

Effect of Two-module-docked Spacecraft Configurations on Spatial Orientation

By

CLAIRE CIZAIRE

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2007

© 2007 Massachusetts Institute of Technology. All right reserved.

Author: _____

Claire Cizaire
Department of Aeronautics and Astronautics
January 31, 2007

Certified by: _____

Charles M. Oman
Senior Lecturer
Department of Aeronautics and Astronautics
Thesis Supervisor

Accepted by: _____

Jaime Peraire
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

Effect of Two-module-docked Spacecraft Configurations on Spatial Orientation

By

Claire Cizaire

Submitted to the Department of Aeronautics and Astronautics on February 2, 2007 in
Partial Fulfillment of the Requirements for the degree of Master of Science in
Aeronautics and Astronautics

Abstract

This thesis used virtual reality techniques to investigate how differences in visual vertical direction of two docked spacecraft affect the ability of a person in one module to mentally visualize the relative orientation of the other module, and spatial relationships of surfaces in it. Spacecraft and space station modules are typically connected differently in space from the way they would be in training simulators on Earth. The local visual vertical is the direction that appears to be “up” as defined by panel and rack orientation, labeling, and placement. In space, the local visual verticals of adjacent modules are not always consistently aligned and astronauts say they find it hard to orient themselves within those configurations. We investigated how relative module orientation determines performance in a spatial memory and visualization task. An experiment compared six different attachment configurations of two modules. Subjects ($n = 20$) wearing a color stereo head mounted display first learned the interiors of two modules separately. They then learned six flight configurations sequentially. In each configuration, subjects located in the first module were shown one “cue” wall in that module, so they could determine their orientation, and were then asked to visualize, place and orient a “target” wall within a wireframe view of the adjacent second module. The total time to respond to each trial was recorded, along with the percentage of correct responses, and the subject’s head orientation. The analysis of time to respond and percentage of correct responses showed that certain configurations were statistically significantly different. As expected, the easiest configurations were “terrestrial like” where the visual verticals of the two modules were co-aligned. Including a 180 deg pitch between the modules made the task harder. The hardest were those that included a 90 deg pitch – in which the local visual verticals of the two modules were orthogonal. Comparing the easiest (terrestrial like) configurations with hardest, subjects needed 3 seconds more to orient and accomplish the task. This represents a significant amount of time given that we perform this task without much thinking about it and almost instantaneously on Earth. Subjects relied heavily on the visual verticals, and often tilted their heads toward alignment with the local visual vertical. Performance could be predicted, based on the number of pitches and yaws relating the two modules in the configurations tested. Supported in part by NASA Cooperative Agreement NCC9-1 with the National Space Biomedical Institute.

Thesis Supervisor: Charles Oman
Title: Senior Lecturer

Acknowledgments

To my parents Yee Ling and Yves Cizaire, my brother Tanguy and my sister Christel, for always being by my side.

To my roommates Amanda Engler, Jessica Li and Rachel Peletz for turning 15E into a home full of wonderful memories.

To all the members of the Dinner Crew for all the great evenings we spent together.

To Patricia Pina for her friendship and wisdom.

To my officemate and great friend Alejandra Menchaca, for sharing my love for chocolate.

To Dr. Charles Oman, my thesis advisor, for all his insights, judicious comments and suggestions on my work.

To Dr. Alan Natapoff, for not only his patience and help throughout the different stages of my experiment, but also for being an endless source of knowledge on sciences and literature.

To Dr. Hiro Aoki, for always being there when something goes wrong.

To Dr. Andy Liu, for all his hardware support.

To David Beneviste and Dan Buckland for tracing the path of this work.

To my subjects, for their time and data.

To Barbara Lechner and Marie Stuppard for their dedication on making my and other graduate students' lives so much simpler.

To MIT and the Arthur Gelb Fellowship for making this possible...

Table of Contents

Abstract.....	3
Acknowledgments.....	4
Table of Contents.....	5
List of Figures.....	6
List of Tables.....	6
Introduction.....	7
Chapter 1: Background.....	10
1.1 Previous work.....	10
1.2 Definition of Relative Module Orientation.....	10
1.3 Pilot experiments and rationale for experiment design.....	12
1.3.1 Original design and pilot experiment.....	12
1.3.2 Pilot experiment findings and ensuing modifications.....	13
Chapter 2: Methods.....	16
2.1 Participants.....	16
2.2 Equipment.....	16
2.3 Environment.....	17
2.4 Experimental design.....	22
2.4.1 Overall architecture.....	22
2.4.2 Trial procedure in a single module.....	23
2.4.3 Trial procedure in the combined modules.....	25
2.5 Dependent Variables.....	27
Chapter 3: Results.....	29
3.1 Total Time to Respond and Percent Correct.....	29
3.2 Data transformation and mixed regression analysis.....	31
3.2.1 Learning effect between successive configurations.....	34
3.2.2 Learning effect within configurations.....	34
3.2.3 Configuration effect.....	34
3.2.4 Cue and orientation effect.....	36
3.2.5 Tendency to be visually upright.....	36
3.2.6 Gender and age.....	38
3.2.7 Standardized tests.....	38
Chapter 4: Discussion and Possible Future Experiments.....	39
4.1 Strategies.....	39
4.2 Differences among the configurations.....	40
4.3 Possible Future Experiments.....	41
Conclusions.....	43
References.....	45
APPENDIX 1: Vizard Code.....	47
APPENDIX 2: Trial Sequences.....	71
APPENDIX 3: Verbal Tour.....	78

List of Figures

Figure 1. The International Space Station. Top: the Shuttle and the US Lab are outlined in red (Source: NASA.) Bottom: schematic view of the Shuttle and the US Lab. The red arrows are the visual vertical of the two modules.....	8
Figure 2. Visual verticals of the different modules of the space station MIR (Source: NASA.). The four black arrows indicate the visual verticals of the different modules.	9
Figure 3. Experimenter wearing the Head Mounted Display and holding the game pad.	17
Figure 4. Layout of the two modules surface textures. The surface texture outlined in red is the surface from the first module that was attached to the second module throughout the entire experiment.	18
Figure 5. Schematic and perspective view of the first three configurations.....	20
Figure 6. Schematic and perspective view of the last three configurations.....	21
Figure 7. Successive views of a trial in a single module.	25
Figure 8. Successive views of a trial in a configuration.	27
Figure 9. Total Time to Respond against Subphases by Configurations, aggregated over the subjects.	30
Figure 10. Percent Correct against Subphases by Configurations, aggregated over the subjects	30
Figure 11. Plots of the residuals for the untransformed and transformed variable.....	33
Figure 12. Differences in Percent Correct due to P_Code (left) and Y_Code (right)	36
Figure 13. Percent Correct for the trials with Orientation 1	38

List of Tables

Table 1. The 23 possible flight configurations made of two modules. The second module is attached to the same wall of the first module and each configuration is obtained by applying a succession 90 deg rotations to the second module.	12
Table 2. Overall architecture of the experiment	22
Table 3. Values of the variables P_Code and Y_Code.....	32
Table 4. The mixed regression analysis of Percent Correct.....	32
Table 5. The mixed regression analysis of Log(Total Time to Respond)	33
Table 6. The mixed regression analysis of Log(Total Time to Respond) for the Orientation 1	37

Introduction

This thesis employed virtual reality techniques to study how differences in visual vertical direction of two docked spacecraft affect the ability of a person in one module to mentally visualize the relative orientation of the other module, and spatial relationships of surfaces in it.

Finding one's way aboard a spacecraft or space station is not as easy as it might seem to be. Keeping track of one's course through the different modules that constitute a spacecraft or space station is challenging.

Gravity is an integral part of our lives. It is of great help, even if we are not always aware of it. Commonly perceived as a burden, gravity is regarded as the reason why we cannot fly and why we inevitably fall, like apples.

However, it greatly simplifies our daily lives. Thanks to gravity we never have any doubt of what "up" or "down" is. Gravity defines a local vertical for all objects and living beings on Earth.

In space, astronauts and cosmonauts are in free fall and there is no perceived gravity. There is no gravitational "down" detected and crewmembers have to rely on visual cues to determine their orientation. Howard (1994) introduced the concepts of visual "frame" and "polarity" cues defining a visual vertical. Ceiling lighting, desks or stowage lockers maintain on Earth a constant orientation consistent with the gravitational "down". Such familiar objects are also present in modules in space and provide cues defining a visual vertical. However, in weightlessness, the "idiotropic" tendency to perceive the visual vertical as aligned with the body axis (Oman, 2003) can interact with the visual cues of the environment. The idiotropic tendency can overcome or change the interpretation of the frame and polarity cues. For example, an astronaut floating upside down in a spacecraft module might think that the surface closest to his feet is the floor, despite the lights indicating that it is instead the ceiling. The real ceiling is then perceived momentarily as a "subjective" floor. This illusion is referred to as a Visual Reorientation Illusion. Based on comparison of astronaut VRI illusions with results of animal experiments (e.g. Knierim, 2000, Taube, 2004) on rats in weightlessness, Oman (2007) noted that populations of rat limbic cells code azimuthal direction and place in a 2D plane whose orientation is labile. This 2D navigational plane likely corresponds to the plane of the "subjective floor" described by astronauts.

Due to engineering constraints on design, the visual verticals of adjacent rooms in a spacecraft may not be co-aligned. The architecture of spacecraft or space stations does not always provide consistent visual verticals, although – as this thesis will show – the spatial orientation performance of the astronauts would improve if they were consistent. If a unique vertical were defined for an entire spacecraft or space station, making it "terrestrial-like", crewmembers would be able to predict the orientation of a new module as they entered it because it would be the same as the module they had just left.

In reality, at present, many crewmembers report difficulties in orienting themselves because real configurations have inconsistent visual verticals (Oman, 2007). They depend on visual cues to orient themselves, and the lack of coherence between the visual verticals of adjacent modules is confusing. Modules are often connected at right angles to one another and being forced to keep track of the visual vertical when passing between modules is burdensome and can lead to errors.

For example, when the Shuttle is docked to the forward end of the International Space Station, the visual verticals defined by the interiors of the two connected modules do not match. The visual vertical of the Shuttle is pitched upward by 90 degrees compared to that of the International Space Station, as shown in Figure 1. Figure 2 shows another example of flight configurations that do not present consistent visual verticals: the space station MIR.

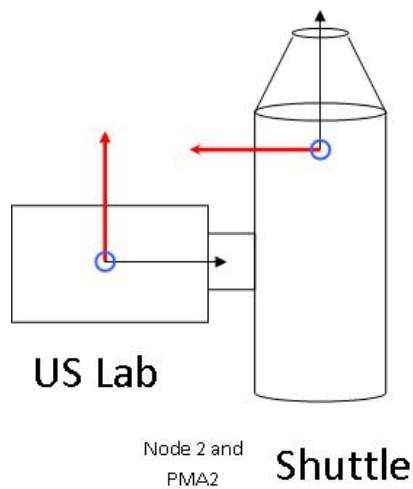
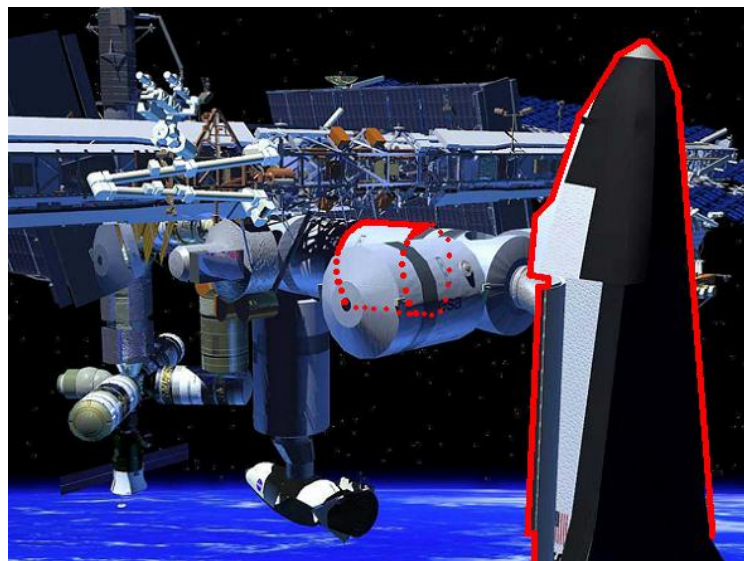


Figure 1. The International Space Station. Top: the Shuttle and the US Lab are outlined in red. Bottom: schematic view of the Shuttle and the US Lab. The red arrows are the visual vertical of the two modules.

There are many different configurations of connected modules in space now. Crewmembers report that it is hard to find their way between modules whose verticals are inconsistent. Apollo, Skylab, Mir and ISS astronauts have all reported momentary disorientation related to incongruity of visual verticals between adjacent modules or even between workstations within a single module (Oman, 2007). One Mir astronaut said that it took about a month till finding one's way around seemed natural and instinctive. Ground trainers weren't helpful, since they aren't in the real relative orientations, and can't be re-oriented. But after about four weeks of living aboard, one learned to [move about] without thinking about it.... Even by the end of the flight, he could not have pointed to places in the other modules from the Base Block, or vice versa. (Richards, et al. 2001) We are interested in knowing if some configurations of connected modules are harder than others to navigate. This study addresses that issue. A three-hour long experiment investigated the effects of six different configurations (pairs of adjacent, differently oriented single modules) on navigation performance within them.

A better understanding of the intrinsic difficulties of configurations could help us reduce the orientation problems that crewmembers experience in space.

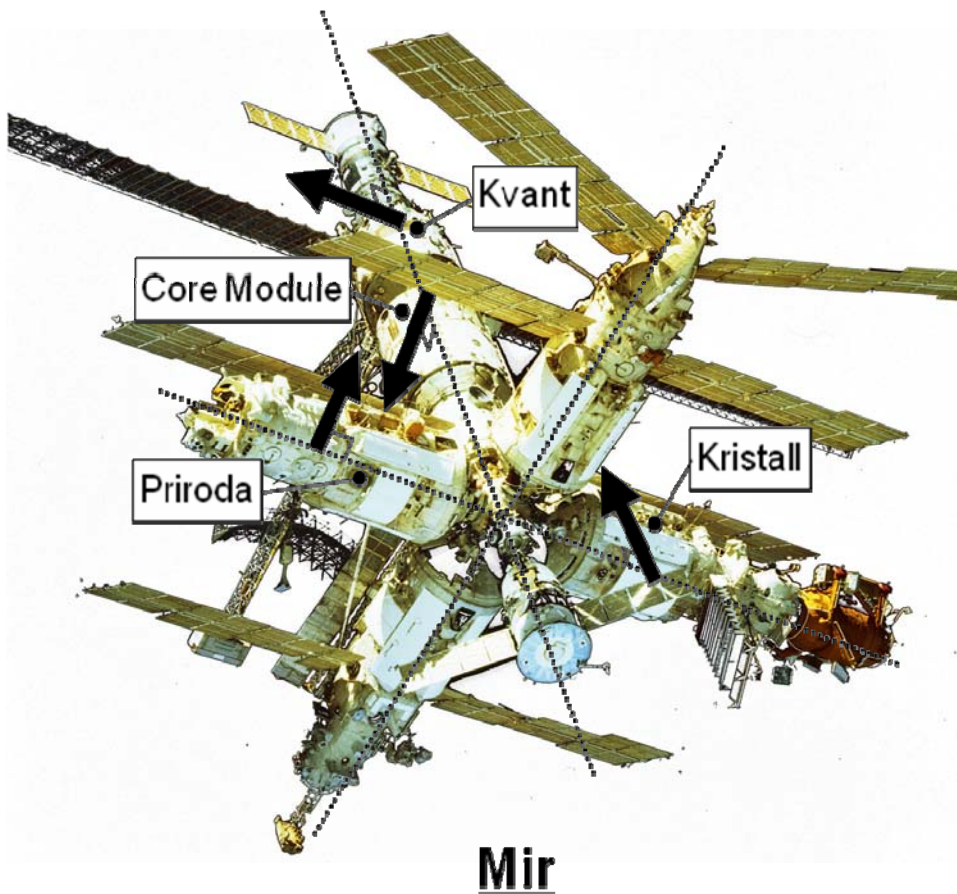


Figure 2. Visual verticals of the different modules of the space station MIR. The four black arrows indicate the visual verticals of the different modules.

Chapter 1: Background

1.1 Previous work

Several experiments have focused on spatial disorientation, using virtual reality techniques. Richards (2000) showed that virtual reality could be used to train subjects to orient in new environments in any relative body orientation and measured their performance when doing so. Benveniste (2004) noted that when astronauts learn the different modules of a spacecraft on Earth, the intrinsic visual verticals of those mock-up modules are necessarily aligned. He postulated that, once in space, unlearning this first perceived “ground” configuration hinders the learning of the real “flight” configuration. His results suggested that after having been exposed to a configuration in which the two modules present a single consistent visual vertical, learning the real “flight” configuration is more difficult than it would be if there had been no such misleading prior training. During pilot experiments, Benveniste tested several different flight configurations separately, with the same protocol. The results suggested that the subjects found some of the configurations to be more difficult than others. Buckland (2006) hypothesized that emphasizing functional relationships between pairs of adjacent walls of two modules during training would help astronauts learn them. Buckland’s experiment tested only one flight configuration.

1.2 Definition of Relative Module Orientation

In everyday life, we rely heavily on gravity. Every room we walk in has a visual vertical that is aligned with the direction of gravity. The “up” and “down” are naturally defined. For each room, the ceiling corresponds to the “up” and the floor to the “down”. Those two opposite walls give a consistent visual cue that defines the vertical.

In space, many research modules replicate the consistent orientation of familiar rooms on Earth. They also consist of six walls, including an easily identifiable ceiling and a floor that encode the vertical axis of the room. It is thus possible for astronauts to clearly identify the visual orientation of the room even though there is no vertical (because there is no perceived gravity.)

Because the visual verticals of two adjacent modules are not necessarily aligned, however, it is not always possible to stay visually upright when transiting between modules. Anecdotal reports documented by Richards (2001) and Oman (2007) suggest that this affects the ease with which astronauts can maintain a sense of the vertical as they pass between such visually inconsistent modules. This study is designed to investigate the differences in performance that those inconsistencies generate and tries to assess the effect the different configurations of module pairs has on them.

There are many ways of docking a pair of modules. Several of those possibilities are realized in existing spacecraft. For example, in order to dock Apollo to Skylab, astronauts must pitch Apollo 90 deg forward and connect its nose to the top of Skylab. The International Space Station has several different configurations within it. The main axis is defined by major modules such as Destiny, Node 1 and 2, Zarya and Zvezda. However, some components such as Soyuz, MPLM, or the docked Shuttle may be pitched by 90 degree with respect to the main axis.

The relative orientation of two rectangular prism spacecraft module interiors can be considered defined by the relative orientation of the two “connecting” or immediately adjacent surfaces. The connecting walls of two adjacent modules could, in principle, be any of the 36 pairs of the twelve walls, one from each module. This study focuses on a subset of those relative configurations. In all the configurations of this study, the connecting wall of the first room remained the same throughout the entire experiment and was a “wall”, i.e. neither its visual ceiling nor its visual floor. Also, we assumed that the six interior surfaces within each module provided cues to the direction of the visual vertical that were directionally congruent – i.e. when subjects looked at a surface, they could readily identify it as a wall, ceiling or floor. These assumptions reduced the variety of possible configurations, and allowed us to compare a coherent subset of related configurations in a single practical experiment.

There are 23 flight configurations obeying the constraints, as shown in Table 1 below. Each configuration can be obtained from the others by a specific sequence of 90 or 180 degree finite rotations (e.g., pitches and/or yaws.) Each transformation, pitch (P), or yaw (Y), or roll (R), represents a 90 deg incremental rotation about the x, y, or z axis. Those finite rotations do not commute and the matrix notation was adopted to clearly identify the order in which the transformations are applied. For example, the P2Y configuration is obtained by first applying

a 90 deg yaw and then two successive 90 deg pitches to the second module of the initial configuration.

Table 1. The 23 possible flight configurations made of two modules. The second module is attached to the same wall of the first module and each configuration is obtained by applying a succession 90 deg rotations to the second module.

Type	Angle	Axis	Number
1 axis	90 deg	Y	2
		P	2
		R	2
	180 deg	Y	1
		P	1
		R	1
2 axis	180 deg	YR = RP = PY	6
		YP = PR = RY	6
	270 deg	P2Y = R2Y	2
Total			23

1.3 Pilot experiments and rationale for experiment design

1.3.1 Original design and pilot experiment

As noted previous studies suggested that some configurations should be more challenging than others but it was not possible to explore the effects of many configurations. Benveniste (2004) focused on the differences between the ground configuration and a single flight configuration. In three pilot experiments, however, he tested three different flight configurations and noticed that they did not seem to be of equal difficulty. To make the experiment more challenging for his subjects, Benveniste chose a single difficult configuration for his final experiment. In this study, we are interested in exploring the challenges posed by to crewmembers navigating in 0-g by a variety of different configurations.

Inspired/motivated by Benveniste's pilot studies, another pilot experiment was designed and performed to identify the properties of the configurations that determine how well subjects could perform, using a spatial orientation and memory task similar to that used by Benveniste and Buckland. The brief overview of the pilot experiment that follows explains the similarities to and differences from earlier experiments conducted by Benveniste and Buckland and the process that eventually

led to the final design of the formal experiment described in later sections. Methodological details are provided in the next chapter.

Six subjects took part in the pilot experiment. All were naïve subjects who had never taken part in any virtual reality experiments in the laboratory before. All subjects wore a color stereo head mounted display, and responded using buttons on a hand held gamepad. The experiment was divided into two sections, as were those of Benveniste (2004) and Buckland (2006). As in these prior pilot studies, in the first section of the pilot experiment, subjects began by learning the interior arrangement of the two modules separately. Then, they learned the configuration of two modules attached. The basic architecture of the present experiment was similar to that used in those earlier studies by Benveniste (2004) and Buckland (2006), except that it tested 6 different configurations — rather than the one or two configurations those experiments investigated.

The second section of this pilot experiment consisted of seven phases. Five different configurations were tested and two were repeated. For every single configuration – i.e., within each phase – the subjects were trained first, then tested. There was feedback, which consisted in displaying the correct answer to the subjects after they had validated their own answer, during the training and none during testing. Otherwise, training and testing were identical. The training and testing methods were similar to those used in the studies referred to above. A detailed description is given in Chapter 2.

The pilot trials for this experiment explored the feasibility of comparing the relative complexity of the configurations. Previous experiments of Benveniste and Buckland determined that subjects typically get tired and/or bored after three consecutive hours of experiment and that the resulting loss of attention reflects in their performance. Thus, we decided to limit ourselves to a three hours long experiment. The seven configurations, including (in the pilot tests) two repetitions, were the most a subject could, apparently, perform adequately in a single session and the most, therefore, that the experiment proper could compare. Configuration repetitions were included in the design to assess potential learning effects across sequential configurations.

1.3.2 Pilot experiment findings and ensuing modifications

The subjects completed the pilot experiment in three hours. One third of the time was spent learning the two modules separately and the other two thirds were spent learning the modules in attached (flight) configuration. A few subjects reported that the experiment was long, but none said it was excessively long. The length chosen stretched the

experiment to its plausible limits and we expected and looked for effects of lowered performance toward the end.

Originally, we feared that frequent changes of configuration would confuse the subjects. In the formal Benveniste and Buckland experiments, subjects had only been exposed to a single attached configuration during any single experiment. Configuration comparisons could only be made by comparing the performance of different subject groups. We tried to strengthen experiment design by using a “within subjects” design, using each subject as their own control and by testing all the subjects on the same set of configurations — a repeated-measures design. However we feared that some subjects might not be able to unlearn the first configurations encountered (which we expected to play an important role in their training). We were concerned that by the third or fourth configuration they might become confused or disoriented. At the end of the experiment, however, the subjects reported that they believed the changes in configuration had not been an issue in their performance. When asked, none reported being confused by the sequence of configurations. The accuracy of their performance supported their subjective reports.

However, the pilot experiment revealed an overall trend toward improvement in performance – apparently a general learning effect over sequential configurations - that subjects also described. In particular, since configurations tested later showed better performance than those performed earlier, the pilot experiment emphasized the importance of the order of presentation of the various configurations

The repetition of two configurations out of the five in the pilot experiment confirmed the importance of the learning effect throughout the experiment, but did not provide other significant information. Moreover, some subjects noticed that a few configurations had been repeated and complained about that repetition (presumably because of boredom.) For that reason, our final design had only one repeated configuration, and six different configurations instead of five. This change increased the diversity of configurations we were able to test, and presumably decreased the subjects’ boredom.

The experimenter noticed that some subjects made additional angular head movements during some of the individual trials. The subjects explained that although they had not been instructed to do this, and the head movements were sometimes uncomfortable while wearing the head mounted display, that the subjects rolled their heads so that their view would be more closely aligned with the local visual vertical of the module, since the experiment simulated weightlessness, and the visual

vertical was often not aligned with their body axis. Some subjects said that they could perform more rapidly when aligned that way.

Chapter 2: Methods

2.1 Participants

The experimental protocol was reviewed and approved by MIT's institutional human subject experiment review board. Twenty subjects completed the experiment. Two others started but could not finish. ¹Eight of the twenty subjects were females. All the subjects were MIT students, post-docs or affiliates. One of the subjects had taken part in one of the virtual reality experiments of the Man Vehicle Laboratory before. The environment and the tasks were, however, completely new to him. None of the other subjects had ever taken part in a similar experiment. The subjects were between 18 and 31 years old. They were paid \$10/hour for participation.

2.2 Equipment

The experiment was conducted in the Man-Vehicle Laboratory using its equipment. To simulate a 0-g environment, we used virtual reality techniques. They allowed subjects, for example, to be visually upside-down in a room.

Subjects wore a NVIS nVisor SX color stereo head mounted display (1280 × 1024 pixels per eye) resolution), as shown on Figure 3, which immersed them into the virtual environments created with 3D modeling software (3D Studio Max), and rendered on a Dell PC (Xeon 2.4 GHz processor, NVIDIA Quadro4 Graphics Card) using a commercially available virtual reality display package (WorldViz, Inc, Santa Barbara, CA, Vizard v.2.5.)

The orientation and position of the head-mounted display was tracked (Intersense Inc., Burlington, MA, Model IS-600, operating in 3-

¹ One subject felt dizzy shortly after starting and exercised the right to stop – that we had explained to them before the start of the experiment. The experiment with a second subject was stopped because of a problem with the equipment. The problem was fixed, but the subject did not have a chance to come back.

DOF). Subjects used a Thrustmaster game pad to select an answer and validate their choice. For additional details see Benveniste (2004) and Buckland (2006).



Figure 3. Experimenter wearing the Head Mounted Display and holding the game pad.

2.3 Environment

The two virtual rooms, or modules, created for the study were the same for the pilot and final experiments. Each virtual room represented a generic spacecraft module and consisted of six walls: one distinctive ceiling, one distinctive floor, and four normal walls. Each room had an unambiguously defined visual vertical.

The configurations tested were obtained by attaching/docking those two rooms/modules together. The second module is attached to the first at a wall in the first module that was the same throughout the experiment.

The textures used to build those two modules are shown in Figure 4. The texture of the first module that was attached to the second module is outlined in red in the figure.

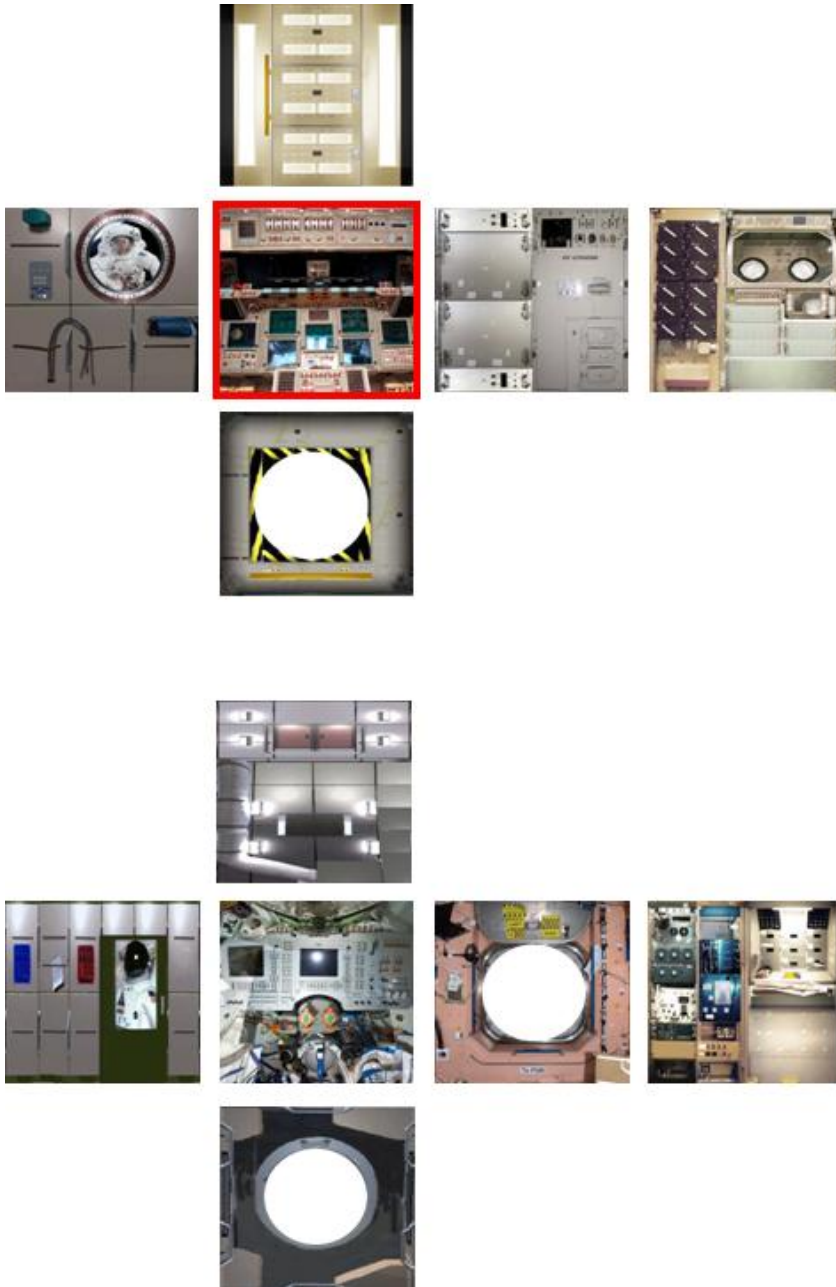


Figure 4. Layout of the two modules surface textures. The surface texture outlined in red is the surface from the first module that was attached to the second module throughout the entire experiment.

Since the pilot experiment showed that it was plausible to include seven phases in a single test session, six different ways of attaching the two modules were tested (i.e., six configurations) and one of them was replicated at the end to assess learning that had taken place during the experiment. Repeating only one configuration rather than two, and putting the replication at the end reduced the risk that we would lose data

on one of our six comparison configurations at the end because of subject fatigue or boredom.

The configuration called “I” (for Intity) below represents the “normal” situation in which both modules are presented precisely as they were presented separately, at the beginning of the experiment—except that they are attached during the experiment proper. For each module, the first wall the subjects saw was the control panel. The first module’s control panel is the connecting wall for that module throughout the entire experiment. In the “I” configuration, as subjects stand visually upright in the first module, looking past the control panel into the second module, they find themselves visually upright in that module too, facing the control panel as well. The remaining configurations are obtained from the I configuration by a combination of yaws (applied first) and pitches applied to the second module with respect to the first.

Referring to the relative module orientation definitions in Table 2, the order in which the configurations appeared in the experiment itself was: P, Y2, P2Y, I, P2, PY, P2Y. Figure 5 and Figure 6 provide a perspective view of the different configurations. P2Y was repeated because it requires the largest number of finite rotations and therefore might prove to be cognitively more challenging to perform. Also, the replication allowed assessment of potential learning effects.

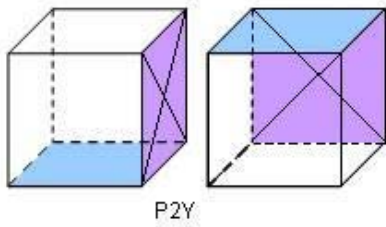
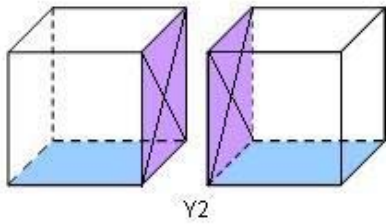
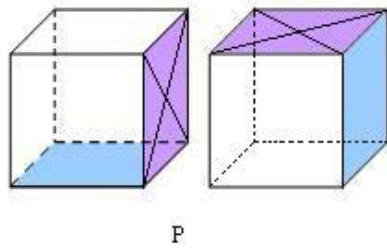


Figure 5. Schematic and perspective view of the first three configurations

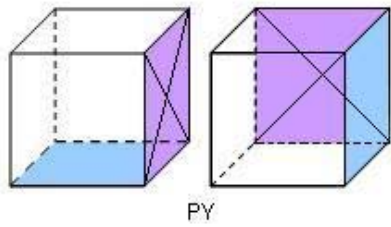
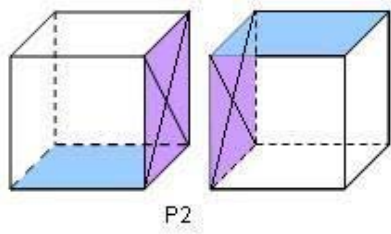
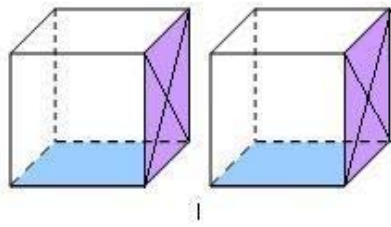


Figure 6. Schematic and perspective view of the last three configurations

2.4 Experimental design

2.4.1 Overall architecture

Like the pilot experiment, the experiment proper consisted of two sections.

In the first section, the subject was introduced to the layout of the two modules separately. In each module, the experimenter verbally explained the use of the equipment pictured on each virtual wall by. This was intended to make the walls easier to learn and recall. The explanations were read from a script in an attempt to provide all the subjects with the same information and instructions. The script is provided in the Appendix. After the verbal tour, the subject was allowed as much time as needed to look around the virtual module and start memorizing its features and their relations. When the subject felt ready, he or she could take off the head-mounted display and tell the experimenter how the module was organized. This helped subjects familiarize themselves with the environment and gently provoked them to ask questions. After this verbal feedback, the subject started the computerized training which accustoms the subject to the game pad and to the experiment's tasks. By the end of this section, subjects should be familiar with the individual module. The training is detailed later in this chapter.

For the trials of the second section, the two rooms were attached. In seven phases, six different configurations were tested, and one of them, P2Y, was tested a second time. For each phase, the subject was first trained and then tested. The training and testing tasks were the same except that feedback that corrected errors in performance was provided during training, but not during testing. The training and testing are detailed later in this chapter. At the very beginning of the second section of the experiment, the subjects were told (reminded) that both accuracy and speed of performance were important.

The overall design is summarized in the following table.

Table 2. Overall architecture of the experiment

Section	1		2						
Environment	Module A	Module B	P	Y2	P2Y	I	P2	PY	P2Y
Phase	Training	Training	Training & Testing	Training & Testing	Training & Testing	Training & Testing	Training & Testing	Training & Testing	Training & Testing

Subjects took two different, relatively short, standardized tests of natural aptitude either at the end of the experiment or on subsequent day. The scores on these tests were obtained for use as possible predictors of performance. The Cube Comparison (“Cube”) Test (Ekstrom, French et al. 1979), a pencil and paper test, measured the ability of subjects to mentally rotate a perspective drawing of a cube in three dimensions. The computer-based Perspective Taking Ability (“PTA”) test (Kozhevnikov and Hegarty 2001) measured the ability of subjects to change perspectives mentally when looking at objects on a 2-D map. Prior studies by Benveniste and Buckland showed correlations between performance and Cube and PTA test scores.

2.4.2 Trial procedure in a single module

Subjects were trained in the same way for both modules. The training was designed to familiarize subjects with the local environment and the equipment. The subjects had not only to learn individual modules in a limited time but also to learn multiple configurations of two modules when joined together. The training was the first occasion for them — they were naïve subjects — to be successively upright, upside-down and tilted, in the virtual module.

The training consisted of 18 trials. The subject’s task was to perform the cue wall self-orient/target surface-place/target surface-rotate task developed by Benveniste and Buckland. In each trial, subjects were shown one (“cue”) surface in the module, then had to determine their relative orientation from this, then call up a picture of a second (“target”) module surface, and place this target image on the appropriate wall and finally rotate it into the correct orientation relative to the cue surface. Figure 7 shows the succession of views a subject saw during an individual trial. At the beginning of each trial, a cue surface of the module the subject was displayed. The module’s 12 edges were shown as a red wireframe, but walls other than the cue wall were missing, and the volume beyond was entirely dark. From the relative orientation of the cue surface, subjects had to infer their orientation within the local module and be able to mentally visualize/reconstruct the position and orientation of other surfaces in the module.

When the subjects established their orientation within the mentally reconstructed module, they pressed an “enter” button on the gamepad, and a picture of a different surface, the target within that module, was displayed. Once subjects had recognized the target wall and were ready to put it in the correct place, they pressed the “enter” button.

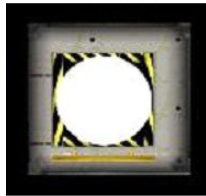
The still picture was then removed and the previous, red-wireframed environment is restored, except that the cue wall was no longer visible and there was a red cross in its place. All the other surfaces of the module were blank and only the room's 12 red edges were outlined. Head tracker information was used to determine which surface the subject's head was pointing at, and a large red cross was displayed on the surface the subject was facing. The subjects moved their heads until they faced the surface on which they thought the target picture belonged and pressed "enter". The picture of the target surface was then pasted onto the surface the subject was facing, replacing the red cross. Finally, the subjects had to rotate the target picture until it had the correct orientation — i.e., the orientation it had in the original module — and validate the displayed target placement by pressing "enter" a final time.

As soon as the subject responded, all the surfaces in the room were presented in their correct orientation. If the placement of the target wall was incorrect, that was immediately obvious to the subject. This placement-error feedback reminded the subjects of the actual module configuration after each training trial and thus enabled the subjects to understand their mistakes and improve the accuracy of their answers.



Cue wall

A cue wall is displayed with a random orientation. All the other walls of the module are blank and only the edges are outlined in red. The subject should infer his orientation

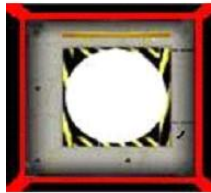


Target wall

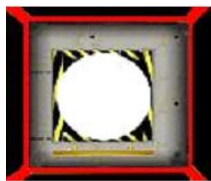
The red edges and the cue wall are removed and a picture of a target wall is displayed. Once the subject has identified the target wall, he presses "enter".



The still picture is then removed and the previous, red-framed, environment restored. There is red cross in the place of the cue wall. The red cross tracks and indicates the wall the subject is facing. The subject moves his head until he faces the wall on which he thinks the target picture belongs and presses "enter".



The picture of the target wall is then pasted onto the wall the red cross that was on and the cross disappears.



The subject then has to rotate the picture within that wall until it has the right orientation. The subject presses "enter" to validate his answer.



Feedback (for the training trials only)

All the walls appear in their correct position and the subject can check his answer.

Figure 7. Successive views of a trial in a single module.

2.4.3 Trial procedure in the combined modules

The trials of Section Two, with the combined modules, were analogous to those of Section One in a single module. As in Section One, the subjects also had to determine their relative orientation relative to a cue surface, and then place a picture of a target surface on the surface where it belonged. However, this time the target surface was in the second module. As in Section One, the subjects' orientation within the first module was varied. They could be upright, upside-down, or orthogonal to the visual vertical of the first room.

At the beginning of each trial, a cue surface from the first module – the module the subject was (virtually) in – was displayed. Only the 12 edges of the first module were outlined, other than the cue surface. All other surfaces in the first module were blank and the second module was not visible at all. Using the cue surface, the subjects inferred their orientation within the first module and predicted the position and orientation of the second module.

When the subjects had identified their orientation in the first module and the direction to look toward the second (but still invisible) module, they pressed “enter”. The second module was then revealed. Its 12 edges appeared in yellow and a picture of one of its surfaces – the target surface – was displayed on a random surface. At the same time, the cue surface was removed. All the surfaces of the first module were then blanked, but its 12 edges were still outlined in red.

With a button on the game pad, the subjects toggled the picture of the target surface onto successive surfaces in the second module until it was displayed on the correct surface, and then used a button to toggle the target surface orientation until it seemed correct, relative to the cue wall and all the unseen (but perhaps visualized) surfaces in both modules. Once subjects were satisfied with both the position and orientation of the target surface, they pressed “enter” to validate their answer.

As during Section One training, at the end of the trial all surfaces (except the two that jointed the two module) were displayed, to provide feedback.

Figure 8 shows the succession of views a subject would see during a trial.

Due to the time constraint, we had to reduce the number of trials per phase compared to previous experiments. Each phase consisted of 24 trials: 6 of the 24 trials had feedback and were hence considered as training trials. To reduce the number of trials, we had to limit the number of cue walls used during the training. Instead of using all the visible surface textures of the first module as cues, we chose two of them: one normal wall and the ceiling. The combination of those two surface textures with the three possible orientations (upright, tilted by 90deg, upside down) enabled us to test all the possible orientation within the first module. There are 6 possible combinations. Therefore, the total number of trials was set to 24 instead of 30 or 36.

Subjects were always offered the opportunity to take a short break between phases.



Cue wall (from the 1st module)

A wall from the 1st module is displayed with a random orientation. While the edges of the 1st module are outlined in red, the 2nd module is invisible. The subject should infer his orientation and predict where the 2nd module should be. Once he has turned his head toward the 2nd module, the subject presses “enter” to go on.



Target wall (from the 2nd module)

The cue wall is removed and, at the same time, the 2nd module is outlined in yellow. A target wall from the 2nd module is displayed on a random wall of the 2nd module, with a random orientation.



The subject uses the game pad to put this target wall on the wall he thinks it belongs to.



Still using the game pad, the subject changes the orientation of the target wall until it correctly oriented, relative to the cue wall. Once the subject is satisfied with his answer, he presses “enter” to validate it.



Feedback (for the training trials only)

All the walls of the two modules appear in their correct position and the subject can check his answer.

Figure 8. Successive views of a trial in a configuration.

2.5 Dependent Variables

Three variables were recorded during the second section of the experiment:

- Accuracy of response
- Time to respond;
- The orientation of the subject’s head. (It indicates the degree to which the subject sought to simplify his task by rotating the environment directly, rather than by rotating it mentally.)

Accuracy was determined using two criteria for a correct answer. The target wall had to be on the correct wall and it had to have the correct orientation. If both criteria were met, the trial was considered correct.

The total time subjects took to respond was also recorded for each trial. The timer was reset at the beginning of every trial, when the cue wall was first displayed, and was stopped when the subject pressed the “enter” button to validate the final answer given. (The time spent on feedback during the training phase was not included in the total time to respond.) Therefore, the total time to respond measured during the training and testing are comparable.

The change in head position between the beginning and the end of the trial was recorded. This measure was introduced as a result of the pilot experiment, to see when subjects tilted their heads toward the visual upright.

Chapter 3: Results

It took three hours, on average, for the subjects to complete the experiment. Since the trials were repetitive and the experiment was long, we were afraid that subjects would be bored by the end of the session and that their performance would degrade on that account. This would have distorted the results for those configurations that came at the end and be difficult to correct for. A few subjects did get tired, but to our surprise, most subjects liked the experiment and found it both fun and challenging.

3.1 Total Time to Respond and Percent Correct

Data from the 20 subjects who completed the entire experiment were analyzed using mixed regression models on the two dependent variables, Total Time to Respond, and Percent Correct. SYSTAT 11 software (SYSTAT Software, Inc., San Jose, CA) was used.

Testing for each module configuration consisted of 24 trials: 6 training, and 18 testing trials. For averaging, they were grouped by 6, thus dividing each set of tests into four subphases: one subphase for training in each configuration, followed by three subphases of testing.

Figure 9 and Figure 10 show the mean Total Time to Respond and the Percent Correct for each the seven phases (configurations) in Section Two, by subphase, averaged over all 20 subjects.

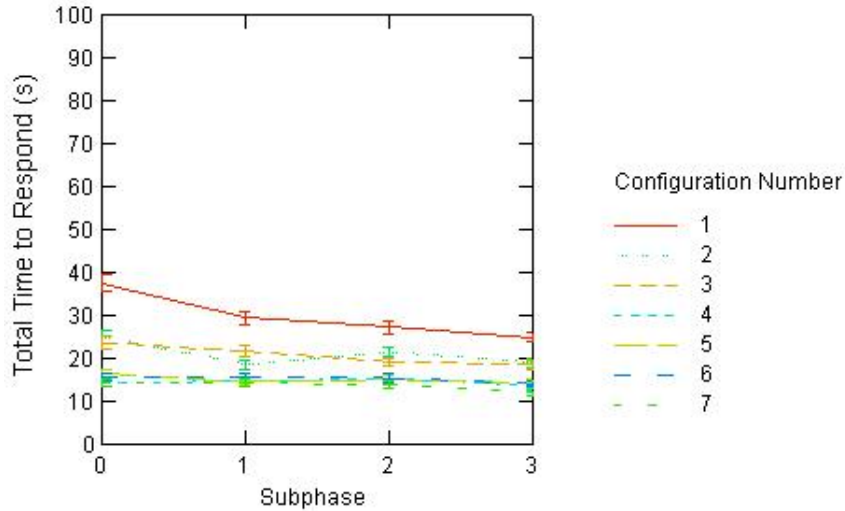


Figure 9. Total Time to Respond against Subphases by Configurations, aggregated over the subjects.

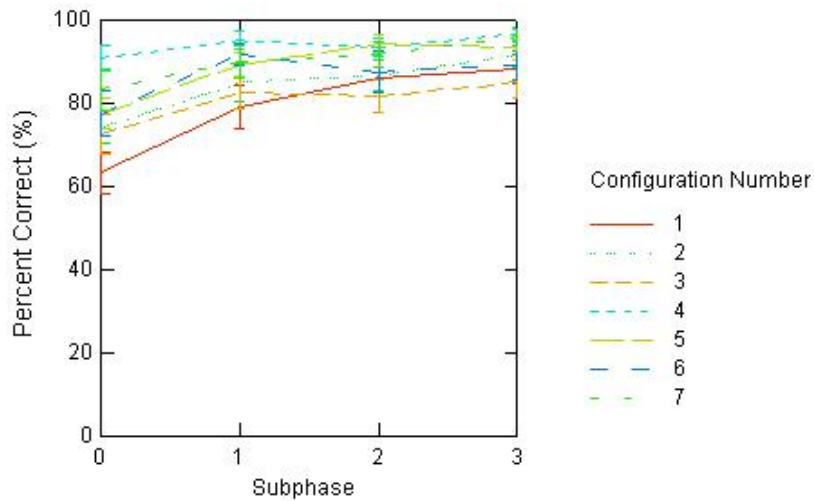


Figure 10. Percent Correct against Subphases by Configurations, aggregated over the subjects .

Figure 10 shows that the Percent Correct for the last subphase is greater than 85% for all configurations. This agrees with the results of Benveniste (2004) and Buckland (2006) who used 36 (instead of 24) trials in each configuration. Despite the 33% shorter phases the subjects were able to achieve relatively high values of Percent Correct in the present experiment too.

3.2 Data transformation and mixed regression analysis

The Total Time to Respond was not normally distributed. In order to achieve normally distributed residuals with stable variances from the model, we fit the log (Total Time to Respond) instead of the Total Time to Respond itself.

For the same reasons, we analyzed the transformed variable $\sin^{-1}\left(\sqrt{\text{Percent Correct}}\right)$ rather than the original measured Percent Correct. This is a standard statistical transformation for percentage data.

These variables were included in the model:

- Subphases;
- Phases;
- Gender;
- Age;
- PTA scores;
- Cube scores;
- The kind of incremental rotation (pitch, yaw) that characterized the configuration;
- Cue wall displayed;
- Orientation in the first module.

As detailed in Chapters 1 and 2, each configuration was characterized by the sequence pitches and yaws of the second module based on its appearance as the subjects learned it separately and alone. During the experiment, configurations consisted of a pitched and/or yawed version of the original second module attached to the original first module. The fourth configuration subjects faced, in fact, combined the original, unpitched and unyawed, version of the second module with the original first module. The other configurations contained second modules that had been pitched (P) or yawed (Y) by some number of 90 degrees rotation increments. Pitches of (0, 90, 180, 270) degrees are indicated by (P0, P1, P2, P3), respectively, (although P3 configuration was not tested). Correspondingly, yaws of (0, 90, 180, 270) degrees were indicated by (Y0, Y1, Y2, Y3), (although only the first three were tested). The symbol PY stands, for example, for the configuration generated by applying a yaw of 90 degrees first, then a pitch of 90 degrees. When yaws and pitches were both applied to generate a configuration, the yaws were always applied first. (A different configuration would be generated, in general, by applying the rotations in another, e.g., the reverse, order.) The number (P_Code) of pitches and (Y_Code) of yaws applied, therefore, uniquely determines the configuration. The analysis, therefore, considers each configuration to be characterized by the variables P_Code and Y_Code.

The values of those two new variables, P_Code and Y_Code for the 6 different configurations tested are summarized in the following table:

Table 3. Values of the variables P_Code and Y_Code

Configuration	P_Code	Y_Code
P	P1	Y0
Y2	P0	Y2
P2Y	P2	Y1
I	P0	Y0
P2	P2	Y0
PY	P1	Y1

The results of the mixed regressions are shown in the following tables:

Table 4. The mixed regression analysis of Percent Correct

Fixed effect	Coeff	Error	Z	p-value	
Gender	0.039	0.048	0.816	0.415 ns	
Age	-0.004	0.008	-0.562	0.574 ns	
PTA	0.014	0.012	1.122	0.262 ns	
Cube	-0.004	0.006	-0.799	0.424 ns	
Phase	0.036	0.008	4.508	0.000	
P_code	P0	0.074	0.026	2.906	0.004
	P1	-0.045	0.02	-2.223	0.026
	P2	-0.029	0.019	-1.545	0.122 ns
Y_code	Y0	0.046	0.017	2.692	0.007
	Y1	-0.024	0.029	-0.823	0.411 ns
	Y2	-0.022	0.034	-0.658	0.510 ns
Subphase	0.071	0.01	6.812	0.000	

Table 5. The mixed regression analysis of Log(Total Time to Respond)

Fixed effect		Coeff	Error	Z	p-value
Gender		0.031	0.042	0.749	0.454 ns
Age		0.01	0.007	1.461	0.144 ns
PTA		-0.026	0.011	-2.465	0.014
Cube		0.006	0.005	1.18	0.238 ns
Phase		-0.053	0.002	-21.617	0.000
P_code	P0	-0.072	0.008	-9.086	0.000
	P1	0.057	0.006	9.181	0.000
	P2	0.015	0.006	2.458	0.014
Y_code	Y0	-0.021	0.005	-4.003	0.000
	Y1	-0.004	0.009	-0.0394	0.693 ns
	Y2	0.025	0.011	2.341	0.019
Subphase		-0.033	0.003	-10.052	0.000
Cue 1		-0.01	0.004	-2.73	0.006
Cue 5					
Orientation 0		-0.115	0.005	-22.487	0.000
Orientation 1		0.036	0.005	7.278	0.000
Orientation 2					
Cue1*Ori 0		-0.025	0.005	-4.71	0.000
Cue1*Ori1		0.006	0.005	1.24	0.215 ns

The residuals of the mixed regressions for Log(Total Time to Respond) and Percent Correct were normally distributed with a constant variance across the predicted values. The use of transformed data for the analysis was justified by the residuals, as shown on Figure 11.

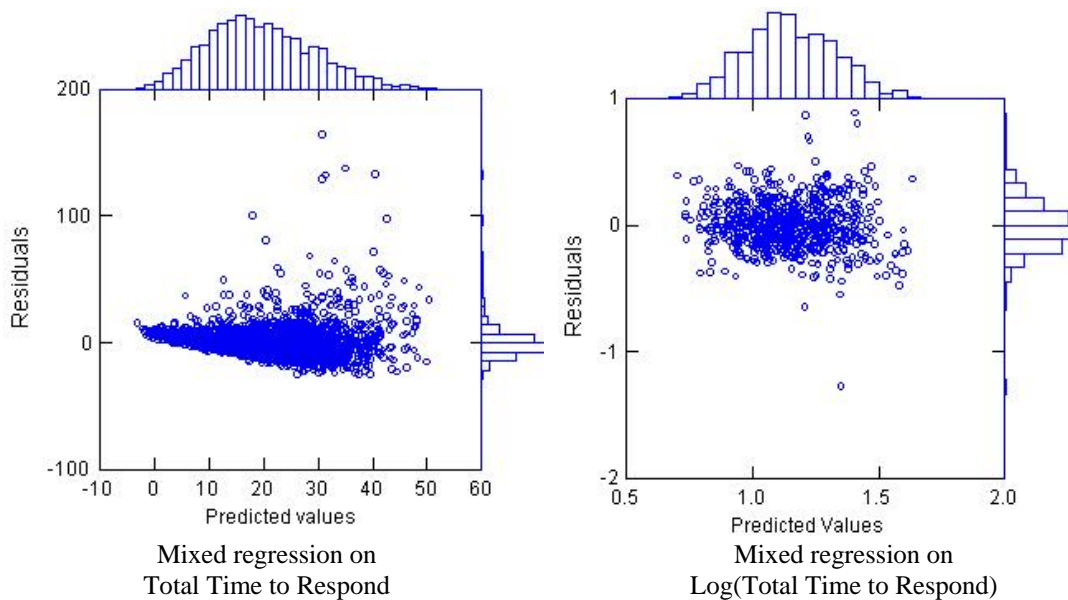


Figure 11. Plots of the residuals for the untransformed and transformed variable

The results of the mixed regressions are presented in the subsequent sections and discussed in Chapter 4.

3.2.1 Learning effect between successive configurations

The pilot experiment, and to a lesser extent the previous studies, emphasized the importance of learning between configurations. As we hoped, there is a clear pattern of learning, as shown on Figure 9 and Figure 10. Although configuration #7 is the same as #3, performance was much better for configuration #7 which had the advantage of being performed after the subject had acquired greater experience with the environment. The rate of learning is not the same throughout the experiment. As shown on Figure 9 and Figure 10, the learning is rapid between the first and second configurations, and the rate of improvement appears to decrease thereafter.

The subjects were aware that they were improving and spontaneously suggested that that trend should be taken into account since it could mask the effect of the differing difficulty of the configurations being tested.

The analysis shows that the learning effect is significant for both measured variables, Total Time to Respond and Percent Correct. The significant (linear) effect of time-in-experiment on the Log(Total time to respond) suggests a classic exponential “learning curve” asymptote to final acquisition of the configurations.

3.2.2 Learning effect within configurations

Figure 10 suggests that learning was also taking place within the 24 trials of each configuration, separately — mostly during the first (training) subphase. The learning curve rises very quickly and then stabilizes for Percent Correct. The effect was statistically significant and important for both variables.

3.2.3 Configuration effect

Beyond the effect of learning, there was a significant effect of configuration type that is associated with the number and type of rotation that generate it.

The configurations that were made from the original second module without yawing are significantly easier than average, but 90 deg yaws or 180 deg yaws were not significantly easier or harder than average for the Log(Total Time to Respond).

By contrast, the different pitches (0, 90, 180) are all significantly (better, much worse, worse), respectively from the average. The configurations (P, PY) are harder than the (P2, P2Y) which are, themselves, harder than (I, Y2). These results are based on inferences from the Time to Respond. According to this analysis, the hardest configuration of all is the (PY) configuration.

The results of the mixed regression analysis were converted back to seconds with the exponential function. According to those results, subjects were about 6.95% faster than average for the (I, Y2) configurations and 5.87% slower than average for the (P, PY) configurations. The time performance for the two remaining configurations was average. The mean time subjects took to complete a trial was computed and we found out that it was about 20 seconds. Based on this computed average, the subjects took about 3 additional seconds to answer in the (P, PY) configurations compared to the (I, Y2) configurations.

The effect of the yaws is not as important. In the absence of yaws, the subjects were on average 2.08% faster than average and with a 180 deg yaw, subjects were about 2.53% slower than average.

The analysis of Percent Correct agrees, generally, with this ordering of configuration types. As in the analysis of the Time to Respond, the unyawed configuration gave significantly better performance than average. Unlike that previous result, yaws of 90 and 180 deg were significantly (not merely apparently) more difficult than average.

The effect of yaws on Percent Correct is half that of pitches, but both are significant. In the ground configuration, for example, subjects performed better, than average (since it has no pitches or yaws) and when the configuration was generated by a pure 90 deg or 180 deg pitch, subjects performed worse. A 90 deg or 180 deg yaw caused subjects to perform significantly worse. 180 deg yaw corresponds to the inversion of left and right and apparently confuses the subjects. That confusion can explain why the Percent Correct fell for that (Y2) configuration but the Total Time to Respond did not decrease significantly.

When comparing the best case and the worst case for Percent Correct in terms of pitches, we found that on average, the percentage of correct answers for the (P, PY) configurations was 6.40% lower than the percent of correct answers for the (I, Y2) configurations. Similarly, the percentage of correct answers for the (Y2) configuration was 2.8% lower than the percentage of correct answers for the configurations with no yaws at all. The following figure shows the percentage of correct answers for each the values of P_Code and Y_Code, by subphase, averaged over all 20 subjects.

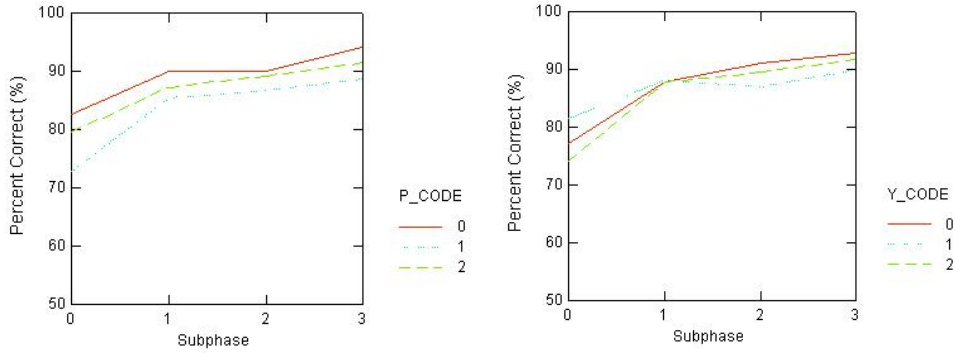


Figure 12. Differences in Percent Correct due to P_Code (left) and Y_Code (right)

3.2.4 Cue and orientation effect

The results show significant effects of cues and orientation on performance. The orientation 0, subjects started the trial upright, seems to be the easiest, for all cue walls. Orientation 2, the subjects started the trial upside down, seems to be the hardest, for all cue walls. Cue wall 1, a normal wall, seemed to give better than average performance, except in the combination Cue 1 and Orientation 2.

3.2.5 Tendency to be visually upright

During the pilot experiment, we noticed that some subjects would roll their heads in order to be visually upright in their virtual module. For that reason, we recorded their head position during the experiment proper and found that when subjects turn their head in order to be visually upright, they perform the task faster than when they do not.

When subjects began the trial with body-axis orthogonal to the visual vertical of the first module, they could easily tilt their heads in order to reorient and be visually upright in the module. For such trials, we defined a new variable, Upright, indicating whether the subjects had stayed orthogonal to the visual vertical or whether they had tried to be visually upright. The results of a mixed regression analysis on Log(Total Time to Respond) on the subset of trials for which the starting orientation was orthogonal to the visual vertical in the first module are shown in the following table:

Table 6. The mixed regression analysis of Log(Total Time to Respond) for the Orientation 1

Fixed effect	Coeff	Error	Z	p-value	
Gender	-0.044	0.031	-1.1424	0.154	
Age	0.014	0.007	1.893	0.058	
Conf_order	-0.056	0.003	-16.833	0.000	
P_Code	P0	-0.06	0.011	-5.611	0.000
	P1	0.04	0.008	4.76	0.000
	P2	0.02	0.008	2.487	0.013
Y_Code	Y0	-0.014	0.007	-2.002	0.045
	Y1	0.011	0.012	0.902	0.367
	Y2	0.004	0.014	0.246	0.806
Sixth	-0.022	0.004	-5.6	0.000	
Cue 1	-0.004	0.005	-0.858	0.391	
Cue 2					
Upright0	0.019	0.007	2.967	0.003	
Upright1					

Subjects were significantly slower when then stayed orthogonal to the visual vertical than when they reoriented their heads in order to be more visually upright.

This analysis was performed for the trials where the subjects' body-axis were originally orthogonal to the first module's visual vertical only. Performing the same analysis on the trials with the Orientation 0 or 2 would not have been relevant. In one case, Orientation 0, the subjects were already visually upright in the local module at the beginning of the trial. In the other case, Orientation2, the subjects began the trial visually upside-down. It is almost impossible to physically turn one's head upside down with the Head Mounted Display on.

A similar mixed regression analysis was performed on Percent Correct for Orientation 1, but no effects were found to be significant. However, for the trials in which subjects' body-axis were originally orthogonal to the local visual vertical, the percent of correct answers was higher for the group of trials in which the subjects had turn their heads to be visually upright than the group of trials in which the subjects had remained orthogonal to the visual vertical, as shown on Figure 13.

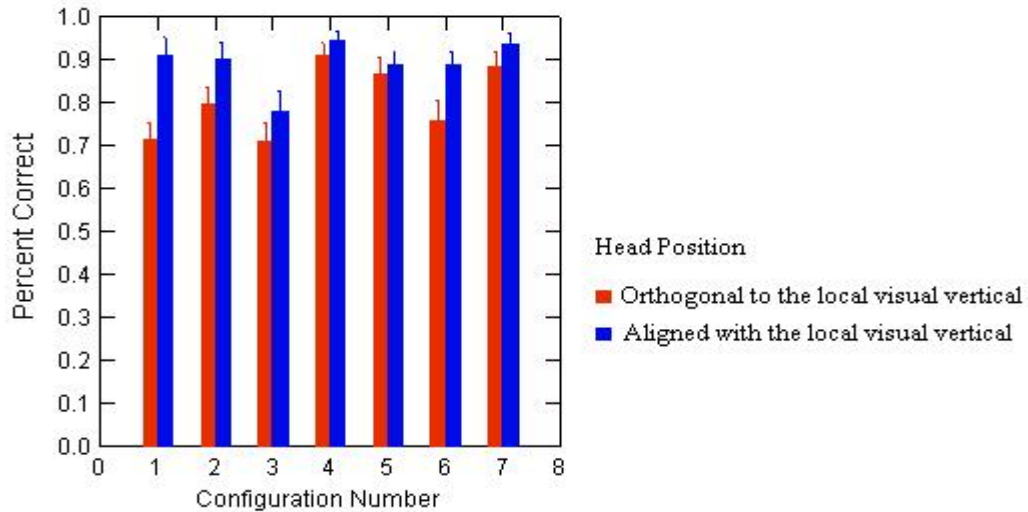


Figure 13. Percent Correct for the trials with Orientation 1

All the subjects did at one point try to be visually upright in the local module rather than stay orthogonal to the visual vertical, but the subjects did not move their head systematically.

3.2.6 Gender and age

Twelve males and eight females took part in the experiment. The subjects were between 18 and 31 years old age. Neither gender nor age were found to have a significant effect on Percent Correct or Total Time to Respond.

3.2.7 Standardized tests

The results of the Perspective Taking Ability test have a significant effect on the total time to respond performance. Subjects with higher scores on the test performed faster than average. No significant effect of score, however, was found on Percent Correct. The other test, the Cube Comparison test, showed no significant effect on either of the performance measures.

Chapter 4: Discussion and Possible Future Experiments

4.1 Strategies

One of the goals of the study was to understand how subjects proceed to mentally reconstruct/visualize the second module from the first one. A few strategies emerged from discussions with the subjects. Most subjects reported using several strategies as the configurations were changing.

One such set of strategies involves what some subjects called “bridge-surfaces”. These were adjacent pairs of surfaces (that share an edge in common) in which one edge of the pair bridges the two modules the configuration. Subjects would remember either a pair of bridge-surfaces or one bridge surface and the far surface of the second module. By remembering “bridge-surfaces”, subjects mentally reconstructed the second module from the first one based on two elements of the second room.

Another kind of strategy emerged from discussion with subjects who did the pilot experiment. One of these subjects reported using solely mental rotations. During training the subject would identify and remember the sequence of rotations that had to be applied to the second module to yield the configuration displayed. During each trial this subject would mentally rotate the second module, and then determine the target location and orientation in the second module. This subject generally responded more quickly than all the other ones. Nevertheless, none of the other 20 subjects who did the formal experiment used this “mental rotation” strategy. Many subjects indicated using rotations, but those were rotations of their head as a way to recapture their sense of the visual vertical of the first module, the module they were in. Subjects found that being visually upright simplified the mental reconstruction of their module.

Therefore, once they were given the cue wall, they would find the visual “up” and try to align their head axis with the visual vertical. They would then use either two pairs of walls or one pair and the far wall to reconstruct the second module from the first.

The results (Section 3.2.5) show that subjects performed faster when they had aligned their head axis with the visual vertical of the first module.

Many subjects spontaneously reported that they felt that, by the end of the experiment, they could handle any configurations with the strategies they had learned.

4.2 Differences among the configurations

We were interested in knowing whether some configurations are inherently harder than others. The results suggest that it is the case.

The results show that, on average, subjects took 3 seconds more to perform the task for the (P, PY) configurations than for the terrestrial-like configurations. Three seconds might not seem to be a very long time at first. However, we shall be aware that, on Earth, we complete the task without thinking much about it, almost instinctively; and that, although 3 seconds are not an eternity, they represent a significant amount of time. We can usually pick up our cell phone in less than 3 seconds. The 3 seconds of additional thinking time that subjects need to perform this common and frequent task shows that the inherent difficulty of the configurations is real.

Subjects were not only slower for the (P, PY) configurations. They also gave less correct answers. On average, the percentage of correct answers given decreased by 6.4% for the (P, PY) configurations compared to the (I, Y2) configurations. The additional time subjects took to answer did not enable them to provide as many good answers as for the terrestrial-like configurations.

The two types of rotation (yaw and pitch) tested seem to have a significant effect on the inherent difficulty of the configurations. Pitches, however, have a greater impact than yaws. The difference in the percentage of correct answers between the best case and the worst case for pitches is 6.4%, whereas the difference between the best case and the worst case for yaws is 2.8%. The effect of yaws on the time performance is also half that of pitches. The difference in the percentage of correct answers between the best case and the worst case for pitches is 6.4%, whereas the difference between the best case and the worst case for yaws is 2.8%. The effect of yaws on the time performance is also half that of pitches. There are several possible interpretations to this result:

- As terrestrial animals, humans are used to yaw mental rotations and yaw perspective changes. We perform many yaws daily as we move around a room, looking for something or simply turning to the door to exit and enter a new room. We also perform yaws as we navigate in the streets of a city or in the corridors of a building.
- Yaws, by definition, preserve the vertical axis, whereas pitches change its direction. One of the first things that subjects would do is to find where the visual up and down were in the first module. According to subjects' feedback after the experiment and the analysis of their head movements, subjects kept track of the vertical axis of the first room, whether or not they tried to align their head with it. When the visual axis in the second room was changed, subjects must change their reference frame to keep track of it. Tilting their head helps them track those changes.

When the second module is not pitched, a configuration is handled better by the subjects, as we might have expected. A 180 degree pitch in the way the two modules are

combined is significantly less difficult than a 90 degree pitch. One might have thought, instead, that the difficulty would increase as the angle of rotation increases, and this is indeed the case with yaws—but not with pitches. There are several possible explanations:

- With a 180 deg pitch between the two modules, the visual vertical axes have opposite senses but – ignoring direction – they stay aligned. To a certain extent, then the visual verticals remain consistent. With a 90 deg pitch, by contrast, the direction of the visual verticals of the two modules are orthogonal and nothing is preserved from the original for them to track.
- The six sides of the module can be grouped in two sets: the floor and the ceiling, and the four other common walls. When the second module is upside down with respect to the first, those groupings are preserved, but the roles of floor and ceiling are reversed. It is easy, then, to deduce where they are from remembering where the ceiling and floor were in the first module. When, however, the second module is pitched at 90 degrees, the coherence of the two groups is compromised. The floor and the ceiling become “common” walls and half of the former change to a floor and ceiling.

4.3 Possible Future Experiments

- A new training strategy emerged from this study. Subjects could be trained to identify and mentally replicate the sequence of rotations that are applied to the original modules to make the configurations. This strategy did not seem to be intuitive since most subjects used other simpler strategies. However, subjects using it seem to perform faster. We could evaluate the feasibility and the effectiveness of this training.
- In order to have a comprehensive ranking of the order of difficulty of the possible configurations, we could test configurations that include the third kind of rotation: roll. We expect that rolls are harder than yaws. It would, however, be interesting to know if rolls are simpler, harder or comparable to pitches.
- This experiment also focused exclusively on incremental rotations of 90 deg. They seemed to be the most common in space. Nevertheless, there are some other cases in which a module is rotated by only 45 deg relative to an adjacent module. The visual vertical of the Soyuz cockpit, for example, was neither aligned nor at a right angle with the visual vertical of the base block module of MIR, which confused the cosmonauts. Configurations involving intermediate angles of rotation should also be considered. Those configurations could prove harder than the one we tested and subjects might have to develop new strategies. The notion of “bridge-surfaces” might not be

as clear in the 45 deg configurations since the walls of the two adjacent modules do not have any edges in common.

- Due to time constraints, this experiment focused on the difficulties in passing from the first module of a configuration to its second module. . We could reverse the task and have subjects put a target wall from the first room in the right place while they are standing, virtually, in the second room. According to the hypothesis that preserving the visual verticals makes a configuration easier, the inherent difficulty should not depend on the cognitive roles of the two modules.

Conclusions

Spacecraft and space stations are potentially challenging venues in which to orient oneself. They are assembled from different modules and, when assembled, their local visual verticals are oriented differently, relative to one another, from the way they would be oriented on Earth. On Earth, the visual verticals are vertical. In space, there is no vertical. Spacecraft engineering constraints frequently require that the visual verticals of adjacent attached module interiors be inconsistently aligned. Astronauts and cosmonauts report that such inconsistencies create difficulties in orienting, and in visualizing their spatial relationship to the entire spacecraft, rather than just to the local module.

Previous MIT-MVL studies by Benveniste and Buckland investigated the effect of learning on orientation performance in flight configurations. Several training methods were evaluated in order to find out how to best teach crewmembers the flight configurations of their modules prior to launch, using VR techniques. Those studies focused on the difference between the ground configuration and a single flight configuration. In the flight configuration, a second module was attached to the first, reference, module with a visual vertical that was inconsistent with the visual vertical of the first module. These studies found that some configurations seemed harder than others for subjects to acquire. It was not clear, however, which properties of those configurations made them more challenging. It was the objective, in the present study, to define the critical features of a configuration that define its intrinsic difficulty. To explore the simplest possibilities, a subset of six interesting configurations was chosen. Twenty subjects learned the configurations, and the effects of configuration, trial number (both within and between configurations) and physical orientation of the head were assessed using mixed regression.

We found — as we would have expected — that the easiest configuration is the ground configuration, the configuration in which the second, attached, module is in the same presentation as it was when it was learned, at first, singly — in the first stages of training. In the other configurations tested, that second module was rotated — pitched or yawed or both — from its original presentation orientation before it was attached to the first, reference module.

The difficulty of orienting in a configuration increases as the yaw angle of rotation between the two original adjacent rooms increases from 0 to 90 deg and then 180 deg. This difficulty, however, is smaller than that generated by pitches. The variation in time performance attributed to yaws is equal to about 2% and the variation due to pitches is about three times greater. The configurations in which the second module is pitched by 180 deg pitch before being attached to the first module are harder than the configurations that are not pitched at all, and configurations with a 90 degrees pitch are the hardest of the three. In general, it took about three additional seconds to the subjects to perform the

task in the configurations with a 90 deg pitch than in the configurations with no pitches at all. This represents a significant amount of time given that we perform this task without much thinking about it and almost instantaneously on Earth. Similarly, the percentage of correct answers given by the subjects was about 6.4% lower in the configurations with a 90 deg pitch than in the configurations with no pitches. That is, performance is worst when the directions of the visual verticals of the two adjacent modules are orthogonal.

The data shows that, apart from the relative orientation of the connected modules, the orientation of subjects' body-axis with respect to the visual vertical of the local room was itself of importance. When the body-axis was perpendicular to the visual vertical of the room, subjects would complete the task more slowly unless they had reoriented their head in order to be visually upright in the room. That is, the visual verticals play an important role in spatial performance within a module in addition to creating obstacles to orientation when passing between modules.

When designing spacecraft or space stations, engineers should try to keep the visual vertical unchanged between adjacent modules and reduce the number of configurations in which there is a right angle between the vertical axes of two adjacent modules. The results of this study suggest that those measures should help crewmembers orient and navigate more easily aboard.

References

- Benveniste, D. (2004). Cognitive Conflict in Learning Three-Dimensional Space Station Structures. *Aeronautics and Astronautics*. Cambridge, MA, Massachusetts Institute of Technology. SM: 128.
- Buckland, D.A. (2006). A Training Methodology for Spatial Orientation in Spacecraft. *Aeronautics and Astronautics*. Cambridge, MA, Massachusetts Institute of Technology. SM: 72.
- Hegarty M. and Waller D. (2004). A Dissociation Between Mental Rotation and Perspective-Taking Spatial Abilities. *Intelligence*, 32 (2004): 175-191.
- Howard I.P., *Human Visual Orientation*. 1982, Toronto: Wiley.
- Howard I.P. and Childerson L., *The contribution of motion, the visual frame, and visual polarity to sensations of body tilt*. *Perception*, 1994. **23**: p. 753-762.
- Ishikawa T. and Montello D. R. (2006). Spatial Knowledge Acquisition From Direct Experience In the Environment: Individual Differences in the Development of Metric Knowledge and the Integration of Separately Learned Places. *Cognitive Psychology*, 52 (2006): 93-129.
- Knierim J.J, et al (2000). *Three-dimensional spatial selectivity of hippocampal neurons during space flight*. *Nat Neurosci*, 3: 209-210.
- Kozhevnikov M. and Hegarty M. (2001). A Dissociation Between Object Manipulation Spatial Ability and Spatial Orientation Ability. *Memory & Cognition*, 29 (5): 754-756.
- Leone G. and Taine M.C. and Droulez J. (1993). The Influence of Long-Term Practice on Mental Rotation of 3-D Objects. *Cognitive Brain Research*, 1 (1993): 241-255.
- Linn M. C and Petersen A. C. (1985). Emergence and Characterization of Sex Differences in Spatial Ability: A Meta-Analysis. *Child Development*, Vol. 56, No. 6: 1479-1498.
- Oman C.M. (2007) *Spatial Orientation and Navigation in Microgravity*, in *Spatial Processing in Navigation, Imagery and Perception*, F. Mast and L. Jancke, Editors. 2007, Springer: New York.
- Oman C.M., *Human Visual Orientation in Weightlessness*, in *Levels of Perception*, L. Harris and M. Jenkin, Editors. 2003, Springer Verlag, NY. p. 375-398.

- Pani J. R. and Chariker J. H. and Dawson T. E. and Johnson N. (2005). Acquiring New Spatial Intuitions: Learning To Reason About Rotations. *Cognitive Psychology*, 51 (2005): 285-333.
- Richards J.T., et al. (2001). *Neurovestibular Effects of Long-Duration Spaceflight: A Summary of Mir Phase 1 Experiences*. 2001 Neurovestibular Adaptation Research Team Report, National Space Biomedical Research Institute., Baylor College of Medicine, Houston, TX.
- Richards, J. T. (2000). Three-Dimensional Spatial Learning in a Virtual Space Station Node. *Aeronautics and Astronautics*. Cambridge, MA, Massachusetts Institute of Technology. SM: 144.
- Shepard, R. N. and Metzler J. (1971). Mental Rotation of Three-Dimensional Objects. *Science*, New Series, Vol. 171, No. 3972: 701-703.
- Taube J.S., Stackman R.W., Calton J.L., Oman C.M. (2004). Rat Head Direction Cell Responses in Zero-Gravity Parabolic Flight. *J Neurophysiol*, 92: 2887-2997.

APPENDIX 1: Vizard Code

```
#####  
# TRAINING IN THE INDIVIDUAL MODULES  
#Created by David Benveniste  
#Modified by Dan Buckland and Claire Cizaire  
#####
```

```
ON = 1  
OFF= 0  
HMD = ON
```

```
# timer flags
```

```
START_TRIAL = 0  
SHOW_TARGET = 1  
SHOW_CUE = 2  
RECORD_WALLS = 3  
MEMORY_TASK = 4  
SEARCH_TARGET = 5  
ORI_TARGET = 6  
DISP_FEEDBACK = 7  
END_EXP = 8  
END_TRIAL = 9  
WAIT = 10
```

```
# game pad buttons to be used
```

```
B1 = 1  
B2 = 2  
B3 = 4  
B4 = 8  
B5 = 16  
B6 = 32  
B7 = 64  
B8 = 128
```

```
import ipc  
import viz  
import sid  
import time  
import vizmat  
import math  
from string import *
```

```
# Choose which stimulus file to use  
FILENAME = 'nouveauetest.txt'
```

```
#####  
# VR SETUP (HMD AND TRACKER)
```

```

#####

if HMD == ON:
    viz.cursor(viz.OFF)
    viz.go(viz.STEREO|viz.HMD)
#    headTrack = viz.addsensor('is600')
#    # Uses only 3 dof to prevent drifting of the scene
#    # To switch back to 6 dof use command(1)
#    headTrack.command(11)

    headTrack = viz.add('intersense.dls')
    headTrack.command(3)

    viz.tracker()

else:
    viz.cursor(viz.OFF)
    viz.go()

viz.eyeheight(0)
viz.override()

##Settings
PHASE = viz.input('Phase?')
if PHASE == 1:
    PHS = '1'
elif PHASE == 2:
    PHS = '2'
else:
    PHS = '6'

NAME = viz.input('Subject Name')
SUBJECT = '\\'+NAME + '_mod1_phs' + PHS + '.txt'

#Room number (0 for phase 1 (first module) or 1 for phase 2 (second module))
if PHASE == 1:
    room_num = 0
else:
    room_num = 1

#####
#GEOMETRY
#####

viz.clearcolor(0,0,0)

## display for targets loaded in a different scene
target = []

target.append(viz.add('./experiment_rooms/no_object.wrl',viz.HEAD,2))
target.append(viz.add('./experiment_rooms/no_object.wrl',viz.HEAD,2))
target.append(viz.add('./experiment_rooms/no_object.wrl',viz.HEAD,2))
target.append(viz.add('./experiment_rooms/no_object.wrl',viz.HEAD,2))
target.append(viz.add('./experiment_rooms/no_object.wrl',viz.HEAD,2))
target.append(viz.add('./experiment_rooms/no_object.wrl',viz.HEAD,2))

```



```

## plane to display the target
target.append(viz.add('./experiment_rooms/hplane3x3.wrl',viz.HEAD,2))

for i in range(0,len(target)):
    target[i].visible(0)
    target[i].translate(0,1.82,0)
    target[i].rotate(1,0,0,-90)
    target[i].translate(0,0,4)

print target

# Generic room
room = []
for i in range(0, 6):
    print i
    room.append(viz.add('./experiment_rooms/hplane3x3.wrl',viz.WORLD,1))

## 3D objects
object = []
for i in range(0,12):
    object.append(room[i%6].add('./experiment_rooms/no_object.wrl',viz.WORLD,1))

iTex = []
# 1st module's textures
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/side_glovebox2.jpg'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/JEM_10x.jpg'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/EL-2000-00036fixed.jpg'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/side_window.jpg'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/Bnodehatchhole.gif'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/TOpx22.jpg'))

#2nd module's textures
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/RB1ax2Z2.jpg'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/node_aft_pink_hatch.gif'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/050223_controlpanel_3.jpg'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/spacesuit.jpg'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/rus_hatch_hatch.gif'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/Brservertop.jpg'))

# Red transparency texture
rTex = viz.addtexture('./experiment_rooms/textures/reticleBe3.tif')
# Yellow transparency texture
yTex = viz.addtexture('./experiment_rooms/textures/reticleBe4.tif')
# red transparency texture with yellow frame to chose the target
rTexYS = viz.addtexture('./experiment_rooms/textures/reticleBe5&square.tif')
#texture for end of phase
EndOfPhase = viz.addtexture('./experiment_rooms/textures/all_done.jpg')

#Text object to display messages not in final sequence-db
mytext = viz.add(viz.TEXT3D,'text',viz.HEAD,1)
mytext.scale(0.05,0.1,0.1)
mytext.alignment(viz.TEXT_CENTER_BOTTOM)
mytext.translate(0,0,0.5)
mytext.color(0.9,0.2,0.1)
mytext.visible(0)

```

```

#####
#POSITIONS
#####

# Defining the starting configuration of the environment
START_POS = []
START_POS.append(vizmat.Transform())
for i in range(0,12):
    START_POS.append(vizmat.Transform())
    START_POS[i].makeIdent()

START_POS[0].makeRot(0,1,0,180)
START_POS[0].postRot(1,0,0,90)
START_POS[1].postRot(1,0,0,-90)
START_POS[1].postRot(0,1,0,90)
START_POS[2].postRot(1,0,0,-90)
START_POS[3].postRot(1,0,0,-90)
START_POS[3].postRot(0,1,0,-90)
START_POS[5].postRot(1,0,0,180)

if HMD == ON:
    for i in range(0,6):
        START_POS[i].postRot(0,1,0,-90)

# Defining the current configuration of the environment
current_pos = []
for i in range(0,12):
    current_pos.append(vizmat.Transform())
    current_pos[i].makeIdent()

#transform object to go from starting position to current position
trans = vizmat.Transform()
noisePos = vizmat.Transform()

#####
#OPEN AND READ FILE FOR EXPERIMENTAL TRIALS
#####

def InitializeExp():

    global file
    global data
    global allCue
    global allTarget
    global allOri
    global allNoise

    file = open(FILENAME,'r')
    print 'opened stim file: ', FILENAME
    #'r' for reading
    data = open(SUBJECT,'a')
    print 'created output file:', SUBJECT
    #'a' for append
    data.write('%Subject name:' + SUBJECT + '\n')
    data.write('%Test performed on:' + time.ctime() + '\n')
    data.write('%stimulus name:' + FILENAME + ' module'+ str(room_num)+'\n')

```

```

data.write('%columns:'+'\n')
data.write('%Trial#'+'\t'+ 'Cue'+'\t')
data.write('Ori'+'\t'+ 'Target'+'\t'+ 'TargAns'+'\t'+ 'OriAns'+'\t')
data.write('T_Cue'+'\t'+ 'T_targ'+'\t'+ 'T_ans'+'\t'+ 'T_ori'+'\t'+ 'T_FB'+'\n')

#Experiment stimulus file
all = file.readlines()
allCue = []
allOri = []
allTarget = []
allNoise = []
print 'stimulus file with ',len(all)-1,' trials'
for i in range(1,len(all)):
    access = all[i]
    s = split(access)
    allCue.append(eval(s[0]))
    allTarget.append(eval(s[1]))
    allOri.append(eval(s[2]))
    allNoise.append(eval(s[3]))
file.close()

#####
# subroutines
#####

def showTarget(target_num):
    global target
    for obj in target:
        obj.visible(0)
    target[len(target)-1].visible(1)
    target[len(target)-1].texture(iTex[target_num],'plane')
    target[len(target)-1].appearance(viz.DECAL)
    viz.fov(39,1.333)
    viz.setscene(2)
    viz.clearcolor(0,0,0)
    if target_num in range(0,len(target)-1):
        target[target_num].visible(1) # display object

def showCues(cue):
    global fov
    viz.fov(fov,1.333)
    viz.setscene(1)
    viz.clearcolor(0,0,0)
    for wall in room:
        wall.texture(rTex,'plane')
        wall.appearance(viz.MODULATE)
    for obj in object:
        obj.visible(0)
    room[cue].texture(iTex[cue+room_num*6],'plane')
    room[cue].appearance(viz.DECAL)
    object[cue+room_num*6].visible(1)

def orientRoom(cue,ori):
    global current_pos
    global trans

```

```

#defining a intermediate transformation
#that sets the entry face to be behind the user
trans.set(START_POS[cue])
trans.invertOrtho()
trans.postMult(START_POS[2])
if HMD == OFF:
    trans.postRot(0,0,1,ori*90)
elif HMD == ON:
    trans.postRot(1,0,0,ori*90)
for i in range(0,6):
    current_pos[i].set(START_POS[i])
    current_pos[i].postMult(trans)
    room[i].update(current_pos[i])
    object[i+6*room_num].update(current_pos[i])

#####
# TIMER FOR EXPERIMENTAL TRIALS
#####

def ExptTimer(timer):

    global currentTrial
    global trans
    global noisePos
    global startTime
    global T_cue
    global T_targ
    global T_ans
    global T_ori
    global T_FB
    global button
    global dir
    global cue
    global ori
    global target
    global target_num
    global noise
    global task
    global THRESHOLD

    THRESHOLD = 0.75

    if timer == WAIT:
        if sid.buttons() in (B5,B6):
            while sid.buttons() is not 0:
                pass
            else:
                print 'button unpressed, starting task', task
                viz.starttimer(task)
        else:
            viz.starttimer(WAIT)

    if timer == SHOW_CUE:
        print 'starting trial ', currentTrial
        dir = 6

```

```

button = 0
viz.fov(fov,1.333)
startTime = time.time()
cue = allCue[currentTrial]
ori = allOri[currentTrial]
noise = allNoise[currentTrial]
orientRoom(cue,ori)
for i in range(0,6):
    room[i].texture(rTex,'plane')
    room[i].appearance(viz.MODULATE)
    object[i+room_num*6].visible(0)
room[cue].texture(iTex[cue+room_num*6],'plane')
room[cue].appearance(viz.DECAL)
object[cue+room_num*6].visible(1)
viz.setscene(1)
viz.clearcolor(0,0,0)
task = SHOW_TARGET#MEMORY_TASK
viz.starttimer(WAIT)

elif timer == SHOW_TARGET:
    dir = 6
    button = 0
    T_targ = time.time()
    T_cue = round(T_targ - startTime,2)
    target_num = allTarget[currentTrial]+6*room_num
    showTarget(target_num)
    task = MEMORY_TASK #SHOW_CUE
    viz.starttimer(WAIT)

elif timer == MEMORY_TASK:
    T_ans = time.time()
    T_targ = round(T_ans - T_targ,2)
    for i in range(0,6):
        room[i].texture(yTex,'plane')
        room[i].appearance(viz.MODULATE)
        object[i+room_num*6].visible(0)
    button = 0
    viz.setscene(1)
    viz.clearcolor(0,0,0)
    viz.fov(fov,1.333)
    task = SEARCH_TARGET
    viz.starttimer(SEARCH_TARGET)

elif timer == SEARCH_TARGET:
    if (time.time() - startTime) > 180:
        if button == 0:
            dir = 6 # no answer was given
            task = DISP_FEEDBACK
            viz.starttimer(task)
        else :
            if not (button in (B5,B6)): # no button pressed yet
                myL = viz.get(viz.HEAD_LOOK)[0:3]
                message_disp = "
                for i in range(0,6):
                    myV = current_pos[i].get()[4:8]
                    room[i].texture(yTex,'plane')

```

```

room[i].appearance(viz.MODULATE)
prod = -myV[0]*myL[0]-myV[1]*myL[1]+myV[2]*myL[2]
if i <4:
    print myV[i]

message_disp = message_disp+'wall # '+str(i)+'\nscalar

prod:'+str(prod)+'\n'

if prod > THRESHOLD:
    dir = i
    room[i].texture(rTexYS,'plane')
    room[i].appearance(viz.MODULATE)
mytext.message(message_disp)
button = sid.buttons()
mytext.message('looking at face:'+str(dir)+'\nanswer is

face:'+str(target_num))

viz.starttimer(SEARCH_TARGET)
else:
    button = 0
    T_ori = time.time()
    T_ans = round(T_ori - T_ans,2)
    task = ORI_TARGET
    noise = allNoise[currentTrial]
    noisePos.set(current_pos[dir])
    noisePos.preRot(0,1,0,90*noise)
    print target_num
    object[target_num].update(noisePos)
    room[dir].update(noisePos)
    object[target_num].visible(1)
    room[dir].texture(iTex[target_num],'plane')
    room[dir].appearance(viz.DECAL)
    while sid.buttons() is not 0:
        pass
    else:
        print 'button unpressed, starting task', task
        viz.starttimer(task)

elif timer == ORI_TARGET:
    if not (button in (B5,B6)):
        if sid.buttons() == B3:
            noise = (noise + 1)%4
            while sid.buttons()<>0:
                pass
        elif sid.buttons() == B2:
            noise = (noise - 1)%4
            while sid.buttons()<>0:
                pass
        button = sid.buttons()
        noisePos.set(current_pos[dir])
        noisePos.preRot(0,1,0,90*noise)
        object[target_num].update(noisePos)
        room[dir].update(noisePos)
        viz.starttimer(ORI_TARGET,0.1)
    else:
        task = DISP_FEEDBACK
        while sid.buttons() is not 0:
            pass

```

```

else:
    print 'button unpressed, starting task', task
    button = 0
    viz.starttimer(task)

elif timer == DISP_FEEDBACK:

    T_FB = time.time()
    T_ori = round(T_FB - T_ori,2)
    for i in range(0,6):
        room[i].update(current_pos[i])
        object[i+room_num*6].update(current_pos[i])
        room[i].texture(iTex[i+room_num*6], 'plane')
        room[i].appearance(viz.DECAL)
        object[i+room_num*6].visible(1)
    task = END_TRIAL
    viz.starttimer(WAIT)

elif timer == END_TRIAL:
    T_FB = round(time.time() - T_FB,2)
    # Writing the trial results to the result file
    Trial_R = str(currentTrial)+'\t'+str(cue)+'\t'+str(ori)
    Trial_R = Trial_R +'\t'+str(target_num)+'\t'+str(dir+room_num*6)
    Trial_R = Trial_R +'\t'+str(noise)+'\t'+str(T_targ)+'\t'+str(T_cue)
    Trial_R = Trial_R +'\t'+str(T_ans)+'\t'+str(T_ori)+'\t'+str(T_FB)
    Trial_R = Trial_R +'\n'
    data.write(Trial_R)
    #moving to next trial
    currentTrial = currentTrial + 1
    if currentTrial > len(allTarget)-1:# The first line of the file is just comments
        task = END_EXP
    else :
        task = SHOW_CUE
    viz.starttimer(task)

elif timer == END_EXP:
    data.close()
    print 'end of experiment'
    viz.setscene(2)
    viz.clearcolor(0,0,0)
    viz.fov(40,1.333)
    for obj in target:
        obj.visible(0)
    target[len(target)-1].visible(1)
    target[len(target)-1].texture(EndOfPhase, 'plane')

```

```

#####
# KEYBOARD FUNCTION TO START THE EXPERIMENT
#####

```

```

def startExpKeyboard(key):
    global currentTrial
    global button

```

```

global task
global fov

if key == 's':
    InitializeExp()
    currentTrial = 0
    orientRoom(2,0)
    for obj in object:
        obj.visible(0)
    for i in range(0,6):
        room[i].texture(iTex[i+room_num*6],'plane')
        room[i].appearance(viz.DECAL)
        object[i+room_num*6].visible(1)
    task = SHOW_CUE
    fov = 70
    viz.fov(fov,1.333)
    button = 0
elif key == ' ':
    viz.starttimer(task)
    print 'task', task
elif key == viz.KEY_UP:
    fov = fov + 1
    viz.fov(fov,1.333)
    print 'field of view set to', fov, 'deg in the vertical direction'
elif key == viz.KEY_DOWN:
    fov = fov - 1
    viz.fov(fov,1.333)
    print 'field of view set to', fov, 'deg in the vertical direction'
elif key == 'h': #get position of the hand
    print handTrack.get()
elif key == 't':
    message = viz.input('enter new text to display')
elif key == '+':
    viz.translate(viz.HEAD_POS,Tvec[0],Tvec[1],Tvec[2])
elif key == '-':
    viz.translate(viz.HEAD_POS,-Tvec[0],-Tvec[1],-Tvec[2])
elif key == 'z': #back up the view point along z-axis
    Tvec = 0,0,1
elif key == 'y': #back up the view point along z-axis
    Tvec = 0,1,0
elif key == 'x': #back up the view point along x-axis
    Tvec = 1,0,0
elif key == 'p': #pitch the view point 90 deg
    viz.rotate(viz.BODY_ORI,0,90,0)
elif key == 'w': #yaw the view point 90 deg
    viz.rotate(viz.BODY_ORI,90,0,0)

fov = 70
viz.callback(viz.TIMER_EVENT,'ExptTimer')
viz.callback(viz.KEYBOARD_EVENT,'startExpKeyboard')

```



```
#####
#TRAINING AND TESTING IN THE COMBINED MODULES
#Seven possible configurations: P, Y2, P2Y, I, PY, P2.
#Created by David Benveniste
#Modified by Claire Cizaire
#####
#####
#The subject should just press "enter" to start the testing session
# automatically after the training.
#The head movements are recorded.
#####
```

```
ON = 1
OFF= 0
```

```
HMD = ON
```

```
# timer flags
```

```
START_TRIAL = 0
SHOW_CUE = 2
MEMORY_TASK = 4
SEARCH_TARGET = 6
DISP_FEEDBACK = 7
END_EXP = 8
END_TRIAL = 9
WAIT = 10
END_TRAINING = 11
SHOW_FEEDBACK = 12
```

```
# Game pad buttons to be used
```

```
B1 = 1
B2 = 2
B3 = 4
B4 = 8
B5 = 16
B6 = 32
B7 = 64
B8 = 128
B9 = 256
B10 = 512
```

```
import viz
import sid
import time
import vizmat
import math
from string import *
```

```
#Define the configuration numbers.
#Each configuration is assigned a number that
#will be used in the subroutine "place2ndModule"
```

```
P=1 #Simple pitch
```

```
Y=2 #Simple yaw
P2=3 #180deg pitch
PY=4 #Combined yaw and pitch (matrix notation)
P2Y=5 #Combined yaw and 180deg pitch
I=6
Y2=7
```

```
#####
# VR SETUP (HMD AND TRACKER)
#####

if HMD == ON:
    viz.cursor(viz.OFF)
    viz.mouse(viz.OFF)
    viz.go(viz.STEREO|viz.HMD)
    headTrack = viz.add('intersense.dls')
    headTrack.reset()
    headTrack.command(3)
    viz.tracker()

else:
    viz.cursor(viz.OFF)
    viz.go()

viz.eyeheight(0)
viz.override()
fov = 70
viz.fov(fov,1.333)
view = viz.get(viz.MAIN_VIEWPOINT)

#placeholder for number of tries
tries = 1

CONF_ORDER = viz.input('Configuration Order?')

NAME = viz.input('Subject Name?')

if CONF_ORDER == 1:
    condition = P
    FILENAME = 'CizaireTrialsP.txt'
    CONF = '1'
elif CONF_ORDER == 2:
    condition = Y2
    FILENAME = 'CizaireTrialsY2.txt'
    CONF = '2'
elif CONF_ORDER == 3:
    condition = P2Y
    FILENAME = 'CizaireTrialsP2Y.txt'
    CONF = '3'
elif CONF_ORDER == 4:
    condition = I
    FILENAME = 'CizaireTrialsI.txt'
    CONF = '4'
elif CONF_ORDER == 5:
    condition = P2
    FILENAME = 'CizaireTrialsP2.txt'
```

```

        CONF = '5'
elif CONF_ORDER == 6:
    condition = PY
    FILENAME = 'CizaireTrialsPY.txt'
    CONF = '6'
elif CONF_ORDER == 7:
    condition = P2Y
    FILENAME = 'CizaireTrialsP2Y.txt'
    CONF = '7'

SUBJECT = '..\\'+NAME + '_mod2_conf' + CONF + '.txt'

#####
#POSITIONS
#####

# Defining the starting configuration of the environment
# method to set all the walls in one room
# to their correct position
# a room consists of a table with 6 elements
# the numbers follow the mnemonics proposed
# by Jason Richards:
# 0, behind; 1, right; 2, ahead; 3, left;
# 4, floor; 5, ceiling
# These numbers should be considered modulo 6 since
# there are 2 rooms (indexes 0 to 5 and 6 to 11)

START_POS = []
START_POS.append(vizmat.Transform())
for i in range(0,12):
    START_POS.append(vizmat.Transform())
    START_POS[i].makeIdent()

#START_POS[i] describes the ABSOLUTE position of the i_th wall in rotation
START_POS[1].postRot(0,1,0,-90)
START_POS[2].postRot(0,1,0,180)
START_POS[3].postRot(0,1,0,90)
START_POS[4].postRot(0,0,1,180)
START_POS[4].postRot(1,0,0,-90)
START_POS[5].postRot(0,0,1,180)
START_POS[5].postRot(1,0,0,+90)

#the relative position of walls in second module are the same as in the first one this #moves them to starting
position 4m away in z
for i in range(0,6):
    START_POS[i+6].set(START_POS[i])
    START_POS[i+6].postTrans(0,0,4)

current_pos = []
for i in range(0,12):
    current_pos.append(vizmat.Transform())
    current_pos[i].makeIdent()

#transform object to go from starting position to current position
trans = vizmat.Transform()
noisePos = vizmat.Transform()

```

```

#object transform (corrective term for the start position)
OBJ_T =vizmat.Transform()
OBJ_T.makeRot(0,1,0,180)
OBJ_T.postRot(1,0,0,90)

#####
#GEOMETRY
#####

target = []
target.append(viz.add('./experiment_rooms/wall0.wrl',viz.HEAD,2))
for i in range(0,len(target)):
    target[i].visible(0)
    target[i].translate(0,1.82,0)
    target[i].rotate(0,1,0,180)
    target[i].translate(0,0,4)

room = []
for i in range(0,12):
    room.append(viz.add('./experiment_rooms/wall0.wrl',viz.WORLD,1))

#Objects to place room textures on
object = []
for i in range(0,12):
    object.append(room[i].add('./experiment_rooms/no_object.wrl',viz.WORLD,1))

for i in range(0,12):
    object[i].update(OBJ_T)

iTex = []
# 1st module's textures
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/side_glovebox2.jpg'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/JEM_10x.jpg'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/EL-2000-00036fixed.jpg'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/side_window.jpg'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/Bnodehatchhole.gif'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/TOPx22.jpg'))

#2nd module's textures
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/RB1ax2Z2.jpg'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/node_aft_pink_hatch.gif'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/050223_controlpanel_3.jpg'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/spacesuit.jpg'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/rus_hatch_hatch.gif'))
iTex.append(viz.addtexture('./experiment_rooms/textures/tex/Brservtop.jpg'))

# Red transparency texture
rTex = viz.addtexture('./experiment_rooms/textures/reticleBe3.tif')
rTexh = viz.addtexture('./experiment_rooms/textures/reticleBe3h2.gif')
# Yellow transparency texture
yTex = viz.addtexture('./experiment_rooms/textures/reticleBe4.tif')
yTexh = viz.addtexture('./experiment_rooms/textures/reticleBeyelh.gif')
# red transparency texture with red cross to chose the target
rTexYS = viz.addtexture('./experiment_rooms/textures/reticleBe5&square.tif')
#texture for end of phase
EndOfPhase = viz.addtexture('./experiment_rooms/textures/all_done.jpg')

```

```

EndOfTraining = viz.addtexture('./experiment_rooms/textures/Congratulations.gif')
#texture for wrong hatch choice
wrongWall = viz.addtexture('./experiment_rooms/textures/wrongWall.tif')
mytext = viz.add(viz.TEXT3D,'text',viz.HEAD,1)
mytext.scale(0.02,0.05,0.1)

#Rotate textures to start positions and place on 3D planes in [room]
for i in range(0,12):
    room[i].update(START_POS[i])
    room[i].texture(iTex[i],'plane')

#####
# OPEN AND READ FILE FOR EXPERIMENTAL TRIALS
#####

def InitializeExp():

    global file
    global data
    global allCue
    global allTarget
    global allOri
    global allNoise
    global allFB

    file = open(FILENAME,'r')
    print 'opened stim file: ', FILENAME
    #'r' for reading
    data = open(SUBJECT,'a')
    print 'created output file:', SUBJECT
    #'a' for append
    data.write('% Subject name: ' + SUBJECT + '\n')
    data.write('% Test performed on: ' + time.ctime() + '\n')
    data.write('% Stimulus name: ' + FILENAME)
    data.write('% The configuration order was: ' + str(CONF)+'\n')
    data.write('% Columns:+'\n')
    data.write('% Trial'+\t+'Cue'+\t+'Ori'+\t+'Target'+\t+'Noise'+\t')
    data.write('TargAns'+\t+'OriAns'+\t')
    data.write('T_Cue'+\t+'T_target'+\t+'T_total'+\t+'T_FB'+\t')
    data.write('HX1'+\t+'HY1'+\t+'HZ1'+\t')
    data.write('HX_position'+\t+'HY1_position'+\t+'HZ1_position'+\t')
    data.write('HX_ori'+\t+'HY_ori'+\t+'HZ_ori'+\t')
    data.write('HX2'+\t+'HY2'+\t+'HZ2'+\t')
    data.write('HX3'+\t+'HY3'+\t+'HZ3'+\t')
    data.write('HX_vector1'+\t+'HY_vector1'+\t+'HZ_vector1'+\t')
    data.write('HX_vector_position'+\t+'HY1_vector_position'+\t+'HZ1_vector_position'+\t')
    data.write('HX_vector_ori'+\t+'HY_vector_ori'+\t+'HZ_vector_ori'+\t')
    data.write('HX_vector2'+\t+'HY_vector2'+\t+'HZ_vector2'+\t')
    data.write('HX_vector3'+\t+'HY_vector3'+\t+'HZ_vector3'+\t+'\n'+\n')

    #Experiment stimulus file
    all = file.readlines()
    allCue = []
    allTarget = []
    allEntry = []
    allOri = []

```

```

allNoise = []
allFB = []
print 'stimulus file with ',len(all)-1,' trials'
for i in range(1,len(all)):
    access = all[i]
    s = split(access)
    allCue.append(eval(s[0]))
    allTarget.append(eval(s[1]))
    allOri.append(eval(s[2]))
    allNoise.append(eval(s[3]))
    allFB.append(eval(s[4]))

file.close()

#####
# SUBROUTINES
#####

def place2ndModule(cond):
    global hatchface
    global backface
    global aheadface
    global START_POS
    temp = vizmat.Transform()
    if cond == P:
        hatchface = 2
        backface = 11
        aheadface = 10
        temp.makeRot(-1,0,0,90)
    elif cond == Y:
        hatchface = 2
        backface = 9
        aheadface = 7
        temp.makeRot(0,-1,0,90)
    elif cond == P2:
        hatchface = 2
        backface = 8
        aheadface = 6
        temp.makeRot(1,0,0,180)
    elif cond == PY:
        hatchface = 2
        backface = 11
        aheadface = 10
        temp.makeRot(0,-1,0,90)
        temp.postRot(-1,0,0,90)
    elif cond == P2Y:
        hatchface = 2
        backface = 7
        aheadface = 9
        temp.makeRot(0,-1,0,90)
        temp.postRot(1,0,0,180)
    elif cond == I:
        hatchface = 2
        backface = 6
        aheadface = 8
    elif cond == Y2:

```

```

        hatchface = 2
        backface = 8
        aheadface = 6
        temp.makeRot(0,-1,0,180)
temp.postTrans(0,0,3.5)
for i in range(0,6):
    START_POS[i+6].set(START_POS[i])
    START_POS[i+6].postMult(temp)
if HMD == ON:
    for i in range(0,12):
        START_POS[i].postRot(0,1,0,-90)
for i in range(0,12):
    room[i].update(START_POS[i])

def orientRoom(cue,ori):
    global trans
    global current_pos
    trans.set(START_POS[cue])
    trans.invertOrtho()
    trans.postMult(START_POS[2])
    if HMD == OFF:
        trans.postRot(0,0,1,ori*90)
    elif HMD == ON:
        trans.postRot(1,0,0,ori*90)
    for i in range(0,12):
        current_pos[i].set(START_POS[i])
        current_pos[i].postMult(trans)
        room[i].update(current_pos[i])

def showTarget(Ltarget_num):
    global target
    global iTex
    for obj in target:
        obj.visible(0)
    target[0].visible(1)
    target[0].texture(iTex[Ltarget_num],'plane')
    target[0].appearance(viz.DECAL)
    viz.fov(39,1.333)
    viz.setscene(2)
    viz.clearcolor(0,0,0)
    print 'tarrrr'

#####
# TIMER FOR EXPERIMENTAL TRIALS
#####

def ExptTimer(timer):

    global currentTrial
    global trans
    global noisePos
    global start_show_cue
    global start_memory_task
    global start_search_target

```

```

global start_disp_feedback
global start_end_trial
global T_target
global T_total
global T_cue
global button
global dirT
global cue
global ori
global target
global target_num
global target_noise
global noise
global noise2
global task
global hx, hy, hz
global hx2, hy2, hz2
global hx1, hy1, hz1
global hx3, hy3, hz3
global hx_position, hy_position, hz_position
global hx_ori, hy_ori, hz_ori
global hx_vector, hy_vector, hz_vector
global hx_vector_position, hy_vector_position, hz_vector_position
global hx_vector_ori, hy_vector_ori, hz_vector_ori
global hx_vector2, hy_vector2, hz_vector2
global hx_vector1, hy_vector1, hz_vector1
global hx_vector3, hy_vector3, hz_vector3
global THRESHOLD
global feedback

```

```

THRESHOLD = 0.75

```

```

if timer == WAIT: # Waits for the buttons B5 or B6 to be pressed AND released in order to start
the next task

```

```

    if sid.buttons() in (B5,B6):
        while sid.buttons() is not 0:
            pass
        else:
            print 'button unpressed, starting task', task
            viz.starttimer(task)
    else:
        head_ori4 = viz.get(viz.HEAD_EULER)
        hx = head_ori4[0]
        hy = head_ori4[1]
        hz = head_ori4[2]
        print head_ori4[0], head_ori4[1], head_ori4[2]
        head_ori_vector4 = viz.get(viz.VIEW_LOOK)
        hx_vector = head_ori_vector4[0]
        hy_vector = head_ori_vector4[1]
        hz_vector = head_ori_vector4[2]
        viz.starttimer(WAIT)

```

```

if timer == SHOW_CUE: # Gives the opportunity to find his/her orientation in the first module
    button = 0
    viz.fov(fov,1.333)
    start_show_cue = time.time()

```



```

mytext.message('SHOW_CUE')
print 'SHOW_CUE'
cue = allCue[currentTrial]
ori = allOri[currentTrial]
target_num = allTarget[currentTrial]+6
target_noise = allNoise[currentTrial]
noise = allNoise[currentTrial]
feedback = allFB[currentTrial]
dirT = aheadface + allNoise[currentTrial]
orientRoom(cue,ori) #The subject is facing the Cue Wall but cannot see anything yet
for i in range(0,6): #The first module is outlined in red
    room[i].texture(rTex,'plane')
    object[i].visible(0)
    room[i].visible(1)
    room[i+6].visible(0)
#
room[4].texture(rTexh,'plane')
room[cue].texture(iTex[cue],'plane') #The texture of the Cue Wall is now visible
object[cue].visible(1)
viz.setscene(1)
#
head_ori1 = headTrack.get() #Get head position/orientation when a subject looked at the
second module first
task = MEMORY_TASK
viz.starttimer(WAIT) # The subject is standing in the first module, facing the Cue Wall.
# The module is outlined in red and only the Cue Wall has its texture pasted on.
# The next step for the subject is to press the button B5 or B6 to show that he/she now
know
    # he/she is.

elif timer == MEMORY_TASK: # Outlines the two modules and puts the target texture
#on the far wall of the seconde module in testing mode.
    print hx, hy, hz
    hx1 = hx
    hy1 = hy
    hz1 = hz
    hx_vector1 = hx_vector
    hy_vector1 = hy_vector
    hz_vector1 = hz_vector
    start_memory_task = time.time()
    T_cue = round(start_memory_task - start_show_cue,2)
    mytext.message('MEMORY_TASK')
    print 'MEMORY_TASK'

    for i in range(0,6): #The 1rst module is outlined in red and the 2nd is now outlined in
yellow
        room[i].texture(rTex,'plane')
        room[i+6].texture(yTex,'plane')
        room[i+6].visible(1)
        object[i].visible(0)
        object[i+6].visible(0)
        noisePos.set(current_pos[aheadface])
        noisePos.preRot(0,0,1,90*noise) #Defines a transformation that when applied to the
aheadface gives a "random" orientation to it
        print 'aheadface= ', aheadface
        print 'target= ', target_num
        room[aheadface].texture(iTex[target_num],'plane') #Adds the target texture to the far
wall

```

```

orientation
    room[aheadface].update(noisePos) #The target texture is on the far wall, in a "random"

    button = 0
    viz.setscene(1)
    viz.clearcolor(0,0,0)
    viz.fov(fov,1.333)
    task = SEARCH_TARGET
    viz.starttimer(SEARCH_TARGET)

    elif timer == SEARCH_TARGET: # The subject has to put the texture on the right wall with the
gamepad. The position is updated.
    start_search_target = time.time()
    print 'SEARCH_TARGET'
    button = sid.buttons()
    if button == B10:
        dirT = dirT + 1 # dirT is used to change the wall the texture is on
        if (dirT%6) == (backface%6): # if the selected wall is the backface, jump it
            dirT = dirT + 1
        head_ori3 = viz.get(viz.HEAD_EULER)
        hx_position = head_ori3[0]
        hy_position = head_ori3[1]
        hz_position = head_ori3[2]
        head_ori_vector3 = viz.get(viz.VIEW_LOOK)
        hx_vector_position = head_ori_vector3[0]
        hy_vector_position = head_ori_vector3[1]
        hz_vector_position = head_ori_vector3[2]
    elif button == B9:
        dirT = dirT - 1
        if (dirT%6) == (backface%6):
            dirT = dirT - 1
        head_ori3 = viz.get(viz.HEAD_EULER)
        hx_position = head_ori3[0]
        hy_position = head_ori3[1]
        hz_position = head_ori3[2]
        head_ori_vector3 = viz.get(viz.VIEW_LOOK)
        hx_vector_position = head_ori_vector3[0]
        hy_vector_position = head_ori_vector3[1]
        hz_vector_position = head_ori_vector3[2]
    elif button == B3: # noise is used to change the orientation of the texture
        noise = noise + 1
        head_ori2 = viz.get(viz.HEAD_EULER)
        hx_ori = head_ori2[0]
        hy_ori = head_ori2[1]
        hz_ori = head_ori2[2]
        head_ori_vector2 = viz.get(viz.VIEW_LOOK)
        hx_vector_ori = head_ori_vector2[0]
        hy_vector_ori = head_ori_vector2[1]
        hz_vector_ori = head_ori_vector2[2]
    elif button == B2:
        noise = noise - 1
        head_ori2 = viz.get(viz.HEAD_EULER)
        hx_ori = head_ori2[0]
        hy_ori = head_ori2[1]
        hz_ori = head_ori2[2]
        head_ori_vector2 = viz.get(viz.VIEW_LOOK)
        hx_vector_ori = head_ori_vector2[0]

```

```

        hy_vector_ori = head_ori_vector2[1]
        hz_vector_ori = head_ori_vector2[2]
elif button in (B5,B6): #moving on to the next step
    button = 0
    if feedback == 1:
        task = DISP_FEEDBACK
    else:
        task = END_TRIAL
if (dirT%6) == (backface%6):
    dirT = aheadface
dirT = (dirT % 6) + 6
noise = noise%4
noisePos.set(current_pos[dirT])
noisePos.preRot(0,0,1,90*noise)
for i in range(6,12):
    if i <> dirT:
        room[i].texture(yTex,'plane')
        room[i].update(current_pos[i])
    room[dirT].update(noisePos)      # The environment is updated: the texture is on the
selected wall with the selected orientation
room[dirT].texture(iTex[target_num],'plane')
# in case the button is still pressed down
head_ori2 = viz.get(viz.HEAD_EULER) # Euler Angles when the
hx2 = head_ori2[0]
hy2 = head_ori2[1]
hz2 = head_ori2[2]
print hx2, hy2, hz2
head_ori_vector2 = viz.get(viz.VIEW_LOOK) # Euler Angles when the
hx_vector2 = head_ori_vector2[0]
hy_vector2 = head_ori_vector2[1]
hz_vector2 = head_ori_vector2[2]
print hx_vector2, hy_vector2, hz_vector2
while sid.buttons() is not 0:
    pass
viz.starttimer(task,0.1)

elif timer == DISP_FEEDBACK: #Displays all walls in correct orientation
start_disp_feedback = time.time()
print 'DISP_FEEDBACK'
mytext.message('DISP_FEEDBACK')
room[hatchface].visible(0)#spinto(rot_Axis[0],rot_Axis[1],rot_Axis[2],0,3,viz.TIME)
for i in range(0,12):
    room[i].texture(iTex[i],'plane')
    room[i].update(current_pos[i])
    object[i].visible(1)
task = END_TRIAL
viz.starttimer(WAIT)

elif timer == END_TRIAL:
hx3 = hx
hy3 = hy
hz3 = hz
hx_vector3 = hx_vector
hy_vector3 = hy_vector
hz_vector3 = hz_vector
start_end_trial = time.time()

```

```

print 'hatch should be back in place'
if feedback == 1:
    T_target = start_disp_feedback - start_memory_task
    T_total = start_disp_feedback - start_show_cue
    T_FB = start_end_trial - start_disp_feedback
else:
    start_disp_feedback = 0
    T_target = start_end_trial - start_memory_task
    T_total = start_end_trial - start_show_cue
    T_FB = 0
mytext.message('END_TRIAL')

# Writing the trial results to the result file

data.write(str(currentTrial)+'\t'+str(cue)+'\t'+str(ori)+'\t'+str(target_num)+'\t'+str(target_noise)+'\t')
data.write(str(dirT)+'\t'+str(noise)+'\t')
data.write(str(T_cue)+'\t'+str(T_target)+'\t'+str(T_total)+'\t'+str(T_FB)+'\t')
data.write(str(hx1)+'\t'+str(hy1)+'\t'+str(hz1)+'\t')
data.write(str(hx_position)+'\t'+str(hy_position)+'\t'+str(hz_position)+'\t')
data.write(str(hx_ori)+'\t'+str(hy_ori)+'\t'+str(hz_ori)+'\t')
data.write(str(hx2)+'\t'+str(hy2)+'\t'+str(hz2)+'\t')
data.write(str(hx3)+'\t'+str(hy3)+'\t'+str(hz3)+'\t')

data.write(str(hx_vector1)+'\t'+str(hy_vector1)+'\t'+str(hz_vector1)+'\t'+str(hx_vector2)+'\t'+str(hy_vector2)+'\t'+str(hz_vector2)+'\t')

data.write(str(hx_vector_position)+'\t'+str(hy_vector_position)+'\t'+str(hz_vector_position)+'\t')
data.write(str(hx_vector_ori)+'\t'+str(hy_vector_ori)+'\t'+str(hz_vector_ori)+'\t')
data.write(str(hx_vector3)+'\t'+str(hy_vector3)+'\t'+str(hz_vector3)+'\t')
data.write('\n')
# Moving to next trial
currentTrial = currentTrial + 1
if currentTrial > len(allTarget)-1:# The first line of the file is just comments
    task = END_EXP
elif currentTrial == 6:
    task = END_TRAINING
else :
    tries = 1
    task = SHOW_CUE
viz.starttimer(task)

elif timer == END_EXP:
    data.close()
    print 'End of Phase'
    viz.setscene(2)
    viz.clearcolor(0,0,0)
    viz.fov(40,1.333)
    for obj in target:
        obj.visible(0)
    target[len(target)-1].visible(1)
    target[len(target)-1].texture(EndOfPhase,'plane')

elif timer == END_TRAINING:
    print 'End of Training'
    tries = 1
    task = SHOW_FEEDBACK

```

```

viz.setscene(2)
viz.clearcolor(0,0,0)
viz.fov(40,1.333)
for obj in target:
    obj.visible(0)
target[len(target)-1].visible(1)
target[len(target)-1].texture(EndOfTraining,'plane')
viz.starttimer(WAIT)

elif timer == SHOW_FEEDBACK:
    print 'Show feedback before testing'
    task = SHOW_CUE
    viz.fov(fov,1.333)
    for i in range(0,12):
        room[i].texture(iTex[i],'plane')
        room[i].visible(1)
    orientRoom(2,0)
    room[hatchface].visible(0)
    viz.setscene(1)
    viz.starttimer(WAIT)

#####
#KEYBOARD FUNCTION TO START THE EXPERIMENT
#####

def startExpKeyboard(key):
    global currentTrial
    global button
    global task
    global fov
    global cue_sequence
    global condition

    if key == 's':
        InitializeExp()
        currentTrial = 0
        orientRoom(2,0)
        for obj in object:
            obj.visible(1)
        for i in range(0,12):
            room[i].texture(iTex[i],'plane')
        task = SHOW_CUE
        button = 0
        for i in range(0,12):
            room[i].visible(1)
        room[hatchface].visible(0)
        cue_sequence = [0,0]
        viz.starttimer(WAIT)

if HMD == OFF:
    def mousemove(x,y):
        euler = view.get(viz.HEAD_EULER)

```

```

euler[0] += x*0.55
euler[0] = viz.clamp(euler[0],-180.0,180.0)
euler[1] += y*0.5
euler[1] = viz.clamp(euler[1],-90.0,90.0)
view.rotate(euler,viz.HEAD_ORI)

def mousedown(button):
    if button == viz.MOUSEBUTTON_LEFT:
        view.reset(viz.HEAD_ORI)
    elif button == viz.MOUSEBUTTON_RIGHT:
        view.reset(viz.HEAD_ORI)

viz.callback(viz.MOUSEBUTTON_EVENT,mousedown)
viz.callback(viz.MOUSEMOVE_EVENT,mousemove)
viz.mousedata(viz.RELATIVE,viz.RELATIVE)

place2ndModule(condition)
viz.callback(viz.TIMER_EVENT,'ExptTimer')
viz.callback(viz.KEYBOARD_EVENT,'startExpKeyboard')

```

APPENDIX 2: Trial Sequences

Sequence of trials for the training in the individual modules

Cue	Target	Orientation	Noise
2	4	0	2
3	5	0	2
1	5	0	0
0	0	1	1
4	5	1	0
5	4	0	1
3	1	3	2
1	2	1	0
2	3	3	3
5	0	2	0
0	1	3	0
4	3	2	3
1	2	3	0
0	2	2	0
3	4	2	1
5	1	1	1
2	3	2	0
4	0	3	2

Sequence of trials for the I configuration

Cue	Target	Orientation	Noise	Feedback
1	2	0	2	1
5	3	2	1	1
1	4	1	0	1
5	2	0	0	1
1	3	1	1	1
5	4	1	3	1
1	2	0	2	0
5	3	0	1	0
1	4	2	1	0
5	2	1	2	0
1	4	0	0	0
5	3	2	1	0
1	2	2	3	0
5	3	1	2	0
1	4	1	0	0
5	2	2	1	0
1	3	0	0	0
5	4	2	0	0
1	2	1	3	0
5	2	0	0	0
1	3	2	0	0
5	4	1	3	0
1	3	1	1	0
5	4	0	3	0

Sequence of trials for the P configuration

Cue	Target	Orientation	Noise	Feedback
1	4	0	2	1
5	3	2	1	1
1	0	1	0	1
5	4	0	0	1
1	3	1	1	1
5	0	1	3	1
1	4	0	2	0
5	3	0	1	0
1	0	2	1	0
5	4	1	2	0
1	0	0	0	0
5	3	2	1	0
1	4	2	3	0
5	3	1	2	0
1	0	1	0	0
5	4	2	1	0
1	3	0	0	0
5	0	2	0	0
1	4	1	3	0
5	4	0	0	0
1	3	2	0	0
5	0	1	3	0
1	3	1	1	0
5	0	0	3	0

Sequence of trials of the P2 configuration

Cue	Target	Orientation	Noise	Feedback
1	0	0	2	1
5	3	2	1	1
1	5	1	0	1
5	0	0	0	1
1	3	1	1	1
5	5	1	3	1
1	0	0	2	0
5	3	0	1	0
1	5	2	1	0
5	0	1	2	0
1	5	0	0	0
5	3	2	1	0
1	0	2	3	0
5	3	1	2	0
1	5	1	0	0
5	0	2	1	0
1	3	0	0	0
5	5	2	0	0
1	0	1	3	0
5	0	0	0	0
1	3	2	0	0
5	5	1	3	0
1	3	1	1	0
5	5	0	3	0

Sequence of trials for the P2Y configuration

Cue	Target	Orientation	Noise	Feedback
1	3	0	2	1
5	2	2	1	1
1	5	1	0	1
5	3	0	0	1
1	2	1	1	1
5	5	1	3	1
1	3	0	2	0
5	2	0	1	0
1	5	2	1	0
5	3	1	2	0
1	5	0	0	0
5	2	2	1	0
1	3	2	3	0
5	2	1	2	0
1	5	1	0	0
5	3	2	1	0
1	2	0	0	0
5	5	2	0	0
1	3	1	3	0
5	3	0	0	0
1	2	2	0	0
5	5	1	3	0
1	2	1	1	0
5	5	0	3	0

Sequence of trials for the PY configuration

Cue	Target	Orientation	Noise	Feedback
1	4	0	2	1
5	2	2	1	1
1	3	1	0	1
5	4	0	0	1
1	2	1	1	1
5	3	1	3	1
1	4	0	2	0
5	2	0	1	0
1	3	2	1	0
5	4	1	2	0
1	3	0	0	0
5	2	2	1	0
1	4	2	3	0
5	2	1	2	0
1	3	1	0	0
5	4	2	1	0
1	2	0	0	0
5	3	2	0	0
1	4	1	3	0
5	4	0	0	0
1	2	2	0	0
5	3	1	3	0
1	2	1	1	0
5	3	0	3	0

Sequence of trials for the Y2 configuration

Cue	Target	Orientation	Noise	Feedback
1	0	0	2	1
5	1	2	1	1
1	4	1	0	1
5	0	0	0	1
1	1	1	1	1
5	4	1	3	1
1	0	0	2	0
5	1	0	1	0
1	4	2	1	0
5	0	1	2	0
1	4	0	0	0
5	1	2	1	0
1	0	2	3	0
5	1	1	2	0
1	4	1	0	0
5	0	2	1	0
1	1	0	0	0
5	4	2	0	0
1	0	1	3	0
5	0	0	0	0
1	1	2	0	0
5	4	1	3	0
1	1	1	1	0
5	4	0	3	0

APPENDIX 3: Verbal Tour

Informed Consent Form

Procedure briefing

In today's experiment, you will have to orient yourself in a two-module spacecraft.

There are different ways of attaching two modules in space. The floors of the two modules are not necessarily aligned. You will be confronted to different flight situations and asked to orient yourself in the spacecraft.

You will be wearing a head mounted display, so you can look around the inside of each of the modules. Your main task is to remember where equipment and displays are located in order to orient yourself in the module and predict where other equipment should be.

You will first learn the interior layout of each individual module in three steps:

1. I will give you a verbal tour of the individual module while you are wearing the head mounted display;
2. Once you have memorized the interior layout, I will ask you to remove the HMD and tell me what you remember;
3. You'll then put the HMD back on and we will switch to computerized training.

Once you know the two modules, we will attach them to constitute configurations. This will be the second section of the experiment. There will be 7 different configurations.

For each configuration, you will first be trained and then tested. The testing task will consist in orienting yourself from a cue wall, and then put a target wall in the right place with the right orientation.

You will earn \$10/hour for your participation in the experiment.

The potential risks of the experiment are boredom, headaches, motion sickness and dizziness. If at one point of the experiment, you do not feel well, let me know so that I can help you.

Any questions so far?

Individual modules: demonstration, training

Let's start:

I am going to give you a verbal tour of the “Command” module. You will have to tell me what you remember from the tour. You should try to learn the interior layout in order to be able to orient yourself.

After the verbal tour, we will start the computerized training. Before you put on the head mounted display, let’s look at the computer screen together, and I will explain you what the computerized training is like and show you how to use the game pad.

[Experimenter shows what the task is like on the computer screens and explains what buttons to use on the game pad.]

To challenge your memory, you will see only one wall at a time. The rest of the walls will appear just as red-wire framed edges. From your view of this one wall, and your knowledge of what’s where, see if you can figure out how you are oriented in the module. After you think you know how you are oriented based on this cue, push the “enter” button on the left side of the game pad. Next, the computer will show you a picture of one of the missing walls. We call this the “target” picture. Your task now is to place this target pictures on the correct surface, and rotate it so it is in the proper relative orientation. To do this, push the “enter” button a second time. The wire frame will turn yellow, and a big red cross will appear on the wall you are currently looking at. Turn your head so the cross moves to the wall where the picture belongs. Once the X is on the correct surface, push the “enter” button again with your left hand. This locks the target to the wall surface, but the orientation may not be correct. Push the “rotate” button as many times as necessary till the target is in the correct orientation. Once you are satisfied with your answer, push “enter” to have the feedback. All the walls will then appear in their correct orientation, and you will be able to see if you positioned the target correctly or not. If you made a mistake, the computer will correct it so that you can learn. Later on, during the second section, you will not always be told what your mistakes are. We would like you to do the task as quickly as possible, giving equal priority to speed and accuracy throughout the entire experiment.

Any questions?

Please put the HMD on.

Tour of the first module

Here is the first module. Let me give you a tour of this module. At the end of the tour, you will have some time to look around and then I will ask you what you recall. The module is called the “Command” module.

The wall you are facing is the control panel. There are 7 screens and keyboards for the pilots to use. There are some more monitor switches at the top.

Now turn to face the ultrasound panel on your right. This equipment is designed for experiment on animals. There are three lockers at the right bottom corner. They are used to store water and food for the animals. You have a control screen at the top right corner.

Turn to your right again to face the glove box. There are 12 storage boxes on the left. Next to the boxes, there is the glove box with the two white circles for astronauts to put their hands in and conduct experiment on the animals.

Please turn all the way around (so you won't get tangled in the cord.) and you will see another wall with a window. There is a French astronaut outside looking at you through the window. At the right corner, you have an oxygen mask.

Now, if you look up, you have the ceiling with two long lights on each side. In the middle, you have three lockers. Only two of them are full: they have labels. There is also one yellow bar on one side.

When you look down, you can see a yellow and black hatch. There is one yellow bar as well.

Please take your time to look around and start memorizing where things are. Let me know when you are ready to take off the head mounted display to give me feedback.

Tour of the second module

Like in the previous module, I will first give you a guided tour of this module and then ask you what you recall.

This module is called the "EVA" module.

You are facing the control panel. There are only two screens. You have orange emergency kits below the screen in the middle.

Now turn to you right. This is a pink hatch. Notice the speed signs on top of it. Small objects can be kept on the right of the hatch with the Velcro. Do not worry about what might be behind this hatch. We won't look at that part of the spacecraft.

One more turn to the right and you will see a wall equipped with scientific instruments. There are two columns of instruments. On the right, you can see a work bench.

Please turn all the way around (so you won't get tangled in the cord) and you will see a green storage place with a space suit.

Now if you look up, you can see the ceiling with a lot of small lights and a few storage places. There is also a white pipe that runs from one part of the ceiling to another one.

If you look down, you can see another hatch. We will not worry about what might be behind this hatch.

Combined module training and testing:

Before we start the first phase, let me describe the task you will have to achieve.

You will always be in the first module. As before, you will first see one wall of the module that you are in – the cue wall. Based on your prior training, you should be able to figure out how you are oriented, and where things are in the local module. When you know how you are oriented, push "enter". Next, the other module will appear outlined in a yellow wire frame. One picture of a surface texture from the second module will be displayed on a random wall of the second module. This is the "target" surface. You have to determine where the target goes and orient it properly relative to the first wall you saw. This button on the game pad enables you to change the wall the picture is on. Push it as many times as necessary to move the wall to the correct wireframe surface. Once the wall is on the correct wireframe surface push "enter" to anchor it there. You

then have to give the picture the right orientation relative to the cue wall you first saw. You should use the “rotate” button to do so. Once you are satisfied with your answer, press “enter” to validate it.

There will be six training trials. You will have feedback during those trials, like in the prior section. After the sixth trial, a message will inform you that you are about to start the testing. You will not have any feedback during those trials.

Please remember: do the task as quickly as possible, giving equal priority to speed and accuracy throughout the entire experiment.

Any questions?