Recently, several authors have found evidence for the idea that spatial memory is not a unitary mechanism but can be divided into a number of processes. Postma and de Haan (1996; Postma, Izendoorn & de Haan, 1998), for example, suggested at least two processes within spatial memory: (1) a process responsible for the encoding of exact metric locations, independent of the object identities, and (2) a process that is responsible for the assignment of object identities to these locations, using relative relations between objects (e.g., object X is on the left of object Y). Also, there might be a third mechanism for the integration of these two processes. Other authors (Chalfonte & Johnson, 1996; Chalfonte, Verfaellie, Johnson, & Reiss, 1996) also distinguished between the *binding* of location and item information and the *retention* of locations per se, which are comparable with Postma and de Haan's (1996) processing of relations and processing of exact locations.

To overcome the inflexibility of the existing tasks for the assessment of spatial-memory functions, we developed a computer program with which spatial-memory processes for object locations can be easily studied. This was done by creating a 32-bit software package, Object Relocation,

Correspondence concerning this article should be addressed to R. P. C. Kessels, Psychological Laboratory, Department of Psychonomics, Utrecht University, Heidelberglaan 2, 3584 CS Utrecht, the Netherlands (e-mail: r.kessels@fss.uu.nl).

which enables designing experiments, running them, and analyzing the results in a user friendly way. That is, changes in experimental designs and properties of the presented stimuli can be made with the help of a graphical user interface (GUI). This paper will describe the main features of the program.

OBJECT RELOCATION

The basic task procedure in Object Relocation consists of presenting a number of objects within a frame on a computer monitor for a period of time. After the presentation of the stimulus display, the screen is cleared, and the objects reappear above the frame. Subsequently, the subject has to relocate the items to the correct positions, using a computer mouse. Here, three conditions are possible: (1) all the objects are equal (positions only), for the study of positional encoding; (2) all the objects are different, while premarked dots are present in the relocation phase (object-to-position assignment), for the study of the binding of item and position information; and (3) all the objects are different, and no premarked dots are present in the relocation phase (combination), for investigation of a process that might integrate positional and relative encoding. This procedure proved to be effective in demonstrating dissociations between these three processes within spatial memory (Postma & de Haan, 1996; Postma et al., 1998).

In experiments using a computer, the obligatory use of a computer mouse is not always possible, especially when testing elderly subjects or brain-injured patients, who can have problems in properly operating the computer mouse owing to both cognitive and physical limitations (Carr, Woods, & Moore, 1986; Martin & Allan, 1991). Therefore, Object Relocation can also be used in combination with a touch-sensitive screen, which is preferred by many subjects, as compared with using a mouse or a keyboard (Karat, McDonald, & Anderson, 1986). Furthermore, experiments can easily be set up, and experimental changes can be made very rapidly through its GUI. This paper will focus on the main features of this program: setting up new experiments and designing stimulus displays, running the experiments, and analyzing the raw data.

DESIGNING NEW EXPERIMENTS

An experiment in Object Relocation, stored in an ASCII file with the extension STI, contains one or more stimuli, limited only by the system's RAM. A stimulus contains a spatial layout (the coordinates of up to 40 points within a frame stored in an SPL file) and objects, which are Windows bitmap files (the ICO format yields the most satisfying results). The spatial layout files are necessary for, but independent of, the actual stimulus file. This means that the same spatial layout can be assigned to different stimuli that contain different object sets. Before a stimulus is made, a spatial layout must be created. This can be done automatically (at random) by providing the desired number of locations and the minimal distance between the points and between the points and the border, which is especially efficient if many different stimuli have to be made. The locations can also be manually adjusted by clicking and dragging. Since the layout files are independent of the actual stimulus, the relative coordinates within the square always go from 0 to 1, both in *x* and in *y* directions. The actual frame size is stored in the stimulus file. Eventually, the spatial layout ASCII file—with the extension SPL—may look like this:

```
.543
.423
.162
.277
.438
.825.
```

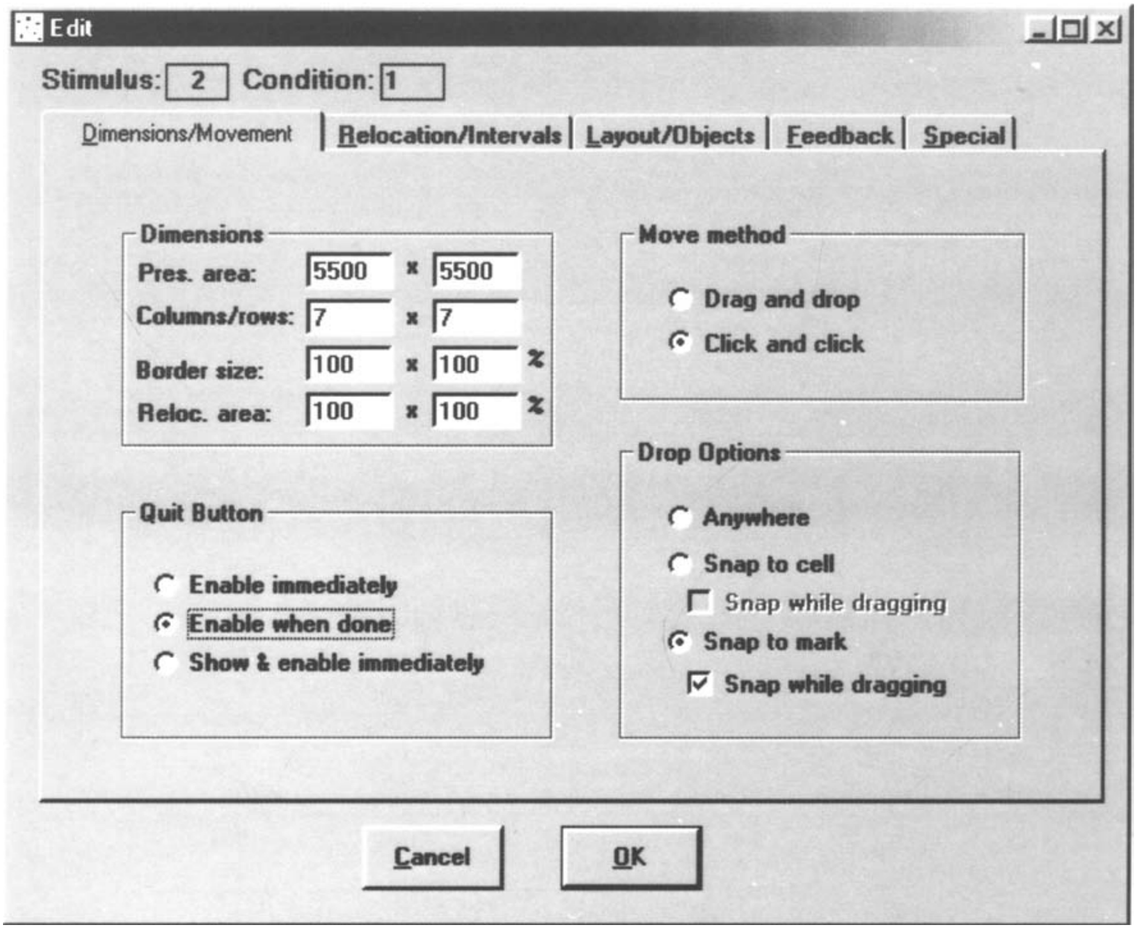
This file indicates that there are three points, with the following coordinates: (0.543, 0.423), (0.162, 0.277), and (0.438, 0.825).

After having made the layout file, a new stimulus can be created. Each stimulus display in an experiment can be identified through a stimulus number. Next, five classes of properties of the stimulus can be adjusted, divided into five sections (Figure 1). In the *Dimensions/Movement* section, features concerning the displayed frame and the methods of moving the objects are present, such as frame size, relocation area within the frame, and the objects' move method (*click and click* or *drag and drop*, since dragging of items is rather inconvenient when using a touch-sensitive screen). Here, it is also possible to select the use of a grid for a more direct comparison of computerized testing with paper-and-pencil tests (Postma, 1996). One can also choose between free relocation and relocation using premarked points.

In the *Relocation/Intervals* section, the duration of the presentation and the length of the delay between the presentation and the relocation phases can be chosen. The presentation and relocation frame can also be presented simultaneously, to test spatial perception and relocation accuracy. The duration between the onsets of the objects can also be changed, which enables serial presentation of objects. The selection of objects to be assigned to a spatial layout can be done in the next section (*Layout/Objects*). Other sections are *Feedback*, where options regarding feedback to the subject can be changed (to see whether subjects can benefit from feedback about their performance), and *Special*, where cuing of external devices through the hardware ports (serial or parallel) is possible and sound alerts can be selected.

Since an experiment will contain more than one stimulus display in most cases, stimuli can be tested individually without having to run the previous stimulus displays. The display is presented, and the objects can be relocated in the same way as in the actual experiment.

The objects that are displayed are Windows icon files that have the extension ICO. Object Relocation can handle all current ICO formats—16 × 16, 32 × 32, or 48 ×



Stimulus: 2 Condition: 1

Dimensions
Pres. area: 5500 × 5500
Columns/rows: 7 × 7
Border size: 100 × 100 %
Reloc. area: 100 × 100 %

Move method
○ Drag and drop
● Click and click

Quit Button
○ Enable immediately
● Enable when done
○ Show & enable immediately

Drop Options
○ Anywhere
○ Snap to cell
□ Snap while dragging
● Snap to mark
☑ Snap while dragging

Figure 1. The properties of a stimulus can be changed easily with the help of the Object Relocation's graphical user interface.

48 pixels—and both 16 and 256 colors. These files can be made easily with the help of an icon editor (many of which can also convert other graphical formats to the ICO format). Moreover, many of these icon files are freely available on the Internet. Since ease of naming (i.e., ease of verbal labeling) may be an important factor in object-location memory (Postma & de Haan, 1996), it is possible to select only pictures that are standardized for naming difficulty, such as the Snodgrass and Vanderwart pictures (Snodgrass & Yuditsky, 1996).

RUNNING EXPERIMENTS

When an experiment is run, the name for the output file (with the extension RES), which stores the raw data for 1 subject, must be entered. Next, a start button appears on the screen. After the button is pressed, the experiment is run, starting with the first stimulus display. The stimulus is displayed; then the square is emptied, and the objects are located above the screen. Now the subject has to attempt to relocate them correctly. The subject then can press the ready button, in order to continue with the next stimulus. Figures 2 and 3 show possible examples of stimuli.

It is possible to use Object Relocation in combination with a touch-sensitive computer monitor: The mouse icon becomes invisible in this case. Subjects can handle the displayed objects directly by clicking and dropping them elsewhere on the screen. This option is especially useful when testing subjects who can have problems with the use of a computer mouse. Of course, using Object Relocation in combination with a normal computer mouse is possible too.

A start button appears before each stimulus in an experiment, and a ready button after each stimulus. No stimulus is present on the screen while the buttons are visible. This enables pauses between stimuli, without having to quit the Object Relocation program. When the experiment is finished, the main menu appears.

ANALYZING THE RESULTS

When an experiment is run, the raw data are stored in an ASCII output file with the extension RES. Stored here

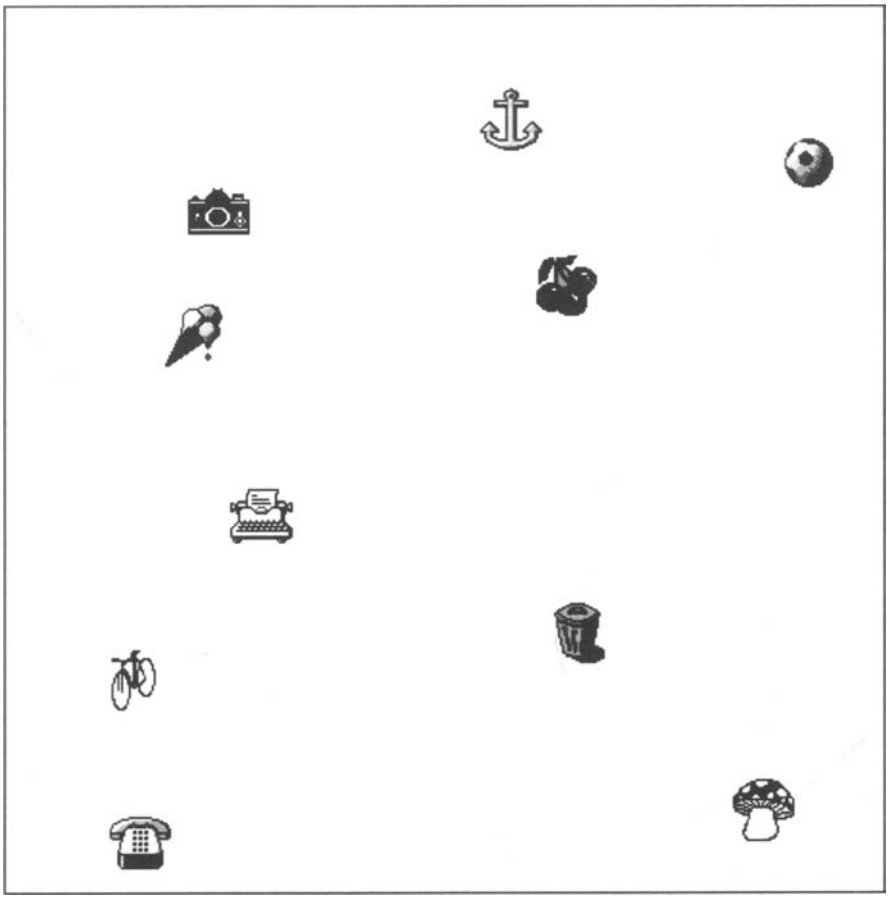


Figure 2. Example of a stimulus displaying 10 everyday objects. For every stimulus, a spatial layout that contains the exact coordinates is linked to a number of bitmaps in ICO format. Stimuli like this one were used by Postma, Izendoorn, and de Haan (1998).

are the presentation and delay durations, the number of columns and rows, and the names of the individual objects, together with their initial position and the relocated position. Also recorded are the reaction times in milliseconds. Although these output files are in ASCII format, they are not very easy to use directly for statistical analyses. Therefore, the output data editor was created, which merges multiple RES files, which contain individual data, and computes error scores, which are stored in a tab-delimited ASCII file that can be imported by statistical software packages (e.g., SPSS). The different error scores that can be computed will be discussed here.

First, the number of correctly relocated objects can be computed. When premarked positions in the relocation phase are used, this is the easiest score to determine: An object is either in the correct or the false location. This error score is referred to as the number of correct objects. However, if the objects have to be relocated in an empty frame, this is not very easy to determine. Even in a correct response, it is likely that there will be a small difference between the original position and the relocated position, since the computer screen is divided into arbitrary screen points. One of the options is to determine whether an object has been relocated within a given critical distance of its original position. If so, the object is regarded as correct; if not, the replacement is incorrect (Sharps & Gollin, 1986). However, this is not a very sensitive error score. The absolute error score is a more concrete error score. The absolute distance between the original and the relocated position is computed (Pezdek, 1983), using the Pythagorean theorem. The actual size in millimeters of the frame on the computer screen can be used to calculate the absolute error score in millimeters.

There are, however, situations in which both the number of correct objects and the absolute error cannot be measured. For example, when the displayed objects are all equal (e.g., for determining positional encoding per se), it becomes difficult to say which relocated object belongs to which original position. Theoretically, it would be possible to assign each relocated object to the original position that is nearest, but the results of this calculation cannot be easily interpreted. Consider, for example, the possibility that every object is consistently misplaced 10 mm in one direction. Now, the configuration of the relocated objects is more important than the position of each individual object. Therefore, the so-called best-fit score can be computed:

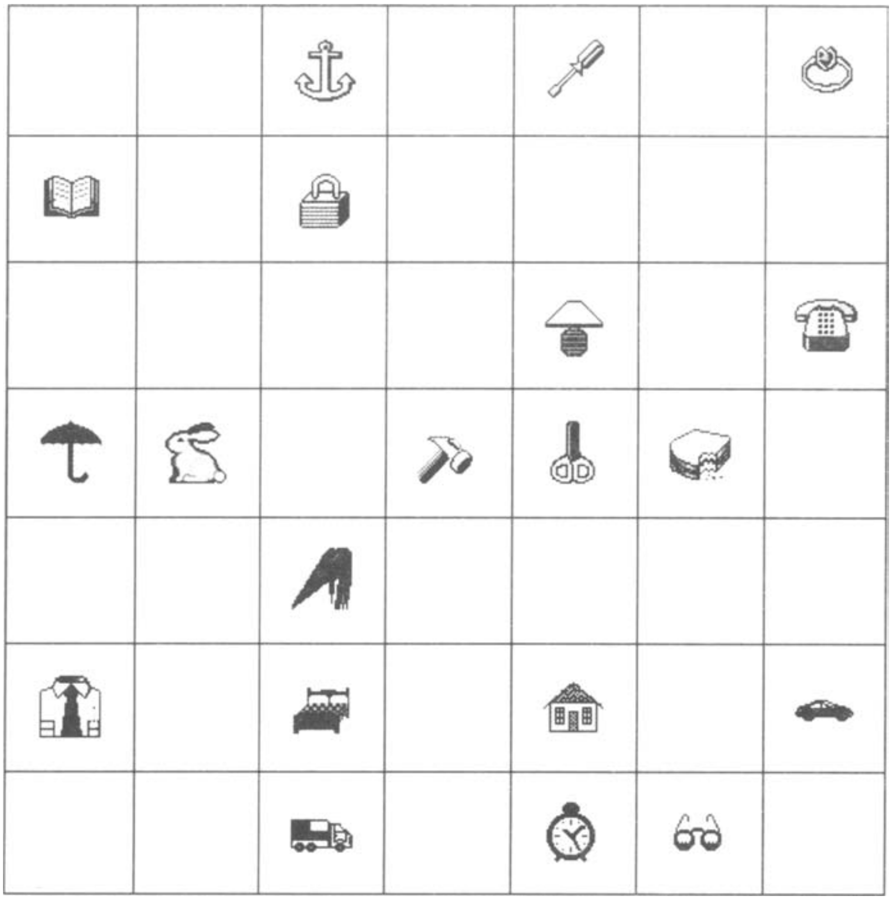


Figure 3. This stimulus consists of a 7 × 7 matrix containing 20 different objects. Chalfonte, Verfaillie, Johnson, and Reiss (1996) used matrices like this in their research on spatial-memory deficits in Korsakoff patients.

All the possible configurations between the original and the relocated positions are computed, and the fit that has the smallest error rate is considered to be the best-fitting configuration (Postma & de Haan, 1996).

The output data editor featured in Object Relocation can compute all these errors for the individual stimuli: the number of correct objects, the absolute error—either using a critical distance or by subtraction—and the best-fit score. The actual frame size in millimeters can be entered, so that the absolute error and the best-fit error are given in millimeters rather than in (arbitrary) screen points. These data are stored in a tab-delimited ASCII file with the extension DAT, which can be imported into many statistical as well as spreadsheet software packages for further analyses.

Currently, the program is being tested under various experimental conditions. In addition, various patient groups are being examined and compared with matched groups of controls. A set of extended norm scores is expected to become available in time.

SUMMARY

In sum, Object Relocation is a software package for designing, running, and analyzing spatial-memory experiments. Its most important feature is its flexibility: Different theoretical paradigms of spatial memory, as well as various processes within spatial memory, can be studied. Also important is the use of a GUI instead of long syntax commands. Various error scores can be computed for analyzing the results, which can be imported by spreadsheet or statistical software packages.

TECHNICAL SPECIFICATIONS AND AVAILABILITY

Object Relocation was written in Microsoft Visual Basic 5. It is a full 32-bit application that runs under Windows 95 or 98 (system requirements, 486 CPU or higher, with a minimum of 8 MB of RAM). The source files were compiled to a native-code executable file that needs the following Microsoft component (OCX) files: COMDLG32, GAUGE32, TABCTL32, COMCTL32, and COMCT232. The DLL file INPOUT32 by Jan Axelson (Lakeview Research) is needed for accessing the hardware ports. A Microsoft help file is available as a manual for on-line help. Other files that are essential are the standard Visual Basic 5 runtime files.

The software package can be obtained from the corresponding author for U.S. $50, and consists of three 3.5-in. disks or one CD–ROM that contain the Windows setup program and all other runtime files that are needed, including a large set of object icons. A sample stimulus is provided for testing purposes. The cost also includes technical support and provision of updates.

REFERENCES

Carr, A. C., Woods, R. T., & Moore, B. J. (1986). Automated cognitive assessment of elderly patients: A comparison of two types of response device. *British Journal of Clinical Psychology*, **25**, 305-306.

Chalfonte, B. L., & Johnson, M. K. (1996). Feature memory and binding in young and older adults. *Memory & Cognition*, **24**, 403-416.

Chalfonte, B. L., Verfaellie, M., Johnson, M. K., & Reiss, L. (1996). Spatial location memory in amnesia: Binding item and location information under incidental and intentional encoding conditions. *Memory*, **4**, 591-614.

Crook, T. [H.], Ferris, S., & McCarthy, M. (1979). The misplaced-objects task: A brief test for memory dysfunction in the aged. *Journal of the American Geriatrics Society*, **27**, 284-287.

Crook, T. H., Youngjohn, J. R., & Larrabee, G. J. (1990). The Misplaced Objects Test: A measure of everyday visual memory. *Journal of Clinical & Experimental Neuropsychology*, **12**, 819-833.

De Renzi, E., Faglioni, P., & Previdi, P. (1977). Spatial memory and hemispheric locus of lesion. *Cortex*, **13**, 424-433.

Jonides, J., Smith, E. E., Koeppe, R. A., Awh, E., Minoshima, S., & Mintun, M. A. (1993). Spatial working memory in humans as revealed by PET. *Nature*, **363**, 623-625.

Karat, J., McDonald, J. E., & Anderson, M. (1986). A comparison of menu selection techniques: Touch panel, mouse and keyboard. *International Journal of Man–Machine Studies*, **25**, 73-88.

Köhler, S., Moscovitch, M., Winocur, G., Houle, S., & McIntosh, R. (1998). Networks of domain-specific and general regions involved in episodic memory for spatial location and object identity. *Neuropsychologia*, **36**, 129-142.

Lezak, M. D. (1995). *Neuropsychological assessment* (3rd ed.). New York: Oxford University Press.

Martin, T. A., & Allan, W. E. (1991). An evaluation of touch-screen input for a HyperCard-based digit-span task. *Behavior Research Methods, Instruments, & Computers*, **23**, 253-255.

Milner, B. (1971). Interhemispheric differences in the localization of psychological processes in man. *British Medical Bulletin*, **27**, 272-277.

Newcombe, F., & Russell, W. R. (1969). Dissociated visual perceptual and spatial deficits of the right hemisphere. *Journal of Neurology, Neurosurgery, & Psychiatry*, **32**, 73-81.

Pezdek, K. (1983). Memory for items and their spatial location by young and elderly adults. *Developmental Psychology*, **19**, 895-900.

Postma, A. (1996). Reconstructing object locations in a 7 × 7 grid. *Psychologische Beiträge*, **38**, 90-100.

Postma, A., & de Haan, E. H. F. (1996). What was where? Memory for object locations. *Quarterly Journal of Experimental Psychology*, **49A**, 178-199.

Postma, A., Izendoorn, R., & de Haan, E. H. F. (1998). Sex differences in object location memory. *Brain & Cognition*, **36**, 334-345.

Sharps, M. J., & Gollin, E. S. (1986). Methods of evaluating performance on spatial memory tasks. *Bulletin of the Psychonomic Society*, **24**, 18-20.

Smith, M. L., Leonard, G., Crane, J., & Milner, B. (1995). The effects of frontal- or temporal-lobe lesions on susceptibility to interference in spatial memory. *Neuropsychologia*, **33**, 275-285.

Smith, M. L., & Milner, B. (1981). The role of the right hippocampus in the recall of spatial location. *Neuropsychologia*, **19**, 781-793.

Snodgrass, J. G., & Yuditsky, T. (1996). Naming times for the Snodgrass and Vanderwart pictures. *Behavior Research Methods, Instruments, & Computers*, **28**, 516-536.

(Manuscript received April 15, 1998;
revision accepted for publication August 20, 1998.)