Rapport no. 1173

Perceived affordances in using a Trackball

M.J. Konert
Perceived Affordances
in Using a Trackball

by
M.J. Konert

Report of the graduation project
Finished: August 1997
Supervision: prof. dr D.G. Bouwhuis (IPO / Philips Research)
dr D.V. Keyson (IPO / Philips Research)ir G.L.J.M. Janssen (Department of Electrical Engineering)

The IPO and the Department of Electrical Engineering of the Eindhoven University of Technology do not accept any responsibility regarding the contents of student projects and graduation reports.
Abstract

Research is continuously being conducted towards improving human-computer interaction. Input devices like mice and trackballs play a major role in interaction with graphical user-interfaces. A user-interface may become more natural and more intuitive if the perceived affordances of an input device are similar to interacting with physical objects in the real world. Since many people have experienced ball characteristics, one could attempt to implement the feedback of the rolling movement of a real ball in a trackball. The current study is focused on the capability of users to recognize the movement characteristics of the trackball as a real ball.

Towards solving this problem, the current work consists of several steps:
- Studying the characteristics of real balls set into a rolling motion by subjects.
- Implement these ball rolling characteristics in trackball control in order to simulate both visual and tactual feedback of the rolling movement.
- Determine the quality of the simulation compared to rolling a real ball in terms of ball rolling accuracy.
- Examine the role of visual feedback in rolling a real ball in comparison with a trackball.

In the implementation both tactual and visual feedback of the ball rolling movement are simulated. The idea is based on giving the user the sensation that the ball is freely rolling, although in reality the actual trackball only rotates during a short interval.

Subject performance in terms of ball rolling accuracy was found to be similar in both the real ball and trackball experiment for shorter distances given visual feedback. However, ball rolling accuracy in the trackball study was found to be lower given no visual feedback as compared to the real-world study. Therefore, it can be concluded:
- Tactual feedback plays a major role in the perception of the user of how to roll a ball. However, simulated visual feedback can compensate for the lack of tactual feedback.
- Real ball rolling does not produce sufficiently accurate target acquisition.
- The simulation of the tactual feedback fell short of reality.

- iii -
In order to use a rolling ball model in user-interface design the simulated feedback should be further improved and should lead to even better performance than rolling of real balls. Therefore, the following recommendations are made: In order to make the trackball more similar to reality, the resistive force of the trackball as subjectively experienced by the user may be reduced. For the same reason the force feedback to the user should be improved. It may also be an option to allow the user to correct ball movements during rolling and focus on the aspect of rolling-direction instead of rolled distance. Furthermore, the ball rolling model developed can be used for an experimental study to examine the benefits of dynamically activated force fields over targets based on prediction of endpoint using the ball rolling model.
Contents

1. Introduction ........................................................................................................................................... 1

2. Problem Analysis ................................................................................................................................... 3
   2.1 Description ........................................................................................................................................ 3
   2.2 Perceived Affordances ..................................................................................................................... 3
   2.3 Introduction in Ball Rolling ............................................................................................................. 5
   2.4 Trackball Technologies .................................................................................................................... 6

3. Ball Rolling Control in the Real World ................................................................................................. 9
   3.1 Experiment ....................................................................................................................................... 9
   3.2 Method ............................................................................................................................................ 9
   3.3 Results ........................................................................................................................................... 12

4. Implementation of the Ball Rolling Model .......................................................................................... 23
   4.1 Requirements ................................................................................................................................. 23
   4.2 Options ......................................................................................................................................... 24
   4.3 Algorithm ..................................................................................................................................... 27
   4.4 Complications ............................................................................................................................... 30
   4.5 Program ....................................................................................................................................... 31

5. Virtual-World Experiment ................................................................................................................... 35
   5.1 Goal of the Experiment .................................................................................................................. 35
   5.2 Method .......................................................................................................................................... 36
   5.3 Results .......................................................................................................................................... 38

6. Real versus Virtual Ball Rolling .......................................................................................................... 43
   6.1 Quantitative Comparison ................................................................................................................ 43
   6.2 Discussion ..................................................................................................................................... 46
Appendixes:
A. Numeric Results of the Experiments:
   A-1 Results of the Real-World Experiment............................................... 61
   A-2 Results of the Virtual-World Experiment........................................... 63

B. Source Files:
   B-1 Instruction Program volgorde.cpp.................................................. 65
   B-2 Class TactileBallRoll........................................................................... 71
1. Introduction

Basically, all sorts of computers and machines have been developed to make human life easier and more comfortable. Due to improved technologies more and more equipment comes available for use in everyday life. Furthermore, the number of functions that can be performed by machines has increased enormously in recent years. Thus, there is a need to pay special attention to the usability of machines. By utilizing system intelligence, in terms of recognition of control movements through an input device, manual control can be enhanced.

Because of the great importance of good human-machine communication it is important that there is an institute such as IPO (Center for Research on User-System Interaction) that focuses its research primarily on user-centred issues. The IPO is a multidisciplinary institute conducting research in the field of perception, cognition and user-system communication. The IPO was originally founded as a cooperation between Eindhoven University of Technology (TUE) and Philips Research. In January 1997 the IPO has been transformed into a research institute of the TUE.

Currently research at IPO is spread over four groups. One of them is "multimodal interaction". This group pays attention to the role of the combination of speaking, seeing, feeling and hearing in user-system interaction.

An important field in user-system interaction is the study of input devices. Often input devices like mice and trackballs are used to give commands to a computer. It is striking that the screen representation of such an input device has very little resemblance with the device itself. Such a device does not feel or act like anything people know from the real world except for previous computer-experiences.

It would be more natural if the behaviour, including feedback, of an input device would be more similar to objects in the real world. Based on everyday experience people have developed a sense for the characteristics and behaviour of numerous physical objects in the real world. An example of such an object is a ball. Everyone has a certain feeling of what is possible with a ball. An input device like a trackball is in fact also a ball. If visual and tactual feedback characteristics of rolling real balls could be implemented in a
trackball, it could make manual control within user-system interaction more intuitive. Because such a system will be easier to use, less learning time should be required.

Furthermore, if users can recognize the movement characteristics of the trackball as a real ball, system intelligence can be further utilized towards providing effective target acquisition feedback.

Keyson (in press) describes that there are generally three approaches available for dealing with the problem of avoiding non-targets in the user-interface:

- Examine the initial velocity profile and direction of a movement to predict the intended target;
- Changing the way of designing interfaces: menu layouts should be positioned to take advantage of human capacity to remember spatial positions;
- Use of adaptive and intelligent interfaces that can interpret user intentions towards dynamically assigning tactual feedback over intended targets.

The study in this report follows the first approach. An earlier experiment conducted by Van der Made (1994) concerning endpoint predicting of manual movement using a rotary dial showed that it is very difficult to base such an analysis on a simulation experiment only. Therefore, it is very important to have a reference in the real world which can be used to compare and build a simulated ball rolling interface.

A more thorough analysis of several aspects of the main problem can be found in Chapter Two. Chapter Three describes the examination of ball rolling in the real world. The implementation of ball-characteristics in a computer system is covered in Chapter Four. Chapter Five pays attention to the testing of the virtual system. Chapter Six pays attention to the differences between the simulated world and the real world. Chapter Seven covers the application of the results of the previous experiments in user-interface navigation. And finally conclusions and recommendations are presented in Chapter Eight.
2. Problem Analysis

2.1 Description

The main objective of the project covered in this report, was to study the characteristics of rolling a ball and implement the relevant parts of a ball rolling model into a user-interface in terms of visual feedback and improved ball control by the user. An important motivation for the whole study is the theory of 'perceived affordances'. This theory details the way in which humans perceive objects. Given the complexity of this study it was necessary to divide the project into several steps:

- Research into the behavioural characteristics of rolling balls in the real world.
- Implementation of the characteristics in a computer system.
- Determine the quality of the simulation compared to rolling a real ball in terms of ball rolling accuracy.
- Examine the role of visual feedback in rolling a real ball as compared to a trackball.

In every step the study focuses primarily on those aspects which are most relevant for an experience that feels most natural to the users. An important consideration is that for implementation a computer system with a sophisticated trackball is available. This system offers a lot more opportunities than an ordinary trackball. Because it is important for the understanding of the project a short description of the IPO-trackball-system is given in this chapter.

2.2 Perceived Affordances

For readers not familiar with the term "perceived affordances" some explanation may be necessary. Norman (1992) describes the essence of the word affordances. "Affordances is a technical term that refers to the properties of objects - what sorts of operations and manipulations can be done to a particular object." Some examples: "A door affords opening and closing... A chair affords support, which means we can sit, stand, or place books and papers on it."
The critical part of those affordances is how they appear to users; the way the user perceives the object's affordances: "We push things that are pushable, pull those that appear pullable". That is why they are called perceived affordances.

Thus, the "perceived affordances" of an object describe the user's perception of what can be done with that object. Perceived affordances are very important in every day life, because thanks to them people can know how to work devices which they never encountered before. According to Norman it is "challenge for designers to make signs unnecessary".

Another example given by Norman are the affordances of a ball: "A ball ... affords throwing. If the ball is made of a hard, dense material, it also affords hammering, smashing. It also affords rolling". One should notice that it would be very difficult, or even impossible, to implement all these properties in a computer system using a trackball as an input device. A major problem is the fact, that the user cannot pick up a ball which is part of trackball device. Thus, it must be realised that a choice has to be made of those affordances that are realistic for implementation in a trackball-device. While the user cannot throw the trackball, being able to roll a ball is certainly a perceived affordance. Therefore, trying to implement the rolling-characteristics is a good option.

Furthermore, in this application it should be obvious for the user that the ball on the screen and the ball in the trackball device are direct mappings of each other. If the ball looks and feels like a natural ball, then it should be obvious to the user as to what can be done with it.

A main application in user-interface design of a ball rolling model can be found in navigating through interfaces using multiple virtual screens. Therefore it is important to examine the performance in rolling over trajectories of several different lengths. Some of those trajectories should fit within the physical size of typical 14"-17" computer display, others should exceed those sizes. Therefore, the study in ball rolling will focus mainly on the aspect of distances. In order to include the effect of different orientations, two 'extreme' situations are taken into account: rolling straight ahead as well as 90° to one side. For the application of a ball rolling model in a multiple virtual screen environment it is important to realise that it is possible that a certain target one would probably want to select, is not visible at the time, because it may be located in a different virtual screen. Because of this, it is important to examine both situations with and without visual feedback.
2.3 Introduction in Ball Rolling

Towards achieving a realistic trackball look and feel, it is important to have knowledge about the characteristics of rolling balls.

On the subject of throwing and catching balls there is much literature available. However, the amount of information on ball rolling is quite limited. One of the two most relevant articles for this research describes the trajectory of a bowling ball (Hopkins and Patterson, 1977). However, it is questionable whether the trajectory of a bowling ball can serve as a standard-model for any ball, for example for a small ball rolled over a table.

The most striking difference is the fact that a bowling ball is not actually rolled by the player, but is in fact thrown onto the alley. Combined with the fact that a bowling ball is considerably heavy, this can be the cause for a spin-effect. Because of this spin-effect the trajectory of the ball is not straight but follows an arc. The question is whether a similar effect is also applicable to smaller balls which are not heavy.

The other interesting article describes the several forces that play a role in rolling a spherical object (such as a ball) over a surface (Witters and Duymelinck, 1986). These forces are:

- a drag force due to the air;
- a resistive force due to the deformation of the surface and the ball in the contact zone;
- a sliding force if the motion is not a pure rolling motion.

Several questions are raised here: Is it necessary to cover all the different forces that may act on a rolling ball in a simulation or may some of them be neglected? What kind of force has to be simulated to give the user the illusion of rolling a real ball?

What really is needed is a model of the performance of ordinary people rolling ordinary balls. It is more interesting to know about the skills of ordinary people than having a model of the performance of trained people practising ball-sports like billiard. Those people are skilled in imposing effects on balls, which would not be done by untrained people. In fact, the same applies to games like playing marbles.

Towards solving this problem the ongoing cooperation between D. Keyson at IPO and professor C. Michaels of the Faculty of Human Movement Sciences ("Faculteit der Bewegingswetenschappen") at the Vrije Universiteit (VU) in Amsterdam was utilized.
The limited amount of published data on ball rolling made it necessary to run an experiment on rolling a real ball in order to collect data about the moving characteristics of a rolling ball and user-control of it. In general two types of data are needed:

- **Subject related variables:** In order to evaluate the performance of rolling a simulated ball, it is essential to have a reference in the real world. Thus, the degree to which people are able to estimate the initial rolling speed they must impose on a ball in order to let it reach an intended target should be measured.

- **Physical characteristics:** Is it necessary to take physical effects into account, like spinning and sliding? Or does a simpler model satisfy to give the user the idea the ball moves naturally are relevant issues here. In the absence of a spinning effect as described by Hopkins and Patterson a ball will roll in a straight line and slows down according to a simple second-order polynomial.

### 2.4 Trackball Technologies

For research on input devices with force feedback, several trackballs with force-feedback capabilities have been developed. An overview of available IPO force feedback devices can be found in Keyson (1996). In this paragraph a description will be provided of trackballs with force feedback which are suitable for the implementation of the current study in this report.

#### 2.4.1 Hardware

A standard commercially available trackball is equipped with sensors only to register the finger movements of the user. Thus, such a trackball only functions as a movement input device. In addition to these sensors the IPO-trackballs are equipped with motors. These motors make it possible to impose forces on the ball. These forces form feedback to the user. The available trackballs include both types with two and three degrees of freedom. The original trackball was a device for input and force feedback in a two-dimensional plane. The basic design consists of two combinations of an optical sensor with a servo motor. One of those combinations handles cursor position and tactual feedback along the x-axis. A similar combination does the same for the y-axis (see fig. 2.1). Opposite to each motor/sensor combination a free-rolling support wheel is placed to keep the ball in position. In laboratory experiments it was shown that such a trackball with contextual force feedback enhances speed and accuracy of pointing and dragging (Engel *et al.*, 1994).
Later modifications to the trackball unit include the adding of optical position sensors opposite to both motors. Thus, making it possible to make measurements of the positions more accurate and independent of the motors. The trackball with three degrees of freedom (fig. 2.2) is in fact a modified 2D-trackball. The addition of an extra motor and position sensor enables the user to move the entire ball up and down with force feedback (Keyson, 1996).
2.4.2 Software

To make the technology of the trackball with force feedback available to experiments with user-interface navigation the flexible TacTool design environment was developed at the IPO. The TacTool-software is written in C++ for the 16-bits MS-Windows environment.

In TacTool experimental user-interfaces can be designed using different types of feedback to the user: visual, auditory and force feedback. In TacTool various predefined objects with force feedback properties are available. Objects providing various types of feedback can be combined to create an experience for the user. In figure 2.3 the design area of TacTool is shown. By clicking on icons in the toolbox on the left side objects can be added. An overview of the available objects for the 2.0 version can be found in the manual written by Klabbers (1996). For the study described in this report the latest available version (2.1) was used.

![Figure 2.3: The workspace of the TacTool design environment.](image)
3. Ball Rolling Control in the Real World

3.1 Experiment

The goal of this first experiment is to develop a basic model of how accurately people can roll a ball along a given distance and orientation in the absence and presence of visual feedback. Additionally, the rolling characteristics of the ball will be measured. Specifically, the following physical variables will be considered:

- relation between ball-speed and deceleration (frictional force)
- check whether the ball rolls in a straight line (absence of spinning)

The subject-related variables that need to be measured are:

- deviation from desired rolling distance
- deviation from desired direction
- influence of visual feedback on rolling accuracy
- influence of direction
- influence of learning given the addition or subtraction of visual feedback

The experiment was conducted at the Vrije Universiteit (VU) in Amsterdam. During the design of the experiment, a main consideration was that the results would be applicable to the trackball configuration at the IPO. Thus, if needed, it would be possible to conduct the same kind of experiment in a simulated environment. In this simulation, a trackball should be experienced by the subjects as similar to a real ball, and feedback of the movement of the ball will be provided by a TV-screen. However, the main difference in a trackball experiment is that the trackball can only be rotated in a fixed position instead of being rolled away from the hand.

3.2 Method

Subjects

Eight subjects, four male, four female, participated in the experiment. The subjects were university students and staff members ranging in age from 20 to 34 with a mean age of
Three of them were left-handed, the others were right-handed, based on preferred hand for writing and tools.

**Apparatus**

The subjects were asked to roll the white plastic ball as used in the IPO-trackball configuration over a table. The ball has a diameter of 57 mm and a mass of 115 gram. A video camera was attached to the ceiling in order to record the trajectory of the ball from above. All recordings were made on S-VHS tape. In order to increase the contrast of the recorded trajectories, the surface of the table is covered with black cloth. The cloth was also used to prevent the user from receiving any auditory feedback on the rolling characteristics of the ball. Furthermore a computer screen was used to inform the subject which task was to be carried out. In fig. 3.1 the apparatus used to conduct the experiment are shown.

**Procedure**

The experiment consisted of 4 main conditions (2 different rolling directions and visual feedback or no visual feedback). The two rolling directions were straight ahead and at 90 degrees to the left (fig. 3.2). Each condition was completed by a subject as a block of 25 trials. The experiment results in 4 blocks of 25 trials per subject. Thus, the total number of trials was 800. The order of the blocks was balanced over the subjects. Within each block five distances (15, 30, 45, 65 and 85 cm) were randomly presented 5 times. For
this purpose a simple C++ program has been written (see appendix B-1). To a certain degree this program tries to avoid that the presented distance is the same as the previous one.

![Figure 3.2: The position of the subjects in the "90 degrees to the left" condition (left), and the position of the subjects in the 0 degrees (straight ahead) condition (right).](image)

The distances were based on the screen size of the 29" VGA-CTV used as part of the trackball configuration at the IPO. The reason to do this was to keep compatibility between this experiment and the simulated experiment described in Chapter Five. By using these five distances a mix has been created of short and long ball trajectories. Three of them are short enough to show both starting and endpoint of a ball trajectory on the screen, the other 2 distances are larger than the screen size. It is conceivable that the user could roll a ball to a desired point in a user-interface beyond the screen border assuming the perceived affordance of the ball is understood in terms of rolling characteristics.

The five different distances are indicated by characters (A-E) on the table. The nearest target was indicated by 'A', the longest distance by 'E' (see also fig. 3.1 for the table layout). On the computer screen a letter was displayed to indicate over which distance the ball should be rolled.

The subjects were instructed to roll the ball over the table, and release it, as soon as the ball started rolling. Figure 3.3 shows the instructed position of the subject's hand. In the experiments without feedback, a curtain was used to prevent the subject from seeing the trajectory of the ball. The subjects could see their arm but not their hand. They were also allowed to have a look at the markers on the table indicating the five targets before they started a trial, as long as they did not see where the ball had stopped rolling.
The tapes with the recorded ball-trajectories were digitized on a VIDIPLUS-system at Nijmegen University (KUN). This VIDIPLUS consists of a S-VHS video recorder combined with a Windows-PC with video-grabbing capabilities and special digitizing software. The software is capable of recognizing the shape of the same object (for example a ball) in successive video frames. It is the same kind of system, that is normally used at the VU for this kind of digitizing tasks. Most of the digitizing work (about 120 hours) was carried out by an assistant hired by the IPO at the KUN. All the digitizing work resulted in a set of xy-coordinates (50 samples per seconds) of the ball-trajectory for each single trial.

3.3 Results

3.3.1 Subject-related Variables

For every combination of conditions (distance, feedback and orientation) the results of all 8 subjects were taken into account. This means that for every combination of conditions a data set of 40 trials was available. Generally the performance in a trial is considered good if the ball is rolled in a straight line towards the intended target, and ends up in the crosses indicated by the letters of the targets. In figure 3.4 the ball is shown on several locations on the table. Figure 3.4-a and b show the ball at the beginning of a trial. Two different ways of ending a trial are shown in figure 3.4-c and d. Since the ball ended almost exactly on the target, fig. 3.4-c shows an example of what can be considered a good trial. In fig. 3.4-d a trial is shown with some deviation from the desired direction.

The aim of the experiment was to examine the effects of the following independent variables on the performance in ball rolling and user-control of it:

- rolling distance;
- orientation;
- visual feedback.
In order to compare the results of several conditions with each other a way has to be chosen to represent the location of the endpoint of a rolling trajectory as two coordinates. It is an option to choose xy-coordinates for this representation. However, the information that is most needed from this experiment is the length of the rolling trajectory. By comparing this length to the distance between starting-point and intended target, one could examine the accuracy of the ball rolling performance. However, direction of the ball trajectory is also important.
Therefore, it has been chosen to present the results of the experiments in the following variables:

- **Path length** indicates the length of the rolling trajectory from starting point to endpoint.
- **Angle** between the line formed by the starting-point and the endpoint of the trajectory, and the line between the starting-point and the intended target, respectively. A deviation to the left is positive, a deviation to the right has a negative sign.
- The "**Absolute angle**" is defined as the average of the absolute values of the angles.

Furthermore, if the ball rolling model is applied to user-interface design, it is interesting to know the absolute **distance** between the intended target and the point where the ball stopped rolling. Figure 3.5 shows the way these dependent variables are related to the location of the ball. These dependent variables are presented in separate charts. Both the mean value and the 95%-confidence interval based on a normal distribution are shown. Furthermore, in appendix A-I the numeric values of mean and standard deviations will be presented for these dependent variables.

![Figure 3.5: Schematic overview of path length, angle and distance.](image)

A two-sample t-test performed on both path length and angle showed that there are no significant differences in the performance of the three left-handed subjects (numbered 1-3) compared to the five right-handed ones (subjects 4-8). A student’s t-test is a statistical function used to determine whether two sample means are equal.
Influence of orientation
In order to examine the influence of orientation on rolling performance, two 'extreme' values have been chosen as orientations (0° and 90° to the left). A two-sample t-test performed on the dependent variables path length and angle showed no significant differences between those orientations. Figures 3.6 and 3.7 serve as illustration that there were only minor differences. Therefore, in the examination of the influence of visual feedback on the rolling performance no distinction is made with regard to orientation.

Figure 3.6: Mean path length as function of the orientation and target distance. The 95% confidence limits around each mean are based on a normal distribution.

Figure 3.7: Mean angle of trajectory as function of the orientation and target distance. The 95% confidence limits around each mean are based on a normal distribution.
Influence of visual feedback on path length
As can be seen in figure 3.8 the mean path length is almost independent of the visual feedback condition. Only in the condition without visual feedback the variance among the results is larger. In both feedback conditions the same trend is visible: For larger distances the path length rises at a lower rate. For shorter rolling trajectories the mean path length is too large, and for the largest rolling trajectories the mean path length is too short. For targets at 65 cm the mean path length almost equals the required rolling distance.

![Figure 3.8: Mean path length as function of visual feedback and target distance. The 95% confidence limits around each mean are based on a normal distribution.](image)

However, it has to be noticed that the table used in the experiment had a limited length. This caused the ball to roll out of camera range in some of the trials in the no-feedback condition. Thus the actual endpoint could not be recorded in some cases. If the camera range would have been large enough to record the actual endpoint of every trajectory, then the calculated mean path length would have been slightly larger. However, this effect is especially applicable to trials aimed at target E at 85 cm (the largest distance) without visual feedback. In this condition in 27.5% of the 80 trials the actual endpoint was not recorded. For trials aimed at the same target distance with visual feedback, in only 7.5% of the trials the trajectory was too long to be entirely recorded. This is a considerably smaller percentage. However, the mean path length is the same with and without visual feedback. Thus it seems that the fact that the end of the trajectory could not be recorded in certain trials did not damage the results. An explanation for the encountered phenomenon could be that the variance in the no-feedback condition is larger. For trials aimed at the other 4 targets the effect of this phenomenon is even smaller. In only 6% of the trials aimed at the target at 65 cm the endpoint of the
trajectory could not be recorded under the condition of no visual feedback. With visual feedback this percentage drops to 2.5%. For trials aimed at the other three targets there was not a problem at all.

**Influence of visual feedback on the angle of the trajectory**

First of all, the results show that for all intended targets the deviation angle of the trajectory have positive values (fig. 3.9). Thus, it seems that people have a tendency to roll the ball with a deviation to the left. It has been checked that the table was horizontal. Therefore an explanation may be that the letters indicating the different targets were located on the left-side of the table only. It might be possible that the subjects developed a tendency to roll towards those letters. Furthermore, there is no general relationship with the distance to the targets. Only the shortest target distance (15 cm) forms an exception.

The results show that the mean angle is almost the same both with and without visual feedback. A two-sample t-test performed on the results without visual feedback showed that only between the target distances 15 and 30 cm a significant difference could be found with respect to the mean angle. For the other distances there are no significant differences.

![Graph showing mean angle of trajectory as function of visual feedback and target distance. The 95% confidence limits around each mean are based on a normal distribution.](image)

**Figure 3.9**: Mean angle of trajectory as function of visual feedback and target distance. The 95% confidence limits around each mean are based on a normal distribution.
Influence of visual feedback on distance

For application of a ball rolling model in user-interface navigation it is also interesting to examine the influence of the visual feedback condition on the distance between actual endpoint of the trajectory and the target. Preventing the user from visual feedback has a clear effect: it decreases the performance (fig. 3.10).

![Graph showing the mean distance between target and the actual endpoint of the trajectory as a function of visual feedback and target distance. The 95% confidence limits around each mean are based on a normal distribution.](image)

Figure 3.10: Mean distance between target and the actual endpoint of the trajectory as function of visual feedback and target distance. The 95% confidence limits around each mean are based on a normal distribution.

For all targets the effect seems about the same: the mean distance is about 50% higher without feedback. However, this effect does not occur in the size of confidence intervals. The mean distances over all targets are 12.0 cm (with feedback) and 18.1 cm (without feedback). It seems a bit odd that the availability of visual feedback has a larger influence on distance than it has on path length (fig 3.8). This difference can be explained by the larger variance in the results of the trials without visual feedback.

Influence of learning by visual feedback

In order to measure the result of the learning effect of rolling with visual feedback, the results of the 4 subjects who started rolling with visual feedback are compared to the results of the four who started with the trials without visual feedback (fig. 3.11). For both groups it can be concluded that rolling with feedback produces better results than rolling without. Furthermore, the group who started without feedback has a slightly better performance both in trials with and without feedback. It is possible that this is due to a sampling-effect since every group consisted of 4 subjects only. A two-sample t-test showed that the differences are not significant. However, it seems there is not a clear advantage in starting with visual feedback. Thus there is no reason to conclude that the order of visual feedback is responsible for a learning effect.
The absolute angle of the trajectory

Generally spoken the angle as shown in fig 3.9 only provides information about the mean direction of the ball in a number of trials. An average value based on both deviations to the left and to the right gives only limited information about the accuracy of rolling in the proper direction. Figure 3.12 shows the mean of the absolute angle as function of target distance. It appears that for shorter distances, the absolute angle is slightly larger. However the differences are very small. For the confidence-intervals the differences are even smaller.

Figure 3.12: Mean absolute angle of the trajectory as function of visual feedback and target distance. The 95% confidence limits around each mean are based on a normal distribution.
3.3.2 Physical variables

Relationship between ball-speed and deceleration

During the motion of the ball, its speed drops from the initial velocity down to zero. In an ideal situation the speed ‘v’ will decrease according to a linear relation:

\[ v(t) = at + b \]  \hspace{1cm} (3.1)

Since it is a situation of deceleration, the acceleration-factor ‘a’ has a negative sign. This linear relation will cause the displacement ‘x’ as a function of time to have the character of a second-order polynomial:

\[ x(t) = \frac{1}{2}at^2 + bt + c \]  \hspace{1cm} (3.2)

In order to check this, the trajectories as functions of time (in fig 3.13 a sample trajectory is shown of a single trial) have been fitted to second-order polynomials. For this task the programs Kaleidagraph and SlideWrite were used. The goodness of a curve fit is described by the \( r^2 \) coefficient of determination. The closer this coefficient is to 1, the better the fit. The curve fits of the ball trajectory have a typical coefficient of 0.99. Thus a second-order polynomial will be a good approximation.

![Figure 3.13: Sample trajectory of a ball trajectory with a target distance of 15 cm.](image)

This fact also means that acceleration-factor is independent of both initial speed and current speed during motion.
Size of the acceleration-factor

In an ideal situation the acceleration-factor 'a' should only be dependent on material of both the ball and surface of the table. That is because the contact zones define the amount of friction. However, in this case there were some differences in the size of the acceleration factor among the 800 digitized trials. Theoretically speaking, every subject rolling the same ball over the same table should experience the same deceleration-factor. However, it will be required then that on every part of the ball and table the properties of the surface are completely the same, otherwise the acceleration-factor will vary depending on the way the ball was rolled.

It appeared that the absolute value of the acceleration-factor was smaller for trials with lower initial velocities. A possible explanation can be found in the fact that the properties of the cloth used to cover the table are less ideal compared to harder materials like glass. The surface of a table covered with cloth is more deformable compared to glass. Furthermore, if a ball rolled on cloth has a higher initial impulse the deformation of the surface will be larger. This will result in a higher deceleration-factor.

However, for implementing a ball rolling model it is not really a problem that an exact factor could not found. Since the deceleration-factor depends on the contact-zones of ball and table-surface (Witters and Duymelinck, 1986) it is not an absolute universal value. If another ball or another table had been used, a different factor would have been found. However, it is likely that rolling a small ball over a table covered with cloth will lead to a deceleration-factor of a similar size. Therefore, the experiments provides a good approximation of what people expect for friction if they roll a ball over a table.

In this experiment the calculated value for the average acceleration factor varied from -0.0698 m/s² for those trials aimed at the nearest target (A at 15 cm) to -0.0962 m/s² for the longest distance (target E at 85 cm). Combining all trials results in a mean value of -0.082 m/s² for the acceleration factor.

Thus, this value has been used for the implementation of the ball rolling model in chapter four. However, it will be necessary to run a pilot-experiment to check whether this is a good value for implementing the real world. If it turns out to be that the resistance in the trackball is higher than the resistance in the real world then the absolute value of the acceleration factor has to be increased.
Otherwise, people think that the resistance in the trackball is equal to the rolling resistance during the motion of the virtual ball. If the frictional resistance in the trackball itself is too high, people will push the ball too hard. Using Newton's formula:

\[ F = m \cdot a \]  

(3.3)

it becomes obvious that by adjusting the acceleration factor, compensation is possible for a too high force.

Based on formula 3.3 and the ball-mass (115 grams) it is possible to calculate the resultant resistive force: This is about \( 9.43 \times 10^{-3} \) N.

Based on this experiment only, it cannot be judged that this is the 'rolling resistive force'. The calculated force may consist of several components. A more thorough physical experiment is needed in order to determine these forces. This kind of experiment is described by Witters and Duymelinck (1986). From their experiment it could be concluded that rolling resistive force due to deformation of the contact zones is nearly constant in the velocity range from 0.5 m/s to zero. The mean value of deceleration is \( 0.09 \) m/s\(^2\). This is about 1% of earth's gravitational acceleration. Although, Witters and Duymelinck's experiment was conducted on a billiard ball of 21.9 gram, it seems the conditions were about the same. In the ball-roll experiment described in this chapter the initial velocity of the ball varied between 0.1 and 0.6 m/s and the mean value of deceleration was \( 0.082 \) m/s\(^2\). So this seems a realistic value.

Since the deceleration-factor almost equals the value of the force component for resistance due to contact zone deformation, this may be an indication that the other two forces:
- drag force
- sliding friction
described by Witters and Duymelinck play only a minor role in this experiment.

Check whether the ball rolls in a straight line (absence of spinning)
As can be concluded from Hopkins and Patterson (1977) a spinning effect of a spherical object will result in a trajectory partially following an arc instead of a straight line. From the experiment it can be concluded that a great majority of the balls followed a straight line, so a spinning effect can be neglected. This supports also the assumption that the sliding friction can be neglected.
4. Implementation of the Ball Rolling Model

4.1 Requirements

Based on the characteristics of ball rolling found in the previous chapter an implementation has to be made in a virtual environment. In fact it is a simulation of the feedback of a rolling ball that has to be implemented. Therefore, it is important to distinguish several types of feedback (Norman, 1988):

- visual feedback;
- tactual feedback;
- auditory feedback.

Objects providing different types of feedback are also called multimodal objects. In what follows, it will be considered what types of feedback are the most important ones for a simulation of the properties of real balls in a virtual environment.

- **Visual feedback:** The user should see the ball slowing down in a proper way from the initial velocity down to zero.

- **Tactual feedback:** Two aspects can be distinguished: (1) the resistive force or friction that is felt when a user tries to roll the ball and (2) the experience that a ball keeps rolling away after releasing it.

- **Auditory feedback:** The sound caused by a rolling ball provides also feedback about the speed of the ball and time during which the ball rolled. The loudness of the sound depends on the kind of material of both the ball and table. If both ball and table are made of hard material the sound will be very clear. However, in the real world experiment the table was covered with soft cloth, in order to minimize the influence of auditory feedback. Therefore, in the simulation the sound of the rolling ball does not have to be taken into account.

Thus, the main objective of the simulation should be implementing both visual and tactual properties of a rolling ball. Furthermore, some additional requirements are important for the implementation:
The implementation should interface in a proper way with existing hard- and software. In this case that will be the 3D IPO-trackball and the latest version of the TacTool design environment (2.1). The 3D trackball has been chosen since in this device a relatively large part of the ball itself is located outside the case (see the cross-section in fig. 4.1).

![Figure 4.1: Schematic cross-section of the 3D-trackball.](image)

The implementation should be suitable to be used in experiments. Therefore, it should be capable of writing data about rolling movements to disk. It is also preferable that the program is able to place the ball at the same starting point if subsequent trials have to be conducted.

The implementation should be as fool-proof as possible. Unexpected behaviour of users should not cause the system to hang-up or skip parts of the experiments.

### 4.2 Options

Independent of what solution will be chosen for the actual implementation the main concept remains the same: In the simulation a ball is displayed on screen that is similar (in size) to the actual trackball. In a good simulation the user should intuitive experience both balls to be the same (single) one. Furthermore, the user should experience the ball to be a real ball with the same properties.

Main problem in the implementation is the fact that the friction perceived by the user when rolling a trackball, is considerably higher than the experienced friction of a real ball. Consequence is that the trackball stops rolling considerably earlier than the real ball when the same initial impulse is imposed on both of them.

For this problem several solutions are possible. Both hardware and software solutions may be options.
**Hardware solutions**

Two different hardware approaches may serve as a solution:

- Decrease the mechanical frictional forces imposed by motor- and support wheels on the trackball;
- Compensate for the frictional forces by imposing a feed forward force on the ball by using the servomotors.

Both hardware solutions have in common that their main objective is to increase the period that the trackball keeps rolling after an initial impulse on the ball imposed by the user. The screen representation of the ball will simply follow the rotation of the trackball and thus keeps rolling as long as the trackball itself is rolling. Thus by this means both visual and tactual feedback are realised in one solution.

The first solution may in fact decrease the resistive force experienced by the user. However, decreasing the contact forces has a considerable disadvantage too: The ball may slide against the sensor wheels instead of correctly rolling. Since the sensors measure displacement and speed of the support- or motor wheels instead of directly measuring the ball's rotational speed, the measurements of the sensors which serve as input for the user-interface would not be reliable any more.

The second solution imposes forces on the ball opposite to the frictional forces. In the current trackball configuration the size of these force is determined by the hardware. One should notice that this so-called feed forward force should not cause the ball to roll from a standstill situation. Only if a user starts the ball to roll, the feed forward forces should be imposed on the ball. Therefore, it is needed that the feed forward forces will be disabled if the velocity of the ball is zero. Therefore a linear function has been implemented in the amplifier which drives the servomotors of the trackball. Thus, the feed-forward force increases linearly with increasing velocity of ball (fig. 4.2). However, the consequence of this linearity is that at a certain velocity of the ball, the feed-forward force will be larger than the frictional force of the ball which is nearly constant (see also fig 4.2). The resultant force will cause the ball to accelerate to the maximum velocity the trackball's servomotors are capable of.

On the other hand, if the velocity if the ball is very slow, the feed-forward is too small to compensate for the frictional force and the ball will stop rolling too early. This a major problem since the initial velocity of the ball rolling movement can vary considerably.

In the previous chapter it turned out to be that a constant deceleration factor is needed in order to simulate the movement of a real ball. Using formula 3.3 it is obvious that the
resultant force of feed forward and friction should be constant. Thus in order to create the correct simulation of ball movement major adjustments should be made to the hardware. Furthermore, it is unlikely that the frictional force of the trackball itself is really completely constant. Thus, it may still be difficult to create a stable and reliable ball rolling model in this way, which will make it difficult to predict the exact endpoint of the rolling trajectory.

Another consideration that has to be made is that hardware feed forward capabilities only have been implemented in the amplifier of the 2D trackball. This trackball is not as suitable for ball rolling experiments as the 3D trackball, because except for a small part of the top almost the entire ball is built-in in the case of the device.

**Software implementation of multimodal synthetic feedback**

Since a hardware solution will cause a lot of troubles, another approach has been chosen to implement the ball rolling model. In a multimodal solution both visual and tactual feedback will be implemented in separate ways:

- **Visual feedback** will be provided by a ball representation that will start with an initial velocity according to the initial velocity imposed on the ball. Next, the ball will slow down according to the second-order polynomial of the ball rolling movement (formula 3.2)

- **Tactual feedback** will be provided by imposing a feed forward force on the ball when the ball has started rolling just before the user is about to let it go. This serves as simulation of the rotational inertia of the ball.
It is imperative to understand the synthetic nature of the feedback. The feedback that is experienced by the user, is not the "natural" feedback of a trackball. Both the visual and tactual feedback are computer generated in order to create the illusion that it is not a trackball but a rolling real ball. The feedback is based on a model of a rolling ball that is continuously being updated by the computer. Therefore, the generated feedback will continue even after the actual trackball has stopped rotating. Because of the relatively high internal friction of the trackball that will be the case after a relatively short time interval. Thus a distinction will be made between the rolling of the actual trackball and the speed of the ball representation on the screen (also called: virtual ball). For this solution an algorithm is provided in the next paragraph.

4.3 Algorithm

Since this implementation of the ball rolling model is to be used for an experiment (similar to the real world experiment in Chapter Three) in the actual implementation two parts can be distinguished:

- the actual ball rolling model;
- a part that keeps track of the number of trials that have been conducted and provides information to the subject what trial has to be carried out (this part is similar to the program used in the real world experiment for randomly determining the order of the targets in the experiment).

The main principle of the system is that when a user touches the ball and rotates it, the velocity increases to a maximum and after the user lets the ball go, the speed will decrease. The program should determine the top of the velocity curve and use this as initial velocity to calculate to the path the ball will follow on the screen. During the movement of the ball on constant time-intervals the computer calculates the new position of the ball until the speed of the ball drops to zero (See fig 4.3).

\[ \text{Calculated speed of the virtual ball} \]

\[ \text{Initial velocity of the trackball} \]

\[ \text{Time (s)} \]

\[ \text{Velocity of the ball (m/s)} \]

Figure 4.3: Measurement of the initial ball velocity.
In the algorithm (fig 4.4) the following parts can be distinguished:

- Determining a target the user has to aim at in the experiment (a);
- Measuring the speed of the trackball until the top of the velocity curve has been determined (b);
- Enable a force field to give a user the experience that the ball starts rolling away (c);
- Show the screen representation of the ball as long as the calculated velocity of the virtual ball is higher than 0 m/s (d).

*ad a:*
Since this implementation is used for an experiment the algorithm repeats itself exact 50 times (the number that is needed in the experiment described in the next chapter). Every new trial a randomized function is used to select a new target. In an implementation that is intended to be used for user-interface navigation this part can be skipped.

*ad b:*
As long as the speed of the ball changes, continuously updated speed-measurement are sent by the trackball to the computer. It is only a matter of determining which one of these speed measurements forms the top of the initial velocity curve. This problem is easy to solve: As long as every measured speed is higher than the previous one, the top of the curve has not been reached yet. The top is reached then, if the next measured speed is lower than the current one.

*ad c:*
This force field is important for the tactual perception. Timing is important: It should not be enabled at a too early stage in the algorithm, otherwise it will influence the speed measurements. However, if the force field is enabled too late, the user has already stopped touching the ball and the effect is lost.

*ad d:*
At constant time-intervals the next location of the ball on the screen is calculated according to the second-order polynomial in formula 3.2. Actually, first the velocity is calculated. If this is not below zero then speed, deceleration factor and the previous position of the screen-ball are used to calculate the next position. This approach is chosen because it is needed to check the velocity of the virtual ball. In order to avoid that the ball starts rolling backwards, the ball on the screen has to be stopped if the calculated speed should become negative. Furthermore, this implementation is meant to be used in an experiment with both feedback and no-feedback conditions. Therefore, before placing the ball at a new position it has to be checked whether it is a trial with feedback or not.
Figure 4.4: Algorithm of the ball rolling implementation.
4.4 Complications

Many problems are caused by the timing needed for a proper interaction between the parts of the algorithm responsible for respectively the visual and tactual feedback. These are problems such as:

- When to stop and start measuring speed-changes in the trackball?
- When to enable and disable the force field for the tactual feedback?

If the timing of those issues is not arranged properly, the force field will change the speed of the ball and thus confound the speed-measurements. For example after the force field has been disabled the ball will not stop to roll immediately. If the system registers this rolling as input for the next trial, the system will conduct the experiment by itself before the subject has had a chance to touch the ball.

To solve this kind of problems it has been decided to define two flags:

- a flag used for speed measurements: "enableSpeed";
- a flag to indicate that the ball is in a safety interval between two trials: "roltijd".

During this interval the flag is inactive.

'EnableSpeed' is used only in the time-interval that input from the user is requested by rolling the trackball. Shortly after enabling the force field the speed-measurements have to be disabled. Just after enabling the force field, during a time-interval of 250 ms the user can still increase the initial impulse that was imposed on the ball, based on his experience of the force field. During the rolling-movement 'roltijd' will be active until the ball on the screen is at the end of its trajectory. Then the force field will be switched off. However, to be sure that the trackball has not any velocity left caused by the force field, the system needs to wait another 1.5 seconds before activating enableSpeed for the newly calculated next target. Since the timing aspect is important for this implementation a timing diagram is shown in figure 4.5.

Another complication is the fact that the trackball sensors are very sensitive. Even if a subject only touches a ball without even rolling it, the system will measure a short increase of speed followed by a short decrease. The algorithm as shown in paragraph 4.3 will consider this as the top of the initial speed and start the trial before the subject has actually started to roll the ball. Since the initial speed was very low the trajectory of the ball will be very short, next the computer will prepare to run the next trial. Thus in fact trials are skipped. In order to avoid that trials are skipped when the ball is just touched, it
has been decided to ignore velocities below 0.085 m/s. This should not cause trouble since most of the initial velocities turned out to be between 0.10 and 0.70 m/s.

![Figure 4.5: Timing diagram of the ball rolling algorithm.](image)

### 4.5 Program

The ball rolling model has been implemented as a part of TacTool. TacTool is based on the object-oriented programming concept and implemented in C++. When designing interfaces in TacTool several types of visual, tactual and auditory can be selected. These objects are defined in separate classes. Therefore, the ball rolling model which is also an object in TacTool has been defined as a separate class called: `TactileBallRoll` and implemented in the file `tacballr.cpp`.

The implementation presented in this chapter will be available in TacTool as the tactual object `BallRoll`. The visual option in the object submenu of a `BallRoll` object can be used to attach a picture of a ball to the `BallRoll` object to let the requested ball appear on screen. Several settings can be adjusted using the same object submenu: The “Size”-option enables the user to adjust the area and direction of the force-field. The “Modify”-option is needed to increase or decrease the force experienced by the user. The force field, i.e. the application of a motor-induced force on the trackball, should be rather small in order not to conflict with the speed measurements. However, on the other hand the force field should not be too small, otherwise the user would not feel it. Therefore, a value was chosen between these two extremes.

The implementation of the algorithm is based on an event-driven concept as is usual in TacTool. Since a considerable part of `tacballr.cpp` contains standardized procedure
needed to interact with other TacTool classes, in this report only the part of the class \texttt{TactileBallRoll} will be covered in which the actual algorithm is implemented. This section of the class can be found in appendix B-2.

In this paragraph the implementation of the algorithm will be covered by discussing the different types of events:

- \texttt{TM\_STARTEXEC}
- \texttt{TM\_CHAR}
- \texttt{TM\_SPEEDCHANGE}
- \texttt{TM\_TIMER}
- \texttt{TM\_STOPEXEC}

\textbf{TM\_STARTEXEC}
This part is carried out only once during program-execution. The purpose of this part is assign initial values to several variables. Various groups of variables can be distinguished:

- velocity variables like $V_{ox}$ and $V_{oy}$ in which both dimensions of the measured speed of the ball are stored;
- values needed to convert units from pixels/s to meters/s. For example $d_s$ is the diameter of the trackball and $n_p$ the number of pixels measured by the system when the trackball is rotated once 360 degrees;
- coordinates that indicate where to place the ball on the screen (e.g.: $X_{new}$, $Y_{new}$)
- counters to store the number of trials that have been conducted (e.g.: $trials$)

Also the value of the deceleration-factor is initialized here in $a_w$. It has to be noticed that $2 \times \text{aw}$ is equal to the acceleration-factor ‘a’ in the previous chapter. Furthermore, it should be noticed that the absolute value of the deceleration-factor in this implementation is twice as high as it should be according to the theoretical situation of formula 3.2. Thus the actual acceleration-factor in the implementation is equal to $4 \times \text{aw}$. This was necessary in order to compensate for the higher friction of the trackball experienced by the users compared to a real ball. This value is based on an informal pilot-test with 5 subjects.

\textbf{TM\_CHAR}
Every time a key is pressed \texttt{TM\_CHAR} will be called. However, in this implementation it is only needed to press just a numeric key once at the start of the experiment. Based on the numeric value (1-8) a specific file is opened to store the results of the subject. The subject-number is also used to determine whether to start the experiment with a no-
feedback condition (odd subject numbers) or in a feedback condition (even subject numbers). Furthermore, the target to aim at in the first trial is displayed in this part.

**TM_SPEEDCHANGE**

Every time the speed of the ball changes, this routine is called. First the speed of the ball is captured in pixels/s ($v_{ox}$, $v_{oy}$). These values are converted in m/s ($v_x$, $v_y$). Based on these two x- and y-components the resultant value ($v_{new}$) is calculated. If this $v_{new}$ is larger than the previous measured speed ($v_{last}$), it is considered that this may be the topspeed in the velocity curve. In that case a timer is started which has to trigger the TM_TIMER routine after a constant time-interval of 50 ms (this module will draw the rolling ball on the screen). If the next measured speed is higher than the previous one, this one is considered to be the top and the timer will be restarted. If the last measured speed is lower than the previous one, the force field needed to provide tactual feedback to the user is enabled. This force field is a variant of the object-type conveyor-belt in TacTool (see also Klabbers 1996).

**TM_TIMER**

The Timer-routine is called 20 times a second in order to draw the new location of the ball on the screen. More updates per screen are not necessary for a good impression of the ball-movement, and a higher frequency of updates will cause a too high system load. Based on the initial highest velocity measured in TM_SPEEDCHANGE ($v_{et}$), the new $v_{et}$ after 50 ms is calculated. Using this value the distance $D$ covered in 50 ms is calculated. Next the screen positions of the virtual ball are calculated and the screen position of the ball is updated. Furthermore, in this routine the speed measurements are disabled after an interval of 250 ms.

An important feature in this routine is the speedcheck. If the new calculated velocity $v_{et}$ is still positive, the timer is called again in order to update the screen again after 50 ms. Otherwise, the force field (conveyor-belt) is disabled. After a safety-interval of 1.5 seconds, a routine similar to the volgorde.cpp program (used for the real world experiment, appendix B-I) is called and the next target to aim at is displayed to the user. However, if already 50 trials have been conducted the experiment will end.

**TM_STOPEXEC**

This routine is called at the end of program-execution and its only task is to disable the speed-measurements.
Virtual-World Experiment

5.1 Goal of the experiment

Given the assumption that rolling a trackball is experienced as in the real world, one can compare the quality of ball movement control and role of movement feedback in both worlds. The best way to do this is to run an experiment and use the results of the experiment described in chapter three as a reference of the real world performance. Therefore, a similar experiment using the simulated trackball world was conducted. In what follows the simulated ball will be referred to as virtual world, and the real ball experiment will be called: "the real-world experiment".

In the virtual-world experiment three main goals can be distinguished:

- Examine the quality of the synthetic feedback in the simulation.
- Determine the improvement of endpoint prediction based on rolling ball movement.
- Examine the ability to roll longer distances with and without visual feedback. This means rolling over distances exceeding the size of 15" and 17" computer displays.

If the results of both experiments are similar, this could serve as indication of the quality of the implementation in the virtual world. However, in order to make such a comparison, it is obligatory that as many factors as possible are the same in both experiments. Those factors are:

- ball size (diameter 57 mm)
- number of targets (5)
- both a feedback and a no-feedback situation
- number of subjects (8). The subjects are also balanced in gender. Furthermore the order of the conditions is balanced over the subjects.

There are also some differences between the two experiments:

- The average distance between the starting-point of the ball and targets is shorter, because the size of the display used in this experiment (a 29" CTV-set) is smaller than the table used in the other experiment. However, three out of five distances have
been chosen the same in both experiments in order to make it possible to compare the results. However, a 29" display size is still large enough to examine the effect of ball rolling over distances exceeding the size of standard 15" and 17" computer displays.

- The orientation was not used as a condition in this experiment, because in the real world experiment it turned out to be that the influence of orientation was rather small.

Furthermore it would not be necessary here to pay attention to physical variables. However, before the experiment was run, it has been checked that the implementation behaved according to a second-order polynomial as had been observed in the real world experiment.

Thus, the experiment is focused on measuring to these subject-related variables:
- deviation of desired rolling distance
- deviation of desired direction
- influence of feedback
- influence of learning by visual feedback

This experiment was run at the IPO, using the trackball configuration in the tactual research laboratory.

5.2 Method

Subjects
Eight subjects, all students and staff members from the IPO ranging in age from 21 to 41 with a mean age of 27, participated in the experiment. Four of them were female, four male. One female was left-handed, so was one male. All the others were right handed, based on preferred hand for writing and tools.

Apparatus
The IPO 3D-trackball was used as input device in this experiment. The same ball as used in the first experiment was built in. A 29" CTV-set was put on its side, in order to create an oblong area. This television was connected to a PC to show a representation of the markings on the table in the first experiment. Both the targets, and the ball were shown on the screen. The aim was to make the screen layout as similar as possible to the layout of the table in the first experiment. Therefore, the size of the markers and the ball on the screen are specifically chosen in order to match the size of ball and markers in the real ball experiment (fig. 5.1)
The trajectories of the balls rolled on the screen were stored on disk. Furthermore, an additional computer screen is used to inform the subject about the task that should be carried out. In figure 5.2 a schematic representation of the configuration in the experiment is shown.

**Procedure**

Since the first experiment showed no significant difference between the two orientations, this experiment consisted of 2 main conditions only (visual feedback and no visual feedback). The order of the blocks was balanced over the subjects.

In each block the subjects were asked to roll the ball over five different distances (15 cm, 22.5 cm, 30 cm, 37.5 cm and 45 cm). As can be seen in figure 5.1 the shortest distance is indicated with 'A', the longest distance with 'E'.

---

*Figure 5.1: Screen layout of the simulation. The left image shows the (virtual) ball in start-position, the picture on the right shows the position of the ball at the end of a trajectory (simulated trial).*

*Figure 5.2: The trackball configuration used for the virtual-world experiment. On the host-PC the TacTool design environment is running. The dedicated PC runs control software for the trackball. The amplifier is needed to impose a force field on the ball.*
Reasons to choose these distances were:

- three of these directions are the same as in the previous experiment
- all distances fit in the screen-size of a 29" television
- the distances between the targets are equal

For each distance 5 trials have been conducted. Within a block the trials were carried out in random order. The five different distances were indicated by characters on the screen (see fig. 5.1). On the computer screen a letter was displayed to instruct the user over which distance the ball should be rolled. The experiment resulted in 2 blocks of 25 trials per subject. Thus, the total number of trials was 400.

The subjects were instructed to rotate the trackball in the same way the subjects was asked to roll the ball in the first experiment (fig. 3.3), and release it, as soon as the ball started rolling. In the experiments without visual feedback, the ball on the screen did not move.

5.3 Results

The results of this virtual-world experiment are presented in the same way as in the real-world experiment. Thus, the attention will be mainly focused on path length and angle of trajectory (see also figure 3.5). The main effects of different target distances and the presence and absence of visual feedback will be examined. In the same way as in the Chapter Three the results are presented in both charts and tables. The performance of the subjects will be shown for all five distances (15 - 45 cm). The charts show the mean values and the 95%-confidence-intervals based on a normal distribution. Tables can be found in appendixes A-2 and show the mean value and the standard deviations.

A two-sample t-test performed on path length and angle showed no significant difference in performance between the two left-handed subjects (number 2 and 5) and the right-handed ones.
Influence on path length

Two effects are visible in the influence of target distance on path length (fig. 5.3):

- With visual feedback there is a very good correspondence between produced path lengths and the actual path lengths. The slope of the function is 1.08. Thus there is only a slight overshoot, since the slope of the ideal function is 1.
- Without visual feedback, a consistent overshoot by a constant factor of produced path length is visible. The mean slope of the function is 1.65, thus this is a considerable overshoot. Also the variance is higher without visual feedback.

Although the subjects rolled the ball too far without visual feedback, it is still possible to distinguish the different target-distances. A striking effect is that the variance is much smaller for the trials with visual feedback compared to the ones without.

![Graph showing path length vs. target distance with visual and no visual feedback](image)

*Figure 5.3: Mean path length as function of visual feedback and target distance. The 95% confidence limits around each mean are based on a normal distribution.*

Influence of visual feedback on the angle of the trajectory

The measurement of the angle between the actual trajectory and the line from starting point to target does not show any clear trend (fig. 5.4). It can be judged positively that under both visual feedback conditions the mean angle is very small for all 5 distances. This means that on an average the ball is rolled in the proper direction without a tendency to deviate to either side. Also the variance among the results is rather small. A considerable number of the trials had none or only minor deviation from the straight line.
Influence of visual feedback on distance

The trend is clear: If the target distance increases, the distance between the actual ending point of the rolling trajectory and the intended target increases (fig. 5.5). The other trend is also clear: the variance among the performances of the subjects also increases for the targets at the larger distances. The large deviation measured here, could be a problem for the usability of these results in user-interface design since the distance between two targets is only 7.5 cm. Thus, there will be quite a lot of overlap. This makes it relatively difficult for the computer to use to ball rolling model for desired endpoint prediction.
Both the mean value of the difference between the target and the actual stopping point of the rolling ball and the variance are much higher in the no-feedback condition compared to feedback. In the case of shorter target distances the mean value is almost three times larger for the no-feedback condition (14.0 cm) compared to the feedback condition (5.2 cm). The average values of the deviated distance over all targets are 10.2 cm (with visual feedback) and 24.0 (without feedback). In combination with the high variance in the no-feedback trials, it can be said that the performance of virtual ball rolling without feedback is considerably worse than it is with feedback.

Influence of learning by visual feedback
For the examination of a possible learning effect, only the distance between target and endpoint of the trajectory will be taken into account (fig. 5.6). Most striking is that the same phenomenon occurs as in the real world experiment. The subjects who started with the trials without visual feedback performed better than the other ones. The effect is even larger than in the real ball experiment. However, a two-sample t-test showed that in this experiment the differences are not significant either.

![Figure 5.6: Mean distance between target and the actual endpoint of the trajectory as function of the order of the visual feedback condition. The 95% confidence limits around each mean are based on a normal distribution.](image)

Absolute angle
Figure 5.7 shows that the subjects had only a minor tendency to deviate the rolling trajectory from the direction straight ahead. In both visual feedback conditions the mean angle over all distances is less than 1 degree. For 4 out of 5 target distances there is almost no difference between the two feedback conditions. It seems that for longer distances, the absolute angle increases a bit. However, differences are too small to base conclusions on.
Figure 5.7: Mean absolute angle as function of visual feedback and target distance. The 95% confidence limits around each mean are based on a normal distribution.
6. Real versus Virtual Ball Rolling

6.1 Quantitative Comparison

In order to compare the results of the virtual ball rolling experiment with ball rolling in the real world the main effects of different target distance and the presence or absence of visual feedback on path length and absolute angle are examined in this chapter. For this comparison, three target distances have been chosen to be the same in the experimental design of both experiments. These are the targets at 15, 30 and 45 cm. This equals targets A, B and C in the real world experiment and targets A, C, and E in the virtual world. In order to make trends visible, for both experiments the two other distances have also been printed in the charts.

Path length

Since it turned out in the virtual ball rolling that there is a large difference between the performance with and without visual feedback (see Chapter Five), separate charts have been made for the situation with visual feedback (figure 6.1) and the situation without (fig. 6.2).

Figure 6.1: Comparison between real-world and virtual-world experiment under the condition of visual feedback. Mean path length as function of target distance. The 95% confidence limits around each mean are based on a normal distribution.
The performance in ball rolling with visual feedback is about the same in the real and virtual world. For the targets at 15 and 30 cm the mean path length almost equals the target-distance in the virtual world experiment. However, the variance in virtual ball rolling performance is higher.

The performance in virtual ball rolling without visual feedback is much worse than it is for the other conditions. The mean path length is longer and the variance is relatively high. The function rises much faster for the simulated ball, than it does for the real ball (see figure 6.2). Instead of rising slower, the path length in virtual ball rolling without visual feedback increases faster for increasing target distance. It seems that the effect of improved accuracy for longer distances does not occur in this case.

![Figure 6.2: Comparison between real-world and virtual-world experiment under the condition of no visual feedback. Mean path length as function of target distance. The 95% confidence limits around each mean are based on a normal distribution.](image)

**Absolute angle of the trajectory**

If the attention is focused on the absolute angle between the actual rolling trajectory and the straight line between starting point and the target, it appears that the performance in virtual ball rolling is undoubtedly better than the real world. This applies to both the situation with visual feedback (figure 6.3) and the situation without (figure 6.4). For all three targets that both experiments have in common, both the average value of the angle and confidence interval are considerably smaller in virtual ball rolling. It also seems that the trendlines under the condition of no visual feedback are rather irregular compared to the case with visual feedback. However, it is difficult to base conclusions like this on the available data since both experiments have only three distances in common and the angles are all very small.
Figure 6.3: Comparison between real-world and virtual-world experiment under the condition of visual feedback. Mean absolute angle as function of target distance. The 95% confidence limits around each mean are based on a normal distribution.

Figure 6.4: Comparison between real-world and virtual-world experiment under the condition of no visual feedback. Mean absolute angle as function of target distance. The 95% confidence limits around each mean are based on a normal distribution.
6.2 Discussion

From the difference in the performance with and without feedback in the virtual ball rolling experiment it becomes clear that the visual feedback plays a major role. In the real world the difference between performance in trials with and without feedback is considerably smaller. Apparently other factors than visual feedback are also important for people in order to determine how they should roll a ball in order to reach a certain target. Also must be concluded that these factors are not implemented good enough in the virtual model. It seems that applying feed forward force to the ball in order to decrease the friction of the ball is not enough. Modifications are necessary to create a model that is also comparable under no-feedback circumstances.

However, for the nearest target the results are almost the same in real and virtual world. There is a chance of 95% that trials will end up within about 7 cm from the target. However, the distance between the targets was 7.5 cm in the second experiment. If more accurate results are wanted then the distance between targets should be increased to about 14 cm.

What in fact is wanted is a model of the virtual world with better performance than the real world. The required distance between targets has to decrease to be very usable in user-interface navigation. Furthermore, for larger target distances the distance between targets should also increase in order to keep the same accuracy. Especially the variance in performance for the target at 45 cm in the virtual world was relatively larger. However, it is possible that this is caused by the fact that this target was located near the edge of the screen. Thus if subjects rolled the ball too fast, the visual feedback was limited because the ball moved out of sight.

It might be wise to study a possible relationship between ball-size and the required distance between the targets. A smaller ball-size (and thus cursor size) may be a solution to improve accuracy.

There are several options to improve the results of the virtual model:

- The way force feedback is given to the user should be improved. However, the time to give force feedback is very short. That is due to the fact that a subject does not touch the ball any more after initiating the rolling movement.

- The hardware of the trackball could be improved in order to decrease the friction of the ball. It is an option to improve the relationship between the actual speed of the ball and feed forward force placed on the ball.
• The distance between targets should be kept relatively large when implementing this technique in user-interface navigation. This implies that the number of targets should be limited.

• The performance may also improve if the user is allowed to correct the movement during rolling.

• Since it turned out that the absolute angle of the trajectory in the virtual world experiment was relatively small, it might be an option to focus on direction instead of distance. However, it should be noticed that the mechanical construction of the trackball with two servomotors under an angle of 90° (as shown in figure 2.1) makes it easier for a subject to roll straight ahead without deviation.
7. Further Study

7.1 Application of the Ball Rolling Model

Since the main aim of the study in this report was to develop and implement a ball rolling model in a trackball, the next step will be to use the modified trackball with ball rolling capabilities as an input device in a user-interface. However, until now only data is collected about the performance of ball rolling without changing any parameter setting of the implementation. It may be interesting to examine the influence of changes in parameter-settings such as the size of the feed forward force. Furthermore, no data is available yet about the performance of the trackball with ball rolling capabilities compared to other input devices like mice and commercially available trackballs.

Both the visual feedback of the rolling trajectory and the improved ball-control for the user have been implemented in a computer system in order to guide the user to the intended target in the user-interface. However, in the simulation experiment described in Chapter 5 no target had an attractive force field connected with it. By providing the intended on-screen target with an attractive force field, performance can be improved. In order to improve the guidance of the user to the on-screen targets a two-step approach is required in the application:

- Based on initial velocity and direction the endpoint of the expected movement is predicted.
- Force fields and auditory feedback are dynamically enabled around the predicted endpoint in order to help the user to find the exact target and thus compensate for minor deviations.

The main objective of the use of force fields is to help a user to find objects without depending too much on visual attention. There are several ways for the actual design of force fields:

- Around an object a circle could be defined with forces all pointing to the target;
- A path can be defined along the trajectory towards an object.
Various ways of implementing this techniques have been described by Keyson (1996). Several types of force fields for tactual feedback are available in the TacTool design tool (Klabbers, 1996).

In this study new aspects in the concept of force fields for tactual feedback are:
- The combination of force with ball rolling;
- Force fields are enabled only when and where needed.

Therefore, experiments have to be conducted in order to determine whether the additional ball rolling capabilities improve the performance in using the trackball. Available options for designing such an experiment include:
- A comparison of trackballs configured with different settings of ball-roll capabilities and force feedback;
- A comparison of the performance in using several different input devices like the study described by Epps (1986).

This second option is a good choice for a later experiment. It would be an interesting experiment to compare a trackball with ball rolling capabilities to an ordinary trackball given visual feedback in both cases. However, first it is important to get a good overview of the performance of the trackball with rolling capabilities under different conditions. Such an experiment will be described in paragraph 7.3.

Generally two types of experiments are common in this field of research:
- A serial one dimensional task, subjects move back and forth between two targets of specified width and separated by a specified distance;
- An experiment in which subjects carry out more discrete tasks, similar to common computer tasks. For example: selection of text (Card et al, 1978).

Main advantage of the first option is that it is possible to collect a lot of data quickly and there are no confounding influences which tend to compromise the measurements like reaction time and angle of approach in real tasks. Main argument against the first option is the claim that it lacks similarity with real tasks.

For this kind of experiments it is common to use the Index of Performance (IP) in Fitts' law to measure the performance of users in certain tasks (MacKenzie 1992). The experiments in Fitts' original report (1954) were serial tasks. In similar experiments Fitts and Peterson (1964) used a discrete task for measuring IP. Such an experimental design has been widely adopted in other studies. The experiment proposed in paragraph 7.3 is
also based on discrete tasks, similar to real computer tasks. It has to be noted that this experiment has not been actually conducted since it was a priority first to pay some extra attention to the implementation described in Chapter Four before conducting a more comprehensive experiment.

7.2 Introduction to Fitts' Law

The aim of Fitts' law is to measure human performance in relation to the difficulty of the movement tasks that have to be carried out. Fitts claimed that human movement can be modelled analogous to information transmission (Fitts, 1954; Shannon & Weaver, 1949). Fitts' law has played a major role in experimental psychology, especially in human factors and kinematics. Since the text-selection experiments conducted by Card, English and Burr (1978) Fitts' has also been adopted widely in research in Human-Computer Interaction. Since the use of graphical user-interfaces has become general, the need has grown for a reliable prediction model of movement time.

In Fitts' law the variables Movement Time (MT) and Index of difficulty (ID) play a major role in the calculation of Index of Performance (IP):

\[
IP = \frac{ID}{MT}
\]  

(7.1)

In order to calculate the ID both the distance D to the target as well as the width W of the target are needed (fig. 7.1):

\[
ID = \log_2(D/W+1)
\]  

(7.2)

![Diagram of Fitts' law](image)

*Figure 7.1: Width and Distance in Fitts' law.*

Fitts claimed that while humans are conducting movement tasks, information is transmitted through a "human channel", similar to an information channel in electronic communications systems. According to Fitts, electronic signals (S) are analogous to movement distances (D) and noise (N) is analogous to width (W). Therefore, Fitts' law was based on Shannon's Theorem 17 (Shannon & Weaver, 1949):
In formula 7.3 $I_P$ matches channel capacity ($C$) in bit/s and $MT$ matches $1/B$ with bandwidth ($B$) in Hz (MacKenzie, 1992).

### 7.3 Proposal for Experimental Design

The modified trackball with ball rolling capabilities has been designed for navigating in user-interfaces using two kinds of forces:

- The feed-forward force added to the trackball to give the user the experience that the ball starts rolling (chapter 4);
- Dynamically activated force-fields based on prediction of desired endpoint (paragraph 7.1).

Therefore, it is important to determine the influence of these conditions on the performance. The following design is proposed:

#### Design:

The experiment could consist of 5 main conditions:

- feed forward on, maximum force feedback around the target
- feed forward on, intermediate force feedback around the target
- feed forward on, low force feedback around the target
- feed forward on, target force feedback off
- null-condition: no feed forward, no force feedback

For each condition a block with a number of trials will be carried out. The order of the blocks will be balanced among the subjects.

#### Procedure:

In each block the subjects are asked to roll/move the ball to one of the targets shown on the screen. The targets vary in size (width, height) and distance from the initial position of the ball on the screen. All targets will look like elements in a user-interface. In order to avoid complications with effective target width (MacKenzie, 1992) circular targets will be used. All targets will be labelled with names or icons. In order to get a reliable image of the performance of the modified trackball, size and distances are chosen in specific combinations in order to get five levels of Index of Difficulty.
On request MacKenzie advised to use a $5 * 3 * 3$ factorial design, using the following factors and levels within them:

- Type of feedback (5 levels)
- Target Distance (3 levels)
- Target Width (3 levels)

With three levels for both distance and width it is possible to get 5 different ID's. This results in 9 targets based on the following specifications (fig. 6.2):

- Width: 1, 2 and 4;
- Distance: 8, 16 and 32.

Figure 7.2: Screen-layout for proposed experimental design.

If a television in landscape orientation is used, the units of these factors can be centimeters. If a larger screenheight is available these units can be adjusted. It is important that the size of the ball-cursor does not exceed the size of the largest target. Thus, it may be needed to make the screen-representation of the ball a bit smaller than the actual ball size.
8. Conclusions and Recommendations

In the study reported, based on an experiment in rolling real balls, a model of ball rolling has been implemented in a virtual environment. In the real-world experiment, it turned out that even in the real world it is a difficult task for people to estimate the impulse they should impose on a ball, to roll it to a specified target. However, the performance of subjects in rolling to nearby targets with the help of visual feedback was reasonable. For more remote targets, the performance was decreasing. People were also surprisingly good at ball rolling without visual feedback, although the performance was a little lower compared to the same targets with visual feedback. The direction in which the balls were rolled had no large influence on the performance.

In the experiment concerning rolling real balls also the physical variables have been studied. The deceleration during the rolling movement is almost completely due to frictional forces. The deceleration factor of the ball is constant. Therefore, it is a good model of the reality to describe the position of the ball as a second-order polynomial function of time.

To implement the ball rolling model in a virtual environment multimodal synthetic controllable feedback was the best option. Both visual and tactual feedback had to be implemented separately. To provide the visual feedback the rolling of a ball was implemented using a second-order polynomial. Tactual feedback was provided by imposing feed forward force on the ball. Furthermore, it could be concluded that the higher friction of the trackball compared to a real ball could be compensated by increasing the absolute value of the deceleration-factor. An explanation is that the resistive force of the trackball subjectively experienced by the user, is interpreted as an indication of the size of the resistive force of the virtual ball during rolling. As this felt resistive force is much higher than for a real ball, subjects apply a greater force to the trackball. This would lead to a higher initial rolling speed. To the deceleration-factor a resistive component is added, corresponding to the internal friction of the trackball. Since it is very difficult to reduce the internal (mechanical) friction of the trackball, it is a better option to increase the friction of the virtual ball.
The quality of the simulated ball turns out to be good for ball rolling over short distances with visual feedback. The performance of the subjects is almost similar to the real ball experiment, sometimes even better. Especially the deviation of the intended direction was very small compared to the real world experiment. Although the differences in performance between feedback and no-feedback condition are considerably larger than in the real world experiment, the trend for both visual conditions were similar. It seems that the mental model that the subjects have of ball rolling is different than the implemented model. It appears that the subjects thought that the simulated ball had a larger deceleration-factor than was implemented.

Therefore, can be concluded that:

- Tactual feedback plays a major role in the perception of the user of how to roll a ball.
- Visual feedback can compensate for the absence of tactual feedback, even if it is just synthetic simulated feedback.
- Real ball rolling does not produce sufficiently accurate target acquisition.
- The simulation of the tactual feedback could be further improved.

Thus, for short rolling distance with visual feedback the implementation is a good approximation of the real world. If the simulation is to be used for rolling over longer distance without visual feedback, the implementation of the tactual feedback should be improved.

Since even in rolling real balls the mean distance between the endpoint of the rolling trajectory and the intended target was considerable, it might be difficult to apply this ball rolling model to user-interface design. However, it is interesting to examine the current usability of the ball rolling model based on the results of the simulation experiment. With regard to mean path length and direction the performance in virtual ball rolling is quite reasonable. Focusing on a single one of these two variables provides only limited information about the usability of the ball rolling model in user-interface navigation. Even if several targets can be reached with the same path length, it is still possible to distinguish which target was meant if the difference in direction between the paths is large enough. If the mechanical construction of the trackball, with the orthogonal position of the two servomotors, is taken into account, it must be possible to distinguish at least four different directions (0, 90, 180, 270 degrees). Diagonal directions may turn out to be no problem as well. However, the influence of the construction of the trackball on those directions should be examined.
For the performance in user-interface navigation, the attention should also be focused on the distance between intended target and the actual endpoint of the trajectory. In the simulation this mean distance is about 30 percent of the length of a rolling trajectory for all target distances under the condition of visual feedback. Although this value is similar to the measurements in the real world, the performance is not good enough to reliably distinguish 5 different targets at target distances varying from 15 to 45 cm. However, if the choice is limited to targets at short and long distances respectively, it will be possible to reliably predict which target was the intended one.

For these reasons the usability of the ball rolling model in a multiple virtual screen environment is limited to a distance of two screens, especially because longer distances were not included in the simulation experiment. Also for longer distances start- and endpoint of trajectories are not visible at the same time. Therefore, the decreased performance under the condition without visual feedback should also be taken into account.

Thus can be concluded that the application of the ball rolling model in user-interface design is currently limited to short and long target distances in at least four different orientations. To become more useful for user-interface design a simulation should be created which offers better performance than the real world experiment showed.

Therefore the following recommendations can be made:

- It should be studied what options are available for improving active force feedback to the user. Further study of timing and size of the force may further improve performance. It may be an option too to examine ways to create a variable force field.
- It may be possible to create a better simulation of the resistive force of a real ball by reducing the internal friction of the trackball. If the internal friction is reduced, less force will be necessary to rotate the ball, and thus it will become more similar to the real world. However, in order to do so, the mechanical construction of the ball has to be adjusted, without reducing the accuracy of the built-in velocity and displacement sensors.
- A more comprehensive experiment could be conducted to examine the optimal friction coefficient for the simulation.
Recommendations concerning the application of the ball rolling model:

- If the user is allowed to correct the ball movement during rolling the performance may improve.
- An alternative for further study is to focus on direction instead of distance since the deviation between intended direction and actual direction was only minor in the simulation. However, the influence of the mechanical construction of the trackball should also be taken into account.

Also, a target acquisition study as described in Chapter 7 could utilize the current findings to examine the benefits of dynamically activated force fields over targets based on prediction of endpoint using the ball rolling model.

The very notion that manual control can be enhanced through synthetic multimodal feedback as compared to a direct coupling of physical feedback and device movement remains an interesting area for further study. It would appear that the perceived affordance of rolling a ball can be simulated to a large degree in terms of anticipated tactual and visual feedback.
Bibliography


- 59 -


NORMAN, D.A. (1992). *Turn signals are the facial expressions of automobiles.* Reading, Massachusetts: Addison-Wesley.


### Appendix A-I: Results of the Real-World Experiment

#### Table 1: Mean values of path-length (cm) under the conditions of feedback and 0° orientation

<table>
<thead>
<tr>
<th>Distance</th>
<th>15</th>
<th>30</th>
<th>45</th>
<th>65</th>
<th>85</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
</tr>
<tr>
<td>1</td>
<td>22.3</td>
<td>10.5</td>
<td>43.6</td>
<td>9.2</td>
<td>66.6</td>
</tr>
<tr>
<td>2</td>
<td>20.3</td>
<td>5.9</td>
<td>21.0</td>
<td>9.5</td>
<td>44.3</td>
</tr>
<tr>
<td>3</td>
<td>18.1</td>
<td>3.7</td>
<td>35.1</td>
<td>6.3</td>
<td>54.3</td>
</tr>
<tr>
<td>4</td>
<td>15.0</td>
<td>10.4</td>
<td>37.0</td>
<td>8.8</td>
<td>51.8</td>
</tr>
<tr>
<td>5</td>
<td>19.0</td>
<td>3.2</td>
<td>33.7</td>
<td>6.5</td>
<td>50.3</td>
</tr>
<tr>
<td>6</td>
<td>16.9</td>
<td>2.9</td>
<td>28.3</td>
<td>7.1</td>
<td>48.8</td>
</tr>
<tr>
<td>7</td>
<td>21.5</td>
<td>6.2</td>
<td>34.5</td>
<td>5.3</td>
<td>46.0</td>
</tr>
<tr>
<td>8</td>
<td>18.7</td>
<td>5.7</td>
<td>40.9</td>
<td>13.2</td>
<td>50.5</td>
</tr>
<tr>
<td>all</td>
<td>19.0</td>
<td>6.4</td>
<td>34.3</td>
<td>10.1</td>
<td>51.6</td>
</tr>
</tbody>
</table>

#### Table 2: Mean values of path-length (cm) under the conditions of feedback and 90° orientation

<table>
<thead>
<tr>
<th>Distance</th>
<th>15</th>
<th>30</th>
<th>45</th>
<th>65</th>
<th>85</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
</tr>
<tr>
<td>1</td>
<td>21.1</td>
<td>5.4</td>
<td>38.1</td>
<td>8.4</td>
<td>59.9</td>
</tr>
<tr>
<td>2</td>
<td>13.8</td>
<td>1.2</td>
<td>40.3</td>
<td>18.0</td>
<td>44.6</td>
</tr>
<tr>
<td>3</td>
<td>15.4</td>
<td>5.3</td>
<td>38.3</td>
<td>9.1</td>
<td>54.1</td>
</tr>
<tr>
<td>4</td>
<td>15.6</td>
<td>7.2</td>
<td>33.0</td>
<td>14.6</td>
<td>42.1</td>
</tr>
<tr>
<td>5</td>
<td>23.9</td>
<td>6.3</td>
<td>28.9</td>
<td>4.2</td>
<td>43.5</td>
</tr>
<tr>
<td>6</td>
<td>16.7</td>
<td>5.0</td>
<td>28.5</td>
<td>8.0</td>
<td>42.5</td>
</tr>
<tr>
<td>7</td>
<td>23.5</td>
<td>9.1</td>
<td>35.2</td>
<td>11.8</td>
<td>43.3</td>
</tr>
<tr>
<td>8</td>
<td>25.6</td>
<td>9.0</td>
<td>33.8</td>
<td>11.6</td>
<td>37.1</td>
</tr>
<tr>
<td>all</td>
<td>19.4</td>
<td>7.2</td>
<td>34.5</td>
<td>11.0</td>
<td>45.9</td>
</tr>
</tbody>
</table>

#### Table 3: Mean values of path-length (cm) under the conditions of no feedback and 0° orientation

<table>
<thead>
<tr>
<th>Distance</th>
<th>15</th>
<th>30</th>
<th>45</th>
<th>65</th>
<th>85</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
</tr>
<tr>
<td>1</td>
<td>19.6</td>
<td>21.3</td>
<td>49.9</td>
<td>16.5</td>
<td>76.4</td>
</tr>
<tr>
<td>2</td>
<td>18.8</td>
<td>9.1</td>
<td>34.8</td>
<td>11.7</td>
<td>52.8</td>
</tr>
<tr>
<td>3</td>
<td>25.6</td>
<td>6.4</td>
<td>43.4</td>
<td>7.7</td>
<td>57.4</td>
</tr>
<tr>
<td>4</td>
<td>9.2</td>
<td>6.7</td>
<td>20.9</td>
<td>1.4</td>
<td>37.9</td>
</tr>
<tr>
<td>5</td>
<td>17.7</td>
<td>4.7</td>
<td>32.3</td>
<td>8.3</td>
<td>51.3</td>
</tr>
<tr>
<td>6</td>
<td>10.7</td>
<td>2.3</td>
<td>15.9</td>
<td>3.8</td>
<td>22.9</td>
</tr>
<tr>
<td>7</td>
<td>26.4</td>
<td>15.6</td>
<td>28.8</td>
<td>8.1</td>
<td>38.1</td>
</tr>
<tr>
<td>8</td>
<td>14.1</td>
<td>3.7</td>
<td>25.7</td>
<td>6.9</td>
<td>31.8</td>
</tr>
<tr>
<td>all</td>
<td>17.8</td>
<td>11.2</td>
<td>31.5</td>
<td>13.3</td>
<td>46.1</td>
</tr>
</tbody>
</table>

#### Table 4: Mean values of path-length (cm) under the conditions of no feedback and 90° orientation

<table>
<thead>
<tr>
<th>Distance</th>
<th>15</th>
<th>30</th>
<th>45</th>
<th>65</th>
<th>85</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
</tr>
<tr>
<td>1</td>
<td>24.5</td>
<td>12.3</td>
<td>47.9</td>
<td>8.4</td>
<td>68.4</td>
</tr>
<tr>
<td>2</td>
<td>30.4</td>
<td>6.7</td>
<td>53.0</td>
<td>7.7</td>
<td>65.4</td>
</tr>
<tr>
<td>3</td>
<td>30.7</td>
<td>5.3</td>
<td>49.6</td>
<td>8.4</td>
<td>61.5</td>
</tr>
<tr>
<td>4</td>
<td>19.6</td>
<td>8.0</td>
<td>36.4</td>
<td>17.9</td>
<td>36.4</td>
</tr>
<tr>
<td>5</td>
<td>22.6</td>
<td>7.0</td>
<td>34.0</td>
<td>6.3</td>
<td>60.5</td>
</tr>
<tr>
<td>6</td>
<td>9.5</td>
<td>1.6</td>
<td>16.1</td>
<td>4.2</td>
<td>18.7</td>
</tr>
<tr>
<td>7</td>
<td>13.7</td>
<td>8.5</td>
<td>30.5</td>
<td>11.6</td>
<td>61.9</td>
</tr>
<tr>
<td>8</td>
<td>14.3</td>
<td>5.1</td>
<td>20.0</td>
<td>6.2</td>
<td>37.0</td>
</tr>
<tr>
<td>all</td>
<td>20.7</td>
<td>9.8</td>
<td>35.9</td>
<td>15.4</td>
<td>51.2</td>
</tr>
</tbody>
</table>
Appendix A-I: Results of the Real-World Experiment (continued)

Table 5: Mean angle of deviation (degrees) under the conditions of feedback and 0° orientation

<table>
<thead>
<tr>
<th>Distance:</th>
<th>15</th>
<th>30</th>
<th>45</th>
<th>65</th>
<th>85</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
</tr>
<tr>
<td>1</td>
<td>11,8</td>
<td>6,5</td>
<td>8,4</td>
<td>5,1</td>
<td>8,0</td>
</tr>
<tr>
<td>2</td>
<td>4,6</td>
<td>2,8</td>
<td>2,6</td>
<td>3,8</td>
<td>3,8</td>
</tr>
<tr>
<td>3</td>
<td>-1,0</td>
<td>1,5</td>
<td>0,2</td>
<td>1,3</td>
<td>0,4</td>
</tr>
<tr>
<td>4</td>
<td>-0,2</td>
<td>2,5</td>
<td>1,0</td>
<td>1,9</td>
<td>1,2</td>
</tr>
<tr>
<td>5</td>
<td>2,9</td>
<td>3,8</td>
<td>2,2</td>
<td>1,3</td>
<td>1,8</td>
</tr>
<tr>
<td>6</td>
<td>2,6</td>
<td>2,5</td>
<td>1,1</td>
<td>1,3</td>
<td>1,6</td>
</tr>
<tr>
<td>7</td>
<td>1,9</td>
<td>0,7</td>
<td>2,8</td>
<td>1,8</td>
<td>1,1</td>
</tr>
<tr>
<td>8</td>
<td>0,3</td>
<td>1,6</td>
<td>0,8</td>
<td>0,9</td>
<td>1,8</td>
</tr>
<tr>
<td>all</td>
<td>2,9</td>
<td>4,7</td>
<td>2,4</td>
<td>3,3</td>
<td>2,5</td>
</tr>
</tbody>
</table>

Table 6: Mean angle of deviation (degrees) under the conditions of feedback and 90° orientation

<table>
<thead>
<tr>
<th>Distance:</th>
<th>15</th>
<th>30</th>
<th>45</th>
<th>65</th>
<th>85</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
</tr>
<tr>
<td>1</td>
<td>6,9</td>
<td>3,7</td>
<td>6,7</td>
<td>3,9</td>
<td>4,1</td>
</tr>
<tr>
<td>2</td>
<td>5,4</td>
<td>6,4</td>
<td>4,9</td>
<td>2,1</td>
<td>4,3</td>
</tr>
<tr>
<td>3</td>
<td>-1,4</td>
<td>2,5</td>
<td>0,1</td>
<td>0,5</td>
<td>0,5</td>
</tr>
<tr>
<td>4</td>
<td>0,0</td>
<td>0,7</td>
<td>2,7</td>
<td>3,1</td>
<td>4,0</td>
</tr>
<tr>
<td>5</td>
<td>2,1</td>
<td>1,7</td>
<td>2,9</td>
<td>1,2</td>
<td>1,6</td>
</tr>
<tr>
<td>6</td>
<td>1,7</td>
<td>3,2</td>
<td>1,3</td>
<td>3,0</td>
<td>2,2</td>
</tr>
<tr>
<td>7</td>
<td>1,6</td>
<td>1,1</td>
<td>2,2</td>
<td>2,8</td>
<td>2,0</td>
</tr>
<tr>
<td>8</td>
<td>0,9</td>
<td>1,5</td>
<td>-1,0</td>
<td>3,6</td>
<td>0,7</td>
</tr>
<tr>
<td>all</td>
<td>2,1</td>
<td>3,8</td>
<td>2,5</td>
<td>3,4</td>
<td>2,4</td>
</tr>
</tbody>
</table>

Table 7: Mean angle of deviation (degrees) under the conditions of no-feedback and 0° orientation

<table>
<thead>
<tr>
<th>Distance:</th>
<th>15</th>
<th>30</th>
<th>45</th>
<th>65</th>
<th>85</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
</tr>
<tr>
<td>1</td>
<td>2,0</td>
<td>10,4</td>
<td>5,3</td>
<td>3,3</td>
<td>6,9</td>
</tr>
<tr>
<td>2</td>
<td>6,3</td>
<td>3,5</td>
<td>13,4</td>
<td>6,8</td>
<td>3,0</td>
</tr>
<tr>
<td>3</td>
<td>1,7</td>
<td>4,3</td>
<td>0,3</td>
<td>4,4</td>
<td>2,9</td>
</tr>
<tr>
<td>4</td>
<td>-6,5</td>
<td>6,6</td>
<td>1,5</td>
<td>2,8</td>
<td>1,3</td>
</tr>
<tr>
<td>5</td>
<td>3,0</td>
<td>0,9</td>
<td>1,7</td>
<td>1,4</td>
<td>1,8</td>
</tr>
<tr>
<td>6</td>
<td>3,9</td>
<td>1,5</td>
<td>6,6</td>
<td>1,8</td>
<td>8,5</td>
</tr>
<tr>
<td>7</td>
<td>2,5</td>
<td>2,6</td>
<td>3,3</td>
<td>2,0</td>
<td>4,6</td>
</tr>
<tr>
<td>8</td>
<td>0,0</td>
<td>1,7</td>
<td>1,2</td>
<td>1,8</td>
<td>0,6</td>
</tr>
<tr>
<td>all</td>
<td>1,6</td>
<td>5,7</td>
<td>4,2</td>
<td>5,1</td>
<td>3,7</td>
</tr>
</tbody>
</table>

Table 8: Mean angle of deviation (degrees) under the conditions of no-feedback and 90° orientation

<table>
<thead>
<tr>
<th>Distance:</th>
<th>15</th>
<th>30</th>
<th>45</th>
<th>65</th>
<th>85</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
</tr>
<tr>
<td>1</td>
<td>13,3</td>
<td>3,2</td>
<td>12,9</td>
<td>4,1</td>
<td>8,8</td>
</tr>
<tr>
<td>2</td>
<td>6,3</td>
<td>4,2</td>
<td>10,2</td>
<td>7,8</td>
<td>6,3</td>
</tr>
<tr>
<td>3</td>
<td>1,1</td>
<td>2,4</td>
<td>3,2</td>
<td>2,6</td>
<td>2,2</td>
</tr>
<tr>
<td>4</td>
<td>0,7</td>
<td>1,2</td>
<td>1,2</td>
<td>1,3</td>
<td>1,9</td>
</tr>
<tr>
<td>5</td>
<td>1,6</td>
<td>3,2</td>
<td>2,0</td>
<td>3,1</td>
<td>-0,7</td>
</tr>
<tr>
<td>6</td>
<td>2,9</td>
<td>3,3</td>
<td>7,5</td>
<td>1,8</td>
<td>5,3</td>
</tr>
<tr>
<td>7</td>
<td>3,8</td>
<td>2,2</td>
<td>0,4</td>
<td>2,8</td>
<td>2,4</td>
</tr>
<tr>
<td>8</td>
<td>-5,7</td>
<td>8,7</td>
<td>2,5</td>
<td>3,1</td>
<td>-0,4</td>
</tr>
<tr>
<td>all</td>
<td>3,0</td>
<td>6,2</td>
<td>5,0</td>
<td>5,5</td>
<td>3,2</td>
</tr>
</tbody>
</table>
## Appendix A-2: Results of the Virtual-World Experiment

### Table 1: Mean values of path-length (cm) under the condition of feedback

<table>
<thead>
<tr>
<th>Distance</th>
<th>15</th>
<th>22.5</th>
<th>30</th>
<th>37.5</th>
<th>45</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
</tr>
<tr>
<td>1</td>
<td>18.5</td>
<td>2.3</td>
<td>24.1</td>
<td>4.5</td>
<td>25.7</td>
</tr>
<tr>
<td>2</td>
<td>18.2</td>
<td>12.0</td>
<td>32.1</td>
<td>7.6</td>
<td>27.0</td>
</tr>
<tr>
<td>3</td>
<td>15.1</td>
<td>3.0</td>
<td>24.4</td>
<td>8.8</td>
<td>44.4</td>
</tr>
<tr>
<td>4</td>
<td>18.1</td>
<td>5.3</td>
<td>22.6</td>
<td>4.4</td>
<td>29.6</td>
</tr>
<tr>
<td>5</td>
<td>8.9</td>
<td>3.0</td>
<td>21.0</td>
<td>9.2</td>
<td>22.0</td>
</tr>
<tr>
<td>6</td>
<td>18.7</td>
<td>12.0</td>
<td>29.6</td>
<td>6.0</td>
<td>32.1</td>
</tr>
<tr>
<td>7</td>
<td>15.6</td>
<td>4.0</td>
<td>21.0</td>
<td>3.6</td>
<td>32.4</td>
</tr>
<tr>
<td>8</td>
<td>14.4</td>
<td>5.1</td>
<td>32.7</td>
<td>14.8</td>
<td>44.6</td>
</tr>
<tr>
<td>all</td>
<td>16.0</td>
<td>8.9</td>
<td>25.1</td>
<td>8.4</td>
<td>32.2</td>
</tr>
</tbody>
</table>

### Table 2: Mean values of path-length (cm) under the condition of no-feedback

<table>
<thead>
<tr>
<th>Distance</th>
<th>15</th>
<th>22.5</th>
<th>30</th>
<th>37.5</th>
<th>45</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
</tr>
<tr>
<td>1</td>
<td>27.2</td>
<td>19.2</td>
<td>22.7</td>
<td>4.3</td>
<td>26.1</td>
</tr>
<tr>
<td>2</td>
<td>27.2</td>
<td>11.7</td>
<td>36.2</td>
<td>12.9</td>
<td>50.6</td>
</tr>
<tr>
<td>3</td>
<td>30.4</td>
<td>12.3</td>
<td>39.2</td>
<td>13.4</td>
<td>62.4</td>
</tr>
<tr>
<td>4</td>
<td>24.2</td>
<td>10.0</td>
<td>35.9</td>
<td>8.6</td>
<td>44.9</td>
</tr>
<tr>
<td>5</td>
<td>16.2</td>
<td>8.0</td>
<td>29.8</td>
<td>18.7</td>
<td>40.8</td>
</tr>
<tr>
<td>6</td>
<td>48.6</td>
<td>21.7</td>
<td>82.4</td>
<td>16.9</td>
<td>74.4</td>
</tr>
<tr>
<td>7</td>
<td>10.7</td>
<td>8.8</td>
<td>14.9</td>
<td>5.0</td>
<td>21.9</td>
</tr>
<tr>
<td>8</td>
<td>23.1</td>
<td>8.8</td>
<td>36.2</td>
<td>10.3</td>
<td>43.5</td>
</tr>
<tr>
<td>all</td>
<td>26.0</td>
<td>15.9</td>
<td>37.2</td>
<td>21.7</td>
<td>45.6</td>
</tr>
</tbody>
</table>

### Table 3: Mean angle of deviation (degrees) under the condition of feedback

<table>
<thead>
<tr>
<th>Distance</th>
<th>15</th>
<th>22.5</th>
<th>30</th>
<th>37.5</th>
<th>45</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
</tr>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2</td>
</tr>
<tr>
<td>2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.8</td>
</tr>
<tr>
<td>3</td>
<td>0.0</td>
<td>0.0</td>
<td>-0.7</td>
<td>1.2</td>
<td>0.2</td>
</tr>
<tr>
<td>4</td>
<td>0.0</td>
<td>0.0</td>
<td>-0.2</td>
<td>0.5</td>
<td>-0.5</td>
</tr>
<tr>
<td>5</td>
<td>0.2</td>
<td>3.3</td>
<td>-0.3</td>
<td>1.1</td>
<td>1.0</td>
</tr>
<tr>
<td>6</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>7</td>
<td>0.7</td>
<td>0.9</td>
<td>0.1</td>
<td>0.3</td>
<td>0.2</td>
</tr>
<tr>
<td>8</td>
<td>-0.3</td>
<td>0.8</td>
<td>-1.7</td>
<td>3.0</td>
<td>0.4</td>
</tr>
<tr>
<td>all</td>
<td>0.1</td>
<td>1.1</td>
<td>-0.4</td>
<td>1.2</td>
<td>0.3</td>
</tr>
</tbody>
</table>

### Table 4: Mean angle of deviation (degrees) under the condition of no-feedback

<table>
<thead>
<tr>
<th>Distance</th>
<th>15</th>
<th>22.5</th>
<th>30</th>
<th>37.5</th>
<th>45</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
<td>σ</td>
<td>mean</td>
</tr>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
</tr>
<tr>
<td>2</td>
<td>0.1</td>
<td>0.2</td>
<td>0.7</td>
<td>0.9</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>-0.1</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>-0.9</td>
</tr>
<tr>
<td>4</td>
<td>-1.0</td>
<td>1.4</td>
<td>-0.9</td>
<td>1.2</td>
<td>-0.2</td>
</tr>
<tr>
<td>5</td>
<td>0.0</td>
<td>0.0</td>
<td>1.6</td>
<td>6.9</td>
<td>-0.2</td>
</tr>
<tr>
<td>6</td>
<td>0.2</td>
<td>0.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
</tr>
<tr>
<td>7</td>
<td>-0.5</td>
<td>1.2</td>
<td>1.1</td>
<td>4.4</td>
<td>-0.2</td>
</tr>
<tr>
<td>8</td>
<td>-0.4</td>
<td>1.0</td>
<td>0.1</td>
<td>2.1</td>
<td>-0.6</td>
</tr>
<tr>
<td>all</td>
<td>-0.2</td>
<td>0.8</td>
<td>0.3</td>
<td>2.8</td>
<td>-0.3</td>
</tr>
</tbody>
</table>
Appendix B-1: Instruction Program volgorde.cpp

*****************************************************************************
filename: volgorde.cpp
copyright: IPO, 1997
created by: M.J. Konert
last update: 10-03-1997
description: Instruction program to calculate the order of the trials in an experiment, and show this order as instructions to the subjects
*****************************************************************************

#include <iostream.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include <conio.h>
#include <fstream.h>
#include <string.h>

char* results_file;

void main()
{
    // begin: initialization
    char Distance;
    char NextDistance;
    char block=219;
    unsigned No1;
    unsigned No2;
    unsigned No3;
    unsigned No4;
    unsigned No5;
    unsigned voltooid;
    unsigned feedback;
    unsigned orient;
    int Check;
    randomize();

    // split screen layout in 3 parts
    clrscr();
    constream vn1, vn2, vn3;
    vn1.window(1, 1, 80, 3);
    vn2.window(11, 4, 70, 18);
    vn3.window(1, 22, SO, 25);

    // Enter data about the subject (number)
    // Subject-number must be in the range 1-8, otherwise ask for new number

    int subject = 0;
    while (subject < 1 || subject > 8)
    {
        vn2 << "Enter the number of the subject: ";
        cin >> subject;
        vn2 << endl;
    }

    // Create output-file dependent on subject-number
    switch (subject)
    {
    case 1: results_file = "rolling1.txt"; break;
    case 2: results_file = "rolling2.txt"; break;
    case 3: results_file = "rolling3.txt"; break;
    case 4: results_file = "rolling4.txt"; break;
    case 5: results_file = "rolling5.txt"; break;
    case 6: results_file = "rolling6.txt"; break;
    case 7: results_file = "rolling7.txt"; break;
    case 8: results_file = "rolling8.txt"; break;
    } // results_file = "afstand.txt"
    ofstream to(results_file);

    // code for the experiment goes here...
}

- 65 -
to << "This file contains the data about subject " << subject << endl;

// Use the subject-number to determine the order of the 4 blocks within // the experiment
if ((subject & 2) == 0)
    { /* Start with feedback */
        feedback = 1;
    }
else
    { /* start with no feedback */
        feedback = 0;
    }

if (subject <= 4)
    { /* Start with straight ahead */
        orient = 1;
    }
else
    { /* start with 90 degrees */
        orient = 0;
    }

// Begin main-loop (consists of four 25-trial blocks)
for (unsigned ta=1;ta<5;ta++)
{
    if ((ta & 2)==0)
        { if (orient == 1) orient = 0; else orient = 1;
    }
    if (ta ==3)
        { if (feedback == 1) feedback = 0; else feedback = 1;
            if (orient == 1) orient = 0; else orient = 1;
        }
    vn2 << "This is part " << ta << " of 4" << endl;
to << endl;
to << "Starting of part " << ta << " of 4" << endl;
if (feedback == 0)
    { vn2 << "The following trials are without feedback" << endl;
to << "Feedback = No" << endl;
    }
else
    { vn2 << "Feedback is allowed during the following trials" << endl;
to << "Feedback = Yes" << endl;
    }
    if (orient == 0)
        { vn2 << "The orientation is 90 degrees to the left" << endl;
to << "Orientation = 90 degrees" << endl;
    }
else
        { vn2 << "The orientation is straight ahead" << endl;
to << "Orientation = 0 degrees" << endl;
    }

// initialize local counters inside a block
Distance = 0;
No1 = 0;
No2 = 0;
No3 = 0;
No4 = 0;
No5 = 0;
// Ask user to continue with next trial
while (kbhit()) getch();

vn2 << "Press a key to continue..." << endl;
while (!kbhit());
getch();
vn2.clrscr();

// Display status
vn3 << "subject #" << subject << endl;
if (orient == 0)
{
    vn3 << "Orientation: 90 degrees" << endl;
}
else
{
    vn3 << "Orientation: 0 degrees" << endl;
}

if (feedback == 0)
{
    vn3 << "Feedback = No" << endl;
}
else
{
    vn3 << "Feedback = Yes" << endl;
}

// Local loop: 25 times a trial
for (unsigned ti=1;ti<26;ti++)
{
    Check = 0;

    // Choose a target (1-5) randomly and count the number of completed
    // trials per target
    NextDistance = random(5);
    NextDistance++;
    voltooid = 0;
    if (No1 >= 5) voltooid++;
    if (No2 >= 5) voltooid++;
    if (No3 >= 5) voltooid++;
    if (No4 >= 5) voltooid++;
    if (No5 >= 5) voltooid++;

    // If possible, avoid that two successive trials are the same
    while (Check!=1)
    {
        switch (NextDistance)
        {
            case 1:
            {
                if ((No1 >= 5) || (ti!=1) && (voltooid <= 3) && (Distance==1))
                {
                    while (NextDistance == 1)
                    {
                        NextDistance = random(5);
                        NextDistance++;
                    }
                }
            }
            else
            {
                No1++;
                Check = 1;
            }
        } break;
        case 2:
        {
            if ((No2 >= 5) || (ti!=1) && (voltooid <= 3) && (Distance==2))
                {
                    while (NextDistance == 2)
                    {
                        NextDistance = random(5);
                    }
                }
            }
        }
    }
}
NextDistance++;

} 
else 
{
    No2++;
    Check = 1;
}
break;

case 3:
{
    if ((No3 >= 5) || (ti!=1) && (voltooid <= 3) && (Distance==3))
    {
        while (NextDistance == 3)
        {
            NextDistance = random(5);
            NextDistance++;
        }
    } 
else 
{
    No3++;
    Check = 1;
    
}
break;

case 4:
{
    if ((No4 >= 5) || (ti!=1) && (voltooid <= 3) && (Distance==4))
    {
        while (NextDistance == 4)
        {
            NextDistance = random(5);
            NextDistance++;
        }
    } 
else 
{
    No4++;
    Check = 1;
    
}
break;

case 5:
{
    if ((No5 >= 5) || (ti!=1) && (voltooid <= 3) && (Distance==5))
    {
        while (NextDistance == 5)
        {
            NextDistance = random(5);
            NextDistance++;
        }
    } 
else 
{
    No5++;
    Check = 1;
    
}
break;
} /* end of switch */

// Display information about the next trial to conduct
vn2 << endl;
vnl << "For trial (#" << ti << ") use: " << endl;
vnl << endl;

// Show the next target as large characters
switch (NextDistance)
{
    case 1:
    
    vn2 << " " << blok << endl;
vnl << " " << blok << " " << blok << " " << blok << endl;
vnl << blok << " " << blok << " " << blok << endl;

    - 68 -
vn2 << blok << blok << blok << blok << blok << endl;
vn2 << blok << " " << blok << endl;
break;

case 2:
{
    vn2 << blok << blok << blok << blok << blok << endl;
    vn2 << " " << blok << " " << blok << endl;
    vn2 << " " << blok << blok << blok << endl;
    vn2 << " " << blok << " " << blok << endl;
    vn2 << blok << blok << blok << blok << endl;
}
break;

case 3:
{
    vn2 << " " << blok << blok << blok << blok << endl;
    vn2 << blok << endl;
    vn2 << blok << endl;
    vn2 << blok << endl;
    vn2 << " " << blok << blok << blok << blok << endl;
}
break;

case 4:
{
    vn2 << blok << blok << blok << blok << endl;
    vn2 << " " << blok << " " << blok << endl;
    vn2 << " " << blok << " " << blok << endl;
    vn2 << " " << blok << " " << blok << endl;
    vn2 << blok << blok << blok << blok << endl;
}
break;

case 5:
{
    vn2 << blok << blok << blok << blok << blok << endl;
    vn2 << blok << endl;
    vn2 << blok << blok << endl;
    vn2 << blok << endl;
    vn2 << blok << blok << block << block << endl;
}
break;

} /* end of switch */
vn2 << endl;

// convert target numbers 1-5 to A-E
Distance = NextDistance + 64;
vn2 << "(Distance #: " << Distance << ")" << endl;
to << "For trial (#" << ti << ") use distance #: " << Distance << endl;

// Create safety-interval between two trials to avoid skipped trials

clock_t start, end;
start = clock();
double wachttijd = 0;
while (wachttijd < 2)
{
    end = clock();
    wachttijd = (end - start)/CLK_TCK;
}
while (kbhit()) getch();
vn2 << endl;
vn2 << "Press a key to continue..." << endl;
while (!kbhit());
getch();
vn2.clearscr();
vn1.clearscr();
vn1 << "Previous (#" << ti << "," << Distance << ")" << endl;
v1 << endl;
Distance = Distance - 64;
Appendix B-2: Class TtactileBallRoll

```cpp
#include <math.h>
#include "objc.h"
#include "base.h"
#include "basebld.h"
#include "fastbld.h"
#include "wacmsg.h"
#include "basereqs.h"
#include "basedlg.h"
#include "actiodlg.h"
#include "objc_id.h"
#include "asyncase_id.h"
#include "pointer.h"
#include "timer.h"
#include "common.h"
#include "vistext.h"
#include <stdio.h>
#include <scrollbar.h>
#include "mainwin.h"

class TtactileBallRoll : public TtactileBase
{
public:
    TtactileBallRoll(Tobject*);
    TtactileBallRoll(StreamableInit);
    virtual TtactileBase *copy(Tobject *newParent) const;
    virtual TsizerBase *buildSizer(PTWindow);
    virtual void modify(PTwindowsObject parent);
    virtual unsigned long msgHandler(HWND hWnd, unsigned msg,
                                    unsigned short wParam, unsigned long lParam);
    virtual int hitTest(int x, int y);
    virtual const char *description() const;
    static PTStreamable build();
    static const Pchar name;
    static const char *desc;

protected:
    virtual Pvoid read(Ripstream);
    virtual void write(Ropstream);

private:
    TtactileBallRoll(const TtactileBallRoll&, Tobject *newParent);
    virtual void getBounds(RECT& bounds);
    virtual void paintBlueprint(HDC);
    void setBallPos(int x, int y);
    void enableConveyor();
    void disableConveyor();
    virtual const Pchar streamableName() const;

private:
    TtactileBallRoll(const TtactileBallRoll&, Tobject *newParent);
    virtual void getBounds(RECT& bounds);
    virtual void paintBlueprint(HDC);
    void setBallPos(int x, int y);
    void enableConveyor();
    void disableConveyor();
    virtual const Pchar streamableName() const;
    char Distance, NextDistance; // Target numbers
    int Check; // Controls selection of targets
    unsigned N01, N02, N03, N04, N05, voltooid;
    unsigned feedback; // 0: visual feedback off, 1: visual feedback on
    unsigned uitvoer; // flag for outputfile
    unsigned trials; // number of conducted trials per block
    int ti; // total number of conducted trials
    int subject; // subject-number
    int enableSpeed, roltijd; // flags to avoid timing problems
    int dx, dy; // size object
    float Vox, Voy; // measured speed
    float Vct; // calculated speed
    float aw; // acceleration factor
    float np; // number of pixels on circumference of ball cross-section */
    float ds; // diameter ball in meters */
    float pi;
    float dt; // sample time in seconds*/
    double hoek;
    float Vlaatst; // Previous velocity
    double Vnew;
    float Xnew, Ynew, Xoud, Youd; // screen coordinates
    float radiusInt;
    float piOverR; float cursorD2;
    int radiusSqr;
    void calcPoints(PPOINT p) const;
    void calcPoints(PPPOINT p) const;
    int dx, dy; // size object
    float force; // force applied in object
    float angle; // rotating angle
    Tobject* ball_obj;
    float sin_angle, cos_angle;
    int fsin_angle, fcos_angle;
};
```
TtactileBallRoll::TtactileBallRoll(Tobject *p): TtactileBase(p)
{
    dx = 1;
    dy = 1;
    force = 0;
    angle = 0;
    ball_obj = 0;
}

TtactileBallRoll::TtactileBallRoll(StreamableInit i)
{
}

"Skipped Part: contains the implementation of several member-functions"

static FILE* tmpf = 0;

#pragma argsused
unsigned long TtactileBallRoll::msgHandler(HWND hWnd, unsigned msg,
unsigned short wParam, unsigned long lParam)
{
    switch (msg)
    {
    case TM_STARTEXEC:
        sin_angle = sin(angle);
        cos_angle = cos(angle);
        fsin_angle = force * sin_angle;
        fcos_angle = force * cos_angle;
        
        POINT p = { 0, 0 };
        float f6 = force * 0.125; // 11->8 bits
        int fc = f6 * cos_angle;
        int fs = f6 * sin_angle;
        parentObj->clientToVirtual(p);
        POINT c(4);
        calcPoints(c);
        objId = runPointer->newTacPoly(p.x, p.y, fc, fs, 4, c);
    }

    // Parameter-initialization
    Vox = Voy = 0.0;
    Vct=0.00000001;
    aw=-0.0411;
    np=1900;
    ds=0.057;
    pi=3.1416;
    dt=0.05;
    enableSpeed=1;
    Vlaatst=0;
    Xnew=0;
    Ynew=0;
    Xoud=50;
    Youd=240;
    uitvoer=0;
    Vlaatst=0.0;
    rolttijd=1;
    teller2=0;
    runPointer->requestSpeedNotifications( this );
    randomize();
    trials = 0;
    ti=0;
    subject = 0;
    enabledSetting = 0;
    }
    break;

    case TM_CHAR:

    // If no output file has been opened yet, open one based on keystroke

    if (uitvoer == 0)
    {
        switch (wParam)
        
        - 72 -
case '1':
{
    tmpf = fopen("c:\\marco\\rolling1.txt", "w");
    uitvoer = 1;
    subject = 1;
}
break;
case '2':
{
    tmpf = fopen("c:\\marco\\rolling2.txt", "w");
    uitvoer = 1;
    subject = 2;
}
break;
case '3':
{
    tmpf = fopen("c:\\marco\\rolling3.txt", "w");
    uitvoer = 1;
    subject = 3;
}
break;
case '4':
{
    tmpf = fopen("c:\\marco\\rolling4.txt", "w");
    uitvoer = 1;
    subject = 4;
}
break;
case '5':
{
    tmpf = fopen("c:\\marco\\rolling5.txt", "w");
    uitvoer = 1;
    subject = 5;
}
break;
case '6':
{
    tmpf = fopen("c:\\marco\\rolling6.txt", "w");
    uitvoer = 1;
    subject = 6;
}
break;
case '7':
{
    tmpf = fopen("c:\\marco\\rolling7.txt", "w");
    uitvoer = 1;
    subject = 7;
}
break;
case '8':
{
    tmpf = fopen("c:\\marco\\rolling8.txt", "w");
    uitvoer = 1;
    subject = 8;
}
break;
} /* end of switch wParam */

fprintf(tmpf, "This file contains the data about subject \%i \n", subject);

// use subject number to determine experimental order:
if ((subject % 2) == 0)
{ /* Start with feedback */
    feedback = 1;
}
else
{ /* start with no feedback */
    feedback = 0;
}

// Determine first target of the experiment
Distance = 0;
No1 = 0;
No2 = 0;
No3 = 0;
No4 = 0;
No5 = 0;
Initial position of ball on the screen

```c
int Xscherm = 50;
int Yscherm = 240;
POINT cp = { Xscherm, Yscherm };
setBallPos( cp.x, cp.y );
```

Determine first target (1-5) of the experiment (continued)

```c
Check = 0;
trials = 1;
ti = 1;
NextDistance = random(5);
NextDistance++;
```

Convert target number (1-5) into A-E and show it to the user

```c
Distance = NextDistance + 64;
sprintf(p, "Roll ball to distance: %c", Distance );
if ( parentObj->visual )
{
((VisualText*)parentObj->visual)->setText( p );
}
```

Count number of completed target distances

```c
voltooid = 0;
switch (Distance)
{
    case 1: No1++; break;
    case 2: No2++; break;
    case 3: No3++; break;
    case 4: No4++; break;
    case 5: No5++; break;
}
break;
```

Measure ball-velocity

```c
Vox = ((short*)&lParam)[1];
Voy = ((short*)&lParam)[0];
```

Convert pixels to m/s

```c
float Vx = (Vox*ds*pi)/np;
float Vy = (Voy*ds*pi)/np;
double d2 = pow(Vx,2) + pow(Vy,2);
if (d2 > 0.000001)
    Vnew = sqrt( d2 );
else
    Vnew = 0.0;
```

Skip very low velocities

```c
if (Vnew>0.085) // Skip very low velocities
{
    if(Vnew>Vlaatst) // top of velocity is still to come
    {
        theTimer->cancel(parentObj, DF_TACTILE);
        theTimer->set(parentObj, 50 , parentObj, DF_TACTILE, 0);
        Vct=Vnew;
        fprintf(tmpf, "Vo= %f \n", Vct);
        teller=0;
    }
    float li = 100.0 / Vnew;
    float Vxn = Vx * li;
    float Vym = Vy * li;
    POINT p = { Vxn, Vym };
    hoek = pointAngle(p);
}
else // Maximum velocity has been measured now
{
    enableConveyor(); // enable force-field
}
Vlaatst = Vnew;
break;
```
case TM_TIMER: if (roltijd)
    { /* Calculate new location of virtual ball on screen as long
         as it rolls */
        // Determine new screen-coordinates of virtual ball in meters
        Vct+=4*aw*dt;
        float D=Vct*dt;
        Xnew=(0*sin(hoek));
        Ynew=(0*cos(hoek));
        int Xscherm;
        int Yscherm;
        teller++;
teller2++;
        if (teller2==5) /* disable speedmeasurements 250 ms after top
         in initial velocity. In this interval
         it is possible to measure a new top and
         thus to improve reliability of the program */
        {
            enablespeed=0;
            sprintf(p, " "); // Disable user-instructions
            if ( [parentObj->visual] )
            {
                [TVvisualText*]parentObj->visual)->setText( p );
            }
        }
        if (Vct>O) /* as long as the virtual ball is rolling, every
         50 ms the screen should be updated */
        {
            theTimer->cancel(parentObj, DF_TACTILE);
            theTimer->set(parentObj, 50, parentObj, DF_TACTILE, 0);
        }
        // Convert screen coordinates in meters to pixels for 29": 640*480
        if (feedback==1)
        {
            Xscherm=50+(Xnew*480/0.43);
            Yscherm=240+(Ynew*480/0.43);
            // Avoid the ball to roll over the edge of the virtual table
            if (Xscherm>50)
            {
                Xscherm=50;
                Yscherm=Youd;
            }
            if (Yscherm>50)
            {
                Yscherm=50;
                Xscherm=Xoud;
            }
            if (Yscherm>430)
            {
                Yscherm=430;
                Xscherm=Xoud;
                Xoud=Xscherm;
                Youd=Yscherm;
            }
            else // in no-feedback condition the ball does not roll
            {
                Xscherm = 50;
                Yscherm = 240;
            }
        }
        else
        {
            Xscherm=Xoud;
            Yscherm=Youd;
            disableConveyor();

            if (teller<25) /* Safety margin
            {
                theTimer->cancel(parentObj, DF_TACTILE);
                theTimer->set(parentObj, 50, parentObj, DF_TACTILE, 0);
            }
            else
            {
                teller2=0;
                roltijd=0;
            }
        }
    }
sprintf(p, "Prepare for next trial");
if (parentObj->visual)
{
(TvisualText*)parentObj->visual)->setText(p);
}

// Safety-interval of 1.5 seconds
theTimer->cancel(parentObj, DF_TACTILE);
theTimer->set(parentObj, 1500, parentObj, DF_TACTILE, 0);
}
POINT cp = { Xscherm, Yscherm };
setBallPos(cp.x, cp.y);
else /* Calculate next target distance */
{
enableSpeed=1;
roltijd=1;
sprintf(p, "Working...");
if (parentObj->visual)
{
(TvisualText*)parentObj->visual)->setText(p);
}

If less than 50 trials have been conducted yet, calculate next distance
if (trials<50)
{
if (trials==25) // Change feedback condition after 25 trials
{
  ti=0;
  Distance=0;
  No1=0;
  No2=0;
  No3=0;
  No4=0;
  No5=0;
  if (feedback == 1) feedback = 0; else feedback = 1;
  trials++;
  ti++;
  Check = 0;
}

// Calculate next target
NextDistance = random(5);
NextDistance++;

// Count number of completed target distances
voltooid = 0;
if (No1 >= 5) voltooid++;
if (No2 >= 5) voltooid++;
if (No3 >= 5) voltooid++;
if (No4 >= 5) voltooid++;
if (No5 >= 5) voltooid++;

// Check for two same distances in succession
while (Check!=1)
{
  switch (NextDistance)
  {
  case 1:
    [...
      if ((No1 >= 5) || ((ti!=1) && (voltooid <= 3) && (Distance==1)))
      {
        while (NextDistance == 1)
        {
          NextDistance = random(5);
          NextDistance++;
        }
      }
  case 2:
  [...
}
if ((No2 >= 5) || ((ti!=1) && (voltooid <= 3) && (Distance==2)))
{
    while (NextDistance == 2)
    {
        NextDistance = random(5);
        NextDistance++;
    }
    else
    {
        No2++;
        Check = 1;
    }
    break;
}

case 3:
{
    if ((No3 >= 5) || ((ti!=1) && (voltooid <= 3) && (Distance==3)))
    {
        while (NextDistance == 3)
        {
            NextDistance = random(5);
            NextDistance++;
        }
        else
        {
            No3++;
            Check = 1;
        }
    }
    break;
}

case 4:
{
    if ((No4 >= 5) || ((ti!=1) && (voltooid <= 3) && (Distance==4)))
    {
        while (NextDistance == 4)
        {
            NextDistance = random(5);
            NextDistance++;
        }
        else
        {
            No4++;
            Check = 1;
        }
    }
    break;
}

case 5:
{
    if ((No5 >= 5) || ((ti!=1) && (voltooid <= 3) && (Distance==5)))
    {
        while (NextDistance == 5)
        {
            NextDistance = random(5);
            NextDistance++;
        }
        else
        {
            No5++;
            Check = 1;
        }
    }
    break;
} /* end of Check-switch */

// Show target distance as character A-E
Distance = NextDistance + 64;
sprintf(p, "Roll ball to distance: %c ", Distance);
if ( parentObj->visual )
{
    ((VisualText*)parentObj->visual)->setText( p );
}

fprintf(tmpf, "trial (total, sub, feedback, distance) %u %u %u %c
",
    trials,ti, feedback, Distance);

- 77 -
// Return ball to initial location for next trial
int Xscherm=50;
int Yscherm=240;

POINT cp = {Xscherm, Yscherm};
setBallPos(cp.x, cp.y);
#else
int Xnew=0;
int Ynew=0;
if (parentObj->visual)
    ((TvisualText*)parentObj->visual)->setText(p);
#else
    sprintf(p, "End of experiment");
    if (parentObj->visual)
        ((TvisualText*)parentObj->visual)->setText(p);
#endif
fclose(tmpf);

break;
case TM_STOPEXEC:
    runPointer->resetSpeedNotifications(this);
    break;
Mailing list

Beun
Bongers
Bouwhuis
Eggen
Gigi
Hermes
Kaufholz
Keyson (5 ex.)
Kohlräusch
Pauws
Piwek
Rypkema
Van de Sluis

M.J. Konert (5 ex.)
Isabellalaan 2
5583 XC Waalre

Prof. C. Michaels (2 ex.)
VU Amsterdam
Fac. der Bewegingswetenschappen
a-622, Van der Boechorststraat 7
1081 BT Amsterdam

Ir. G.L.J.M. Jansen
TUE, Fac. Elektrotechniek
EH 9.26

Faculteitsbibliotheek
TUE, Fac. Elektrotechniek
EH 2.06

IPO-directeur
Dr. ir. W. Strijland, Octr. en Merken, WAH

5 archief