

Untersuchung von Effekten Sensumotorischer Transformationen bei grafischen Eingabegeräten

Diplomarbeit am
Institut für Psychologie
Prof. Dr. Jochen Müsseler
Arbeits- und Kognitionspsychologie
in Zusammenarbeit mit dem
Lehrstuhl Medieninformatik
Prof. Dr. Jan Borchers
Fachgruppe Informatik
RWTH Aachen



von
Laszlo Bardos

Matrikelnummer: 211197

Erstgutachter:
Prof. Dr. Jochen Müsseler

Zweitgutachter:
Prof. Dr. Jan Borchers

Datum der Anmeldung: 30.01.2007
Datum der Abgabe: 18.04.2007

Überblick

Bei der Benutzung von Werkzeugen wirken Transformationen, die Bewegungen des Körpers in Bewegungen des Werkzeuges überführen. Im Gegensatz zu physikalischen Werkzeugen sind Computereingabegeräte hinsichtlich ihrer Transformation variabel. Im Rahmen dieser Arbeit wurde eine Experimentalsoftware entwickelt, mit der Fitts' Tapping Aufgaben durchgeführt werden können. Die Cursortransformationen können dabei in den Trials verändert werden. Eine Pilotstudie wurde ebenfalls durchgeführt, um Fragen zu Effekten der wechselnden Transformationen auf die subjektiven und objektiven Leistungsmaße der Probanden zu beantworten. Dabei interessierte vor allem, ob bei konstanten Controllerdistanzen und -zielgrößen die Cursortransformation einen Einfluss auf die Leistung der Versuchspersonen hat.

Abstract

When a tool is used, transformations are in effect, that carry over the movements of the hand into movements of the tool. In contrast to physical tools, most computer input devices are variable with respect to their transformation. In order to conduct experiments based on Fitts Tapping tasks, an experimental software was developed as part of this work. The software allowed to change the cursor transformation between trials. A pilot study was conducted to answer the question, whether the changes in transformation had an effect on the subjective and objective performance of the subjects. The main interest lay in the impact of cursor transformation on the performance, if controller distances and target sizes remain constant.

Eidesstattliche Erklärung

Ich versichere an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie alle wörtlich oder sinngemäß übernommenen Stellen in der Arbeit gekennzeichnet habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Aachen, 18. April 2007

Laszlo Bardos

Danksagung

Bei der Erstellung dieser Arbeit wurde ich von vielen Seiten unterstützt. Hiermit möchte ich mich bei allen Freunden und Bekannten bedanken, die sich als Versuchspersonen zur Verfügung gestellt haben und mir auf diese Weise die Durchführung der Experimente sehr erleichtert haben. Weiterhin bedanke ich mich bei meiner Freundin Cita Weickert, die mich nicht nur beim Korrekturlesen unterstützt hat, sondern auch wertvolle Ratschläge zu Schreibstil und Gliederung gegeben hat. Für die Möglichkeit, diese Diplomarbeit am Lehrstuhl für Psychologie schreiben zu dürfen möchte ich Prof. Jochen Müsseler herzlich danken, sowie Prof. Jan Borchers für die Betreuung als Zweitgutachter. Mein ausgesprochener Dank geht an Dr. Christine Sutter, die mich intensiv betreut hat und sich immer Zeit nahm, um meine Fragen zu beantworten oder Anregungen zu geben. Auch bei den Herausforderungen der statistischen Auswertung stand sie mir helfend zu Seite.

Inhaltsverzeichnis

1	Einführung	8
2	Theoretische Grundlagen	10
2.1	Untersuchungen von Fitts	10
2.1.1	Fitts' Tapping Aufgabe	11
2.1.2	Fitts' Gesetz	11
2.2	Werkzeugtransformation	13
2.3	Cursortransformation	14
2.4	Stimulus-Response Kompatibilität	18
2.5	Theorie der Ereigniskodierung	20
2.6	Fragestellung	24
3	Umsetzung der Software	26
3.1	Spezifikationen des Experiments	26
3.2	Matlab als Entwicklungsumgebung	30
3.3	Psychtoolbox Erweiterung	31
3.4	Grafische Darstellung	32
3.5	Wechsel der Cursortransformation	33
3.6	Balancierung der Ausdrucksstärken	37
3.7	Zentimeter und Pixel als Maßeinheiten	39
3.8	Detektierung der Treffer - 1. Variante	40
3.9	Detektierung der Treffer - 2. Variante	42
3.10	Protokollierung der Daten	43
3.11	Verwaltung der Blöcke	45

3.12	Distanzschätzung	47
3.13	Abschlusstest	48
3.14	Datenreduktion	48
3.15	Instruktionen	54
3.16	Zusammenfassung	55
4	Pilotstudie	56
4.1	Methode	56
4.2	Ergebnisse	62
4.2.1	Hypothese 1: Transformation und Bewegungszeit	62
4.2.2	Hypothese 2: Transformation und Distanzschätzung	65
4.2.3	Hypothese 3: Nacheffekte bei Transformationswechsel	68
4.3	Diskussion	69
5	Diskussion	71
5.1	Einordnung der Ergebnisse	71
5.2	Software und Versuchsanordnung	72
5.2.1	Matlab und Psychtoolbox	72
5.2.2	Verzerrungen des Bildschirms	73
5.2.3	Fluktuation der Zielgrößen	74
5.2.4	Abweichungen in den Transformationen	75
5.2.5	Zeitliche Verzögerung	76
5.2.6	Registrierung von Under- und Overshoots	77
5.2.7	Handposition auf dem Tablett	78
5.2.8	Hoher Fehlerquotient	79
5.2.9	Distanzschätzung	80
5.2.10	Datenreduktion und Korrektur der Fehlerquoten	80
5.2.11	Instruktionen	82
5.3	Ausblick	83
5.3.1	Fragestellungen für zukünftige Studien	83
5.3.2	Erweiterung der Software	84
5.4	Zusammenfassung	85

6 Fazit

86

Literaturverzeichnis

88

1 Einführung

In der heutigen, modernen Welt benutzen wir Menschen immer seltener unsere ganze Muskelkraft, um eine Aufgabe zu bewältigen. Selbst in den meisten Alltagssituationen nehmen uns elektrische oder mechanische Geräte die körperliche Arbeit ab. Diese Erleichterung steht jedoch der Herausforderung bei der Bedienung der Maschinen gegenüber. Jeder, der den Führerschein erwerben möchte, muss sich beispielsweise in stundenlangen Fahrstunden mit der Steuerung des Fahrzeuges vertraut machen und lernen, welche Bewegungen zu welchen Resultaten führen. Für eine fehlerfreie und routinemäßige Beherrschung eines Autos sind sogar Jahre des Übens notwendig. Nicht jedes Gerät ist gleich schwer zu bedienen. Daher wird intensiv geforscht, um herauszufinden, welche Faktoren bei der Mensch-Maschine-Interaktion für die Schwierigkeit der Aufgabe eine Rolle spielen.

Schon in den 50er Jahren des letzten Jahrhunderts hat Paul Fitts einen Zusammenhang gefunden, zwischen der Entfernung und Größe eines Zielobjektes und der benötigten Zeit, die für das manuelle Erreichen dieses Objektes benötigt wird. Diese Untersuchungen wurden später auf viele Bereiche ausgeweitet, unter anderem auch auf Computereingabegeräte. Bei der Interaktion mit einem Computer geschieht die Interaktion bis auf wenige Ausnahmen, wie z.B. dem Touchscreen, immer durch ein indirektes Eingabegerät, das als Werkzeug dient. Neben der häufig verwendeten Maus und Tastatur, kommen auch Grafiktablets, Trackpoints, oder Joysticks zum Einsatz. Jedes dieser Geräte setzt die Bewegungen des Körpers anders in Bewegungen des Bildschirmcursors um. Diese Werkzeugtransformationen besitzen im Gegensatz zu den klassischen Werkzeugen, wie Hammer oder Schraubendreher variable Eigenschaften. Eine Computermaus kann auf verschiedene Empfindlichkeiten eingestellt werden. Bei Anwendungen die mit dem Joystick gesteuert werden sind die vertikalen Bewegungsrichtungen häufig vertauscht.

Um den Umgang mit Computereingabegeräten zu verbessern, werden von Forschern und Produktentwicklern häufig Benutzertests durchgeführt. Mit den Ergebnissen

aus diesen Tests werden die Geräte schrittweise verbessert, bis sie einem bestimmten Qualitätsstandard entsprechen. Dieses praxisnahe Vorgehen lieferte neue Erkenntnisse über das Design von Eingabegeräten (Douglas und Mithal, 1997). Ein Nachteil dieser Methode ist jedoch, dass sie nach dem Try-and-Error Prinzip arbeitet und sich nicht einfach auf neue Eingabegeräte übertragen lässt. Aus diesem Grund wird in der Kognitionspsychologie versucht, die Grundlagen der menschlichen Wahrnehmung und Motorik besser zu verstehen. Sind diese Zusammenhänge einmal verstanden, dann können Vorhersagen über zukünftige Eingabegeräte getroffen werden, bevor diese überhaupt gebaut werden.

In dieser Arbeit wurde ein Experiment mit einem Grafiktablett auf der Grundlage von Fitts' Versuchen durchgeführt. Die Versuche fanden am Computer statt und die dafür benötigte Experimentalssoftware wurde selbst entwickelt. Mit Hilfe der Software konnte die Transformation zwischen Eingabegerät und Bildschirmausgabe variiert werden, so dass die Effekte der Werkzeugtransformation untersucht werden konnten. Die Ergebnisse aus dieser Arbeit werden in zukünftige Forschungsreihen einfließen, in denen die gefundenen Effekte genauer betrachtet und validiert werden.

2 Theoretische Grundlagen

In diesem Kapitel werden die Konzepte beschrieben, die die Grundlage für die aktuelle Arbeit bildeten. Hierbei wird sowohl auf Untersuchungen eingegangen, die seitens der Methodik, als auch seitens des theoretischen Rahmenwerks für diese Arbeit relevant waren. Angefangen von Fitts' Experimenten bis hin zu ganz aktuellen Ergebnissen im Bereich der Ereigniskodierung, wird der Stand der Forschung wiedergegeben. Am Ende des Kapitels werden aus den vorgestellten Forschungsergebnissen die Fragestellungen abgeleitet, die für die Entwicklung der Experimentalsoftware und der Pilotstudie als Ausgangspunkt dienen.

2.1 Untersuchungen von Fitts

Die Arbeit von Fitts (1954) bildete, nicht nur auf dem Forschungsgebiet der Psychologie und der Mensch-Maschine-Interaktion, die Grundlage für eine Vielzahl von Untersuchungen. Fitts' Gesetz ist nach Schmidt (1982) beachtlich, weil es eine robuste, mathematische Vorhersage für viele motorische Handlungen des Menschen liefert. Aufgrund der hohen Komplexität des menschlichen Bewegungsapparates sind solch präzise Vorhersagen ansonsten nur selten möglich. Fitts' Gesetz behält für ein sehr breites Spektrum von motorischen Handlungen seine Gültigkeit, wenn auch die mathematischen Formulierungen an die jeweiligen Domänen angepasst werden mussten.

Die aktuelle Untersuchung hatte Fitts' Gesetz zwar nicht als Kernpunkt, aber bediente sich sowohl einer ähnlichen Versuchsmethodik, als auch einer vergleichbaren Operationalisierung der Variablen. Aus diesem Grund wird nicht in aller Tiefe auf die Herleitung der Formeln und den informationstheoretischen Hintergrund eingegangen.

2.1.1 Fitts' Tapping Aufgabe

In Abbildung 2.1 ist die Versuchsanordnung von Fitts zu sehen. Die Apparatur hat zwei rechteckige Bereiche, die von den Probanden abwechselnd mit einem Stift berührt („tapped“) werden. Hierfür haben die Probanden 15 Sekunden Zeit, um so viele Treffer wie möglich zu erzielen. Sensorflächen links und rechts der Zielbereiche registrieren fehlerhafte Treffer, also Bewegungen die zu kurz (Undershoot) oder zu lang (Overshoot) waren. Die Versuchspersonen werden instruiert, sich mehr auf Genauigkeit als auf Geschwindigkeit der Bewegung zu konzentrieren.

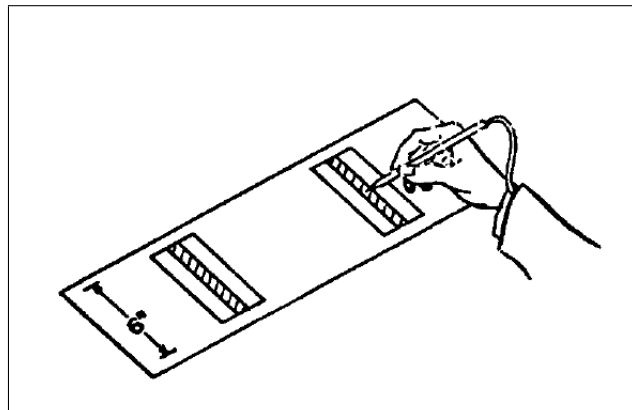


Abbildung 2.1: Versuchsanordnung von Fitts' Tapping Aufgabe (Fitts, 1954)

Fitts variierte als unabhängige Variable den Abstand der Zielbereiche (gemessen von den Mittelpunkten aus) und deren Breite. Die Höhe der Zielbereiche war konstant und so hoch gewählt, dass ungewollte Abweichungen der Zielbewegungen zu keinen Fehlern führten. Die Untersuchung beschränkte sich also auf eindimensionale Bewegungen in horizontaler Richtung. Als abhängige Variablen wurden die durchschnittliche Bewegungszeit und die Anzahl der fehlerhaften Zielbewegungen definiert (Fitts, 1954).

2.1.2 Fitts' Gesetz

Aufbauend auf den Theorien von Shannon und Weaver (1949) aus dem Bereich der Informationstheorie, entwickelte Fitts ein objektives Maß, mit der die Schwierigkeit der Zielbewegungen beschrieben werden konnte. Aus der Entfernung der Mittelpunkte der Zielbereiche (A) und deren Breite (W), wurde die Schwierigkeit der

Aufgabe (ID) wie folgt definiert:

$$ID = \log_2 \left(\frac{2A}{W} \right)$$

Aus diesen Überlegungen folgte Fitts eine Formel, mit der sich die Bewegungszeit (MT) einer Zielbewegung vorhersagen lässt:

$$MT = a + b \cdot ID$$

Die Konstanten a und b sind empirisch zu bestimmen.

Die von Fitts' Gesetz vorhergesagten Bewegungszeiten können die in Experimenten tatsächlich gemessenen Zeiten sehr genau vorhersagen. Die Schwierigkeit der Aufgabe ist also durch die Entfernung des Ziels und seiner Größe bestimmt. Darüber hinaus besteht ein linearer Zusammenhang zwischen der Aufgabenschwierigkeit und der Ausführungszeit.

Fitts' Gesetz wurde seit seiner Entdeckung in vielen empirischen Untersuchungen bestätigt, wenn auch Modifikationen der Formel nötig waren, um die Vorhersage weiter zu präzisieren (MacKenzie und Buxton, 1992). Weiterhin haben Card, English, und Burr (1978) gezeigt, dass Fitts' Gesetz selbst dann gültig ist, wenn statt der direkten Interaktion mit einem Stift auf einer Platte die Interaktion durch ein Zwischenmedium wie einem Computer stattfindet. Bewegungen eines Bildschirmcursors mit der Maus scheinen den gleichen motorischen Prinzipien zu gehorchen, wie Bewegungen im nicht-virtuellen Raum. Für experimentelle Studien hat das eine große Signifikanz, da viele Untersuchungen am Computer leichter zu realisieren sind.

Grosjean, Shiffrar, und Knoblich (2006) haben in einer aktuellen Untersuchung herausgefunden, dass Fitts' Gesetz nicht nur gültig ist wenn Bewegungen ausgeführt werden, sondern auch wenn fremde Bewegungen beobachtet werden. Die Probanden bekamen Fotos von Zielbewegungen präsentiert und mussten schätzen, ob eine bestimmte Zeitspanne ausreichen würde, um diese Zielbewegung erfolgreich zu absolvieren. Die Gültigkeit von Fitts' Gesetz für wahrgenommene Bewegungen stärkt die Annahme, dass Handlungsausführung, -wahrnehmung und motorische Vorstellung ähnliche Prozesse zugrunde liegen.

2.2 Werkzeugtransformation

Bei Bewegungshandlungen ist zu unterscheiden, ob ein Werkzeug als zwischengeschaltetes Medium verwendet wird oder nicht. Wird beispielsweise eine Tasse mit der Hand angehoben, dann wird das Objekt direkt mit der Hand manipuliert. In diesem Fall ist die Bewegung der Körpertransformation zuzuschreiben. Die Werkzeugtransformation hingegen beschreibt die Relation zwischen Körperbewegung und Bewegung des Werkzeuges, mit dessen Hilfe das Zielobjekt manipuliert wird. Dieses Werkzeug kann etwas sehr einfaches sein, wie z.B. ein Schraubenzieher oder Stift, aber auch eine komplexe Maschine, wie ein Kran oder ein Flugzeug.

Heuer (1983) unterscheidet zwei Aspekte von Werkzeugtransformationen. Diese nennt er Bahntransformation und räumlich-zeitliche Transformation. Die Bahntransformation beschreibt wie die Bewegungsbahn des Körpers in Bewegungsbahnen des Werkzeuges übersetzt wird. Hierbei können alle Teile des Körpers zur Steuerung des Werkzeuges benutzt werden, nicht nur Hände oder Füße – wobei diese hauptsächlich bei Objektmanipulationen benutzt werden. Eine Kurbel für die Fensterscheibe im Auto transformiert eine Kreisbewegung der Hand in eine lineare Bewegung der Fensterscheibe. Im Falle eines elektrischen Fensterhebers wären sowohl Körper- als auch Werkzeugbewegung lineare Bewegungen. Zu den linearen bzw. drehenden Bewegungen kommt noch die Bewegungsrichtung als dritter Faktor hinzu. Wird die Kurbel der Fensterscheibe im Uhrzeigersinn gedreht, dann kann sich das Fenster nach unten aber auch nach oben bewegen – abhängig von der Konstruktion. Möglich ist auch, dass die Drehung der Hand im Uhrzeigersinn in einer Drehung des Werkzeuges gegen den Uhrzeigersinn resultiert. Auf die Auswirkungen einer solchen inkompatiblen Transformation wird in Kapitel 2.4 eingegangen.

Der räumlich-zeitliche Charakter einer Werkzeugtransformation legt fest, wie die Bewegung des Körpers als Eingangsgröße $e(t)$ in Bewegungen des Werkzeuges als Ausgangsgröße $a(t)$ im zeitlichen Verlauf übersetzt werden. Die einfachste Transformation ist die Proportional-Transformation

$$a(t) = k \cdot e(t),$$

bei der die Körperbewegung um einen Faktor k verstärkt wird. Das bedeutet, dass (im Falle einer linear zu linear Transformation) eine Bewegung der Hand um 10 cm, durch das Werkzeug in eine Bewegung von $k \cdot 10$ cm übersetzt wird. Die Versuche, die im Rahmen dieser Arbeit durchgeführt wurden, haben sich auf diese Art von

Transformation konzentriert. Neben der Proportional-Transformation bezeichnet Heuer sowohl die Totzeit- also auch die Integral-Transformation als bedeutsam. Die Totzeit-Transformation

$$a(t) = e(t - \tau)$$

reagiert erst mit einer Verzögerung (τ) auf die Bewegungen des Körpers. Komplexe Transformationen lassen sich über die Integral-Transformation beschreiben

$$a(t) = \int e(t)dt.$$

Im Falle des elektrischen Fensterhebers wirkt eine Integral-Transformation. Wird der Knopf gedrückt, bewegt sich die Scheibe von diesem Augenblick an mit konstanter Geschwindigkeit nach oben, ohne dass der Finger eine weitere Bewegung durchführt. Natürlich lassen sich wesentlich komplexere Transformationen beschreiben, vor allem da sich die verschiedenen Arten von Transformationen beliebig kombinieren lassen.

Heuer beschreibt die Schwierigkeit einer Transformation unter anderem als davon abhängig, wie sehr sie von der Proportional-Transformation abweicht. Aber auch die Anpassung an veränderbare Werkzeugtransformationen erhöht die Schwierigkeit der Interaktion. Im alltäglichen Gebrauch haben die Werkzeuge eine Proportional-Transformation mit einem Verstärkungsfaktor, der immer den gleichen Wert hat. In Experimenten von Young (1969) wurde der Verstärkungsfaktor der Transformation so verändert, dass er sich in den Zielbewegungen der Probanden niederschlug. War der Verstärkungsfaktor kleiner als erwartet, passierte ein Undershoot (siehe Abb. 2.2 oben), während ein großer Faktor zu Overshoots führte (Abb. 2.2 unten). Nach ein bis zwei Zielbewegungen hatten sich die Probanden jedoch an die neue Verstärkung angepasst, so dass keine Under- bzw. Overshoots dieser Form mehr registriert wurden.

2.3 Cursortransformation

Im Gegensatz zu physikalischen Werkzeugen sind die Parameter – insbesondere die Transformation – bei der Interaktion mit einem Computer variierbar. Aktuelle Betriebssysteme erlauben dem Benutzer beispielsweise, die „Mausempfindlichkeit“ zu verändern. Diese Funktion verändert bei einer Proportional-Transformation der Maus den Verstärkungsfaktor k . Es gibt aber auch die sog. „Power-Mouse“, der

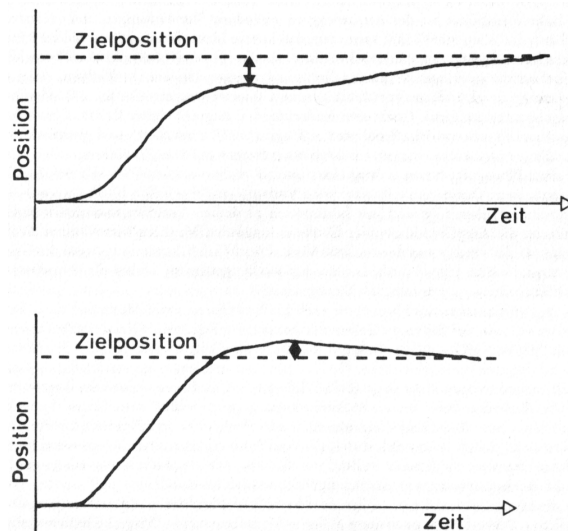


Abbildung 2.2: Under- und Overshoot bei unerwarteter Verstärkung in der Transformation (Young, 1969)

keine Proportional- sondern eine Integraltransformation zu Grunde liegt. Abhängig davon, wie schnell die Maus bewegt wird, ändert sich der Verstärkungsfaktor. Mit schnellen Bewegungen können größere Strecken zurückgelegt werden, als mit einer herkömmlichen Maus.

Graham (1996) verglich drei verschiedene Transformationsarten in einer Fitts' Tapping Aufgabe. Die Proportional-Transformation betrug 1:1, die zwei Integral-Transformationen variierten zwischen 1:1 und 1:2. Graham fand heraus, dass die beiden Integral-Transformationen schlechter abschnitten als die Proportional-Transformation. Die Bewegungszeit, die eingespart wurde weil der Cursor die Distanz zum Ziel schneller überwinden konnte, wurde am Ende der Zielbewegung zunichte gemacht. Die Ansteuerung des Zieles erfolgte durch Feinjustierungen. In dieser Phase veränderte sich die Transformation bei den Integral-Transformationen, so dass die Feinjustierung gestört wurde.

Die Transformation wird bei Eingabegeräten mit dem Display/Control Gain angegeben (Arnaut und Greenstein, 1990):

$$Gain = \frac{Displaydistanz}{Controllerdistanz}.$$

Der Gain bzw. die Transformation ergibt sich aus der auf dem Bildschirm sichtbaren Bewegung des Cursors und der mit dem Eingabegerät zurückgelegten Distanz.

Diese Definition ist aber nicht eindeutig, da die gleiche Transformation auf unterschiedliche Weisen erreicht werden kann. Eine 1:1 Transformation eines Systems kann auf 2:1 erhöht werden, in dem entweder die Ausgabedistanz vergrößert, oder die Eingabedistanz verkleinert wird. Konkret kann eine Vergrößerung der Ausgabedistanz erreicht werden, wenn ein doppelt so großer Bildschirm an das System angeschlossen wird. Die gleiche Bewegung des Eingabegerätes resultiert nun in einer längeren Distanz auf dem Bildschirm. Auf der anderen Seite kann auch die Empfindlichkeit des Eingabegerätes erhöht werden. Die Bewegungen des Eingabegerätes werden im System mit einem Faktor verstärkt, so dass für die gleiche Displaydistanz eine kürzere Bewegung des Eingabegerätes benötigt wird. Diese beiden Methoden haben den gleichen Gain-Wert, sind aber durch unterschiedliche Konfigurationen erreicht worden.

In ihrer Arbeit untersuchten die beiden Forscher die Auswirkungen, wenn Displaydistanz, Controllerdistanz und die Zielgrößen separat variiert werden. Sind die Display- und Controllerdistanz festgelegt, dann bestimmt die Transformation auch das Verhältnis von Display- und Controllerzielgröße. In Abbildung 2.3 ist dieser Zusammenhang verdeutlicht.

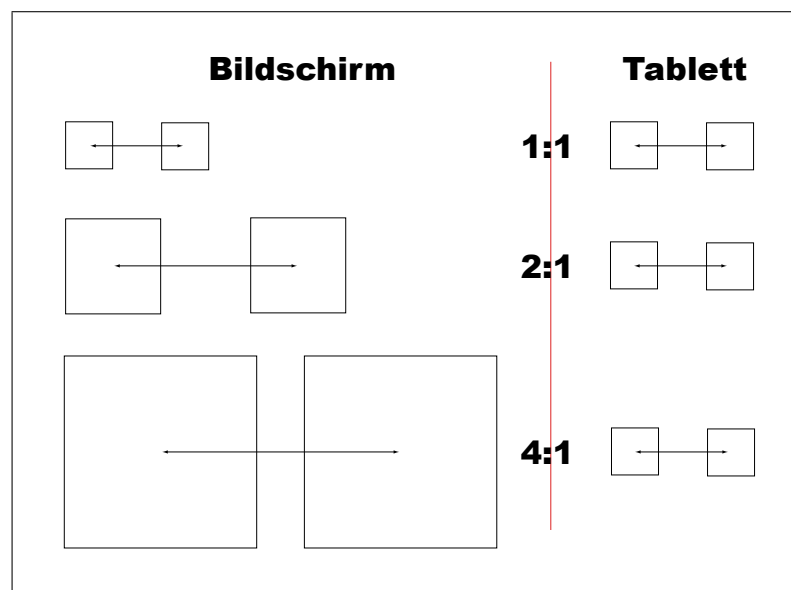


Abbildung 2.3: Control- und Displaydistanzen, sowie Zielgrößen bei unterschiedlichen Transformationen

In den Experimenten wurde den Probanden an vier Positionen des Bildschirms ein Cursor dargeboten, den sie in das Zielkästchen bewegen mussten. Das Zielkästchen

befand sich jeweils auf der gegenüberliegenden Seite des Stimulus. Die Endposition des Cursors musste mit einem Tastendruck bestätigt werden. In den Trials wurden drei Cursortransformationen von 1:1, 1:2 und 1:3 verwendet. Die Ergebnisse der Untersuchung zeigten, dass trotz konstantem Gain bzw. Transformation die Displayamplitude einen Effekt auf die Ausführungszeit hatte. Der Gain alleine ist daher kein genaues Maß, um ein Interface eindeutig zu beschreiben. Vielmehr müssen alle Faktoren von Cursor- und Displayamplitude, sowie Cursor- und Displayzielgröße in Betracht gezogen werden. Werden zwei dieser Freiheitsgrade nicht spezifiziert, können diese Werte – bei konstantem Gain – beliebig variiert werden. Hierbei treten Diskrepanzen zwischen den visuell und motorisch wahrgenommenen Bewegungen auf. In den Experimenten wurde auch untersucht, ob bei gleich schweren Aufgaben (nach Fitts ID) der Gain einen Effekt auf die Bewegungszeit hat. Hierbei variierten die Bewegungszeiten bei Trials mit gleicher ID (Arnaut und Greenstein, 1990).

Wie bereits erwähnt, zeigten die Untersuchungen von Card et al. (1978), dass Fitts' Gesetz auch für Zielbewegungen bei Computereingabegeräten ihre Gültigkeit hat. Graham und MacKenzie (1996) kamen jedoch teilweise zu anderen Ergebnissen. In ihren Experimenten mussten die Probanden eine typische Fitts' Tapping Aufgabe bearbeiten, in der anstatt Zielkästchen, Kreise als Targets präsentiert wurden. Die Probanden konnten im ersten Durchgang die Bewegung ihres Fingers direkt sehen, im zweiten Durchgang wurde die Sicht auf die Hand durch eine halbdurchlässige Scheibe versperrt. Unter dieser Bedingung wurden die Bewegungen des Fingers von einer Kamera erfasst und durch einen Computer auf die Scheibe projiziert. Die Probanden sahen anstatt ihres Fingers einen virtuellen Cursor an gleicher Stelle. Die Bewegungen des Fingers wurden in Echtzeit berechnet und sorgten für eine wahrheitsgetreue Darstellung der Bewegungen. Die Ergebnisse zeigten, dass sich die Ausführungszeiten zwischen realer und virtueller Bedingung unterschieden. Die erste Phase der Zielbewegungen zeigte noch keine Unterschiede bezüglich des Geschwindigkeitsprofils. Erst in der zweiten Phase der Bewegung, in der die visuelle Kontrolle der Fingerposition für eine erfolgreiche Zielbewegung am wichtigsten ist, zeigte sich ein Unterschied im Profil. Die Ausführung in der zweiten Phase war verlangsamt, was insgesamt zu einer längeren Bewegungszeit führte. Graham und MacKenzie vermuteten, dass die fehlenden räumlichen Informationen der virtuellen Bedingung dazu führten, dass die Bewegungen langsamer waren. Während die Probanden den Finger nach unten bewegten, erzeugten diese Bewegungen keine Rückmeldung durch den Cursor, da dieser nur zweidimensional abgebildet wurde.

Die Untersuchungen zu Effekten der Cursortransformationen haben gezeigt, dass am Computer durchgeführte Fitts' Tapping Aufgaben geeignet sind, Leistungsunterschiede in den Zielbewegungen aufzuzeigen. Phänomene, wie Over- und Undershoots nach Wechseln der Cursortransformationen und die unterschiedlichen Leistungswerte bei identischen Fitts' ID sollen im Rahmen dieser Arbeit untersucht werden.

2.4 Stimulus-Response Kompatibilität

Bei Untersuchungen fanden Fitts und Seeger (1953) heraus, dass Antworten auf Stimuli schneller und genauer erfolgten, wenn der Stimulus räumlich passend zur geforderten Reaktion war. Rechtsseitige Stimuli wurden mit rechtsseitigen Reaktionen (ipsilaterale Bedingung) mit einer geringeren Reaktionszeit beantwortet, als mit linksseitigen Reaktionen (contralaterale Bedingung). Dieser Effekt trat auch auf, wenn die Position des Stimulus für die Reaktion keine Bedeutung hatte und von den Probanden ignoriert werden sollte (Kunde, Müsseler, und Heuer, 2006). Aus Fitts' Ergebnissen wurde geschlossen, dass es leichter ist auf den Stimulus zu reagieren, wenn Stimulus und Response (S-R) zueinander räumlich kompatibel sind.

Bei einer Computermaus bewegt sich der Cursor auf dem Bildschirm in die gleiche Richtung wie die Hand auf dem Tisch. Erscheint z.B. ein anzuklickendes Dialogfenster auf dem Bildschirm, dann liegen Dialogfenster, Cursorbewegung und der Effekt der Bewegung in einer kompatiblen Anordnung vor. Das Ziel der Bewegung befindet sich ebenfalls auf der Seite auf die die Cursorbewegung hinzielt. Es liegt also eine S-R kompatible Zuordnung vor. Bei manchen Werkzeugen gibt es diese übereinstimmenden Bewegungsrichtungen jedoch nicht. Wird bei einer Operation mit einem Endoskop gearbeitet, dann sind die Hand- und die Werkzeugbewegungen entgegengesetzt: Bewegt der Operateur die Hand nach links, dann bewegt sich das Ende des Endoskops im Körper nach rechts. Diese Unterschiede in der Bewegungsrichtung führen zu einer S-R Inkompatibilität (Kunde et al., 2006)

Hommel (1993) fand in seinen Experimenten jedoch heraus, dass S-R Kompatibilität nicht immer eine schnellere Reaktion begünstigt. Er ließ Probanden abhängig von den präsentierten Stimuli Knöpfe drücken. Erschien ein Stimulus auf der rechten Seite, mussten die Versuchspersonen den linken Knopf drücken. Für einen Stimulus auf der linken Seite musste analog der rechte Knopf gedrückt werden. Wurde ein

Knopf gedrückt, ging gleichzeitig auf der gegenüberliegenden Seite eine Lampe an – also immer auf der Seite des Stimulus. Eine Versuchsgruppe bekam die Instruktion, einen linken Stimulus mit einem *rechten Knopfdruck* zu beantworten, während die zweite Gruppe instruiert wurde, bei einem linken Stimulus die *linke Lampe anzuschalten* (für rechtseitige Stimuli war alles jeweils gespiegelt). Die beiden Gruppen unterschieden sich nur in den Instruktionen, so dass beide eine S-R inkompatible Anordnung vorliegen hatten (Knopfdruck auf der anderen Seite als der Stimulus). Es zeigte sich, dass die Instruktion einen Einfluss auf die Reaktionszeiten der Probanden hatte. Nicht die S-R Kompatibilität, sondern die Übereinstimmung von Stimulus und Handlungseffekt (S-E) schien für schnellere Reaktionszeiten zu sorgen. Die Gruppe, die mit dem Knopfdruck instruiert wurde, hatte eine S-E inkompatible Anordnung. Der Effekt ihrer Handlung (Knopfdruck) war auf der gegenüberliegenden Seite des Stimulus. Die zweite Gruppe hatte eine S-E kompatible Anordnung, da der beabsichtigte Effekt (Lampe anschalten) auf der Seite des Stimulus stattfand.

Für die endoskopische Operationsmethode besteht die Frage, welchen Einfluss die S-R Inkompatibilität der Hand- und Werkzeugbewegung auf die Leistung des Benutzers hat. Kunde et al. (2006) versuchten diese Frage zu beantworten, in dem sie einen Versuchsaufbau nach dem Prinzip der endoskopischen Werkzeuge wählten. In Abbildung 2.4 ist der schematische Aufbau zu sehen. Das Werkzeug ist um den Angelpunkt in der Mitte drehbar und kann zum linken oder rechten Stimulus bewegt werden. Die Probanden mussten das Werkzeug entweder am näheren Griff (indirekte Bedingung) oder am entfernten Griff (direkte Bedingung) festhalten. In der direkten Bedingung bewegte sich die Spitze des Werkzeugs in die gleiche Richtung wie die Hand der Probanden und in der indirekten Bedingung entgegengesetzt. Abhängig von der Farbe des Stimulus musste die Spitze zum Stimulus hin, bzw. vom Stimulus weg bewegt werden. Es gab insgesamt vier Bedingungen: S-R kompatibel bzw. S-R inkompatibel und S-E kompatibel und S-E inkompatibel.

Die Ergebnisse zeigten, dass die S-E Kompatibilität für die Reaktionszeit entscheidend war. Weiterhin hat sich gezeigt, dass die Reaktionen der indirekten Bedingung langsamer waren als die der direkten. Stimmt die Handbewegung nicht mit der Bewegung der Werkzeugspitze überein, waren die Reaktionszeiten länger. In einer zweiten Studie konnten diese Effekte bestätigt werden, obwohl hier die räumliche Lage des Stimulus für die Reaktion nicht relevant war. Als Ursache für die gefundenen Effekte wurde die räumliche Aufmerksamkeit vermutet. Die räumlichen Positionen von Stimulus, Werkzeugspitze und Hand mussten während der

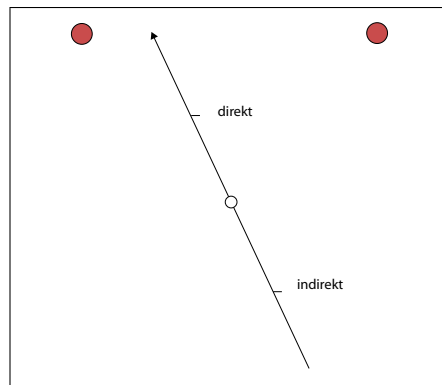


Abbildung 2.4: Drehbares Werkzeug mit direkter und indirekter Bedingung

Planungsphase der Handlung beachtet werden. Lagen diese auf entgegengesetzten Seiten, dann sorgte der Wechsel der Aufmerksamkeit für Verzögerungen. Interessanter Weise waren die beobachteten Effekte weiterhin vorhanden, wenn die Probanden während den Experimenten ihre eigene Hand nicht sehen konnten. Es wurde vermutet, dass die vorgestellte Position der Hand wichtiger war, als die tatsächlich wahrgenommene visuelle Position.

Die Kompatibilität der Stimulusposition und des Effektes der Handlung hat offenbar eine positive Wirkung auf die auszuführende Bewegung. Die mentale Vorstellung der Bewegung hat ebenfalls Einfluss auf die motorische Ausführung. Im Rahmen dieser Untersuchung wurde die Kompatibilität von Stimulus und Effekt durch die unterschiedlichen Cursortransformationen realisiert.

2.5 Theorie der Ereigniskodierung

In vielen Modellen der kognitiven Psychologie sind die Bereiche Handlung und Wahrnehmung strikt voneinander getrennt (z.B. Posner (1978)). So wird in den linearen Stufenmodellen angenommen, dass Prozesse der Wahrnehmung völlig isoliert von späteren handlungsbezogenen Verarbeitungsabläufen betrachtet werden können. Es wird auch angenommen, dass die Repräsentationen der Stimuli im Gehirn an anderer Stelle abgelegt sind als die Repräsentationen der Antworten des Körpers auf den Stimulus. Mit der Zeit wurden die Modelle erweitert. Mehrere parallele Verarbeitungsprozesse sollen für die Wahrnehmung verantwortlich sein und liefern als Ergebnis das Perzept (Milner und Goodale, 1995). Erst im Folgenden werden die Handlungseffekte bearbeitet, jedoch ohne direkte Kommunikation zwi-

schen Wahrnehmung und Handlung. Neuere Arbeiten kamen zum Schluss, dass in manchen Bereichen scheinbar doch eine Verbindung zwischen wahrgenommenen Handlungen und der Erzeugung von Handlungsmustern existieren muss. In Studien konnten Probanden anhand von Schriftmustern die dynamischen Bewegungsmuster passend zu diesen Mustern reproduzieren (Babcock und Freyd, 1988). Weitere Belege sind in der Gehirnforschung von Affen zu finden (Rizzolati et al., 1996). Es wurden Spiegel-Neuronen gefunden, die sowohl bei der Wahrnehmung von motorischen Abläufen, als auch bei der Ausführung dieser Handlungen aktiv waren (Übersicht in Hommel, Müsseler, Aschersleben, und Prinz (2001)).

Die Theorien über Handlungsplanung verfolgen ebenfalls zwei unterschiedliche Ansätze. In den sensumotorischen Modellen (Sanders, 1998) werden Handlungen als Reaktionen auf Stimuli betrachtet. Im Gegensatz dazu erklären die ideomotorischen Modelle (Gollwitzer und Bargh, 1996) Handlung als Ergebnis von internen zielorientierten Prozessen. Die sensumotorische Sicht erklärt Handlungen anhand von physikalischen Prinzipien. Nach diesem Modell sind bei strukturierten Aufgabenstellungen die Stimuli sowohl zwingend erforderlich als auch ausreichend, um eine angemessene Antwort der Probanden auszulösen. Der ideomotorischen Sicht folgend, sind Handlungen in willensgesteuerten Prozessen verwurzelt. Äußere, stimulusgesteuerte Einflüsse werden als Ursache in dem ideomotorischen Modell nicht in Betracht gezogen. Dem Modell zufolge ist die Vorstellung einer durchgeführten Bewegung die mentale Ursache für ihre Ausführung und nicht ein wahrgenommener Stimulus. Im Gehirn spielen daher aus ideomotorischer Sicht die Repräsentationen der Zielsetzungen die gleiche zentrale Rolle wie in der Sensumotorik die Repräsentation der Stimuli. Der interne Charakter von Zielsetzungen erschwert jedoch deren experimentelle Untersuchung, da sich diese Prozesse im Gegensatz zu stimulusbezogenen Abläufen nur schwer beobachten lassen. Aus diesem Grund gibt es keine ganzheitlichen Ansätze für das Verständnis vom Zusammenspiel von Zielsetzungen und Handlungsabläufen (Hommel et al., 2001).

Diese unterschiedlichen Modelle können zwar viele der in Experimenten beobachteten Effekte erklären, aber eben nicht alle Phänomene. Aus diesem Grund haben (Hommel et al., 2001) den Versuch unternommen, die verschiedenen Theorien zu einer einheitlichen Theorie der Ereigniskodierung zusammenzuführen. Hierzu werden Handlungen als Bindeglied zwischen Bewegungen und Zielsetzungen betrachtet. Zusammen mit Wahrnehmung, Aufmerksamkeit und Intention operieren sie alle auf einer gemeinsamen mentalen Domäne der Repräsentation. Weiterhin wird angenom-

men, dass die Wahrnehmung und die Planung einer bestimmten Handlung funktionell äquivalent sind, da sie beide interne Repräsentationen von externen Ereignissen sind. In klassischen Theorien über die Repräsentation von wahrgenommenen Stimuli werden diese anhand von einzelnen Merkmalen beschrieben, die in verschiedenen Bereichen des Gehirns verarbeitet werden. Die Theorie der Ereigniskodierung folgert daher analog, dass für einen Handlungsplan verschiedene Handlungsmerkmale existieren, aus denen sich der vollständige Plan zusammensetzt. Es wird betont, dass die Ereigniskodierung auf einer distalen und daher abstrahierten Ebene abläuft. Das bietet den Vorteil, dass Wahrnehmung und Handlungsplanung sich – anstatt auf domänenspezifische Modalitäten – auf den Informationsinhalt eines Ereignisses beziehen können.

Die distalen Merkmale – oder auch Merkmalskodes – sind nicht spezifisch für einen Stimulus oder eine Reaktion. Diese Merkmalskodes können beispielsweise Farben sein oder relative Positionen wie „Links“ oder „Rechts vom Körper“. In Abbildung 2.5 ist zu sehen wie die Interaktion von sensorischen und motorischen Kodes über die gemeinsamen Merkmalskodes ablaufen. Der Merkmalskode f_1 soll beispielsweise repräsentieren, dass ein Stimulus links vom Betrachter erscheint, und f_2 steht für einen hochfrequenten Ton. Die sensorischen Kodes s_1 bis s_3 stellen das visuelle System dar und s_4 bis s_6 das auditive System. Die Interaktion des sensorischen Systems mit den Merkmalskodes zeigt, dass sowohl das auditive System in der Positionsbestimmung mitwirkt, als auch dass das visuelle System zur Erkennung des hohen Tons aktiv ist (z.B. durch Erkennen des Instrumentes). Wenn nun ein Handlungsplan entsteht, dann bestehen diese aus genau den gleichen Merkmalskodes, auf die sich das sensorische System bezogen hat. Eine motorische Reaktion auf das Wahrgenommene wird durch die Merkmalskodes beeinflusst. Das System für die Steuerung der Hände (m_1 bis m_3) wird durch f_1 stimuliert, eher linksgerichtete Handlungen auszuführen, während gleichzeitig das Sprachsystem (m_4 bis m_6) zur Erzeugung der Worte „links“ vorbereitet wird. Diese Interaktion zeigt, dass die Merkmale sich untereinander beeinflussen. Es kann hierbei vorkommen, dass sich verschiedene Merkmalskodes beeinflussen, die eigentlich zu unterschiedlichen Ereignissen gehören. Diese Beeinflussung kann dabei für die jeweiligen Handlungen hinderlich oder förderlich sein. Dies entspricht z.B. den empirisch beobachteten S-R Effekten.

Wird ein bestimmtes Objekt wahrgenommen, dann aktiviert dieser Ereigniskode die entsprechenden Merkmalskodes. Eine wahrgenommene Kirsche aktiviert Kodes

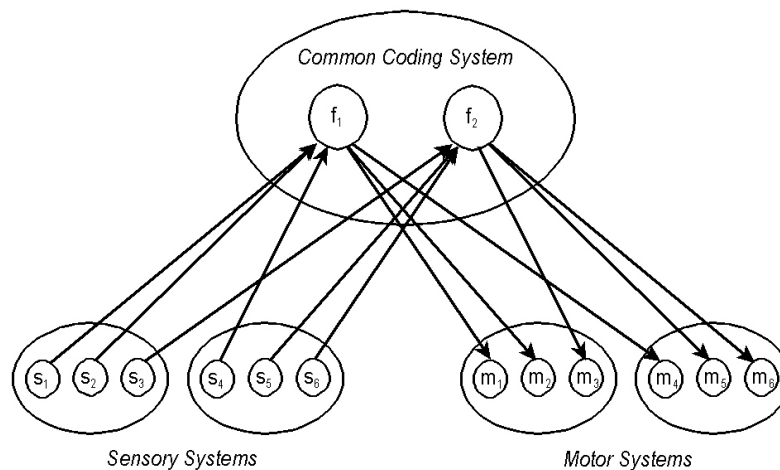


Abbildung 2.5: Merkmalskodierung. Sensorische Information s_i , motorische Codes m_i und gemeinsame Merkmalskodes f_i (Hommel et al. (2001) S. 862)

wie *klein*, *rund* und *rot*. Diese Aktivierung erzeugt jedoch noch keinen neuen Ereigniskode. Ansonsten würde die Wahrnehmung ein anderes Objekt wie z.B. ein Apfel mit den Merkmalen *groß*, *rund* und *grün* dazu führen, dass fünf Merkmalskodes gleichzeitig präsent sind. Die Merkmalskodes könnten nicht mehr zugeordnet werden, so dass diese Codes zuerst integriert werden müssen. Dieser Integrationsprozess verbindet die zusammengehörigen Merkmale temporär miteinander zu Ereignisrepräsentationen.

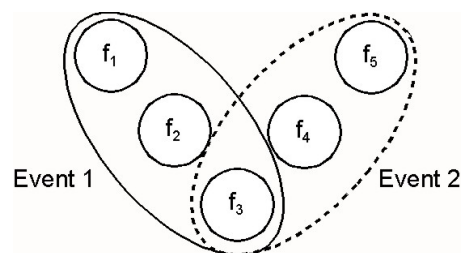


Abbildung 2.6: Integration von Merkmalskodes zu Ereignisrepräsentationen (Hommel et al. (2001) S. 863)

In Abbildung 2.6 ist eine solche Integration von Merkmalskodes zu sehen. Ereignis 1 wurde aus den Merkmalen f_1 , f_2 und f_3 integriert. Das konkurrierende Ereignis 1, das Merkmal f_3 aktiviert hat, behindert nun die Kodierung von Ereignis 2. Mit diesem Szenario können sowohl Interferenzeffekte erklärt werden, als auch die

zeitlichen Abläufe dieser Prozesse. Es wird angenommen, dass die stimulusbezogenen Merkmalkodes im ersten Schritt parallel aktiviert werden. Im nachfolgenden Schritt werden die aktivierten Merkmale integriert, und sind nicht mehr für andere Prozesse verfügbar. Die Integration der Merkmale geschieht auf der Seite der Wahrnehmung durch Steuerung der Aufmerksamkeit. Auf der Seite der Handlungsplanung ist die Integration mit spezifischen Vorbereitungen auf die Handlung verbunden. Um den selektiven Charakter der Wahrnehmung einzubeziehen, wird in der Theorie der Ereigniskodierung hervorgehoben, dass diese Kodierungen in hohem Maße von den Zielen der Person abhängig sind. Die einzelnen Merkmalkodes werden situationsbedingt unterschiedlich gewichtet und beeinflussen dadurch die resultierenden Ereigniskodes. Die für eine bestimmte Aufgabe relevanten Merkmale werden bevorzugt aktiviert, so dass die Gewichtung auf die spätere Integration indirekten Einfluss hat. Diese Gewichtung geschieht bei Wahrnehmungsprozessen durch die Aufmerksamkeit und bei der Handlungsplanung durch willensgesteuerte Prozesse.

Die Theorie der Ereigniskodierung ist in der Lage, viele empirisch beobachtete Effekte erklären zu können. So sind die in früheren Kapiteln erwähnten S-R und S-E Kompatibilitätseffekte mit Hilfe dieser Theorie sehr gut erklärbar.

2.6 Fragestellung

Seit den Experimenten von Fitts (1954) wurden etliche Untersuchungen durchgeführt, die sowohl praxis- als auch theorierelevante Aspekte der Motorik allgemein, und der Sensumotorik im speziellen, beleuchtet haben. Die Arbeiten von Arnaut und Greenstein (1990) haben für ihre Untersuchung von Cursortransformationen ebenfalls ein Paradigma gewählt, das auf Fitts' Experimenten basierte. Arnaut und Greenstein haben jedoch versucht die Frage zu beantworten, ob Gain zur Spezifikation eines Interfaces ein geeignetes Maß ist. Sie haben dabei weniger die motorischen Effekte der unterschiedlichen Transformationen untersucht. Die Experimente von Hommel (1993) haben gezeigt, dass es Unterschiede zwischen S-R kompatiblen und inkompatiblen Bedingungen gibt, abhängig von der Kompatibilität von Stimulus und Effekt der Handlung. Die Theorie der Ereigniskodierung erklärt diese Effekte durch das gemeinsame Kodierungssystem der Wahrnehmung und des motorischen Systems. Wahrgenommene Stimuli können auf nachfolgende Bewegungen hinderlich sein, wenn die Merkmalkodes der Bewegung in konkurrierenden Ereignissen

nisrepräsentationen gebunden sind. Bei veränderten Cursortransformationen eines Eingabegerätes ist die visuelle Rückmeldung der Cursorbewegung unterschiedlich zu der Bewegung der Hand. Wenn die Bewegung der Hand nicht sichtbar ist, beschränkt sich die Wahrnehmung auf die Bewegung des Cursors. Die Inkompatibilität von Hand- und Cursorbewegung kann nur durch die propriozeptorische Rückmeldung wahrgenommen werden. Der Theorie der Ereigniskodierung folgend, würde diese Wahrnehmung andere Merkmalkodes aktivieren, als die visuelle Rückmeldung durch den Bildschirmcursor. Basierend auf diesen Überlegungen stellten sich für die aktuelle Arbeit folgende Fragen:

- Hat die Cursortransformation bei gleicher Schwierigkeit der Aufgabe einen Effekt auf die Leistung? Verändert die visuelle Rückmeldung also die Schwierigkeit der motorischen Aufgabe?
- Führt die durch die Cursortransformation verfälschte visuelle Rückmeldung zu Überlagerungen der propriozeptorischen Empfindungen?
- Sind nach Wechsel einer Transformation die Probanden schnell in der Lage sich der veränderten Werkzeugtransformation anzupassen?

Der Fokus der Arbeit lag zu einem großen Teil in der Entwicklung der für die Experimente benötigte Software. Hierzu wurde eine Spezifikation ausgearbeitet, auf deren Grundlage das Programm entwickelt wurde. Die durchgeführte Pilotstudie sollte eventuelle Schwächen der Methodik sowie des Versuchsaufbaus herausstellen, so dass die zukünftig durchgeführten Experimente verlässliche Daten liefern. Im nächsten Kapitel wird sowohl auf die Spezifikation als auch auf deren Umsetzung detailliert eingegangen.

3 Umsetzung der Software

Ziel der gesamten Arbeit war es, nicht nur eine Pilotstudie durchzuführen, sondern zusätzlich auch das experimentelle Werkzeug für diese Studie zu entwickeln. Um die Effekte während der Interaktion mit Computereingabegeräten zu untersuchen, muss das Experiment nahe liegender Weise ebenfalls computergestützt ablaufen. Bei der Softwareentwicklung sind die größten Herausforderungen die abhängigen und unabhängigen Variablen präzise umzusetzen, und die Störeinflüsse zu eliminieren. In diesem Kapitel werden besondere Aspekte der Software beschrieben, die sich während der Konzeptionierung und Modellierung ergeben haben.

3.1 Spezifikationen des Experiments

Bevor auf die konkreten Umsetzungen der psychologischen Vorgaben eingegangen wird, soll an dieser Stelle erst ein Überblick über die Spezifikationen des Experiments dargestellt werden. Das soll helfen, später die Zusammenhänge und die Herausforderungen seitens der Software leichter zu überblicken.

Mit Hilfe der entwickelten Software sollte eine Fitts' Tapping Aufgabe mit einem Grafiktablett realisiert werden. In einer solchen Aufgabe sieht die Versuchsperson auf dem Bildschirm zwei Kästchen und soll hintereinander horizontale Zielbewegungen aus dem einen in das andere Kästchen ausführen. Sie steuert den Bildschirmcursor dabei mit dem Stift auf dem Grafiktablett. Im Gegensatz zu Fitts' Experimenten können auf einem Grafiktablett jedoch keine ballistischen Bewegungen durchgeführt werden. Die Bewegungen erfordern ständigen Kontakt zum Tablett und sind daher lineare Bewegungen. Ist eine Zielbewegung beendet, muss der Proband von dieser Endposition aus die nächste Zielbewegung in das gegenüberliegende Kästchen beginnen. Hier soll keine Pause zwischen den Zielbewegungen entstehen, so dass eine flüssige Bewegung entsteht. In Abbildung 3.1 ist zu sehen, wie die Aufgabe auf

dem Bildschirm, in der selbst entwickelten Software, aussah. Sowohl bezüglich der Form, als auch den Farben musste die Software die Aufgabe spezifikationsgetreu wiedergeben.

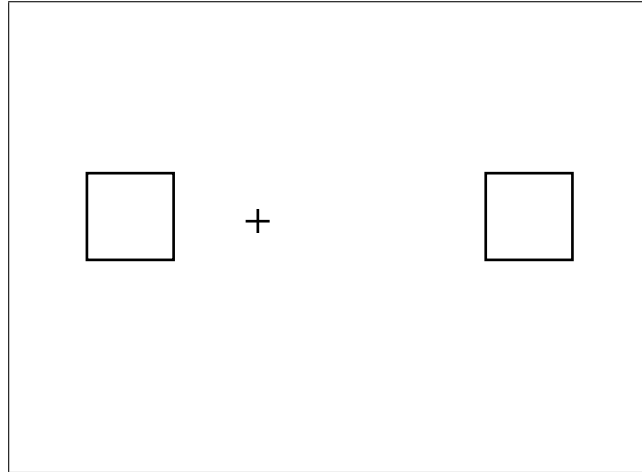


Abbildung 3.1: Fitts' Tapping Aufgabe auf dem Bildschirm

Die unabhängigen Variablen im Experiment die von der Software verändert werden mussten:

Zielgröße (ZG): Die Größe der Zielkästchen wurde variiert. Auf dem Bildschirm erschienen dabei zwei identische Kästchen, deren Größe 0,5 cm, 1 cm, 2 cm oder 4 cm betrug.

Cursortransformation (CT) meint den Verstärkungsfaktor zwischen zurückgelegter Distanz auf dem Grafiktablett und die resultierende Bewegung des Cursors auf dem Bildschirm. Die Transformationen betragen 1:1, 2:1 und 4:1. In der 4:1 Bedingung legte der Cursor also eine viermal längere Strecke auf dem Bildschirm zurück, als der der Proband mit dem Stift zurückgelegt hatte.

Tablettdistanz (TD) ist die zurückgelegte Strecke auf dem Tablett, die für eine Bewegung zwischen den Mittelpunkten der Kästchen erforderlich war. Die Tablettdistanzen waren hierbei 2 cm, 4 cm und 6 cm groß. Sie wurden blockweise dargeboten, so dass innerhalb eines Blocks die Displaydistanzen mit der jeweiligen Transformation angepasst werden mussten. Mit einer 2:1 Transformation musste der Abstand der Kästchen im Vergleich zur 1:1 Transformation verdoppelt werden, damit die zurückzulegende Tablettdistanz bei beiden Transformationen gleich groß war. Siehe hierzu auch Abbildung 2.3. Auf dem Bildschirm wurden die Größen und

Entfernungen der Kästchen so angepasst, dass die Targetgrößen und Distanzen auf dem Tablett gleich blieben.

Das Experiment hatte ein 3 (TD) x 3 (ZG) x 3 (CT) Design. Insgesamt waren daher 36 Bedingungen vorhanden, die jeweils 25 mal erfolgreich absolviert werden mussten.

Während des Experimentes musste die Software die geforderten abhängigen Variablen messen und speichern. Diese Variablen waren:

Bewegungszeit (BZ): Für jede Zielbewegung musste gemessen werden, wie viel Zeit (Sekunden) zwischen Anfang und Ende der Bewegung vergangen war. Das Ende einer Bewegung wurde eine Umkehrung der Richtung der Cursorbewegung definiert. Dieser Zeitpunkt war gleichzeitig der Anfang der Zeitmessung der nachfolgenden Bewegung.

Zielpräzision (ZP): Messung des Abstands (Pixel) des Bewegungsendpunktes vom Mittelpunkt des Zielkästchens.

Fehlerquotient (FQ): Anzahl der fehlerhaften Zielbewegungen im Vergleich zu allen Zielbewegungen. Der Fehlerquotient wurde nicht bezüglich einer Zielbewegung, sondern pro ZG und CT Kombination berechnet. Innerhalb eines Blocks musste die Versuchsperson bei jeder Kombination von ZG und CT Zielbewegungen durchführen, bis 25 gültige Treffer erreicht wurden. Verfehlte der Proband das Zielkästchen, war dies ein fehlerhafter Treffer. Die nachfolgende Zielbewegung wurde nicht als gültige Zielbewegung gezählt. Traf die Versuchsperson bei dieser Zielbewegung das Zielkästchen erneut nicht, galt das trotzdem als Fehler. Der Fehlerquotient wurde anhand dieser 25-er Durchgänge berechnet und setzte sich aus den fehlerhaften Zielbewegungen und den gesamten Zielbewegungen zusammen.

Geschätzte Tablettedistanz (GT): Nachdem die Probanden einen Block beendet hatten, wurden sie instruiert, die Tablettedistanz jenes Blockes zu schätzen (Zentimeter).

Maximale Cursorgeschwindigkeit (MV): Die größte Geschwindigkeit (Pixel/s) mit der der Cursor über den Bildschirm bewegt wurde.

Zeitpunkt der höchsten Cursorgeschwindigkeit (TV): Zeitpunkt (Sekunden) in der Zielbewegung, an dem die Cursorgeschwindigkeit am höchsten war.

Maximale Cursorbeschleunigung (MA): Die größte Beschleunigung (Pixel/s²) mit der der Cursor über den Bildschirm bewegt wurde.

Zeitpunkt der höchsten Cursorbeschleunigung (TA): Zeitpunkt (Sekunden) in der Zielbewegung, an dem die Cursorbeschleunigung am höchsten war.

Richtung der Zielbewegung (RZ): Position des Zielkästchens und damit die Richtung, in der die Zielbewegung ausgeführt werden musste. Da mehrere Zielbewegungen hintereinander durchgeführt wurden, wechselten sich die Richtungen einander ab.

Am Anfang des Experiments wurden den Probanden die Instruktionen vorgelesen und die Versuchsapparatur erklärt. Im Anschluss folgte der Testdurchlauf. In diesem Probelauf galt es für drei ZG (0,5 cm; 2 cm, 4 cm) und CT-Bedingungen (1:1; 2:1; 4:1) die 25 erfolgreichen Zielbewegungen zu erreichen. Nun war es Aufgabe der Probanden, die randomisierten TD-Blöcke zu bearbeiten.

Zwischen den Blöcken wurde den Probanden die Instruktionen für die Schätzung der Tablettedistanz vorgelesen. Die Versuchspersonen sahen zur Erinnerung für jede der Cursortransformationen ein Beispiel aus dem vorangegangenen Block auf dem Bildschirm. Hierbei interessierte, ob den Probanden auffiel, dass die auf dem Tablett zurückgelegte Strecke unabhängig von der Cursortransformation unverändert geblieben war. Zum Schätzen der Tablettedistanz sollten die Probanden einen Abstandsmarker bewegen. Der Abstand der Marker auf dem Bildschirm sollte der geschätzten Tablettedistanz entsprechen.

Nach der Schätzung, gab es für die Versuchspersonen eine kurze Pause, bevor der nächste Block begann.

Die letzte Aufgabe der Probanden war der Abschlusstest. Anstatt der Cursortransformation wurde jedoch nun die Tablettedistanz verändert. Die Größe der Zielkästchen war konstant, so dass die Tablettedistanz die einzige unabhängige Variable war. Die abhängigen Variablen blieben die gleichen. Die Versuchspersonen hatten die Aufgabe zuerst die üblichen 25 erfolgreichen Zielbewegungen mit der 4 cm Tablettedistanz auszuführen, bevor die nächsten drei Tablettedistanzen (2 cm; 6 cm; 7cm) mit jeweils fünf erfolgreichen Trials folgten. In diesem Teil des Experimentes sollte die Versuchsperson mit einer Cursortransformation die 4 cm Tablettedistanz üben. Nachfolgend wurde – im Gegensatz zum Rest des Experimentes – die Cursortransformation beibehalten und die Tablettedistanz verändert. Der Abschlusstest sollte zeigen, wie der Lerneffekt bei konstanten Transformationen sich entwickelt.

3.2 Matlab als Entwicklungsumgebung

Matlab als Entwicklungsumgebung für eine psychologische Untersuchung zu verwenden ist auf den ersten Blick sicherlich ungewöhnlich. Traditionell liegt die Stärke von Matlab in mathematischen Kalkulationen, speziell auf dem Gebiet der Matrixoperationen. Matlab ist dabei eine Skriptsprache, d.h. Programme die in der „M“ genannten Sprache geschrieben sind, werden vor der Ausführung nicht in Maschinencode übersetzt. Dieser Vorgang des Kompilierens ist in der Softwaretechnik eine gängige Praxis. Durch sie werden die resultierenden Programme u.a. schneller ausgeführt. Nicht nur die Geschwindigkeit bei der Ausführung, sondern auch der Gesamtumfang der Software spielt bei der Entscheidung für oder gegen eine Skriptsprache eine Rolle. Häufig werden Skriptsprachen nur für kleinere Projekte verwendet, was aber für diese Experimentalsoftware kein limitierender Faktor war.

Für den Einsatz von Matlab sprachen aber andererseits mehrere Argumente:

- **Erfahrung und Infrastruktur:** Matlab war am Institut, an dem die Experimente durchgeführt wurden, eine gängige Programmiersprache für die Entwicklung von Experimentalsoftware. Der Erfahrungsschatz aus diesen Versuchen bildete eine bewährte Grundlage – eine Art „Proof of Concept“ – für die Entwicklung der aktuellen Software. Weiterhin konnte auf eine bereits ausgebauten Infrastruktur zurückgegriffen werden, da Matlab auf allen relevanten Rechnern installiert und betriebsbereit war. Damit war auch auf diesem Bereich eine Basis für die Softwareentwicklung vorhanden. So wurde durch die zeitliche Kosten-Nutzen-Rechnung insgesamt klar Matlab favorisiert.
- **Plattformunabhängigkeit:** Das Institut verfügte über eine heterogene Computerstruktur. Ein Teil der Computer lief mit dem Windows Betriebssystem, während auf dem anderen Teil MacOS 9 bzw. MacOS X installiert war. Erschwerend kam hinzu, dass selbst die Experimentalrechner unterschiedliche Installationen aufwiesen. In Matlab geschriebene Programme haben den Vorteil, plattformübergreifend zu sein und ohne Anpassungen auf Windows und MacOS Systemen betrieben werden zu können. Später, in Kapitel 5.2.1 wird beschrieben, wie es zu unerwarteten Problemen beim Betrieb auf den unterschiedlichen Rechnern gekommen ist.
- **Wartung und Erweiterung:** In der Vergangenheit wurden am Institut mehrere Experimente auf Basis von Matlab Programmen durchgeführt. Von per-

soneller Seite war also gewährleistet, dass auch nach Abschluss dieser Arbeit die Software gewartet und erweitert werden konnte. Eine zeitintensive Einarbeitung in eine andere Entwicklungsumgebung entfällt auf diese Weise.

Eines der inhärenten Nachteile einer Skriptsprache musste aber noch ausgeglichen werden, nämlich die Geschwindigkeit und Präzision mit der die Programme ausgeführt werden können. Für psychologische Experimente ist neben einer sehr genauen Zeitmessung auch eine niedrige Latenzzeit unabdingbar, d.h. die Reaktionszeit des Systems auf eine Benutzereingabe muss so klein wie möglich gehalten werden. Wie bereits erwähnt sind diese Forderungen nicht ohne weiteres in Matlab realisierbar, so dass hierfür eine zusätzliche Erweiterung zum Einsatz kam. Im folgenden Kapitel wird näher auf dieses Programmpaket eingegangen.

3.3 Psychtoolbox Erweiterung

Das Psychtoolbox Programmpaket wurde speziell für computergestützte Experimente auf dem Feld der visuellen Psychophysik entwickelt. Es ist in der klassischen Programmiersprache „C“ geschrieben worden und bietet daher die nötige Schnelligkeit und Präzision bei der Ausführung. Das Paket bildet eine Schnittstelle zwischen der Hardware und Matlab, und kann sehr einfach aus Matlabskripten aufgerufen werden. Diese Synergie erlaubt die einfache Handhabung und Wartung der Experimentalssoftware, ohne im Bereich der Stimulusdarstellung und Messung an Genauigkeit zu verlieren (Brainard, 1997; Pelli, 1997). Hätten die benötigten Funktionen für die Grafikausgabe und Zeitmessung selbst entwickelt werden müssen, dann wäre der zeitliche Rahmen einer Diplomarbeit wohl überschritten worden. Gerade die Probleme mit der millisekundengenauen Zeitmessung sind unter Windows nicht trivial zu lösen (Chambers und Brown, 2003).

Psychtoolbox bietet eine große Anzahl von Funktionen, von denen aber nur eine begrenzte Auswahl in der Experimentalssoftware zum Einsatz kam. Diese waren Zeitmessung, Routinen zur Grafikausgabe und Manipulation des Bildschirmcursors. Ein Auszug aus dem Programmcode zeigt, welche Funktionen besonders häufig verwendet wurden:

```
% Mauszeiger auf Koordinaten (200,300) setzen  
setmouse(200,300);
```

```
% Einlesen der Mauskoordinaten in die zwei Variablen
[mouse_xcoord,mouse_ycoord] = getmouse;

% Aktuelle Zeit in Variable Speichern
trial_started = getsecs;

% Zeichnet einen Quader auf den Bildschirm
screen(display_screen,'FrameRect',[0 0 0],sourcebox,1,1);
```

Es ist sofort zu erkennen, dass die leicht erlernbare Programmsyntax von Matlab auch auf die Aufrufe der Psychtoolboxroutinen zutrifft. Der `screen` Befehl ist hierbei eine der vielfältigsten Funktionen der gesamten Toolbox. Neben der Verwaltung von Fenstern und Ausgabebildschirmen können grafische Objekte wie z.B. Ovale, Rechtecke, Polygone, Bögen oder Bitmapdaten auf dem Monitor dargestellt werden. Es liegt wohl an der großen Vielseitigkeit dieser Funktion, dass in den Windows und MacOS Versionen geringfügige Unterschiede in der Implementation zu finden sind. Die Unterschiede beschränken sich auf nebensächliche Bereiche, wie den Betrieb von mehreren Bildschirmen oder das Umschalten auf eine andere Farbtiefe. So hat es sich auch im Nachhinein bewährt, Psychtoolbox einzusetzen und auf diese Weise sowohl von der Plattformunabhängigkeit als auch von der hohen Geschwindigkeit zu profitieren.

3.4 Grafische Darstellung

Wie bereits erwähnt, werden die Psychtoolbox-Funktionen unter anderem dazu verwendet, Objekte auf den Bildschirm zu zeichnen. In den folgenden Kapiteln wird genauer auf die einzelnen Aspekte der Interaktion eingegangen. An dieser Stelle werden vorher die grundlegenden Techniken der grafischen Ausgabe beschrieben.

Der vorhin erwähnte `screen` Befehl ist in der Lage, z.B. ein Rechteck auf den Monitor zu zeichnen. Für die aktuelle Software reichte es aber nicht aus, Objekte statisch zu zeichnen. Der Cursor musste bei Positionsänderungen des Eingabegerätes auf eine neue Bildschirmkoordinate bewegt werden. Hierfür wurde eine recht simple Methode verwendet, bei der zuerst alle Objekte auf den Bildschirm

gezeichnet wurden. Das Programm wartete in einer Programmschleife auf eine Bewegung des Eingabegerätes. Erfolgte diese, dann wurde der Cursor „gelöscht“ und anhand der aktualisierten Position neu gezeichnet. Die Form des Cursors bestand aus zwei zueinander rechtwinkligen Linien und ließ sich daher sehr einfach durch Psychtoolbox-Funktionen zeichnen. Zur Sicherheit wurden die statischen Objekte – also die Zielkästchen – nach einer Cursorbewegung neu gezeichnet, falls durch das Löschen deren Konturen teilweise überschrieben wurden.

Mit dieser einfachen Methode konnte die Bewegung des Cursors realisiert werden. Im nächsten Kapitel wird beschrieben, warum der Cursor von der Software selbst gezeichnet werden musste, anstatt den normalen Betriebssystemcursor zu verwenden.

3.5 Wechsel der Cursortransformation

Wie der Titel dieser Arbeit schon ankündigt, sind (Werkzeug-)Transformationen ein zentraler Aspekt dieser Arbeit. In den Hypothesen (siehe Kapitel 2.6) bezieht sich aus diesem Grund eine der unanhängigen Variablen auf den Wechsel der Cursortransformation. Als die Grundlagen für die Software ausgearbeitet wurden, stellte sich die Frage auf welche Art und Weise dieser Wechsel technisch realisiert werden kann.

Seitens des Betriebssystems gibt es diverse Möglichkeiten, die Transformation des Eingabegerätes zu beeinflussen. Bei aktuellen Mäusen z.B. können neben der „Geschwindigkeit“ auch Funktionen wie „Zeigerbeschleunigung“, „Auflösung“ und „Abtastrate“ manipuliert werden. Die nahe liegende Idee war, direkt diese Geräteparameter zu benutzen, um während des Experiments die Transformation zu verändern. Der Funktionsumfang der Eingabegeräte ist jedoch sehr stark vom jeweiligen Modell abhängig, speziell vom Treiber des Herstellers. Die gewünschte Plattformunabhängigkeit der Software stellte ein zusätzliches Problem dar, da Windows und MacOS Treiber sich auch bei identischen Geräten stark unterscheiden können. Vergleichsweise schwierig gestaltete es sich auch, auf diese Parameter aus einem Programm heraus zuzugreifen. Psychtoolbox bietet für dieses Problem auch keine Lösung an, so dass ein anderer Weg gesucht werden musste, um die Transformation wechseln zu können.

Der Bildschirmcursor ist bei den heutigen Betriebssystemen ein fester Bestandteil

der Systemarchitektur und kann daher nur in begrenztem Umfang beeinflusst werden. Zwar kann z.B. das Aussehen des Cursors unter Windows und MacOS fast beliebig verändert werden, aber für die anderen Eigenschaften sind das Betriebssystem und die Gerätetreiber zuständig. In Abbildung 3.2 ist zu sehen, welchen Weg eine Bewegung des Eingabegerätes nimmt, bis ein Ergebnis sichtbar wird.

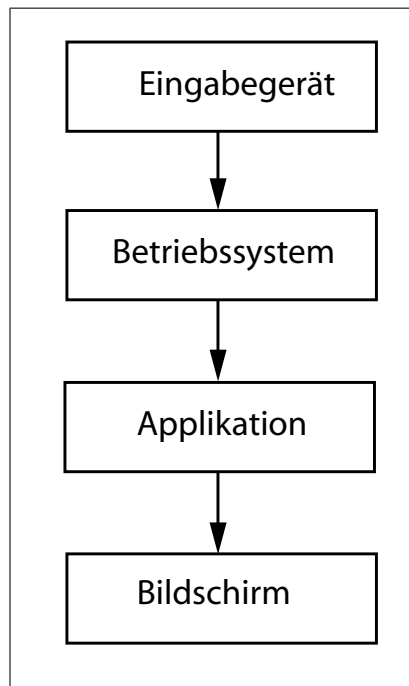


Abbildung 3.2: Stufen der Verarbeitung bei der Bewegung des Eingabegerätes

Grundsätzlich kann an jedem dieser vier Punkte die Transformation verändert werden. Durch mechanische oder optische Modifikationen am Eingabegerät ist es z.B. möglich, dass eine Maus beim Zurücklegen einer festen Strecke jeweils unterschiedliche Rückmeldungen an das Betriebssystem gibt. Genauso ist es vorstellbar, zwischen Bildschirm und Versuchsperson ein optisches System zu setzen und die visuelle Ausgabe des Computers auf diese Weise zu manipulieren. Diese Modifikationen der Ein- bzw. Ausgabeseite sind aber zu aufwändig und unhandlich um ernsthaft in Betracht zu kommen. Nachdem der Zugriff auf die Betriebssystemfunktionen ebenfalls ausgeschlossen wurde, kann die Transformation daher nur auf der Ebene der Applikation sinnvoll beeinflusst werden.

Abgesehen von Forschungsarbeiten wie (Westergaard, 2002), ist der Bildschirmcursor eines Computers einzigartig und kann nicht durch weitere Cursor ergänzt

werden. Für unsere Untersuchung galten jedoch spezielle Regeln, da die Interaktion nicht durch den üblichen Widgets abgewickelt wurde. Die Probanden sahen lediglich zwei schwarze Kästchen auf weißem Hintergrund, anstatt die gewohnten Komponenten wie Knöpfe, Menüs oder Fenster. Sie mussten keine Klicks ausführen oder Drag&Drop Aufgaben lösen, so dass die einzig relevante Interaktion in der Positionsänderung des Cursors bestand. Wird eine Applikation im Vollbildmodus ausgeführt und dabei der Cursor des Betriebssystems ausgeblendet, kann ohne großen Aufwand ein virtueller Cursor vom Programm simuliert werden. Hierzu wird ständig die Position des (echten) Cursors abgefragt und der virtuelle Cursor an diesen angepasst. Eine höhere oder niedrigere Transformation lässt sich auf diese Weise leicht implementieren: wurde der Cursor um X Pixel in eine Richtung bewegt, wird der virtuelle Cursor um $X \cdot k$ Pixel bewegt. Die gewünschte Transformation von Bewegung des Eingabegerätes in Bewegungen des Cursors wird dabei durch den Faktor k vorgegeben. In Abbildung 3.3 ist zu sehen, wie die Bewegungen des echten Cursors (Pfeil) von der Applikation durch die entsprechenden Faktoren beeinflusst werden. Der Benutzer sieht dabei nur den virtuellen Cursor (Fadenkreuz).

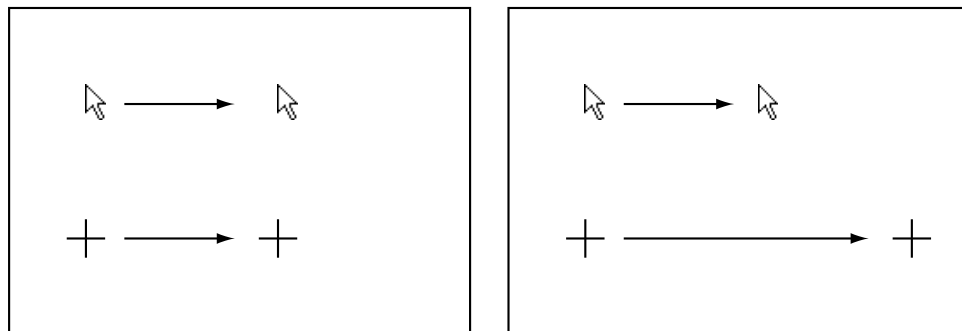


Abbildung 3.3: Cursorbewegungen mit Verstärkungsfaktor eins bzw. zwei

Psychtoolbox liefert für diese Art der Transformationswechsel auf Applikationsebene das passende Werkzeug. Der `screen` Befehl kann im Vollbildmodus betrieben werden und die `hidecursor` Anweisung kann der Pfeil des Betriebssystems ausblenden. Im Programm kann nun durch Aufruf des `getmouse` Befehls überprüft werden, auf welcher Koordinate sich der Pfeil befindet und ob er (und damit das Eingabegerät) seit dem letzten Aufruf von `getmouse` bewegt wurde. Ist das der Fall, wird aus der Differenz der alten und neuen Position die Bewegung in Pixel berechnet und gemäß der gewünschten Transformation mit dem entsprechenden Faktor

multipliziert.

An dieser Stelle ist es wichtig, den Unterschied zwischen Transformation und den Faktoren hervorzuheben. Die Transformation beschreibt die Quotienten aus der zurückgelegten Strecke auf dem Bildschirm und der Strecke, die mit dem Eingabegerät zurückgelegt wurde. Das Betriebssystem des Rechners hat eine bestimmte Transformation zwischen Eingabegerät und Bildschirmcursor (also des Pfeils). Wird die Experimentalsoftware gestartet, so werden die Bewegungen des Pfeils mit verschiedenen Faktoren multipliziert. Die Bewegungen des Fadenkreuzes führen dann zu einer anderen Transformation zwischen Eingabegerät und Fadenkreuz. Beträgt die Transformation zwischen Pfeil und Eingabegerät 2:1, und beträgt der Verstärkungsfaktor der Software vier, dann ergibt sich eine Gesamttransformation von 8:1. Eine Bewegung des Eingabegerätes um 5 cm resultiert in einer Bewegung des Fadenkreuzes von 40 cm. Der ausgeblendete Pfeil bewegt sich hierbei um 10 cm.

Diese Implementierung erfüllt die gewünschte Eigenschaft der Transformationswechsel sehr präzise, hat aber einige Nebeneffekte, die beachtet werden müssen. Wird ein Multiplikationsfaktor verwendet, der kleiner als eins ist, kann das Fadenkreuz nicht mehr alle Punkte auf dem Bildschirm erreichen. Bei einem Faktor von 0,5 bewegt sich das Fadenkreuz nur noch um die Hälfte der Strecke, den der Pfeil zurückgelegt hat. Startet die Bewegung z.B. vom linken Rand des Bildschirms, führt eine horizontale Bewegung des Pfeils über den gesamten Bildschirm dazu, dass das Fadenkreuz sich vom linken Rand nur bis zur Mitte des Bildschirms bewegt. Die rechte Seite des Bildschirms ist mit dem Fadenkreuz nicht mehr zu erreichen, da der Pfeil bereits den rechten Rand erreicht hat. Um Faktoren, die kleiner als eins sind, zu vermeiden, gibt es Abhilfe. Die Cursorgeschwindigkeit des Betriebssystems muss hierbei auf einen sehr niedrigen Wert gestellt werden, so dass eine Bewegung des Eingabegerätes in sehr kleinen Bewegungen des Pfeils resultieren. Das System hat daher vor dem Starten der Applikation eine sehr niedrige Transformation und wird erst durch die Multiplikation mit den entsprechenden Faktoren auf die gewünschte Transformation gebracht. Da die verwendeten Faktoren nun größer als eins sind, tritt der beschriebene Effekt nicht mehr auf. Die Einstellungen am Betriebssystem können einmal manuell vom Versuchsleiter vorgenommen werden und bedürfen daher keiner weiteren Behandlung.

Ein weiterer Umstand macht es nötig, eine niedrige Transformation des Betriebssystems zu verwenden: es können nur ganzzahlige Faktoren verwendet werden, so dass Faktoren, die kleiner als eins sind, ausgeschlossen sind. Sich auf ganzzahlige

Faktoren zu beschränken ist nötig, da ein Faktor von z.B. 1,5 erfordern würde, das Fadenkreuz um 4,5 Pixel zu verschieben, wenn der Pfeil um drei Pixel bewegt wurde. Nur durch Rundung der Werte könnte dieses Problem gelöst werden, doch das würde zu inkonsistenten Bewegungsmustern führen und ist daher zu vermeiden. Für diese Arbeit war die Einschränkung akzeptabel, da die Transformationen alle ganzzahlig waren.

Ein weiterer Effekt dieser Art der Transformationswechsel ist komplexer und wird im nächsten Kapitel genauer beschrieben.

3.6 Balancierung der Ausdrucksstärken

Für die Experimentalsoftware wurde ein Transformationswechsel programmiert, der die Bewegungen des Pfeils mit einem Faktor multipliziert. Auf diese Weise wird die Strecke verlängert, die vom Fadenkreuz auf dem Bildschirm zurückgelegt wird. Wird ein Faktor von beispielsweise vier verwendet, führt die Bewegung des Pfeils um einen Pixel zu einer Bewegung des Fadenkreuzes von vier Pixel. Das Fadenkreuz „überspringt“ hierbei drei Pixel. Je höher der angelegte Faktor ist, umso mehr Pixel werden bei einer Bewegung übersprungen, was problematisch sein kann. Es können nicht mehr alle Pixel auf dem Bildschirm mit dem Fadenkreuz angesteuert werden, so dass die Menge der möglichen Positionen abnimmt. In der Definition des Design Spaces wird das als Verminderung der Ausdrucksstärke des Eingabegerätes bezeichnet (Card, Mackinlay, und Robertson, 1990). In Abbildung 3.4 ist sichtbar, welchen Einfluss dieser Effekt auf das Zieltarget hat. Die Kreuzungen der Linien stellen die möglichen Positionen des Fadenkreuzes dar. Die Kreuzungen der dünnen Linien entsprechen dem Faktor von eins und die Kreuzungen der dickeren Linien dem Faktor vier. Mit dem größeren Faktor können viele Positionen nicht erreicht werden.

Für einen gültigen Treffer im Sinne von Fitts' Tapping Aufgabe muss das Fadenkreuz sich innerhalb des Zielkästchens (rote Fläche in der Abbildung) befinden. Können wegen der niedrigeren Ausdrucksstärke nicht alle Pixel angesteuert werden, ergibt sich eine effektive Targetgröße, (grüne Fläche) die kleiner ist als die visuell wahrgenommene Zielgröße. Der ID der Aufgabe verändert sich trotz konstanter Zielgröße und -entfernung, je nachdem welcher Faktor für die geforderte Transformation gewählt wurde.

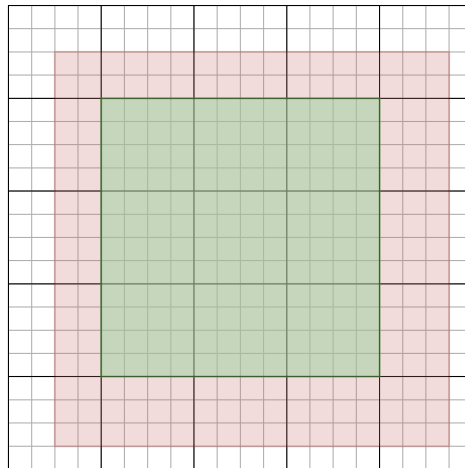


Abbildung 3.4: Effektive Zielgröße bei reduzierter Ausdrucksstärke

Diese Störgröße konnte eliminiert werden, indem die Ausdrucksstärke bei allen Transformationen auf ein konstantes Level angepasst wurden. Hierbei wurde in den niedrigen Transformationen (mit kleinen Faktoren) die Ausdrucksstärke künstlich auf die der höchsten Transformation heruntersetzt. In der Pilotstudie betrug der höchste Faktor vier und der niedrigste eins. Lag der Faktor vier an, führte jede Bewegung des Pfeils von einem Pixel zu einer Bewegung von vier Pixel des Fadenkreuzes. In den kleinsten Faktoren wurde dieser Effekt ebenfalls erreicht, in dem nicht bei jeder Bewegung des Pfeils die Position des Fadenkreuzes neu berechnet wurde. Vielmehr wartete das Programm so lange, bis der Pfeil vier Pixel in eine Richtung zurückgelegt hatte, bevor das Fadenkreuz um vier Pixel bewegt wurde. Bei den übrigen Faktoren wurde analog verfahren. Auf diese Weise war bei allen Transformationen die kleinste mögliche Bewegung des Fadenkreuzes auf vier Pixel begrenzt.

Damit die Ausdrucksstärken gleichgesetzt werden konnten, gab es eine Voraussetzung an die verwendeten Faktoren. Der größte Faktor musste durch alle kleineren Faktoren teilbar sein. Das Problem lässt sich an einem Beispiel leicht verdeutlichen. Ist der höchste Faktor Fünf und einer der kleineren Faktoren Zwei, dann kann mit diesem kleinen Faktor kein Sprung von fünf Pixeln erreicht werden, da entweder vier Pixel oder sechs angesprungen werden können. Diese Einschränkung war für die aktuelle Untersuchung akzeptabel, da die Faktoren diese Bedingung von vorne herein erfüllten.

3.7 Zentimeter und Pixel als Maßeinheiten

Der interdisziplinäre Charakter dieser Arbeit bedeutete, dass die Konzepte aus der Psychologie und der Informatik in Einklang gebracht werden mussten. Nicht immer war dieses Vorhaben unproblematisch. So werden in der Informatik die Eigenschaften von Peripheriegeräten häufig in „Dots per Inch“ bzw. Pixeln beschrieben, während in der Psychologie Transformationen bezüglich der Distanz in Zentimeter angegeben werden.

Für die psychologisch relevante Transformation zählt das Verhältnis von zurückgelegter Strecke mit dem Eingabegerät, zu der Strecke die sich der Cursor bewegt hat. Wenn 10 cm mit der Maus zurückgelegt werden und der Cursor auf dem Bildschirm dabei sich ebenfalls 10 cm bewegt, dann entspricht das der Transformation 1:1. Wird jetzt ein größerer Monitor an den Computer angeschlossen, vergrößert sich bei gleicher Bildschirmauflösung das dargestellte Bild. Wird die Maus jetzt 10 cm weit bewegt, resultiert die Cursorbewegung jetzt in einer größeren Distanz auf dem Bildschirm von vielleicht 14 cm. Die Transformation hätte sich in diesem Beispiel von 1:1 auf 1,4:1 erhöht (siehe auch Kapitel 2.3). Für das Betriebssystem, die Anwendungen oder den Programmierer gibt es keine Möglichkeit, diese Veränderung festzustellen, da die physikalischen Dimensionen des Monitors dem Computer nicht bekannt sind. Aus Sicht der Computertechnik ist in diesem Zusammenhang die Anzahl der darstellbaren Pixel bzw. die aktuelle Auflösung des Bildschirms relevant. Wird das Eingabegerät um 10 cm bewegt, so verarbeitet der Computer diese Strecke in zurückgelegte Pixel. Nochmals deutlich wird der Unterschied zwischen den psychologischen und computertechnischen Maßen, wenn die Auflösung der Grafikkarte verdoppelt wird. Die 10 cm Bewegung mit der Maus entsprechen zwar immer noch der gleichen Anzahl an Pixel, aber durch die doppelte Auflösung legt der Cursor nur noch die Hälfte der Strecke auf dem Bildschirm zurück.

In der Pilotstudie sind – wie üblich – alle genannten Parameter bekannt und werden während der Versuche auch nicht verändert. Insofern hätte verzichtet werden können, auf dieses Problem näher einzugehen. Hierzu hätten einfach alle unabhängigen Variablen von Zentimetern in Pixel umgerechnet und später die gemessenen abhängigen Variablen zurückgewandelt werden müssen. Da es für die Versuchsleiter aber ungewohnt ist, mit Pixel als Längenmaß umzugehen, wurde aus Gründen der Benutzerfreundlichkeit die Software umgeschrieben. Alle unabhängigen Variablen können unverändert eingegeben werden, genauso wie fast alle abhängigen Varia-

blen im metrischen System ausgegeben werden. Ausnahmen bilden hierbei abhängige Variablen wie z.B. die maximale Geschwindigkeit und Beschleunigung, die in der psychologischen Literatur ebenfalls in Pixel/s bzw. Pixel/s² angegeben werden.

Seitens der Software wurde das Problem gelöst, indem der Versuchsleiter zu Beginn der Versuchsserie misst, wie viele Pixel einem Zentimeter auf dem Bildschirm entsprechen. Dieser Wert wird in die Konfigurationsdatei eingetragen und kann von der Software verwendet werden, um alle metrischen Werte in Pixel umzurechnen. Je nach verwendetem Bildschirm kann es zu Schwierigkeiten kommen, wenn ein Zentimeter einer nicht ganzzahligen Anzahl von Pixel entspricht. In diesem Fall muss gerundet werden, so dass eine gewisse Ungenauigkeit entsteht. Später in Kapitel 5.2.2 wird auf die Probleme mit der ungenauen Bildschirmdarstellung noch näher eingegangen.

3.8 Detektierung der Treffer - 1. Variante

Die meisten Untersuchungen über Werkzeugtransformationen und ihre Effekte arbeiten mit Aufgaben, in denen die Probanden Objekte auf dem Bildschirm selektieren oder verschieben müssen Arnaut und Greenstein (1986). Diese Aufgaben erfordern von den Versuchspersonen, dass sie zu gegebener Zeit dem System signalisieren, wann eine Selektion ausgeführt werden soll, bzw. wann die Verschiebung beginnt und endet. In den meisten Fällen kommt hierzu eine Taste zum Einsatz, aber es gibt Ausnahmen wie beispielsweise das Haptic Pen (Lee, Dietz, Leigh, Yera-zunis, und Hudson, 2004). Auf alle Fälle sind für eine Fitts' Tapping Aufgabe diese Bestätigungsaktionen aus mehreren Gründen hinderlich:

- Die Zielbewegungen sollen entsprechend des Open Loop Prinzips, durch Festlegung der motorischen Kommandos und deren Parameter ausgeführt werden. Wenn eine Bestätigung der Endposition benötigt wird, können die Probanden Korrekturbewegungen durchführen die nach dem Closed Loop Prinzip arbeiten. Schnelle und flüssige Zielbewegungen sind auf diese Weise nicht realisierbar.
- Eine zusätzliche Bestätigung der Endposition des Cursors erhöht die gesamte Ausführungszeit, so dass die Zeiten nicht mit den Vorhersagen aus Fitts' Gesetz verglichen werden können. Hinzu kommt, dass eine flüssige Bewegung von

Kästchen zu Kästchen verhindert wird, da zwischendurch angehalten werden muss.

- Die Bestätigung ist eine zusätzliche Fehlerquelle, falls die Probanden vergessen, die Bestätigung auszuführen oder sie nicht komplett ausführen.
- Hat die Versuchsperson das Zielkästchen getroffen, kann durch die nachfolgende Bestätigungsaktion der Cursor aus dem Zielkästchen herausrutschen und den eigentlich gültigen Trial zu einem fehlerhaften machen.

Aus diesen Gründen musste die Software so konzipiert sein, dass sie in einer Serie von Zielbewegungen eigenständig erkennt, wann eine neue Bewegung beginnt. Einer der nahe liegenden Ideen war, die Geschwindigkeit des Cursors fortwährend zu berechnen. Hält der Proband mit dem Eingabegerät an, erreicht die Cursorgeschwindigkeit den Wert Null. In diesem Moment kann ausgewertet werden, ob ein Treffer vorliegt, oder nicht.

Eine erste vereinfachte Implementation der Geschwindigkeitsmessung las bei jedem Durchlaufen der Programmschleife die Position des Cursors aus. Wenn sich diese im Vergleich zum vorangegangenen Durchlauf nicht verändert hat, wurde ein Anhalten signalisiert. Während diese Methode für Geschwindigkeitsmessungen in der realen Welt geeignet ist, offenbaren sich in der diskreten Welt der Computer ihre Schwächen. Bei einer sehr langsamen Bewegung des Cursors, bleibt dieser für eine gewisse Zeit auf einer Bildschirmposition stehen, bevor er sich zum nächsten Pixel weiter bewegt. Werden zwei Messungen der Cursorposition in diesem Zeitfenster ausgeführt, ist die resultierende Geschwindigkeit gleich Null. In der echten Welt hätte ein langsam bewegendes Objekt trotzdem eine winzige Strecke zurückgelegt und daher eine Geschwindigkeit ungleich Null geliefert. Auf einem Computerbildschirm jedoch beträgt die kleinstmögliche Positionsänderung genau ein Pixel.

Um nicht komplizierte Analysen der gemittelten Geschwindigkeiten durchführen zu müssen, wurde stattdessen die bestehende Methode abgeändert. Die Software maß weiterhin die Bildschirmpositionen, aber mit einer erweiterten Bedingung: erst wenn sich über die Zeit von 100 ms hinweg die Cursorposition nicht verändert hatte, wurde das als Anhalten gewertet. Mit diesem zusätzlichen Zeitfenster sollten auch langsame Bewegungen nicht fälschlicher Weise als Ende der Bewegung gezählt werden. Das Zeitfenster durfte dabei nicht zu hoch angesetzt werden, da ansonsten Korrekturbewegungen – die das Ergebnis verfälschen – möglicher Weise unentdeckt

geblieben wären. Auf der Gegenseite hätte ein zu kleines Wartefenster erneut zu falschen Positiven geführt. Diese oberen und unteren Grenzen auszutarieren, erwies sich als unerwartet schwierig. Bei sehr kleinen Zielgrößen müssen die Probanden ihre Bewegungen zwangsweise verlangsamen, um die Zielkästchen überhaupt treffen zu können. Ein Zeitfenster von 100 ms erwies sich als zu niedrig und musste wesentlich erhöht werden. Hierdurch wurden jedoch, wie zu erwarten war, viele Korrekturbewegungen nicht mehr registriert.

Als weiteres Hindernis stellte sich heraus, dass die Versuchspersonen gelegentlich zum Ende einer Zielbewegung im Zielkästchen mit dem Cursor gezittert haben. Durch das große Zeitfenster wurden diese minimalen Bewegungen als Teil der ursprünglichen Zielbewegung gewertet. Die abhängige Variable der Ausführungszeit wurde hierdurch verfälscht, so dass versucht wurde, mit einem weiteren Zeitfenster diesem Effekt entgegenzusteuern. Wenn der Cursor sich mindestens 500 ms im Zielkästchen aufhielt, wurde das erste Zeitfenster auf 0 ms herabgesetzt. Sehr langsame Cursorbewegungen wurden daher erneut als Anhalten interpretiert, was aber hinnehmbar schien, da diese Bedingung nur am Ende einer Bewegung auftreten konnte.

Nach einer Revision der ersten Version der Software wurde entschieden, dass die Praxistauglichkeit der Messmethode nicht gegeben ist. Aus diesem Grund war es nötig, einen gänzlich anderen Weg zu gehen, um das Ende einer Zielbewegung zu messen.

3.9 Detektierung der Treffer - 2. Variante

Aus den Erfahrungen mit der zeitbasierten Messmethode wurde eine neue Methode entwickelt, die nicht auf die Messung der Cursorgeschwindigkeit angewiesen war. In den hintereinander durchgeführten Zielbewegungen ändert sich abwechselnd die Richtung der Bewegung. Der Umkehrpunkt befindet sich dabei jeweils an der Stelle, an der die eine Zielbewegung geendet hat, und die nächste beginnt. Die Implementation einer solchen Methode zur Erfassung der Umkehrpunkte ist wesentlich einfacher, so dass die Umstellung gleichbedeutend war mit wesentlich weniger Programmcode und einfacheren Strukturen.

Während die Probanden die Zielbewegungen ausführen, analysiert die Software die Position des Cursors. Interessant ist dabei nur die X-Koordinate der Bewegung,

da die Bewegungsrichtungen von links nach rechts ausgeführt werden. Solange die neue X-Koordinate größer oder gleich der alten Position ist, wurde der Cursor von der Versuchsperson nach rechts bewegt. Erst wenn die neue X-Koordinate einen kleineren Wert im Vergleich zur vorigen Position angenommen hat, ist das gleichbedeutend mit der Umkehrung der Bewegungsrichtung.

Das Ende einer Zielbewegung kann durch diese Methode natürlich erst erkannt werden, wenn die erste Bewegung der neuen Zielbewegung begonnen hat. Im Normalfall bereitet das keine Probleme, aber wie im folgenden Kapitel zu sehen sein wird, musste für die Protokollierung eine spezielle Lösung ausgearbeitet werden. Kapitel 5.2.7 befasst sich zusätzlich damit, welche unvorhergesehenen Seiteneffekte aufgrund dieser Meßmethode aufgetreten sind und wie sie behoben werden können.

3.10 Protokollierung der Daten

Für die spätere Analyse mussten die Aktionen der Probanden während des Experiments genau protokolliert bzw. mitgeloggt werden. Matlab bietet hierfür den aus anderen Programmiersprachen bekannten Befehl `fprintf`, um schnell und einfach Daten in eine Datei zu schreiben. Jedes Mal, wenn die Software eine Positionsänderung registriert, wird die Information über diesen neuen Zustand in die Logdatei geschrieben. Für gewöhnlich werden die Logdaten nach der Durchführung der Versuche automatisiert ausgewertet. Trotzdem ist es empfehlenswert, die Daten teilweise redundant zu speichern, um die Lesbarkeit der Logdaten zu verbessern. Unter Umständen muss sowohl die Validität der Software, als auch spezielle Vorkommnisse in den Daten manuell ausgewertet werden. Hierbei ist es sehr hilfreich, wenn die relevanten Daten nicht in irgendeiner Form kodiert vorliegen.

Ein Auszug aus den Daten, die bei der aktuellen Untersuchung gewonnen wurden:

```
6 0.39848 316 392 1 0 0 8 8 17 2 6.0 4 0 left
6 0.41830 312 392 1 0 0 4 8 17 2 6.0 4 0 left
6 0.46628 312 388 1 0 0 4 4 17 2 6.0 0 4 left
6 0.49840 308 388 1 0 0 0 4 17 2 6.0 4 0 left
6 0.52648 308 384 1 0 0 0 0 17 2 6.0 0 4 left
6 0.52648 308 384 1 0 1 0 0 17 2 6.0 0 4 left Treffer!
```

Die Spalten enthalten hierbei folgende Informationen:

- 1. Spalte: Nummer für die aktuelle ZG & CT Bedingung im Block.
- 2. Spalte: Verstrichene Zeit in Sekunden seit dem Beginn der aktuellen Zielbewegung.
- 3.-4. Spalte: X und Y Koordinaten des Cursors auf dem Bildschirm.
- 5. Spalte: Gültiger bzw. ungültiger Trial, repräsentiert durch 1 und 0.
- 6. Spalte: Binäre Variable für einen fehlerhaften Treffer.
- 7. Spalte: Binäre Variable für einen Treffer.
- 8.-9. Spalte: Entfernung in X und Y Richtung von Zielkastenmittelpunkt in Pixeln.
- 10. Spalte: Größe des Zielkästchens in Pixel.
- 11. Spalte: Verstärkungsfaktor des Cursor.
- 12. Spalte: Tablettedistanz in Zentimetern.
- 13.-14. Spalte: Zurückgelegte Distanz seit der letzten Bewegung in X und Y Richtung in Pixel.
- 15. Spalte: Seite auf der sich das Zielkästchen befindet.
- 16. Spalte: Treffer oder nicht. Diese Ereignisse werden markiert um sie leichter wiederfinden zu können.

Wie bereits erwähnt, enthalten nicht alle Spalten neue Informationen. Die unabhängigen Variablen sind in den Spalten 10-12 vorhanden, während die abhängigen Variablen in den Spalten 2-4 und 6-7 enthalten sind. Diese Informationen würden prinzipiell ausreichen, um später die restlichen Daten rekonstruieren zu können. Seitens der Auswertungssoftware würde das natürlich einen höheren Programmieraufwand erfordern, da die Trials quasi nachsimuliert werden müssten. Mit Hilfe der zusätzlichen Informationen kann aber sowohl der Mensch als auch die Software leichter erkennen, wann beispielsweise ein Treffer erfolgt ist, und ob dieser Treffer in einem gültigen Durchgang aufgetreten ist.

Die Spalten 5-7 bedürfen noch kurz näherer Erklärung: für den Ablauf der Experimente wurde spezifiziert, dass nach einem fehlerhaften Trial der nächste Treffer

noch als ungültig gewertet werden soll. In Spalte 5 wird aus diesem Grund darüber Buch geführt, ob dem aktuellen Trial ein Treffer vorangegangen ist oder nicht. In Spalte 6 und 7 ist fast immer der Wert Null verzeichnet. Nur wenn eine Bewegung beendet wurde, wird in diesen Spalten ein Treffer bzw. fehlerhafter Trial mit einer „1“ markiert. In diesem Moment wird auch der Wert in Spalte 5 neu ausgewertet und für den nächsten Trial festgesetzt.

Bei genauer Betrachtung der obigen Logdaten fällt eine Unregelmäßigkeit auf: die letzten beiden Zeilen enthalten die gleichen Zeit- und Positionsdaten und unterscheiden sich nur durch Spalte 7 und 16. Der Grund hierfür liegt in der verwendeten Methode, mit der Treffer registriert werden. Bei jeder Bewegung des Cursors wird eine Zeile in die Logdatei geschrieben. Erst wenn sich die Bewegungsrichtung des Cursors umdreht, wird das von der Software als Ende der ursprünglichen Bewegung ausgewertet. Das bedeutet, dass ein Treffer erst dann ausgewertet werden kann, wenn eine neue Zielbewegung bereits angefangen hat. Damit die eventuelle Pause zwischen den zwei Bewegungshandlungen nicht fälschlicher Weise zu der alten Bewegungszeit hinzugerechnet wird, wurde eine Umgehungslösung erstellt. Die Software speichert alle Informationen einer Bewegung zwischen und im Falle einer Bewegungsumkehr werden die letzten Daten erneut in die Logdatei geschrieben. Nur die Spalten 5-7 werden hierbei mit den jeweiligen Werten belegt und Spalte 16 mit dem Vermerk „Treffer“ bzw. „Angehalten außerhalb“ versehen. Das Programm für die Reduktion der Daten muss auf diese Eigenart der Logdaten angepasst werden, da ansonsten die Geschwindigkeits- und Beschleunigungsberechnung falsche Werte liefern würden. Diese Übergangslösung könnte durch eine gepufferte Protokollierung ersetzt werden. Hierbei würden bei jeder Cursorbewegung die Informationen anstatt direkt in die Logdatei, erstmals in einen Zwischenspeicher geschrieben. Erst bei der nächsten Positionsänderung fest steht, ob die Bewegungsrichtung sich umgekehrt hat würden die entsprechenden Daten in die Datei geschrieben. Zu Verbesserungen diesbezüglich wird in Kapitel 5.2 noch weiter eingegangen.

3.11 Verwaltung der Blöcke

In psychologischen Untersuchungen ist es üblich, die Abfolge der Bedingungen zu randomisieren, um Lerneffekte ausschließen zu können. Für die vorliegende Arbeit waren die Reihenfolge der Tablett-Distanzen und innerhalb dieser Bedingung, die Abfolge der Stimuli zu beachten. Aufgrund der geringen Anzahl der Tablett-Distan-

zen wurde beschlossen, diese nicht mit Hilfe der Software zu randomisieren. Statt dessen musste der Versuchsleiter vor jedem Versuch die Reihenfolge der drei Distanzen gemäß seiner vorher erstellten Liste manuell eingeben. Die Randomisierung der Stimuli innerhalb der Blöcke wurde in der Software realisiert.

```
ratios = [1 2 4];

sizes = [0.5 1 2 4];

block_combinations =
[ratios(1) sizes(1);
 ratios(1) sizes(2);
 ratios(1) sizes(3);
 ratios(1) sizes(4);
 ratios(2) sizes(1);
 ratios(2) sizes(2);
 ratios(2) sizes(3);
 ratios(2) sizes(4);
 ratios(3) sizes(1);
 ratios(3) sizes(2);
 ratios(3) sizes(3);
 ratios(3) sizes(4)];
```

Die Variable `ratios` definiert die drei Transformationsfaktoren. `Sizes` definiert die Größen der Zielkästchen in Zentimetern. In der `block_combinations` Variable sind alle Kombinationen von Faktoren und Zielgrößen festgelegt. Diese Kombinationen könnten zwar ohne weiteres auch automatisch generiert werden, aber wurden aus Gründen der besseren Kontrolle auf diese Weise definiert. So kann der Versuchsleiter bestimmte Kombinationen ausschließen oder einzelne hinzufügen. Nachdem das Programm startet, wird ein Element der Liste zufällig ausgewählt und aus der Liste gestrichen nachdem die Trials ausgeführt wurden.

Bei bestimmten Konstellationen kann es in den Trials vorkommen, dass sich Ursprungs- und Zielkästchen überlappen. In der 1:1 Transformation muss z.B. die Größe der Zielkästchen kleiner oder gleich der aktuellen Tablettedistanz sein, um eine Überlappung zu vermeiden. Eine 4:1 Transformation mit 4 cm Zielgröße und eine Tablettedistanz von 2 cm führt zu solch einer Überlappung. Nach Problemen

während der ersten Versuchsdurchläufe wurde die Software durch eine Überprüfungsroutine erweitert. Diese Routine vergleicht die Distanzen und überspringt diese Kombination, falls sich die Kästchen überlappten.

3.12 Distanzschätzung

Nachdem die Probanden einen Block abgeschlossen hatten, wurden sie gebeten, die Tablettedistanz des Blockes zu schätzen. Diese Tablettedistanz war während allen Bedingungen des Blocks konstant gewesen. Den Probanden wurde ein Beispiel aus dem bearbeiteten Block nochmals gezeigt. Die Probanden konnten den darunter befindlichen Abstandsmarker mit dem Eingabegerät verstellen. Für jede der drei Cursortransformationen wurde eine Distanzschätzung benötigt. In Abbildung 3.5 ist zu sehen, wie dem Probanden die zwei Kästchen zur Erinnerung dargeboten wurde. Die Funktionsweise der Abstandsmarker lief dabei wie folgt ab: wurde das Eingabegerät bewegt, so bewegten sich beide Enden des Abstandsmarkers um den gleichen Betrag in entgegengesetzter Richtung. Diese Technik ist an die Experimente von Sangals (1997) angelehnt. Sie sollte verhindern, dass die Probanden die Enden des Abstandsmarkers dazu verwendeten die Zielbewegung nachzusimulieren. Die Abstandswerte hätten auf diese Weise nicht der Tablettedistanz entsprochen, an die sich die Versuchspersonen erinnern haben sondern lediglich einem Wert, der sich aus der angelegten Transformation ergab. Die Transformation zwischen Eingabegerät- und Abstandsmarkerbewegung betrug im Übrigen immer 2:1.

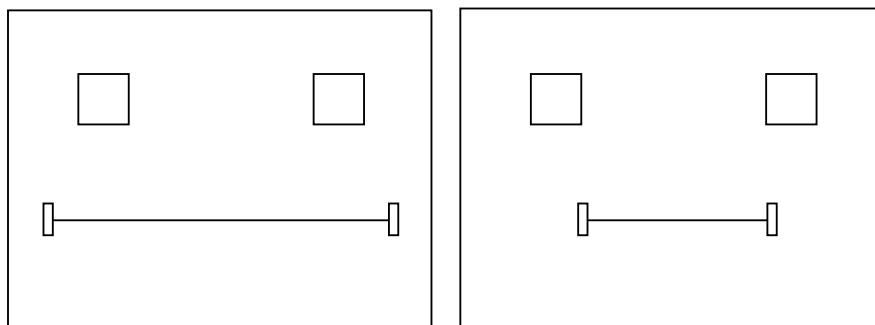


Abbildung 3.5: Abstandsmarker bei der Distanzschätzung.

Nachdem der Proband die Distanz geschätzt hatte, bestätigte der Versuchsleiter mit einem Mausklick die eingestellte Entfernung. Die Software rechnete aus den Bildschirmpositionen der Marker-Enden aus, wie groß der Abstand zwischen ihnen

in Pixel betragen hat. Der Software war bekannt wie viele Pixel einem Zentimeter auf dem Bildschirm entsprachen (siehe Kap. 3.7), so dass der Abstand als Zentimetermaß in eine separate Logdatei geschrieben werden konnte.

3.13 Abschlusstest

Im Laufe des Experiments bekamen die Probanden Aufgaben, in denen die Tablettedistanz gleich blieb und unter anderem die Transformationen variierten. Genau umgekehrt verhielt es sich im Abschlusstest. Hier blieb die Transformation die Gleiche und die Tablettedistanzen wurden verändert (siehe Kap. 3.1). Die Probanden mussten eine Cursortransformation 25 mal üben und anschließend auf andere Tablettedistanzen absolvieren. Für diesen Teil der Software wurden ähnliche Routinen verwendet, wie in den Blöcken zuvor. Da die Reihenfolge der Tablettedistanzen nicht randomisiert wurde und die Zielgrößen konstant auf 1 cm gehalten waren, vereinfachte sich die Implementation. Darüber hinaus blieben die Rahmenbedingungen verglichen mit den HauptROUTINEN gleich, so dass aus technischer Sicht nicht näher auf den Abschlusstest eingegangen werden muss.

3.14 Datenreduktion

Werden mit der Software Experimente durchgeführt, liegen die gewonnenen Versuchsdaten in den mitgeloggten Dateien vor. In diesen Dateien sind sämtliche Cursorbewegungen aufgezeichnet, die die Probanden während der Versuche getätigt haben. Für die Datenanalyse sind diese Daten aber noch nicht zu gebrauchen, da die abhängigen Variablen erst herausgefiltert bzw. berechnet werden müssen. Für diese Aufgabe wurde ein eigenes Modul entwickelt, das ebenfalls in Matlab geschrieben wurde.

Die Datenreduktion führt das Programm durch, indem es die Logdaten zeilenweise einliest und dabei die einzelnen Zielbewegungen separiert. Wie in Kapitel 3.10 beschrieben wurde, ist das Ende einer Zielbewegung eindeutig durch eine „1“ in der „Treffer“- bzw. „fehlerhafter Treffer“-Spalte der entsprechenden Datenzeile der Logdatei markiert. Die Ausführungszeit kann direkt aus der ersten Spalte dieser Datenzeile ausgelesen werden, genauso wie die Abstände zum Mittelpunkt des Zielkästchens. Im Gegensatz dazu müssen die Daten für die Geschwindigkeits- und

Beschleunigungswerte getrennt berechnet werden. Hierzu wird für jede Datenzeile die zurückgelegte Strecke (13. und 14. Spalte) aus den zwei Koordinaten mit Hilfe vom Satz des Pythagoras berechnet. Die Zeit, die für diese Strecke benötigt wurde, lässt sich aus den Zeitdifferenzen der aktuellen und der vorangegangenen Datenzeilen berechnen (2. Spalte). Auf diese Weise wird für jede Datenzeile die Cursorgeschwindigkeit ermittelt.

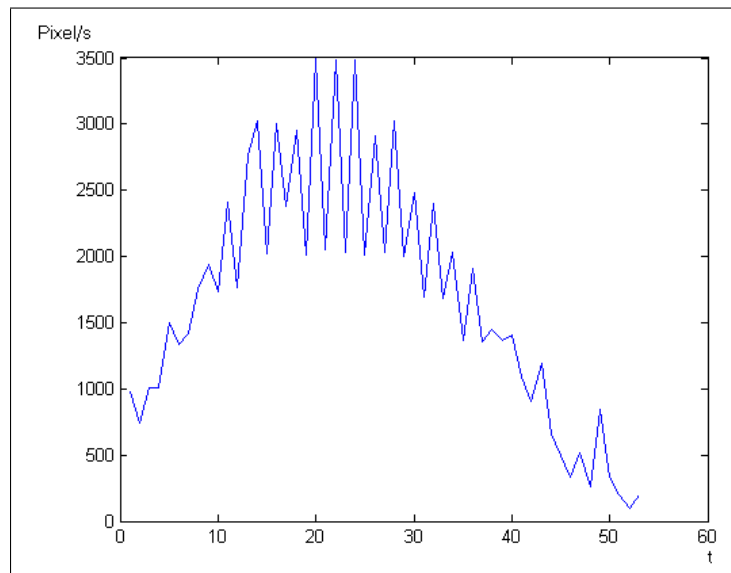


Abbildung 3.6: Cursorgeschwindigkeiten während einer Zielbewegung

Die Cursorgeschwindigkeiten für eine Zielbewegung sind in Abbildung 3.6 dargestellt. Es fällt auf, dass die Kurve sehr ausgeprägte Zacken hat, anstatt eines erwarteten gleichmäßigen Verlaufs. Dies kann in mehreren Ursachen begründet liegen. Schon in den Experimenten von Douglas und Mithal (1997) zeigten die Versuchsdaten ein ausgeprägtes „Jitter“. Die genaue Ursache für diese sprunghaften Messdaten vermuteten die Forscher in der Hardware des Eingabegerätes. Es ist also möglich, dass die vom Tablett an den Computer übermittelten Positionsdaten bei schnellen Bewegungen eine gewisse Ungenauigkeit aufweisen. Diese kleinen Abweichungen würden sich bei der Division mit den ebenfalls kleinen Zeitabständen zu großen Schwankungen im Ergebnis aufsummieren. Eine andere Erklärung von Douglas und Mithal war, dass der Mikrotremor der Finger durch das Eingabegerät verstärkt wiedergegeben wurde. Diese Erklärung für den Effekt ist aber im vorliegenden Fall aufgrund der ausgeprägten Art des Jitters eher auszuschließen.

Eine zweite Erklärung für die Schwankungen kann in der Berechnung der zurück-

gelegten Strecke liegen. Die Bewegungen des Cursors werden getrennt in X- und Y-Koordinaten aufgezeichnet, so dass die gesamte zurückgelegte Strecke mit dem Satz des Pythagoras berechnet werden muss. Bewegt sich beispielsweise der Cursor ausschließlich waagrecht um 2 Pixel pro 10 ms über den Bildschirm, so beträgt die zurückgelegte Gesamtstrecke auch nach Pythagoras $\sqrt{2^2 + 0^2} = 2$ Pixel. Die Geschwindigkeit des Cursors beträgt also 200 Pixel / Sekunde. Würde in dieser waagerechten Bewegung der Cursor zwar weiterhin 2 Pixel pro 10 ms zurücklegen, aber gelegentlich zusätzlich um einen Pixel nach oben oder unten abweichen, wäre die Gesamtstrecke: $\sqrt{2^2 + 1^2} \approx 2,24$. Trotz der gleichmäßigen Bewegung würde durch diese kleine Abweichung die Cursorgeschwindigkeit sich auf 224 Pixel / Sekunde erhöhen – ein Anstieg um über 10%. Für gewöhnlich treten solche Abweichungen in der vertikalen Richtung selten gleichzeitig hintereinander auf, sondern eher zufällig und verstreut. Wird der gesamte Geschwindigkeitsverlauf betrachtet, führen diese kleinen Abweichungen zu Sprüngen in der Cursorgeschwindigkeit. Dieser Umstand könnte daher für den zitterigen Kurvenverlauf verantwortlich sein.

Für sich alleine betrachtet, ist ein unruhiger Kurvenverlauf keine Angelegenheit, die eine gesonderte Behandlung benötigt. Sollen jedoch aus den Geschwindigkeitsdaten die Beschleunigungswerte errechnet werden, gibt es ein Problem. Diese Berechnung erfordert, dass die Geschwindigkeitskurve differenziert wird, also die erste Ableitung gebildet wird. Da diese Ableitung die Steilheit der Geschwindigkeitskurve beschreibt, führen selbst kleine „Zacken“ in der Kurve zu großen Schwankungen im Ergebnis. In der Ableitungskurve sind die Schwankungen quasi verstärkt wiedergegeben.

Die einzige Lösung für dieses Problem besteht darin, die Geschwindigkeitsdaten zu filtern bzw. zu glätten. Dabei erwies sich die interne `smooth`-Funktion von Matlab als recht effektiv. Die LOESS (locally optimal estimating and smoothing scatter plots) Analyse mit einem Fenster von 30% der Gesamtdaten lieferte eine gute Annäherung an die Geschwindigkeitskurve, bei der das Jitter nicht mehr auffindbar war. Versuche, einen Tiefpassfilter als Glättungsfunktion einzusetzen, lieferten keine zufrieden stellenden Ergebnisse. Die LOESS-Analyse wurde daher beibehalten. In Abbildung 3.7 ist das Ergebnis der Glättung der Daten aus Abbildung 3.6 zu sehen.

In der weiteren Berechnung der abhängigen Variablen wird die Ableitung der Geschwindigkeitskurve ebenfalls geglättet, um auch hier Ausreißer zu eliminieren. Mit den geglätteten Geschwindigkeits- und Beschleunigungsdaten kann das Programm

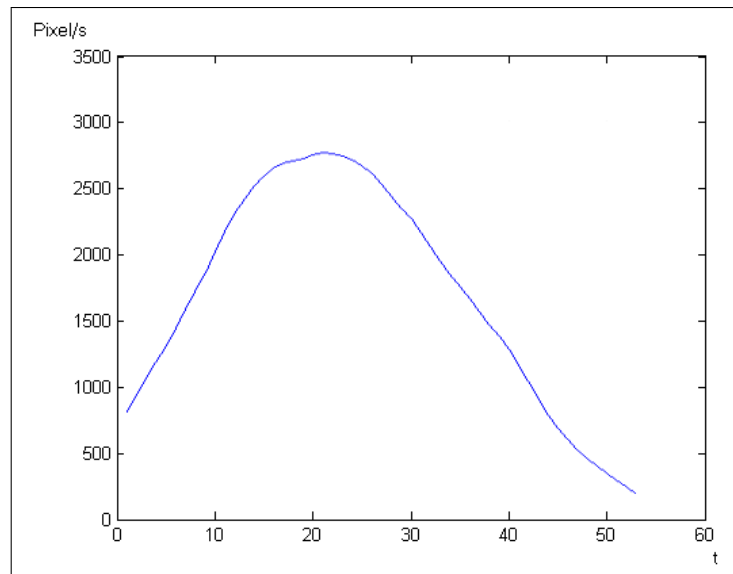


Abbildung 3.7: Cursorgeschwindigkeiten nach der Glättung

sehr einfach die Maximumwerte berechnen, die als abhängige Variablen benötigt werden.

Nachdem alle abhängigen Variablen einer Zielbewegung berechnet wurden, gibt die Reduktionssoftware diese Ergebnisse in eine Ausgabedatei aus:

```
003 2 2 4 17 1 0.5666 4 0 3991 0.0628 748 0.1990
003 2 2 4 17 r 0.5783 4 0 2591 0.0942 698 0.2705
003 2 2 4 17 1 0.5852 4 4 4557 0.0813 758 0.1898
003 3 2 4 34 r 1.3023 0 0 1734 0.6462 619 0.9345
003 3 2 4 34 1 0.7580 4 0 2747 0.0025 688 0.3301
003 3 2 4 34 r 0.6865 4 4 3319 0.1145 757 0.1828
003 3 2 4 34 1 0.6262 4 0 7039 0.0025 974 0.1503
```

Die Spalten beinhalten dabei folgende Informationen:

- 1. Spalte: Vpn-Nummer
- 2. Spalte: Block ID. Nummerierung der einzelnen Bedingungen innerhalb eines Blocks. Pro Block gab es zwölf unterschiedliche Bedingungen
- 3. Spalte: Transformation die in dieser Bedingung verwendet wurde

- 4. Spalte: Tablettedistanz des aktuellen Blocks in Zentimetern
- 5. Spalte: Targetgröße in Pixel
- 6. Spalte: Lage des Zielkästchens auf der linken oder rechten Seite
- 7. Spalte: Ausführungszeit der Zielbewegung in Sekunden
- 8.-9. Spalte: Entfernung des Endpunktes vom Zielkästchenmittelpunkt in Pixel
- 10. Spalte: Höchste Beschleunigung in Pixel / Sekunde²
- 11. Spalte: Zeitpunkt der höchsten Beschleunigung in Sekunden
- 12. Spalte: Höchste Geschwindigkeit in Pixel / Sekunde
- 13. Spalte: Zeitpunkt der höchsten Geschwindigkeit in Sekunden

Die Software gab in dieser Datei nur die Daten der erfolgreichen Trials aus. In der Pilotstudie wurden 25 erfolgreiche Zielbewegungen benötigt, bevor eine neue Bedingung präsentiert wurde. Pro Block gab es zwölf Bedingungen. Bei drei Blöcken pro Versuch ergaben sich insgesamt (exklusive Übungsdurchlauf und Abschlusstest) 900 Einträge pro Versuchsperson.

Als abhängige Variable interessierte aber auch der Fehlerquotient innerhalb einer Bedingung, also die Frage: Wie viele Zielbewegungen waren nötig, bis die 25 gültigen Zielbewegungen erreicht wurden. Hierbei ist es wichtig nochmals zu erwähnen, dass nach einer fehlerhaften Zielbewegung die nächste erfolgreiche Zielbewegung nicht als gültig gezählt wurde. Die Software rechnet für jede Bedingung den Fehlerquotienten aus und gibt diese in eine gesonderte Datei aus. Die Ausgabe sieht dabei wie folgt aus:

```
003 1 1 68 4 0.0357 1 28 25
003 2 2 17 4 0.1250 4 32 25
003 3 2 34 4 0.2619 11 42 25
003 4 4 17 4 0.6049 98 162 25
```

Der Inhalt der Spalten ähnelt sehr der obigen Ausgabe:

- 1. Spalte: Vpn-Nummer

- 2. Spalte: Block ID
- 3. Spalte: Transformation
- 4. Spalte: Targetgröße in Pixel
- 5. Spalte: Tabletdistanz in Zentimeter
- 6. Spalte: Fehlerquotient
- 7. Spalte: Anzahl der Fehlerhaften Zielbewegungen
- 8. Spalte: Anzahl der Zielbewegungen insgesamt
- 9. Spalte: Anzahl der erfolgreichen Zielbewegungen

Die Auswertung der Daten kann nicht nur für eine Versuchsperson gleichzeitig durchgeführt werden, sondern es können die Versuchsdaten mehrerer Probanden hintereinander reduziert werden. Für die einfachere Bedienung dieses Programms wurde hierfür eine grafische Oberfläche implementiert, die in Abbildung 3.8 dargestellt ist.

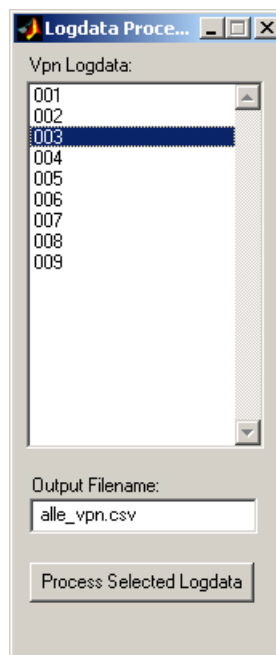


Abbildung 3.8: Grafische Oberfläche der Reduktionssoftware

Mit Hilfe dieser Oberfläche ist die Auswahl der zu bearbeitenden Daten wesentlich einfacher, als durch eine kommandozeilenbasierte Steuerung.

3.15 Instruktionen

Für eine erfolgreiche und präzise Untersuchung ist es sehr wichtig, die Instruktion der Versuchspersonen sorgfältig zu planen. Obwohl die Instruktionen mit Bedacht formuliert wurden, zeigten sich teilweise Probleme bei diesen, wie später in Kapitel 5.2.11 beschrieben wird. Die Instruktionen vor dem Experiment lauteten:

Herzlich Willkommen zum Experiment!

Sie werden gleich zwei Kästchen auf dem Bildschirm sehen. Ihre Aufgabe ist es das Fadenkreuz aus dem einen, in das andere Kästchen zu bewegen und anschließend die gleiche Aufgabe von diesem Kästchen ausgehend zu wiederholen. Versuchen Sie die Aufgabe so schnell, aber gleichzeitig auch so genau wie möglich auszuführen. Im Zweifelsfall ist die Genauigkeit wichtiger als die Geschwindigkeit. Bitte heben sie den Stift während den Bewegungen nicht vom Tablett ab, und achten Sie auf eine möglichst flüssige Bewegung. Versuchen Sie auch möglichst die Mitte der Kästchen zu treffen und nehmen Sie den kürzesten Weg zwischen den Kästchen. Also vermeiden Sie Schlangenbewegungen.

In den Instruktionen wurden verschiedene Anweisungen gegeben, um Fehlerquellen zu minimieren und die Vergleichbarkeit der Daten zu gewährleisten. Damit die Probanden nicht zu schnelle Zielbewegungen - mit einer hohen Fehlerquote - machten, wurde auf die Balance zwischen Geschwindigkeit und Genauigkeit hingewiesen. Die Anweisung zu flüssigen Bewegungen sollte verhindern, dass Korrekturbewegungen durchgeführt werden.

Die Instruktionen für die Schätzung der zurückgelegten Distanzen bestanden aus zwei Teilen. Die erste Instruktion galt für alle drei Transformationen. In den Schätzungen der jeweiligen Transformationen waren speziell angepasste Instruktionen nötig. In diesem Beispiel ist die Instruktion für die „lange“ Distanz, also der größten Transformation zu sehen:

Die Kästchen die Sie vorhin gesehen haben, kamen in drei unterschiedlichen Abständen vor. Bitte versuchen Sie auf den nächsten Bildschirmen jeweils für den kurzen, mittleren und

langen Abstand die Strecke zu schätzen, die sie mit dem Stift auf dem Tablett zurückgelegt haben.

Sie haben vorhin verschieden große Kästchen in unterschiedlichen Abständen gesehen. Sie sehen unten eines der von Ihnen vorhin absolvierten Aufgaben mit dem langen Abstand. Stellen Sie bitte die darunter befindlichen Abstandsmarker so ein, dass sie die Strecke zeigen die Sie für diesen Fall mit dem Stift auf dem Tablett zurückgelegt haben.

Das Wesentliche bei diesen Instruktionen war, dass die Probanden verstanden, welche Strecke mit den Abstandsmarkern eingestellt werden sollte. Der Begriff der Transformation durfte nicht erwähnt werden, um den Versuchspersonen keinen Hinweis über die unabhängigen Variablen zu geben. Da bei den großen Transformationen die Kästchen einen größeren *Abstand* voneinander einnahmen, wurde diese Bezeichnung in den Instruktionen verwendet. Auf die Erfahrungen mit den Instruktionen wird in Kapitel 5.2.11 näher eingegangen.

3.16 Zusammenfassung

In diesem Kapitel wurden die technischen Herausforderungen bei der Umsetzung der psychologischen Spezifikationen und deren Lösungen geschildert. Scheinbar funktionierende Methoden, wie die zeitbasierte Detektierung von Treffern musste, im Nachhinein als unpassend für die Zwecke der experimentellen Untersuchung betrachtet werden. Sie wurde daher durch die umkehrpunktbasierte Methode ersetzt. Unabhängige Variablen wie die Änderung der Transformation ließen sich nicht ohne Umwege realisieren, die ihrerseits verschiedene Effekte auf andere Bereiche der Software hatten. Im nächsten Kapitel wird auf die Durchführung der Pilotstudie eingegangen, das als Testlauf der Experimentalsoftware gesehen werden kann.

4 Pilotstudie

In den folgenden Kapiteln werden die Methodik, sowie die Ergebnisse der durchgeführten Versuchsreihe beschrieben. In der Diskussion wird hauptsächlich auf die Punkte eingegangen, die in direktem Zusammenhang mit den Ergebnissen stehen. Eine ausführliche Diskussion wird später in Kapitel 5 geführt. Dort wird auf die Versuchsausrüstung und deren Effekte auf das Experiment eingegangen.

4.1 Methode

Im letzten Kapitel wurde die Entwicklung der Experimentalsoftware beschrieben und auf welche Art die psychologischen Spezifikationen ihre technische Umsetzung fanden. Ebenso wurde auf die abhängigen und unabhängigen Variablen eingegangen (siehe Kap. 3.1), so dass sie in diesem Kapitel nur kurz genannt werden. Gleiches gilt für die Hypothesen, die in Kapitel 2.6 detailliert geschildert wurden.

Wie bereits in Kapitel 3.1 erläutert wurde, war die **Aufgabe der Probanden**, für jede der drei Tablettedistanzen genau 25 erfolgreiche Zielbewegungen auszuführen, und zwar für jede der 12 möglichen Kombinationen aus Cursortransformation und Zielgröße (siehe auch Tab. 4.1). Das Fadenkreuz auf dem Bildschirm musste mit Hilfe des Eingabegerätes aus dem Startkästchen in das Zielkästchen, auf der anderen Seite bewegt werden. Anschließend wurde der Cursor aus dem aktuellen Kästchen zurück in das ursprüngliche Kästchen geführt werden. Diese Abwechselnden Zielbewegungen mussten ohne Pause zwischen den Bewegungen kontinuierlich wiederholt werden. Eine Bedingung – also eine ausgewählte Kombination aus CT und ZG – wurde dabei so lange dargeboten, bis die Anzahl von 25 korrekten Zielbewegungen erreicht wurde. Wurde das Zielkästchen nicht getroffen, dann zählte der darauf folgende Treffer zwar als gültige Zielbewegung, wurde aber nicht zu den benötigten 25 erfolgreichen Treffern gezählt.

Die **Operationalisierung der Variablen** ist wie folgt geschehen:

1. **UV Tablettedistanz (TD)**: Strecke, die mit dem Eingabegerät zurückgelegt wurde. (2 cm; 4 cm; 6 cm)
2. **UV Cursortransformation (CT)**: Verhältnis von der Bewegung des Cursor auf dem Bildschirm zu der Bewegung des Eingabegerätes. (1,22:1, 2,44:1, 4,88:1)
3. **UV Zielgröße (ZG)**: Größe der Zielquadrate hinsichtlich der Seitenlänge. (0,5 cm; 1 cm; 2 cm; 4 cm)
4. **UV Fitts ID (ID)**: Jede feste Kombination aus Tablettedistanz und -zielgröße hat einen Fitts'schen Schwierigkeitsindex. Dieser Wert ergibt sich aus den UV Zielgröße und Tablettedistanz und ist daher keine eigenständige Variable, wird aber für Vergleiche gleich „schwerer“ Trials benötigt.

Die UV Cursortransformation ist hier anders angegeben als in Kapitel 3.1. Der Treiber des Eingabegerätes ließ sich nicht hinreichend genau einstellen, so dass es nicht möglich war die 1:1 Transformation herzustellen. Als beste Annäherung an diesen Wert der erreicht werden konnte, war die 1,22:1 Transformation. Näheres dazu in Kapitel 5.2.4.

In Fitts' Tapping Aufgaben ist es üblich, die verschiedenen Tablettedistanzen jeweils zu verdoppeln. Im aktuellen Experiment musste die größte Distanz auf 6 cm beschränkt werden, da sich mit der größten Transformation und 8 cm Tablettedistanz eine Displaydistanz von über 39 cm ergeben hätte. Die Breite des Bildschirms betrug jedoch nur 30 cm, so dass diese Bedingung nicht hätte dargestellt werden können.

In Tabelle 4.1 sind alle Kombinationen aus den UV aufgelistet. Als vierte UV ist für jede dieser Bedingungen der Fitts ID und die Tablettzielgröße angegeben. Diese Werte ergeben sich zwingend aus den drei UV, und sind lediglich für die spätere statistische Auswertung interessant. Die Bedingung, deren ID in der Tabelle mit einem Stern markiert ist, wurde im Experiment nicht dargeboten. Der $ID < 1$ zeigt an, dass sich in dieser Bedingung die Zielkästchen (wegen der geringen Distanz und der maximalen Zielgröße) überschneiden hätten. Die Reihenfolge der Tablettedistanzen wurde jedem Probanden geblockt und randomisiert dargeboten. Die Bedingungen innerhalb eines Blocks wurden ebenfalls randomisiert.

Tabelle 4.1: Alle Kombinationen der UV. Zusätzlich wurden aus diesen jeweils Tabletzielgrößen und IDs berechnet.

Tablettdistanz	Transformation	Displayzielgröße	Tabletzielgr.	ID
2 cm	1,22:1	0,5 cm	0,41 cm	2,56 bits
	1,22:1	1,0 cm	0,82 cm	1,78 bits
	1,22:1	2,0 cm	1,64 cm	1,15 bits
	1,22:1	4,0 cm	3,28 cm	0,69 bits*
2 cm	2,44:1	0,5 cm	0,20 cm	3,46 bits
	2,44:1	1,0 cm	0,41 cm	2,56 bits
	2,44:1	2,0 cm	0,82 cm	1,78 bits
	2,44:1	4,0 cm	1,64 cm	1,15 bits
2 cm	4,88:1	0,5 cm	0,10 cm	4,39 bits
	4,88:1	1,0 cm	0,20 cm	3,46 bits
	4,88:1	2,0 cm	0,41 cm	2,56 bits
	4,88:1	4,0 cm	0,82 cm	1,78 bits
4 cm	1,22:1	0,5 cm	0,41 cm	3,43 bits
	1,22:1	1,0 cm	0,82 cm	2,56 bits
	1,22:1	2,0 cm	1,64 cm	1,78 bits
	1,22:1	4,0 cm	3,28 cm	1,15 bits
4 cm	2,44:1	0,5 cm	0,20 cm	4,39 bits
	2,44:1	1,0 cm	0,41 cm	3,43 bits
	2,44:1	2,0 cm	0,82 cm	2,56 bits
	2,44:1	4,0 cm	1,64 cm	1,78 bits
4 cm	4,88:1	0,5 cm	0,10 cm	5,36 bits
	4,88:1	1,0 cm	0,20 cm	4,39 bits
	4,88:1	2,0 cm	0,41 cm	3,43 bits
	4,88:1	4,0 cm	0,82 cm	2,56 bits
6 cm	1,22:1	0,5 cm	0,41 cm	3,97 bits
	1,22:1	1,0 cm	0,82 cm	3,06 bits
	1,22:1	2,0 cm	1,64 cm	2,22 bits
	1,22:1	4,0 cm	3,28 cm	1,50 bits
6 cm	2,44:1	0,5 cm	0,20 cm	4,95 bits
	2,44:1	1,0 cm	0,41 cm	3,97 bits
	2,44:1	2,0 cm	0,82 cm	3,06 bits
	2,44:1	4,0 cm	1,64 cm	2,22 bits
6 cm	4,88:1	0,5 cm	0,10 cm	5,93 bits
	4,88:1	1,0 cm	0,20 cm	4,95 bits
	4,88:1	2,0 cm	0,41 cm	3,97 bits
	4,88:1	4,0 cm	0,82 cm	3,06 bits

- **AV Bewegungszeit (Sekunden) (BZ):** Zeit, die für eine Zielbewegung benötigt wurde, gemessen von der ersten Bewegung des Eingabegerätes bis zur letzten Bewegung im Zielkästchen.
- **AV Fehlerquotient (Prozent) (FQ):** Anzahl der Zielbewegungen, in denen das Zielkästchen nicht getroffen wurde, geteilt durch alle Zielbewegungen. Berechnet auf eine Serie von Zielbewegungen, die bis Erreichen der 25 gültigen Trials einer Bedingung ausgeführt wurden.
- **AV Geschätzte Tablettedistanz (Zentimeter) (GT):** Die zwischen den Blöcken durchgeführte Schätzung der Tablettedistanz nach Darbietung von vorher gesehenen Displaydistanzen. Die geschätzte Distanz entspricht dem Abstand der Marker, den die Probanden mit dem Eingabegerät auf die geschätzte Distanz eingestellt haben. Nach jedem Block werden den Versuchspersonen Displaydistanzen entsprechend der drei Transformationen präsentiert.

Die Bewegungszeit und der Fehlerquotient sind motorische Leistungsmaße, und die geschätzte Tablettedistanz ein subjektives Maß.

Zur **Kontrolle der Störvariablen** wurden die Versuche im gleichen Raum und am gleichen Computer durchgeführt. Die Position und Einstellungen des Bildschirms waren bei allen Probanden gleich. Die Parameter des Eingabegerätes wurden ebenfalls konstant gehalten. Innerhalb der Blöcke war die Reihenfolge der Bedingungen randomisiert, genauso wie die Reihenfolge der Blöcke selbst. Nur im Abschlusstest war stets die gleiche Abfolge von Bedingungen vorgegeben, während die Cursortransformation hier den Probanden zufällig zugewiesen wurde. Jede Transformation im Abschlusstest wurde von genau drei Versuchspersonen bearbeitet.

Die der Untersuchung zugrunde liegenden Hypothesen waren, wie bereits erwähnt, die folgenden:

- Hypothese 1: Die Cursortransformation beeinflusst bei konstanten Tablettedistanzen und -zielgrößen die Bewegungszeit.
- Hypothese 2: Eine Änderung der Cursortransformation und dadurch auch der Displaydistanz führt zu Unterschieden in der Schätzung der zurückgelegten Tablettedistanz durch die Versuchspersonen.
- Hypothese 3: Nach einem Wechsel der Cursortransformation kommt es mit der neuen Transformation nur bei den ersten fünf Trials zu einem Nacheffekt.

Die **Stichprobe** bestand aus neun Versuchspersonen im Alter von 25 bis 32 Jahren, von denen sechs männlich und drei weiblich waren. Vier von ihnen waren Informatikstudenten, ein Maschinenbaustudent, ein Diplom-Physiker, ein Mathematisch-Technischer-Assistent, eine Sozialpädagogin und eine Hotel-Fachfrau. Zwei Probanden waren Linkshänder, was aber keine Anpassung seitens der Versuchsapparatur erforderte. Alle Probanden waren Freiwillige und haben nach dem Versuch als Belohnung Süßigkeiten erhalten.

Versuchsbeschreibung: Der Versuch wurde im Zeitraum vom 12.01.2007 bis zum 19.01.2007 am Institut für Psychologie der RWTH durchgeführt. Ungefähr 25 Minuten dauerte der Versuch pro Proband. Als Experimentalsoftware kam das im Rahmen dieser Diplomarbeit entwickelte Programm zum Einsatz.

Versuchsapparatur: Die Hardware bestand aus einem Windows XP PC, an den ein 17" EIZO F563-T Röhrenmonitor angeschlossen war. Für eine Lage in Augenhöhe wurde der Monitor auf einen 8 cm hohen Sockel platziert (Abbildung 4.1) und befand sich ca. 55 cm von den Augen der Probanden entfernt. Als Eingabegerät wurde ein Wacom intuos2 A3 Grafiktablett mit dem dazugehörigen Stift verwendet. Das Grafiktablett wurde direkt vor dem Monitor auf dem Tisch platziert. Damit die Probanden die Bewegung ihrer Hände nicht sehen konnten, wurde ein Sichtschutz verwendet, das das gesamte Tablett verdeckte. Der Sichtschutz war so konstruiert, dass die Bewegungen auf dem Tablett nicht beeinträchtigt wurden. In Abbildung 4.2 ist die Apparatur samt Sichtschutz abgebildet.

Nachdem den Probanden die Instruktionen vorgelesen wurden, haben sie drei Bedingungen (mit 7 cm Tablettedistanz) als Übungsdurchgang bearbeitet. Hiernach folgten die drei Testblöcke, mit jeweils zwölf Kombinationen aus Cursortransformation und Zielgröße. Die Versuchspersonen haben insgesamt 35 Bedingungen bearbeitet, die jeweils 25 gültige Zielbewegungen erforderten. Zwischen den Blöcken bearbeiteten die Probanden die Tablettedistanzschätzungen und konnten danach eine kurze Pause machen. Im Anschluss an die drei Testblöcke wurde der Abschlusstest durchgeführt, der aus einer Übungsbedingung (mit 25 erfolgreichen Zielbewegungen) und drei Testbedingungen (mit 5 erfolgreichen Zielbewegungen) bestand.

Während der Experimente wurden sämtliche Cursorbewegungen mitgeloggt, die vor der **Auswertung der Daten** zuerst Reduziert werden mussten. Es wurde hierfür die selbst entwickelte Reduktionssoftware verwendet, die als Ausgabe CSV Dateien lieferte.



Abbildung 4.1: Versuchsaufbau: Monitor mit Sockel und Wacom Grafiktablett



Abbildung 4.2: Versuchsaufbau mit angebrachtem Sichtschutz

Tabelle 4.2: Tablettzielgrößen in Abhängigkeit von Transformation und Displayzielgrößen

Transformation	Tablettedistanz			
	0,5 cm	1 cm	2 cm	4 cm
1,22	0,41 cm	0,82 cm	1,64 cm	3,28 cm
2,44	0,2 cm	0,41 cm	0,82 cm	1,64 cm
4,88	0,1 cm	0,2 cm	0,41 cm	0,82 cm

Für sämtliche Bedingungen wurde der Fitts'sche ID der Aufgabe bezüglich der Bewegung auf dem Tablett berechnet. Sämtliche Analysen der Daten erfolgten in SPSS der Version 12.0 bzw. 13.0.

4.2 Ergebnisse

In diesem Kapitel werden die Ergebnisse zu den im letzten Kapitel beschriebenen Hypothesen vorgestellt. Als statistische Methoden kamen lineare Regressionsanalysen, sowie Varianzanalysen zum Einsatz.

4.2.1 Hypothese 1: Transformation und Bewegungszeit

Für die Untersuchung der Daten bezüglich Hypothese 1 mussten diese genauer betrachtet werden. Innerhalb eines Blocks war die Tablettedistanz über alle Bedingungen konstant. Die Displayzielgröße und die Cursortransformation wurden als unabhängige Variablen bei jeder Bedingung verändert. Dies führte zu unterschiedlichen Tablettzielgrößen, die von diesen beiden Variablen bestimmt wurden. In Tabelle 4.2 sind die Tablettzielgrößen bei jeweiligen Transformationen und Displayzielgrößen verzeichnet. Die Tablettzielgrößen 0,41 cm und 0,82 cm kamen dabei bei jeweils drei unterschiedlichen Bedingungen vor. In diesen drei Bedingungen waren also sowohl die Tablettedistanz, als auch die Tablettzielgröße konstant, und ergaben daher identische Fitts' ID-Schwierigkeiten bezüglich der motorischen Leistung. Siehe hierzu auch Tabelle 4.1 Für die folgenden Analysen wurden von allen Versuchspersonen nur die Trials mit diesen sechs Bedingungen in die Berechnungen einbezogen. Die Bedingung mit der 0,41 cm Tablettzielgröße entsprach der „kleinen Zielgröße“ und die Bedingung mit 0,82 cm der „großen Zielgröße“.

Zunächst wurde untersucht, inwiefern ID und Bewegungszeit miteinander korrelieren. Die Regressionsanalyse für die kleine kleine Zielgröße ergab $R^2 = 0,95$ (kor-

rigiert), d.h. 95% der Varianz der Bewegungszeit wird durch den ID erklärt. Für die große Zielgröße war $R^2 = 0,91$ (korrigiert). In den Abbildungen 4.3 und 4.4 sind für beide Zielgrößen der Zusammenhang zwischen den ID und der Bewegungszeit dargestellt, mit $BZ_{0,41} = -0,34 \text{ s} + 0,28 \text{ s/bits}$ und $BZ_{0,82} = -0,18 \text{ s} + 0,27 \text{ s/bits}$. Hierbei steigt die Regressionsgerade an, so dass für steigende ID sich steigende Bewegungszeiten ergeben. Die Fitts'sche Aufgabenschwierigkeit hat somit einen hochsignifikanten Effekt auf die Bewegungszeit.

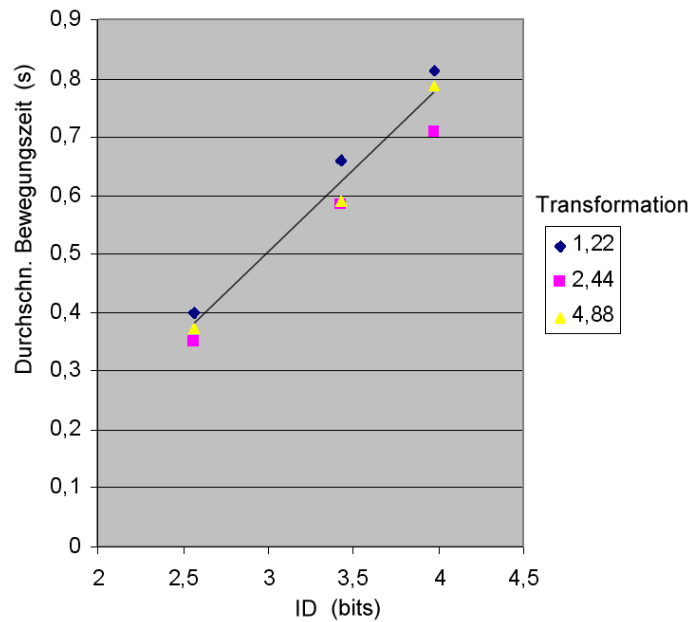


Abbildung 4.3: Bewegungszeiten in Abhängigkeit der Schwierigkeit (kleine Zielgröße; 0,41 cm)

Die durchschnittlichen Bewegungszeiten für die zwei Tablettzielgrößen sind in Abbildung 4.5 zu sehen. Die linken drei Säulen sind die Zeiten für die kleine (0,41 cm) und die rechten für die große Zielgröße (0,82 cm). Die drei Säulen einer Bedingung stehen für die einzelnen Cursortransformationen. Es ist zu erkennen, dass bei beiden Größenbedingungen eine ähnliche Systematik in den Bewegungszeiten zu finden ist. Die 2,44:1 Transformation führt in beiden Fällen zu niedrigeren durchschnittlichen Bewegungszeiten (0,548 s für die kleine und 0,441 s für die große ZG) als die 1,22:1 (kleine ZG: 0,625 s; große ZG: 0,506 s) bzw. 4,88:1 Transformation (kleine ZG: 0,583 s; große ZG: 0,524 s).

Die Varianzanalyse für die kleine Zielgröße ergab, dass die Transformation keinen

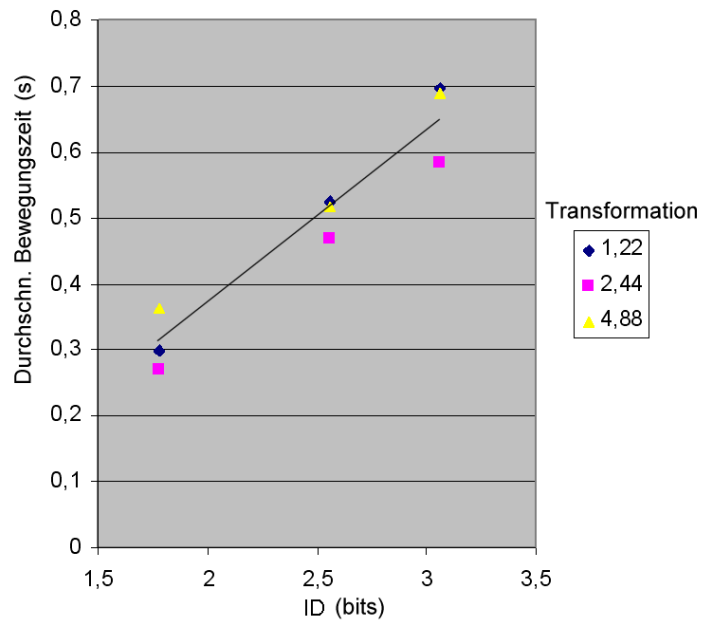


Abbildung 4.4: Bewegungszeiten in Abhängigkeit der Schwierigkeit (große Zielgröße; 0,82 cm)

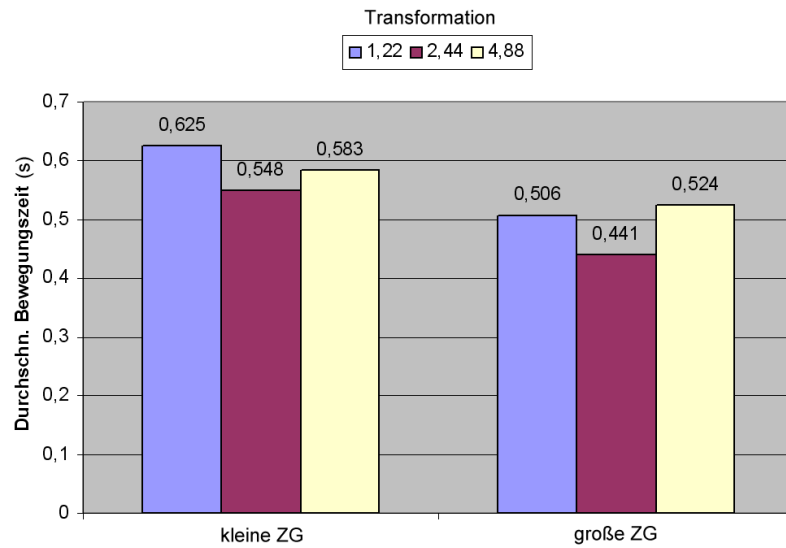


Abbildung 4.5: Durchschnittliche Bewegungszeiten bei zwei verschiedenen Zielgrößen

signifikanten Effekt auf die Bewegungszeit hat $F_{(2,72)} = 1,985$; $p = 0,145$. Bei der Untersuchung der vorhin genannten Systematik, also der Analyse von Transformation*ID, ergab sich kein Effekt auf die Bewegungszeit $F_{(4,72)} = 0,215$; $p = 0,93$. Im Gegensatz dazu wurde bei dem Fitts'schen ID ein hochsignifikanter Effekt gefunden $F_{(2,72)} = 52,752$; $p < 0,01$.

Bei der großen Zielgröße war der Effekt der Transformation tendenziell signifikant $F_{(2,72)} = 2,643$; $p = 0,078$, während der Zusammenhang von Transformation und ID wie zuvor keinen Effekt zeigte $F_{(4,72)} = 0,346$; $p = 0,846$. Der Effekt des IDs auf die Bewegungszeit war auch bei der großen Zielgröße hochsignifikant $F_{(2,72)} = 41,601$; $p < 0,01$. In Bezug auf die Hypothese konnte also in einem Fall ein tendenziell signifikanter Effekt gefunden werden. Im zweiten Fall wurde zwar kein solch signifikanter Effekt gefunden, die Systematik der durchschnittlichen Bewegungszeiten (Abb. 4.5) weist jedoch auf eine hypothesenkonforme Tendenz der Daten hin.

Zusätzlich zur Untersuchung der Effekte von ID und Transformation auf die Bewegungszeit wurden die gleichen Analysen auch bezüglich des Fehlerquotienten durchgeführt. Im Fall der kleinen Zielgröße konnte für die Transformation ein signifikanter Effekt auf den Fehlerquotienten gefunden werden $F_{(2,72)} = 3,693$; $p < 0,05$. Keinen Effekt hatte der ID $F_{(2,72)} = 0,699$; $p = 0,501$ und die Interaktion von Transformation und ID auf den Fehlerquotienten $F_{(4,72)} = 0,212$; $p = 0,931$. Für die große Zielgröße war kein Effekt der Transformation auf den Fehlerquotienten zu finden $F_{(2,72)} = 2,169$ $p = 0,122$, genauso wenig wie für die Interaktion von Transformation und ID $F_{(4,72)} = 0,756$; $p = 0,558$. Für den ID wurde ein hochsignifikanter Effekt gefunden $F_{(2,72)} = 7,346$; $p = 0,01$.

Die durchschnittliche Fehlerquote für die große Zielgröße betrug insgesamt 5,4% und für die kleine Zielgröße 12%. Die drei IDs der kleinen Zielgröße hatten jeweils die mittleren Fehlerquoten von: 10,1% für ID 2,56 bits; 13,1% für ID 3,43 und 12,8% für ID 3,97. Die IDs der großen Zielgröße hatten mittlere Fehlerquoten von: 1,9% für ID 1,78; 6% für ID 2,56 und 8,1% für ID 3,06. In Tabelle 4.3 sind die durchschnittlichen Fehlerquotienten nach Aufgabenschwierigkeit und ID aufgelistet.

4.2.2 Hypothese 2: Transformation und Distanzschätzung

In Hypothese 2 wurde angenommen, dass die Probanden die (konstante) Tablettidistanz bei unterschiedlichen Displaydistanzen unterschiedlich beurteilen würden. Die

Tabelle 4.3: Fehlerquotienten nach ID

Zielgröße	ID	Mittelwert Fehlerquotient
klein	2,56 bits	10,1%
	3,43 bits	13,1%
	3,97 bits	12,8%
groß	1,78 bits	1,9%
	2,56 bits	6,0%
	3,06 bits	8,1%

Tabelle 4.4: Test der Schätzungen vs. Tablettedistanzen

Tablettedistanz	Displaydistanz	Schätzung	T-Test Wert
2 cm	kurz = 2 cm	1,97 cm	T(8) = -0,106; n.s.
	mittel = 4 cm	3,80 cm	T(8) = 3,676; p < 0,05
	lang = 8 cm	6,26 cm	T(8) = 5,88; p < 0,01
4 cm	kurz = 4 cm	5,85 cm	T(8) = 2,085; n.s.
	mittel = 8 cm	8,83 cm	T(8) = 5,76; p < 0,01
	lang = 16 cm	12,86 cm	T(8) = 6,48; p < 0,01
6 cm	kurz = 6 cm	8,14 cm	T(8) = 1,69; n.s.
	mittel = 12 cm	11,19 cm	T(8) = 6,49; p < 0,01
	lang = 24 cm	18,45 cm	T(8) = 6,62; p < 0,01

unterschiedlichen Displaydistanzen kommen durch die Variation der Transformationen zustande. Es wurde abhängig von der kurzen, mittleren bzw. langen Displaydistanz in allen drei Fällen ein hochsignifikanter Effekt auf das Urteil der Probanden gefunden, mit $F_{(2,9)} = 16,34$; $p < 0,01$ für die 2 cm Tablettedistanz, $F_{(2,9)} = 11,05$; $p < 0,01$ für die 4 cm Distanz und $F_{(2,9)} = 14,58$; $p < 0,01$ für die 6 cm Distanz.

In Abbildung 4.6 ist dieser Zusammenhang deutlich zu sehen. Pro drei Tablettedistanzen fielen die Schätzungen der Versuchspersonen sehr unterschiedlich aus, abhängig davon welche Distanz sie auf dem Bildschirm wahrgenommen haben. In den drei Säulen entspricht die linke (blau) der kurzen, die rechte (gelb) der langen und die mittlere (violett) der mittleren Displaydistanz bzw. Cursortransformation. In Tabelle 4.4 sind die Werte nochmals zusammengefasst. Die T-Tests ergaben, dass bei allen Tablettedistanzen, nur bei der kurzen Displaydistanz (also 1,22:1 Transformation) die Schätzung sich nicht signifikant von der echten Tablettedistanz unterschied. Bei diesen Schätzungen war die Displaydistanz gleich der Tablettedistanz.

Für die mittleren und langen Displaydistanzen wurde die umgekehrte Analyse durchgeführt. Es wurde geprüft, ob die Probanden bei der Schätzung nur die auf

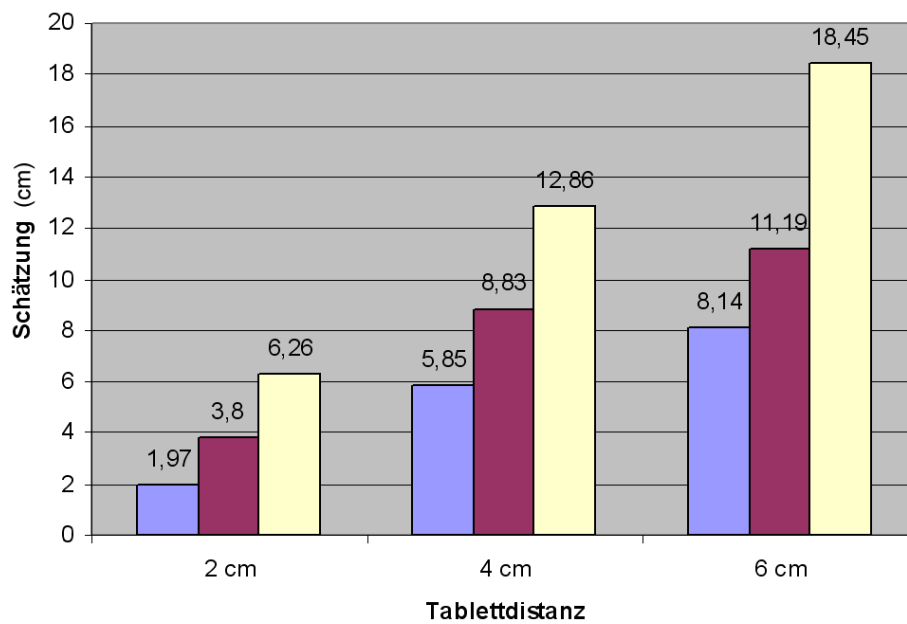


Abbildung 4.6: Schätzung der Tablettdistanzen bei verschiedenen Transformationen

dem Bildschirm gesehene Distanz reproduzierten. In Tabelle 4.5 sind die Ergebnisse der T-Tests wiedergegeben. Für die langen Displaydistanzen ergab sich bei allen drei Tablettdistanzen ein signifikanter Effekt der Displaydistanz auf die Schätzung der Probanden. Bei den Displaydistanzen der mittleren Entfernungen ergaben sich keine signifikanten Effekte.

Die unterschiedlichen Displaydistanzen spiegelten sich in den Urteilen der Probanden wieder. Es gab signifikante Abweichungen der Urteile von der tatsächlich zurückgelegten Tablettdistanz. Die Schätzung der kurzen Displaydistanz entsprach der tatsächlichen Tablettdistanz, hatte aber aufgrund der Transformation auch an-

Tabelle 4.5: Test der Schätzungen vs. Displaydistanzen

Tablettdistanz	Displaydistanz	Schätzung	T-Test
2 cm	mittel = 4 cm	3,80 cm	n.s.
4 cm	mittel = 8 cm	8,83 cm	n.s.
6 cm	mittel = 12 cm	11,19 cm	n.s.
2 cm	lang = 8 cm	6,26 cm	$T(8) = -2,4; p = 0,043$
4 cm	lang = 16 cm	12,86 cm	$T(8) = -2,3; p = 0,051$
6 cm	lang = 24 cm	18,45 cm	$T(8) = -2,95; p = 0,018$

Tabelle 4.6: Mittelwerte und Standardabweichungen der Zielbewegungszeiten

Zielbewegungen	Mittelwert	Standardabweichung
1.-5.	0,638 s	0,115 s
6.-25.	0,548 s	0,118 s
Gesamt	0,593 s	0,122 s

nähernd eine 1:1 Beziehung zwischen Display- und Tablettedistanz. Die Schätzung der mittleren Displaydistanz entsprach der gesehenen Distanz auf dem Bildschirm. Für die lange Displaydistanz ergab sich eine Unterschätzung von der gesehenen Displaydistanz.

4.2.3 Hypothese 3: Nacheffekte bei Transformationswechsel

Um herauszufinden, ob Hypothese 3 zutrifft, also ob es nach einem Wechsel der Cursortransformation zu einem Nacheffekt kommt, wurden die Bewegungszeiten der Zielbewegungen untersucht. Jede Bedingung erforderte 25 gültige Treffer, bevor eine neue Bedingung, und damit eine andere Transformation präsentiert wurde. Für die Entscheidung, ob es zu einem Nacheffekt kommt oder nicht, wurden die ersten fünf Zielbewegungen mit den restlichen 20 verglichen. Hierzu wurden die Mittelwerte der Bewegungszeiten gebildet (siehe Tabelle 4.6). Die ersten fünf Zielbewegungen hatten eine durchschnittliche Bewegungszeit von 0,638 s (STD 0,115) im Gegensatz zu den übrigen 20 Zielbewegungen (0,548 s; STD 0,118). Der Mittelwert der Bewegungszeit betrug 0,595 s (STD 0,122) auf alle 25 Bewegungen berechnet. Obwohl die Bewegungszeiten der ersten fünf Zielbewegungen gegenüber den nachfolgenden 20 höher waren, war diesbezüglich kein signifikanter Effekt zu finden ($F_{(1,9)} = 2,687$; $p = 0,121$).

Es wurde eine genauere Analyse der Daten durchgeführt, bei dem die Zielbewegungen der ersten fünf (durchschnittlich 0,639 s; STD 0,115), mit den letzten fünf verglichen (durchschnittlich 0,531 s; STD 0,12) (siehe Tabelle 4.7) wurden. Auch hier ist zu sehen, dass die Standardabweichung sehr hoch ist. Es ergab sich auch hier kein signifikanter Effekt in den paarweisen Vergleichen ($p = 0,609$).

Obwohl ein Übungseffekt in den mittleren Bewegungszeiten zu beobachten war, ließ sich kein Nacheffekt in den ersten Zielbewegungen feststellen, die als ein Nacheffekt des Transformationswechsels hätte gewertet werden können.

Tabelle 4.7: Gesonderter Vergleich der ersten und letzten fünf Zielbewegungen

Zielbewegungen	Mittelwert	Standardabweichung
1.-5.	0,638 s	0,115 s
21.-25.	0,531 s	0,12 s

4.3 Diskussion

Nachdem die Ergebnisse der statistischen Auswertung vorgestellt wurden, werden in diesem Kapitel diese Ergebnisse hinsichtlich der Hypothesen diskutiert.

Die **erste Hypothese** lautete, dass bei konstanten Tablett-Distanzen und -zielgrößen die Cursortransformation einen Effekt auf die Ausführungszeit hat. Bezüglich der großen Zielgröße (0,82 cm) konnte ein tendenzieller Effekt auf die Bewegungszeit gefunden werden. Kein Effekt der Transformation auf die Bewegungszeit konnte bei der kleinen Zielgröße (0,41 cm) gefunden werden. Grundsätzlich zeigt sich in den Daten aber eine hypothesenkonforme Tendenz. Der Effekt müsste an einer größeren Stichprobe verifiziert werden um hier Sicherheit zu erlangen.

Bei der Transformation fand sich mit der kleinen Zielgröße ein signifikanter Effekt auf den Fehlerquotienten. Für die große Zielgröße fand sich jedoch kein Effekt. Letzteres könnte allerdings durch die sehr niedrige Schwierigkeit (also niedrige ID) in diesen Bedingungen begründet sein. Die große Zielgröße hat mit 5,4% auch insgesamt eine wesentlich niedrigere Fehlerquote, als die kleine Zielgröße mit 12%. Die Probanden konnten bei großen Zielgrößen und kleinen Displaydistanzen kaum Fehler machen, so dass dieser Effekt die Ergebnisse beeinflusst haben könnte.

Der signifikante Effekt der Transformation auf den Fehlerquotienten liegt möglicherweise in den Instruktionen begründet. Die Versuchspersonen haben die Geschwindigkeit ihrer Bewegung scheinbar nicht der empfindlicheren Transformation angepasst, was zu mehr Fehlern geführt hat. In zukünftigen Untersuchungen müssten die Probanden daher explizit instruiert werden, damit sie ihre Geschwindigkeit auch an die Transformation anpassen. Es muss ihnen deutlicher vermittelt werden, dass sie im Zweifelsfall eher genau als schnell arbeiten sollen.

Als **zweite Hypothese** wurde angenommen, dass die wechselnden Cursortransformationen einen Effekt auf die empfundene, zurückgelegte Tablett-Distanz haben würden. Hierbei wurde für alle drei Tablett-Distanzen ein hochsignifikanter Effekt der Displaydistanz auf die Schätzung der Probanden gefunden. Die während der

Schätzung gezeigten Displaydistanzen entsprachen den drei verschiedenen Transformationen. Der Schluss, dass die Probanden die Tablettedistanzen durch die verschiedenen Transformationen anders wahrnahmen, wäre durch die Ergebnisse gestärkt. Andererseits gibt es begründete Zweifel, dass andere Faktoren für diese Ergebnisse verantwortlich waren. Der Grund ist die Analyse, von signifikanten Unterschieden zwischen dargebotener Displaydistanz und der Schätzung durch die Probanden. Für die kurzen und mittleren Displaydistanzen unterschieden sich die Längenschätzung nicht signifikant. Die Probanden haben also möglicherweise nicht die Tablettedistanz versucht zu rekonstruieren, sondern haben nur die visuell wahrgenommene Distanz wiedergegeben. Das trifft zwar nur bedingt auf die lange Distanz zu, kann aber durch die ungewöhnlich große Tablettedistanz von bis zu 24 cm erklärt werden. Diese ungewöhnlich langen Distanzen kamen den Probanden womöglich zu unrealistisch vor, so dass sie ihre Schätzung nach unten korrigiert haben.

Die Instruktionen für die Distanzschätzung müssen dahingehend verbessert werden, dass die Versuchspersonen besser verstehen, welche Distanzen sie schätzen sollen. Möglicherweise muss sogar die Methodik modifiziert werden, um verlässlichere Daten über die subjektiv empfundene Tablettedistanzen zu erhalten.

In **Hypothese 3** wurde ein Nacheffekt bei einem Wechsel der Transformation vorhergesagt. Die Analyse der Daten lieferte keine signifikanten Befunde. Wurden die ersten fünf Zielbewegungen mit den restlichen 20 bzw. mit den letzten fünf Bewegungen verglichen, ergab sich kein signifikanter Effekt auf die Bewegungszeit, trotz Abnahme der durchschnittlichen Bewegungszeiten. Tendenziell scheinen die Ergebnisse die Hypothese zu stützen, wenn auch hier erst eine nachfolgende Untersuchung mit einer größeren Stichprobe Gewissheit schaffen kann.

Nachdem in diesem Kapitel die Ergebnisse aus der statistischen Analyse diskutiert wurden, wird im nächsten Kapitel näher auf die softwarespezifische Evaluation der Pilotstudie eingegangen.

5 Diskussion

Im vorherigen Abschnitt wurden Methodik, Durchführung und die Ergebnisse des Versuchs vorgestellt und diskutiert. Dieses Kapitel bewertet die aufgetretenen Probleme und zeigt mögliche Lösungen. Wenn möglich, werden Wege aufgezeigt, um sowohl die Software als auch die Methodik für zukünftige Studien zu verbessern. Weiterhin werden als Ausblick auf zukünftige Untersuchungen neue Hypothesen vorgeschlagen, die sich direkt aus den aktuellen Experimenten ableiten ließen.

5.1 Einordnung der Ergebnisse

Bei der in dieser Arbeit entwickelten Software stand die Variation der Cursortransformation im Mittelpunkt. In den durchgeführten Fitts' Tapping Aufgaben wurde untersucht, inwiefern die Unterschiede von motorisch ausgeführter Bewegung und Handlungseffekt auf dem Bildschirm die Leistung der Versuchspersonen beeinflussten. Die Ergebnisse haben tendenziell gezeigt, dass bei gleichen motorischen Bewegungen die veränderte visuelle Rückmeldung einen Effekt auf die Bewegungszeit und die Fehlerrate hatte. Diese Tendenz spiegelte sich auch in den Distanzschätzungen der Probanden wieder. Obwohl nicht mit Sicherheit feststand, ob der Effekt durch Probleme in der Instruktion hervorgerufen wurde, entsprachen diese Ergebnisse ebenfalls den Erwartungen. Nach den klassischen Modellen im Sinne von Fitts ist zu erwarten, dass für die Bewegungszeit und Fehlerrate die motorische Schwierigkeit (Controlamplitude und -zielgröße) einer Aufgabe ausschlaggebend ist. In diesem Sinne dürfen Aufgaben, welchen gleiche Tablettamplituden und -distanzen zugrunde liegen, keine Variation der Leistungsmerkmale ausweisen. Die Ergebnisse dieser Untersuchung hingegen stützen die Überlegungen der Theorie der Ereigniskodierung (siehe Kap. 2.5). Basieren Wahrnehmung und Handlungsplanung auf den gleichen Mechanismen, so sind genau die Effekte zu erwarten, die sich im Experiment gezeigt haben. Die Planung einer Bewegung führt zu Aktivierung und Integration der

entsprechenden Kodes, die mit dem wahrgenommenen Stimulus interferieren. Umgekehrt sind Inkompatibilitäten auch in umgekehrter Richtung zu erwarten, so dass die visuelle Rückmeldung mit der propriozeptorischer Rückmeldung interferiert. Die Schätzung der zurückgelegten Distanzen hatte eine starke Neigung in Richtung der gesehenen Distanzen. Dieser Effekt entstand möglicher Weise durch eine stärkere Gewichtung der visuellen, gegenüber den propriozeptorischen Informationen.

Grundsätzlich hat sich die Software für den Zweck der Untersuchung der sensumotorischen Effekte bewährt. In den folgenden Kapiteln werden Aspekte der Software und der Experimentalmethodik betrachtet, deren Verbesserung präzisere Ergebnisse ermöglichen. Durch zukünftige Experimente kann mittels der Software weiter untersucht werden, welche Effekte die Werkzeugtransformation auf die Wahrnehmung und die Handlungsplanung hat.

5.2 Software und Versuchsanordnung

In Kapitel 3 wurde beschrieben, wie die von der Software geforderten Funktionen konkret umgesetzt wurden. Nach der Durchführung der Pilotstudien und der Auswertung der Versuchsdaten konnten neue Erkenntnisse bezüglich dieser Funktionen gewonnen werden. Diese Erkenntnisse werden genutzt, um ausgewählte Aspekte der Methodik und der Software kritisch zu reflektieren.

5.2.1 Matlab und Psychtoolbox

Die Entscheidung, Matlab als Entwicklungsumgebung zu verwenden, erwies sich in fast allen Belangen als richtig. Kleinere Probleme gab es bei der Anpassung an die unterschiedlichen Versionen von Psychtoolbox, die es jeweils für Windows und MacOS gibt. Manche Funktionen sind in der einen Version noch nicht verfügbar, während andere Funktionen aufgrund der technischen Differenzen zwischen den Betriebssystemen nicht auf die gleiche Art implementiert sind. Ein Beispiel ist die GetMouse Funktion, die in der MacOS-Version nur einen, und in der Windows Version drei Mausknöpfe ansprechen kann. Weiterhin verlangt die MacOS-Version von Matlab, dass boolesche Variablen mit '0' und '1' anstatt 'false' und 'true' definiert werden. Da die Windows-Version beide Schreibweisen akzeptiert, konnte das Problem einfach gelöst werden, indem die restriktivere Notation verwendet wurde.

Ungelöst blieb das Problem, warum die grafische Benutzeroberfläche des Datenreduktionsprogramms nicht unter MacOS lauffähig war. Prinzipiell spricht seitens der Technik her nichts gegen eine plattformunabhängige Benutzeroberfläche. Möglicherweise liegt der Grund in den verschiedenen Versionsnummern der Windows- bzw. MacOS-Versionen von Matlab. Unter Windows wurde die 6er Version verwendet, unter MacOS die neuere 7er Version. Nach Rücksprache wurde vereinbart, dass die Datenreduktion nur auf den Windows-Versionen lauffähig sein muss. Aber selbst mit dieser Einschränkung war ein reibungsloser Betrieb nicht von vorn herein gegeben. Die Datenreduktion verwendet zum Glätten der Cursordaten Funktionen, die in der normalen Installation von Matlab nicht enthalten sind und spezielle Zusatzpakete verlangt. Aus dem Grund konnte die Datenauswertung nur auf Rechnern mit den entsprechenden Installationen durchgeführt werden. Dieser Umstand könnte behoben werden, wenn anstatt diesen speziellen Matlab-Funktionen eine selbst geschriebene Glättungsfunktion verwendet würde. Für die erwähnte Inkompatibilität der grafischen Benutzeroberfläche, könnte auf die Verwendung Jener gänzlich verzichtet werden, da sie lediglich für eine höhere Benutzerfreundlichkeit programmiert wurde. Ohne weiteres kann ein textbasierter UI die gleichen Aufgaben übernehmen.

5.2.2 Verzerrungen des Bildschirms

Wie in Kapitel 3.7 bereits geschildert, muss für jeden Monitor gemessen werden, wie viele Pixel einem Zentimeter entsprechen. Diese Messung wurde auch an den Bildschirmen vorgenommen, an denen später die Experimente durchgeführt wurden. Eher zufällig ist hierbei aufgefallen, dass mehrere Messungen nicht das gleiche Ergebnis lieferten. Je weiter am Rand des Bildschirms gemessen wurde, umso mehr Pixel passten in einen Zentimeter. Die Ursache hierfür liegt in einer Verzerrung des Bildes durch den Monitor und ist in Abbildung 5.1 schematisch dargestellt.

Die Gitterstruktur soll zeigen, wie Objekte auf dem Bildschirm gestaucht werden und erklärt, warum ein Kästchen am Rand des Bildschirms kleiner ist als in der Mitte. Der Effekt ist von der menschlichen Wahrnehmung zwar nicht direkt registrierbar, kann aber die Größe und Entfernung der Kästchen in einer Größenordnung von 5% beeinflussen. Grund für diesen Effekt ist das hohe Alter des eingesetzten Monitors, so dass ein neues Gerät sämtliche Probleme in diesem Bereich lösen würde. Ideal wäre ein TFT Bildschirm, bei der es technisch bedingt zu keinerlei geometrischen Verzerrungen kommen kann. Für manche psychologischen Experimente sind aber TFT Monitore wegen ihrer geringen Wiederholfrequenz nicht geeignet.

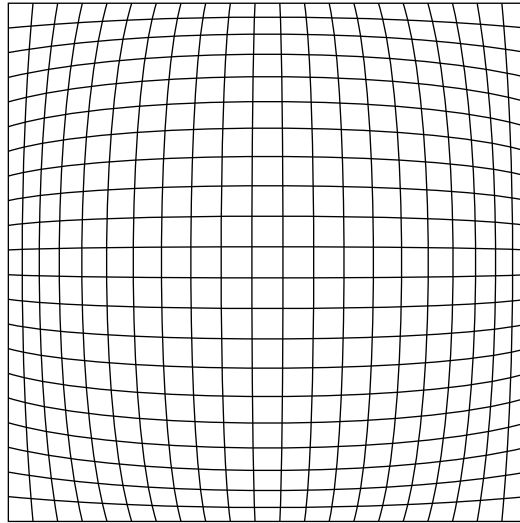


Abbildung 5.1: Kissenartige Verzerrungen am Bildschirm

5.2.3 Fluktuation der Zielgrößen

In den Trials mit unterschiedlichen Transformationen wurde durch die Software die Cursorausdrucksstärke konstant gehalten (siehe Kapitel 3.6). Die effektiven Targetgrößen waren daher (unabhängig von der Transformation) gleich groß. Ein anderer Effekt beeinflusste die effektive Targetgröße jedoch ebenfalls und war von der jeweiligen Position der Zielkästchen auf dem Bildschirm abhängig. In Abbildung 5.2 sind zwei Beispiele für Targetkästchen dargestellt. Das hellgraue Raster beschreibt die einzelnen Bildschirmpixel während das schwarze Raster für die effektiv erreichbaren Pixel steht. Durch das rot gefärbte Kästchen ist das Zielkästchen so abgebildet, wie sie auf dem Bildschirm von den Probanden gesehen wird. Ihre Breite und Höhe ist hier in beiden Fällen 17 Pixel groß. Die reduzierte Auflösung des Eingabegerätes erlaubt es, den Cursor nur auf diejenigen Pixel zu positionieren, die durch das schwarze Raster markiert sind. Jetzt wird der Unterschied zwischen den zwei Beispielen deutlich. Im linken Fall beträgt die effektive Targetgröße nur 12 x 12 Pixel während sie im rechten Beispiel 16 x 16 Pixel beträgt. Die Zielgröße ist daher im zweiten Fall 80% größer als im ersten, und das nur bedingt durch die etwas andere Bildschirmposition des Kästchens.

Der Effekt ist natürlich bei den kleinen Zielgrößen am stärksten ausgeprägt, so dass nicht alle Trials gleich stark betroffen waren. Dennoch ist es wünschenswert, diesen

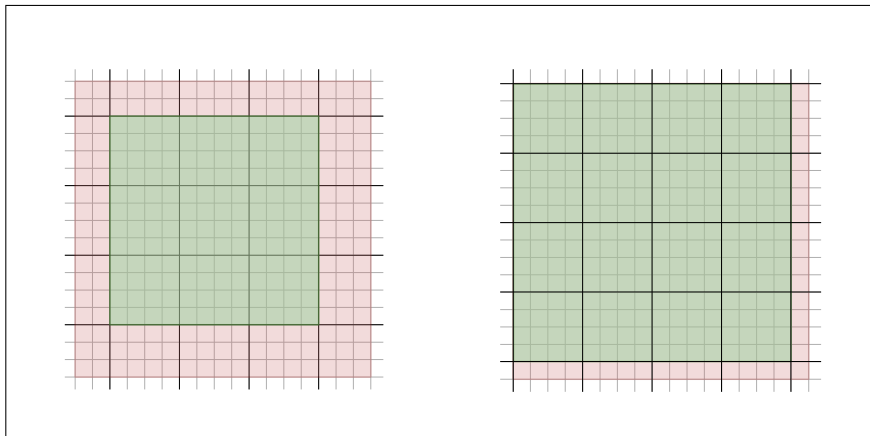


Abbildung 5.2: Variationen der effektiven Targetgrößen

Störeinfluss in den nachfolgenden Studien stärker zu kontrollieren. Eine mögliche Maßnahme besteht darin, in jedem Trial die beiden Kästchen am gröberen Raster auszurichten. Zwar verschieben sich die beiden Kästchen auf diese Weise ein wenig vom Mittelpunkt des Bildschirmes, aber da es sich nur um eine Verschiebung von ein bis drei Pixel handelt, kann das in Kauf genommen werden.

5.2.4 Abweichungen in den Transformationen

In den Formulierungen der unabhängigen Variablen waren ursprünglich die Transformationen eins, zwei und vier vorgesehen. Obwohl die Implementierung der Cursortransformation (siehe Kapitel 3.5) erwartungsgemäß funktionierte, gab es Probleme, die gewünschten Transformationen exakt einzuhalten. Der Grund hierfür liegt in der indirekten Weise, in der die endgültigen Transformationen erzeugt werden. Die Experimentalsoftware benutzt im Hintergrund die Cursorpositionen, die vom Betriebssystem geliefert werden und vervielfacht die Bewegungen um den jeweiligen Verstärkungsfaktor. Für den Fall, dass in einem Trial die 1:1 Transformation dargeboten werden soll, übernimmt die Software die Cursorpositionen des Betriebssystems ohne weitere Modifikationen. Ein Problem entsteht hierbei, wenn der Treiber des Eingabegerätes nicht optimal eingestellt ist, bzw. sich nicht optimal einstellen lässt. Genau dieses Problem tauchte während der Pilotstudie auf, als der Versuch scheiterte, zwischen Tablett und Betriebssystemcursor eine 1:1 Transformation herzustellen. Der Treiber erlaubt die Einstellung der Zeigerempfindlichkeit lediglich

auf einer achtstufigen Skala. Keines dieser Werte entsprach der gewünschten Transformation, so dass die zweitkleinste Empfindlichkeit gewählt wurde. Sie entspricht einer Transformation von 1,22:1 und ist am nächsten an der 1:1 Transformation gelegen.

Zwar ist es prinzipiell möglich, in der Experimentalssoftware die Transformationen so anzupassen, dass die 1,22:1 Transformation des Betriebssystems durch 1,22 geteilt wird und damit im Experiment eine echte 1:1 Transformation erreicht werden kann. Dieses Vorgehen verursacht jedoch Probleme an einer anderen Stelle der Software. Wie in Kapitel 3.6 beschrieben, wird die Auflösung des Eingabegerätes nachträglich reduziert. Diese Reduktion hängt von der größten Transformation ab, die in der Software definiert wurde. Sie bestimmt, wie viele Pixel in den niedrigeren Transformationen während einer Bewegung ausgelassen werden. Würden die Transformationen von 1, 2 und 4 durch den Faktor 1,22 geteilt, wären die resultierenden Transformationen nicht mehr ganzzahlig. Bei der Berechnung der übersprungenen Pixel ergeben sich hierbei Rundungsfehler, die später zu ungenauen Berechnungen führen.

5.2.5 Zeitliche Verzögerung

Eine der Versuchspersonen berichtete nach dem Experiment, dass sie gelegentlich eine leichte Verzögerung bei den Cursorbewegungen empfunden hat. Der nahe liegende Schluss wäre, dass die Ursache für diesen Effekt in einem vorübergehenden Leistungseinbruch des Computers begründet liegt. Der Proband jedoch empfand diese Verzögerung bei kleinen Transformationen und geringen Zielentfernungen stärker, als bei anderen Bedingungen. Aus diesem Grund konnte ein zufälliger Effekt ausgeschlossen werden. Nachdem die fraglichen Trials genauer untersucht wurden, konnte die reduzierte Cursorgenauigkeit als Ursache identifiziert werden: bei kleinen Transformationen und geringen Bildschirmabständen bewegt sich der Cursor vergleichsweise langsam über den Schirm. Durch die künstlich verringerte Genauigkeit des Cursors resultiert aber nicht jede Bewegung des Stiftes in einer Bewegung des Cursors. Abhängig von der jeweiligen Transformation wurden im Experiment null, zwei oder vier Pixel übersprungen. Bei den niedrigen Transformationen erschien es für die Probanden daher, als würde das System erst mit einer kurzen Verzögerung auf ihre Eingaben reagieren. Im Vergleich zu der 4:1 Transformation dauerte es bei der 1:1 Transformation viermal so lange, bis der Cursor auf dem Bildschirm eine neue Position einnahm.

Wie stark die Ergebnisse von diesem Effekt beeinflusst wurden, ist zu diesem Zeitpunkt nicht bezifferbar. Klar ist aber, dass der Einfluss abhängig von der verwendeten Transformation unterschiedlich ist. Eine gesonderte Untersuchung muss durchgeführt werden, um diese Gegebenheiten zu klären. Der hier beschriebene Effekt entfällt auch nicht vollständig, wenn das System auf den in Kapitel 3.5 genannten auflösungsbasierten Wechsel der Transformation umgestellt würde. Wird nämlich die Auflösung des Eingabegerätes herabgesetzt, muss der Proband ebenfalls eine längere Strecke zurücklegen, bevor sich der Cursor auf dem Bildschirm weiterbewegt. Einen minimalen Vorteil hätte die auflösungsbasierte Methode jedoch: sie vervielfacht nicht die zeitliche Verzögerung des Gesamtsystems. Benötigt der Rechner mit der höchsten Transformation beispielsweise 5 ms, um eine Bewegung des Cursors neu darzustellen, so beträgt der Wert in der niedrigen Transformation 20 ms. Der Grund hierfür liegt darin, dass in der niedrigen Transformation drei Cursorbewegungen zwar registriert (und verarbeitet) werden, aber erst bei der vierten Bewegung die neue Position auf dem Bildschirm ausgegeben wird. Wie MacKenzie und Ware (1993) gezeigt haben, ist bei einer Verzögerung von über 75 ms die Leistung der Probanden vermindert. Für die hier entwickelte Experimentalssoftware müssten daher die Latenzzeiten analysiert werden. Sind die Zeiten über 75 ms, so ist zu erwägen, in zukünftigen Experimenten Gegenmaßnahmen zu ergreifen.

5.2.6 Registrierung von Under- und Overshoots

Wie schon in Kapitel 3 beschrieben, war es nicht einfach, eine geeignete Methode zu finden, mit der ein Treffen des Zielkästchens, gemäß den psychologischen Rahmenbedingungen, von der Software registriert werden konnte. Nachdem die Experimentalssoftware auf die umkehrpunktbasierte Meßmethode umgestellt wurde, verbesserte sich die vorher sehr hohe Fehlerrate bei den Probanden. Einen Schwachpunkt hat die neue Methode jedoch: es können keine Fälle von Korrekturbewegungen nach einem Undershoot registriert werden.

In Abbildung 5.3 ist zu sehen, wie ein Under- bzw. Overshoot von dem Probanden korrigiert wird. Diese Korrekturen verfälschen die Ausführungszeiten und sind daher nach Möglichkeit zu unterbinden. Es ist leicht ersichtlich, dass ein Overshoot von der Software als fehlerhafter Trial gewertet wird, da der Umkehrpunkt der Bewegung außerhalb des Zielkästchens liegt. Im Gegensatz dazu führt ein Undershoot zu keinem expliziten Umkehrpunkt, so dass die Software die zweite Bewegung ins

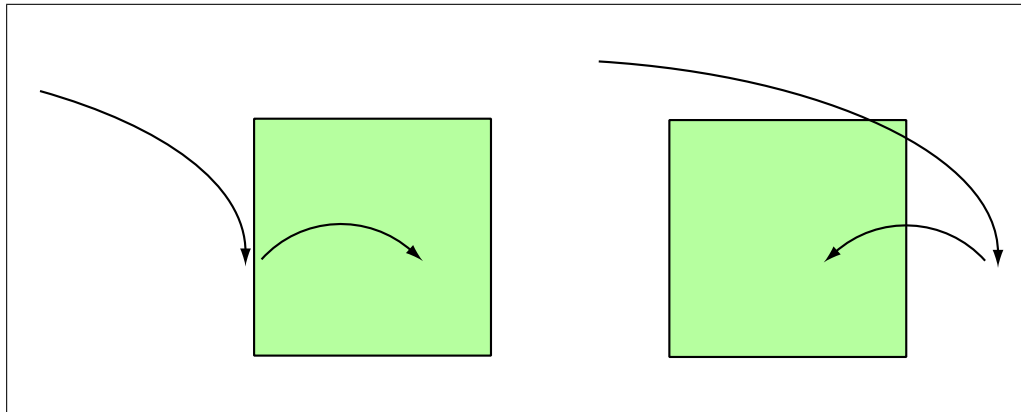


Abbildung 5.3: Under- oder Overshoot des Zielkästchens

Zielkästchen als Teil der ursprünglichen Bewegung registriert. Obwohl in den Experimenten die Probanden instruiert wurden, keine Korrekturbewegungen zu machen, ist nicht auszuschließen, dass sie sich im Einzelfall entgegen der Instruktionen verhalten haben. Die einzige Möglichkeit, solche Korrekturbewegungen zu erkennen, wäre eine zusätzliche Messung der Cursorgeschwindigkeit während der Trials. Ein solches Hybridmodell muss aber sehr genau abgestimmt werden, damit die in Kapitel 3.8 beschriebenen negativen Effekte nicht auftreten. Alternativ ist es möglich, die mitgeloggtten Daten im Nachhinein auf Undershoots zu untersuchen und diese Trials nachträglich als fehlerhaft zu markieren. Auf diese Weise entfällt jedoch die Möglichkeit, eine konstante Anzahl von gültigen Trials zu erreichen.

5.2.7 Handposition auf dem Tablett

In den Experimenten wurden alle Probanden darauf hingewiesen, dass sie den Stift mittig auf dem Tablett positionieren sollten. Ansonsten bestand die Gefahr, dass gerade bei den großen Tablettedistanzen die Probanden an den Rand des Tablett stößen würden. Wenn sie die Bewegungen von der Mitte des Tablett starteten, würden die Hand nach jedem zweiten Trial in die Mitte zurückkehren. Bis auf wenige Ausnahmen hat sich diese Annahme auch bestätigt. Die Ausnahme bildete der Fall, wenn der Proband 25 erfolgreiche Trials mit einer bestimmten Bedingung durchgeführt hat. Bevor ihm die neue Bedingung auf dem Bildschirm dargestellt wurde, blieb dieser für ungefähr zwei Sekunden weiß. Diese kurze Wartezeit war

vorgesehen, um dem Probanden nicht überraschend eine neue Bedingung zu zeigen. Während dieser Pause wurde die Cursorposition aber nicht neu berechnet; der virtuelle Cursor behielt also seine letzte Position bei. Da die Probanden nicht wussten, wann die Anzahl von 25 gültigen Trials erreicht wurde, haben sie stets nach dem letzten Treffer eine neue Bewegung begonnen. Bevor die Versuchspersonen registrierten, dass ein neuer Durchgang gestartet wurde, hatten sie mit dem Stift schon eine gewisse Strecke auf dem Tablett zurückgelegt. Das Programm synchronisierte diese Bewegung aber nicht mit dem Cursor, so dass die Hand sich nicht mehr mittig auf dem Tablett befand, als die neue Bedingung begann. Nach mehreren solchen Wechsel der Bedingungen konnte es vorkommen, dass die sich Probanden mit dem Stift so weit in eine Richtung entfernten, dass sie an den Rand des Tablets stießen. Die strikten Instruktionen behinderten teilweise eine Kompensation dieses Effekts, zumindest bei einigen Probanden (siehe auch Kapitel 5.2.11). Abhilfe für dieses Problem kann zukünftig geschaffen werden, indem die Cursorpositionen während des gesamten Experiments ohne Unterbrechung aktualisiert werden. Alternativ wäre es zwar möglich, die Instruktionen zu erweitern, so dass den Probanden ausdrücklich erlaubt wird, vor jedem neuen Durchgang den Stift abzuheben und mittig auf dem Tablett zu platzieren. Hierbei tritt das Problem auf, dass die Handabdeckung nur wenig Spielraum zum Anheben des Stiftes bietet. Wird dabei der Stift mit einem Abstand von ein bis zwei Zentimetern über das Tablett bewegt, registriert das Tablett teilweise weiterhin diese Bewegung. Dieser Effekt verhindert ein einfaches Neupositionieren des Stiftes und würde erwartungsgemäß von der Software als fehlerhafter Trial gewertet werden.

5.2.8 Hoher Fehlerquotient

Werden in den Daten der Pilotstudie die Fehlerquotienten betrachtet, fallen bei manchen Bedingungen sehr hohe Fehlerzahlen auf. Anstatt der benötigten 25 Trials mussten manche Probanden über 160 Zielbewegungen in einer bestimmten Bedingung durchführen, bis die nächste Bedingung präsentiert wurde. Diese sehr hohe Anzahl an Wiederholungen kann die Ergebnisse verfälschen, da die Probanden möglicherweise gegen Ende des Versuchs ermüden, oder aber aufgrund der höheren Anzahl an Iterationen besser geübt sind.

Da die Bedingungen mit den sehr hohen Fehlerquotienten aus statistischer Sicht ohnehin problematisch sind, könnte die Software modifiziert werden. Die Bedingung würde nach Erreichen eines bestimmten Fehlerquotienten abgebrochen und es würde

zur nächsten Bedingung übergegangen. Sowohl die Probanden als auch die Qualität der Versuchsdaten würde von einer solchen Modifikation profitieren.

5.2.9 Distanzschätzung

Für die größte Verwirrung unter den Versuchspersonen führte ohne Zweifel die Schätzung der zurückgelegten Tablettabstände, welche nach jedem Block abgefragt wurden. Trotz sorgfältiger Instruktionen fühlten sich die meisten Probanden verunsichert, was genau von ihnen abverlangt wurde. Manche von ihnen nahmen irrtümlich an, dass sie den Abstand zwischen den zwei Beispieltastchen mit Hilfe der Abstandsmarker einstellen sollten. Andere waren überzeugt, dass aufgrund der unterschiedlichen „Cursorgeschwindigkeiten“ in den vorangegangenen Trials es nicht möglich sei dem gezeigten Beispiel eine feste Strecke zuzuordnen. Für zukünftige Studien müssen sowohl die Instruktionen, als auch die Methodik der Distanzschätzung überarbeitet werden. Einen interessanten Ansatz hat Sangals (1997) in seinen Versuchen angewandt, bei denen die Probanden eine Trackingaufgabe zuerst mit visueller Rückmeldung geübt haben. Anschließend musste die gleiche Aufgabe ohne Rückmeldung durchgeführt werden. Aus den Abweichungen wurde später abgeleitet, inwiefern die Versuchspersonen die Transformation des Eingabegerätes gelernt haben. Diese Methode kann auch auf die aktuelle Untersuchung angewandt werden, in dem anstatt der Distanzschätzung die Probanden mehrere Trials „blind“ bearbeiten.

5.2.10 Datenreduktion und Korrektur der Fehlerquoten

Die Datenreduktion wird von einem separaten Programm durchgeführt (siehe Kapitel 3.14). Ursprünglich sollte es ausschließlich die Datensätze zusammenfassen und nebenbei Parameter wie maximale Geschwindigkeit und Beschleunigung des Cursors berechnen. Nachdem die Daten aus den Experimenten zum ersten Mal analysiert wurden, zeigte sich bei einigen Durchgängen eine ungewöhnlich hohe Fehlerquote. Die Software hatte gelegentlich eine Serie von kurz aufeinander folgenden fehlerhaften Zielbewegungen registriert. Bei genauerer Betrachtung offenbarte sich, dass es sich bei diesen fehlerhaften Trials im Grunde genommen um keine echten Zielbewegungen handelte. Die Ursache für die hintereinander aufgetretenen Fehler war vielmehr ein kurzes Zittern mit dem Stift im Start- bzw. Zielkästchen. Schon kleine Bewegungen mit dem Stift in abwechselnd entgegengesetzte Richtungen wurden

von der Software stets als ein Trial gewertet, so dass diese Zitterbewegungen in einer fälschlich erhöhten Fehlerquote resultierten. In die Datenauswertungsroutinen wurde eine zusätzliche Überprüfung eingebaut, um die unerwünschten Trials aus der endgültigen Fehlerquote herauszufiltern. Hierzu berechnete das Programm bei jedem fehlerhaften Trial, ob die Endposition der Bewegung sich außerhalb des Startkästchens befand oder nicht. Traf letzteres zu, wurde dieser Trial bezüglich der Fehlerquotenberechnung ignoriert. Offensichtlich hat eine unbeabsichtigte Bewegung der Versuchsperson zu diesem fehlerhaften Trial geführt. Für nachfolgende Studien ist es von Vorteil, die Software zu modifizieren, damit schon während der Experimente diese unerwünschten Trials herausgefiltert werden.

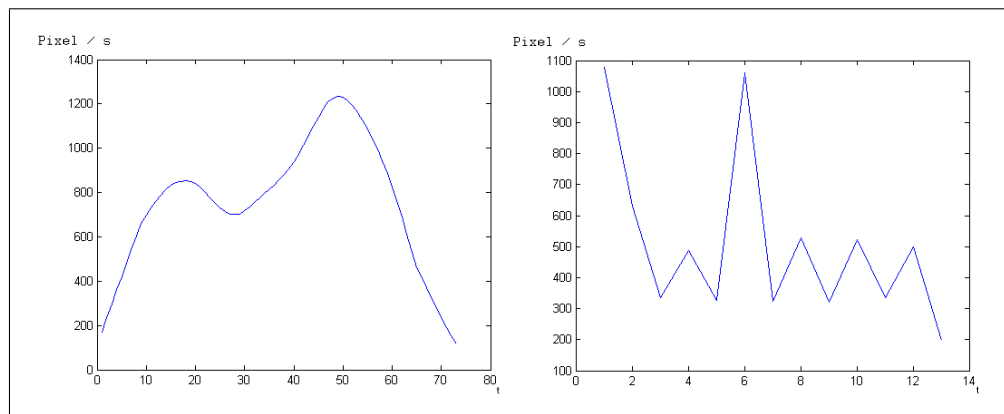


Abbildung 5.4: Cursorgeschwindigkeiten in Trials mit vielen und wenigen Datenpunkten

Obwohl in der Pilotstudie die Geschwindigkeit und Beschleunigung des Cursors nicht als abhängige Variablen einbezogen wurden, könnten diese Parameter während der Datenreduktion durch die Software berechnet werden. Bei der Reduktion der Experimentaldaten zeigte sich jedoch eine Schwäche bei der Kalkulation dieser Parameter. Abbildung 5.4 zeigt die Cursorgeschwindigkeiten zweier Trials, nachdem diese geglättet wurden. Hierbei ist sichtbar, dass die Glättungsfunktion im rechten Beispiel immer noch eine zackenartige Struktur aufweist. Ursächlich hierfür ist die Tatsache, dass bei einigen Trials nur sehr wenige Datenpunkte erzeugt wurden. Gerade bei Trials mit geringen Tablettedistanzen und niedrigen Transformationen liegen die Start- und Zielkästchen auf dem Monitor nur wenige Pixel voneinander entfernt. Durch die künstlich reduzierte Auflösung des Tabletts werden während der Bewegung zusätzlich Pixel übersprungen. So liegen auf der Strecke zwischen den

Zielkästchen nur sehr wenige mögliche Punkte in denen sich der Cursor überhaupt befinden kann. Diese wenigen Datenpunkte reichen später in der Datenreduktion nicht aus, um die Werte hinreichend zu glätten. Unter solchen Umständen ist es daher nur eingeschränkt möglich, den Zeitpunkt der maximalen Cursorgeschwindigkeit zu ermitteln, da die ungeglätteten Daten mehrere Maxima anzeigen können. Um das Problem der wenigen Datenpunkte zu beheben, müssten während der Datenreduktion zwischen den existierenden Punkten zusätzliche Werte interpoliert werden. Auf diese Weise würde sich die Gesamtanzahl der Datenpunkte erhöhen und eine Glättung der Werte wäre bei allen Trials möglich.

5.2.11 Instruktionen

Die Instruktion der Versuchspersonen wurde den Experimenten vorangehend sorgfältig geplant und ausgearbeitet. Ungeachtet dessen offenbarte die Pilotstudie auch hierbei einzelne Stellen, an denen die Instruktionen verbessert werden können:

- Die Aufforderung den Stift nicht vom Tablett abzuheben, wurde von manchen Probanden als Hinweis interpretiert, den Stift sehr fest auf das Tablett zu drücken. Das Resultat waren schmerzende Finger und eine schnelle Ermüdung der Hand. Hier sollte die Instruktion erwähnen, dass das Tablett auch mit wenig Druck zuverlässig arbeitet und der Stift lediglich kontinuierlichen Kontakt zur Oberfläche haben muss.
- Die Versuchspersonen waren verwundert, warum nach den angekündigten drei Blöcken das Experiment noch nicht zu Ende war. Der Abschlusstest sollte daher in den Instruktionen erwähnt werden.
- Die Instruktionen zu den Schätzungen der Tablettedistanzen lösten bei den meisten Teilnehmern Verwirrung darüber aus, was von ihnen verlangt wurde. Aus diesem Grund muss die Instruktion zusammen mit der Methodik die Aufgabe besser beschreiben.

Neben den inhaltlichen Veränderungen müssen die Instruktionen konsistenter gestaltet werden, indem alle Instruktionen auf dem Bildschirm dargestellt werden. Auf diese Weise kann verhindert werden, dass die Probanden unbewusst den vorgelesenen Instruktionen eine höhere Gewichtung zukommen lassen, als den selbst gelesenen.

5.3 Ausblick

Im vorangegangenen Kapitel wurden verbesserungsfähige Aspekte der Methodik und der Experimentalsoftware vorgestellt. Die vorgeschlagenen Modifikationen ergaben sich dabei direkt aus den Problemen und Erfahrungen aus der Pilotstudie. In diesem Kapitel soll ein weit gefasster Ausblick auf zukünftige Studien gegeben werden, der sich sowohl auf methodischer als auch auf theoretischer Ebene in neue Bereiche erstreckt.

5.3.1 Fragestellungen für zukünftige Studien

Durch ihre Vielseitigkeit bezüglich der Einsatzgebiete und Interaktionsschemen, bieten Computereingabegeräte eine Grundlage für eine Vielzahl von interessanten Hypothesen. Im Rahmen der in dieser Arbeit entwickelten Experimentalsoftware sind folgende Bedingungen – jeweils vor dem Hintergrund der variablen Werkzeugtransformation – für eine Studie prädestiniert:

- **Andere Eingabegeräte:** Sind die Ergebnisse aus dieser Studie bei anderen Eingabegeräten wie Trackpoint, Joystick, Touchpad oder Maus ähnlich, oder wirken sich bei diesen Geräten die unterschiedlichen Transformationen anders auf die Leistung aus?
- **Distale vs. Proximale Bewegung:** Wie unterscheiden sich Bewegungen in Richtung des Körpers gegenüber Bewegungen, die in die umgekehrte Richtung führen? Gibt es signifikante Unterschiede in der Ausführungszeit und den Fehlerraten?
- **Geschwindigkeitsprofil:** Verändert ein Wechsel in der Transformation den räumlich zeitlichen Ablauf der Tapping Aufgabe? Gibt es also einen Zusammenhang zwischen der Transformation und dem Zeitpunkt der maximalen Geschwindigkeit?
- **Streuung im Zielbereich:** Inwieweit wirkt sich die Transformation auf die Streuung der Treffer im Zielbereich aus. Sind horizontale Merkmale gleich ausgeprägt wie vertikale?
- **Altersstruktur:** Können sich junge Menschen schneller an die wechselnden Transformationen anpassen? Sind manche Transformationen für jüngere bzw. ältere Menschen besser geeignet?

- **Persönliche Präferenz:** Zeigen Probanden, die im normalen Umgang mit Eingabegeräten eine hohe bzw. niedrige Transformation bevorzugen, eine schlechtere Leistung in den entgegengesetzten Transformationen? Sind die subjektiv präferierten Transformationen tatsächlich höher in der Leistung als andere Transformationen?

5.3.2 Erweiterung der Software

Sowohl die Experimentalsoftware, als auch das Programm für die Datenreduktion können natürlich in beliebig vielen Punkten erweitert werden. In Anbetracht der oben vorgestellten zukünftigen Fragestellungen drängen sich einige Erweiterungen in den Vordergrund. Die Analyse der Trajektorien würde als zusätzliche abhängige Variable die Auswertung der Daten bereichern und könnte in die Reduktionsroutinen eingebettet werden. In diesem Zuge könnten auch weitere Daten über das Geschwindigkeits- und Beschleunigungsverhalten berechnet werden.

Die Experimentalsoftware unterstützt in seiner aktuellen Form nur Transformationen, die konstante und diskrete Werte annehmen. In der Praxis kommen aber auch dynamische Transformationen, z.B. in Form der „Power Mouse“, vor. Mit einer modifizierten Software könnten auch solche Eingabegeräte untersucht werden. Zusätzlich können Eingabegeräte, die für gewöhnlich keine dynamische Transformation erlauben (z.B. das Touchscreen), ebenfalls in Experimente eingebunden und untereinander verglichen werden.

Sehr wichtig ist auch die Bedienbarkeit der Experimentalsoftware. In den ersten Testversionen der Experimentalsoftware traten wiederholt Fehler in der Durchführung auf, da bei jeder Versuchsperson die Parameter an unterschiedlichen Stellen im Programm verändert werden mussten. Spätere Versionen verminderten dieses Problem, indem alle Einstellungen zentral in einer Konfigurationsdatei abgelegt wurden. Die Transformation, die im Abschlusstest verwendet werden sollte, wurde aber immer noch bei Start der Software abgefragt. Der Versuchsleiter musste dafür Sorge tragen, dass alle Transformationen gleich häufig von den Probanden bearbeitet wurden. Zukünftige Versionen der Software sollten die Möglichkeit bieten, eine Liste der Parameter schon vorab einzugeben, so dass bei der Durchführung der Experimente dem Versuchsleiter keine Fehler mehr unterlaufen können.

5.4 Zusammenfassung

Die aus den Experimenten gewonnenen Erfahrungen haben eine Reihe von Möglichkeiten aufgezeigt, die Methodik und die Software zu optimieren. Störeinflüsse wie die Fluktuation der effektiven Zielgrößen, oder erhöhte zeitliche Verzögerungen müssen auf Softwareebene eliminiert werden. Aber auch die Methodik und speziell die Instruktionen der Versuchspersonen bedürfen sorgfältiger Revision, um beispielsweise die Schätzung der Tablettdistanz transparenter zu gestalten. Probleme bei der mittigen Positionierung der Hand sind ebenfalls auf Ebene der Instruktionen zu beheben.

Neben den Verbesserungspunkten lieferte die Studie auch neue Ansätze für zukünftige Untersuchungen. Zusammen mit den Ideen für neue Hypothesen sind Erweiterungsmöglichkeiten bezüglich der Software entstanden, die ihre Einsatzmöglichkeiten weiter ausbauen und praxisnäher gestalten können.

6 Fazit

Die vorliegende Diplomarbeit verfolgte aufgrund ihres interdisziplinären Charakters zwei Ziele. Das erste Ziel war eine klassische Softwareentwicklung. Sie bestand aus der Programmierung einer Experimentalsoftware, mit der Fitts' Tapping Aufgaben durchgeführt und Effekte sensumotorischer Transformationen untersucht werden können. Das zweite Ziel war psychologischer Natur und sah vor, eine Pilotstudie mit Hilfe der entwickelten Software durchzuführen.

Während der Entwicklung der Software mussten die psychologischen Spezifikationen genau eingehalten werden, um die Validität der späteren Ergebnisse zu gewährleisten. Es mussten Probleme überwunden werden, die sich aus dem Zusammenspiel von Eingabegerät, Betriebssystem, Ausgabegerät und den verschiedenen Werkzeugtransformationen ergaben. Um die Transformation zwischen Eingabegerät und Bildschirmausgabe verändern zu können, wurde nicht der Cursor des Betriebssystems benutzt. Stattdessen wurde der Cursor ausgeblendet und im Vollbildmodus ein virtueller Cursor implementiert, der die gewünschten Transformation realisieren konnte. Zusätzlich wurde bei manchen Transformationen die Ausdrucksstärke dieses Cursors künstlich reduziert, damit die verschiedenen Transformationen untereinander vergleichbar blieben.

Gleichzeitig zu der Softwareentwicklung wurde die Pilotstudie geplant und der Versuchsablauf und die nötigen Instruktionen vorbereitet. Mit der Pilotstudie wurden Hypothesen überprüft, die sich mit den Auswirkungen von veränderten Cursortransformationen auf das subjektive Empfinden von Bewegungsdistanzen, sowie auf die objektiven Leistungsmaße der Probanden beschäftigten. Die Daten der Pilotstudie bestätigten tendenziell die Hypothese, dass bei gleichen motorischen Schwierigkeiten, die Gesamtschwierigkeit der Aufgabe von der visuellen Rückmeldung beeinflusst wird. Auch wurde ein starker Zusammenhang zwischen den Transformationen und dem subjektiven Distanzempfinden der Probanden gefunden.

Von den Erfahrungen aus der Pilotstudie konnten wichtige Verbesserungsvorschläge abgeleitet werden. Diese beziehen sich sowohl auf die verwendete Hardware, als auch auf die hier entwickelte Software. In zukünftigen Studien können ein verzerrungsfreier Monitor und abgestimmte Transformationen helfen, noch präzisere Daten zu erhalten. Mit Hilfe der Experimentalssoftware können in Zukunft Fragen beantwortet werden, inwiefern das Alter eine Rolle bei den verschiedenen Werkzeugtransformationen spielt, oder ob die persönliche Präferenz einer bestimmten Transformation auch die optimale ist. Langfristig können so neue Einsichten gewonnen werden und Theorien wie die der Ereigniskodierung weiter erforscht werden.

Literaturverzeichnis

- Arnaut, L. Y. und J. S. Greenstein (1986). Optimizing the touch tablet: The effects of control-display gain and method of cursor control. *Human Factors* 28(6), 717–726.
- Arnaut, L. Y. und J. S. Greenstein (1990). Is display/control gain a useful metric for optimizing an interface? *Human Factors* 32(6), 651–663.
- Babcock, M. K. und J. J. Freyd (1988). The perception of dynamic information in static handwritten forms. *American Journal of Psychology* (101), 11–130.
- Brainard, D. H. (1997). The psychophysics toolbox. *Spatial Vision* 10, 433–436.
- Card, S. K., W. K. English, und B. J. Burr (1978). Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a crt. *Ergonomics* 21, 601–613.
- Card, S. K., J. D. Mackinlay, und G. G. Robertson (1990). The design space of input devices. In *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, S. 117–124. ACM Press.
- Chambers, C. D. und M. Brown (2003). Timing accuracy under microsoft windows revealed through external chronometry. *Behavior Research Methods, Instruments, and Computers* 35(1), 96–108.
- Douglas, S. A. und A. K. Mithal (1997). *The ergonomics of computer pointing devices*. New York, NY, USA: Springer-Verlag New York, Inc.
- Fitts, P. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* 47, 381–391.
- Fitts, P. M. und C. M. Seeger (1953). S-r compatibility: Spatial characteristics of stimulus and response codes. *Journal of Experimental Psychology* 46, 199–210.

- Gollwitzer, P. und J. A. Bargh (1996). *The psychology of action: Linking cognition and motivation to behavior*. NY, USA: Guilford Press.
- Graham, E. D. (1996). Virtual pointing on a computer display: Non-linear control-display mappings. In *Graphics Interface*, S. 39–46.
- Graham, E. D. und C. L. MacKenzie (1996). Physical versus virtual pointing. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, S. 292–299. ACM Press.
- Grosjean, M., M. Shiffrar, und G. Knoblich (2006). Fitts' law holds for action perception. *Psychological Science (in Press)*.
- Heuer, H. (1983). *Bewegungslernen*. Stuttgart: Kohlhammer.
- Hommel, B. (1993). Inverting the simon effect by intention: Determinants of direction and extent of effects of irrelevant spatial information. *Psychological Research* 55, 270–279.
- Hommel, B., J. Müsseler, G. Aschersleben, und W. Prinz (2001). The theory of event coding (tec): a framework for perception and action planning (with commentary). *Behavioral and Brain Sciences* 24(5), 849–937.
- Kunde, W., J. Müsseler, und H. Heuer (2006). Compatibility effects with tool use. *Human Factors (in Press)*.
- Lee, J. C., P. H. Dietz, D. Leigh, W. S. Yerazunis, und S. E. Hudson (2004). Haptic pen: a tactile feedback stylus for touch screens. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, New York, NY, USA, S. 291–294. ACM Press.
- MacKenzie, I. S. und W. Buxton (1992). Extending fitts' law to two-dimensional tasks. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, S. 219–226. ACM Press.
- MacKenzie, I. S. und C. Ware (1993). Lag as a determinant of human performance in interactive systems. In *CHI '93: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, S. 488–493. ACM Press.
- Milner, A. D. und M. A. Goodale (1995). *The visual brain in action*. Oxford University Press.

- Pelli, D. G. (1997). The videotoolbox software for visual psychophysics: transforming numbers into movies. *Spatial Vision* 10, 437–442.
- Posner, M. I. (1978). *Chronometric explorations of mind*. Hillsdale, NJ, USA: L. Erlbaum.
- Rizzolati, G., L. Fadiga, V. Gallese, und L. Fogassi (1996). Premotor cortex and the recognition of motor actions. *Cognitive Brain Research* (3), 131–141.
- Sanders, A. F. (1998). *Elements of Human Performance: Reaction processes and attention in human skills*. Mahwah, NJ, USA: L. Erlbaum.
- Sangals, J. (1997). *Der Einfluß der Bewegungsrückmeldung auf das Erlernen nicht-linearer Werkzeugtransformationen*. Ph. D. thesis, Philipps-Universität Marburg.
- Schmidt, R. A. (1982). *Motor Control and Learning: A Behavioral Emphasis* (second ed.), S. 55–65. Human Kinetics Publishers.
- Shannon, C. und W. Weaver (1949). *The Mathematical Theory of Communication*. Urbana, Illinois: University of Illinois Press.
- Westergaard, M. (2002). Supporting multiple pointing devices in microsoft windows. In *Proceedings of Summer Research Workshop 2002*.
- Young, L. (1969). On adaptive manual control. *Ergonomics* 4, 635–74.