

# Fitts' Gesetz als programmtechnische und empirische Aufgabe für Studierende der Informatik

Alexander Kröner<sup>1</sup> und Timo Götzelmann<sup>2</sup>

**Abstract.** Fitts' Gesetz ist Gegenstand vieler Lehrveranstaltungen in der Informatik. Neben der ursprünglichen Quelle existieren mehrere Formeln für verfeinerte Approximationen des Phänomens. Dieses Paper beschreibt einen Ansatz zum Einsatz dieses Themenkomplexes für Lehrzwecke für Studierende des 3. Regelstudiensemesters. Zunächst setzen Studierende den Sachverhalt als Programmieraufgabe mit mehreren zu testenden Variablen um. Im Weiteren führen sie selbst mit einer Benutzerstudie eine Verifikation durch welche im Folgenden ausgewertet wird. Bei der gemeinsamen Diskussion werden die Ergebnisse durch verschiedene Formeln sowie empirische Grundlagen wie die Ausreiserproblematik erörtert. Durch dieses Lehrkonzept, welches über mehrere Jahrgänge verfeinert wurde, werden mehrere Themen der Mensch-Computer-Interaktion mit praktischen Tätigkeiten für die Studierenden verknüpft. Ziel dessen war, Studierende in die softwaretechnische Umsetzung, die Durchführung und Auswertung der Studie zu involvieren und ihnen damit eine ganzheitliche Sicht auf die Zusammenhänge unterschiedlicher Sachverhalte aus dem Bereich der Mensch-Computer-Interaktion zu geben.

**Keywords:** Fitts' Gesetz, Programmierung, Benutzerstudie, Evaluation, Korrelation

## 1 Einleitung

Fitts' Gesetz bezeichnet eine von Paul Fitts empirisch ermittelte Gesetzmäßigkeit, die beispielsweise im Bereich der Informatik angewendet wird, um den Schwierigkeitsgrad einer Zeigeaufgabe in Abhängigkeit von Größe und Distanz des Ziels abzuschätzen [Fi54]. Fitts' Gesetz ist Thema zahlreicher Forschungsarbeiten und wird immer wieder auf neue Interaktionsformen übertragen (z.B. spezielle für Touchscreens [SS91]). Die zahlreichen Variablen in den derzeit anerkannten Ausprägungen des Gesetzes und den damit verbunden Arten von Zeigeaufgaben machen aus Sicht der Forschung eine Validierung der empirisch gewonnenen Ergebnisse interessant.

Die in Arbeiten zu Fitts' Gesetz gewonnenen Erkenntnisse sind aber auch von grundlegendem Interesse für Lehre im Bereich der Mensch-Maschine-Interaktion. Ein Beispiel bildet das Fach „Interaktive Systeme“ der Fakultät Informatik der Technischen Hochschule Nürnberg Georg Simon Ohm. Im dritten Regelstudiensemester werden theoretische Grundlagen interaktiver Systeme behandelt und in Laborpraktika auf

---

<sup>1</sup> Technische Hochschule Nürnberg Georg Simon Ohm, Fakultät Informatik, Keßlerplatz 12, 90489 Nürnberg, Alexander.Kroener@th-nuernberg.de

<sup>2</sup> Technische Hochschule Nürnberg Georg Simon Ohm, Fakultät Informatik, Keßlerplatz 12, 90489 Nürnberg, Timo.Goetzelmann@th-nuernberg.de

konkrete Aufgabenstellungen angewendet (vgl. [In17]). Programmtechnische Grundlage der praktischen Arbeiten bildet Microsoft Windows Presentation Foundation (WPF). Fitts' Gesetz als Baustein dieses Fachs soll im Sinne der kompetenzorientierten Lehre von Studenten verstanden, angewendet und ggf. auch auf neue Situationen übertragen werden können.

Die Einbindung von Fitts' Gesetz in die Lehrveranstaltung eröffnet die Möglichkeit, Studierende an Aspekte wissenschaftlichen Arbeitens heranzuführen und zugleich programmtechnische Kenntnisse zu vertiefen. Im Vordergrund des hier behandelten Falls standen dabei die folgenden Lernziele:

1. Vertiefung des vermittelten Wissens zur Implementierung von Dialogen
2. Verständnis, welche Parameter die Schwierigkeit einer Zeigeaufgabe beeinflussen
3. Methodik zur Analyse empirischer Beobachtungen kennenlernen und anwenden
4. Kritisches Hinterfragen bestehender wissenschaftlicher Erkenntnisse

Um diese Ziele zu unterstützen wurde ein Lehrkonzept entwickelt, das Ideen des forschenden Lernens (vgl. [Bu70]) aufgreift. Im Vordergrund steht die „*Ausführung der Methoden bis zur Prüfung [...] der Ergebnisse in selbstständiger Arbeit*“ [HHS09]. Andere Aspekte forschenden Lernens – insbesondere die selbständige Wahl von Thema und Methode, sowie die Freiheit eines unbegrenzten Risikos – wurden zugunsten der Umsetzung modifiziert oder eingeschränkt.

Im folgenden Kapitel 2 werden die Anforderungen an die Umsetzung des Lehrkonzepts erläutert. Die in Kapitel 3 vorgestellte Umsetzung wurde in mehreren Iterationen immer weiter entwickelt. Das Kapitel geht auch auf später verworfene Ansätze ein und erläutert Aspekte, die sich mit Blick auf die Anforderungen als hinderlich erwiesen. Neben einer Zusammenfassung von Ansatz und Ergebnissen wird dann in Kapitel 4 die weitere Vorgehensweise erläutert.

## 2 Ausgangssituation

Fitts' Gesetz wird als Grundlage zur Mensch-Computer Interaktion in vielen Studiengängen gelehrt (z.B. [Gr96, P106]). Neben einer rein theoretischen Vermittlung des Sachverhaltes, können jedoch auch andere Elemente in die Lehre mit aufgenommen werden. In der Lehre lässt sich ein experimenteller Zugang zu Fitts' Gesetz auf unterschiedliche Weise realisieren. So können beispielsweise Zeigeaufgaben in ein Spiel eingebettet und die vom Benutzer benötigte Zeit aufgezeichnet und aufbereitet werden (vgl. [Er17]). Praktische Experimente mit Fitts' Gesetz lassen sich zudem in elektronische Kurse mit Vor- und Nachbereitung einbetten (vgl. [Si12]).

Das vorliegende Lehrkonzept greift solche Ansätze auf und entwickelt sie dahingehend weiter, dass die Lernenden zum einen an der Entwicklung des Testprogrammes zum

Experiment beteiligt werden, und zum anderen die zur Analyse nötigen Schritte selbst durchführen müssen. Eine besondere Rahmenbedingung bildet dabei das Ziel, einen zur Analyse geeigneten Korpus im Rahmen der Lehrveranstaltung zu erzeugen. Den Autoren ist kein Ansatz bekannt, welcher Fitts' Gesetz derart in der Lehre einbindet.

Die Umsetzung des Lehrkonzepts sollte verschiedenen Anforderungen folgen, die in erster Linie durch die Rahmenbedingungen der Hochschullehre motiviert sind.

## 2.1 Organisatorische Anforderungen

**R-Planbarkeit:** Das Experiment soll auf regelmäßiger Basis wiederholt werden können, um den initialen Aufwand der Erstellung der Lehrmaterialien zu rechtfertigen. Damit verbunden ist die Anforderung eines straffen und kontrollierbaren Ablaufs der einzelnen Schritte des Lehrkonzepts, um die zeitlichen Auswirkungen auf die Lehrveranstaltung vorhersagbar und beherrschbar zu machen.

**R-Kosten:** Die Arbeiten sollen mit vorhandener Ausstattung der Fakultät und ohne Einbindung zusätzlichen Personals durchgeführt werden können. In direkter Konsequenz sollen zudem aufwändige Hardware- oder Software-Installationen vermieden werden.

**R-Umfang:** Die Umsetzung des Lehrkonzepts innerhalb einer Lehrveranstaltung soll deren Fortgang nicht übermäßig beeinträchtigen.

## 2.2 Teilnehmerbedingte Anforderungen

**R-Vergleichbarkeit:** Die Teilnehmer beherrschen nach dem den Studiengängen zugrundeliegenden Curriculum die Grundlagen der Programmierung in C#. Der individuelle Kenntnisstand kann aber stark abweichen. Dies betrifft auch die Erfahrung der Studierenden mit den für die Umsetzung des Konzepts erforderlichen Werkzeugen zur Datenanalyse. Gründe können hier unter anderem in Vorwissen aus beruflicher Tätigkeit oder in Abweichungen zwischen den Studiengängen liegen. Im vorliegenden Fall wird beispielsweise nur im Studiengang Medieninformatik das Fach „Software-Ergonomie“ gelesen, in dem Fitts' Gesetz ebenso Erwähnung findet. Im Fach „Interaktive Systeme“ werden auf Basis eines gemeinsamen Skriptums beide Studiengänge getrennt von einem jeweils zugeordneten Dozenten betreut. Mit Blick auf diese Diversität müssen Maßnahmen ergriffen werden, die die Vergleichbarkeit der Vorgehensweise und der Ergebnisse verschiedener Studierender gewährleisten.

**R-Motivation:** Mit Blick auf den erhofften Lernerfolg sollen Studierende sich selbsttätig mit der Materie auseinandersetzen. Studierende sind allerdings nicht zwangsläufig intrinsisch motiviert, die im Rahmen des Konzepts geforderten Tätigkeiten voranzutreiben. Also muss der Wert einer aktiven Teilnahme in geeigneter Form deutlich gemacht werden.

### 2.3 Fachliche Anforderungen

**R-Korpus:** Eine geeignete Menge an Messungen soll durchgeführt werden können, um deren empirische Analyse zu ermöglichen.

## 3 Umsetzung

In diesem Abschnitt werden die aufeinander aufbauenden Abschnitte des Lehrkonzeptes eingeführt. Die Umsetzung wurde im Verlauf von drei Jahrgängen immer weiter entwickelt. Im Folgenden wird die Umsetzung im Wintersemester 2016/17 betrachtet unter Verweis auf Erfahrungen aus vorangegangenen Jahrgängen.

Die Lehrveranstaltung wurde von zwei Dozenten zweizügig gelesen (Informatik, Medieninformatik). Im Vorfeld der Aufgabenstellung wurden deren theoretische Grundlagen in einer Vorlesung eingeführt. Diese Einführung umfasste unter anderem Fitts' Experiment und die von ihm postulierte Formel zur Berechnung der Schwierigkeit einer Zeigeaufgabe, eine Modifikation der Formel nach MacKenzie [Ma92], sowie Erweiterungen zur Übertragung von Fitts' Erkenntnissen auf zweidimensionale Zeigeaufgaben [AZ03][MB92]. In der Vorlesung wurde auf die Prüfungsrelevanz des Themas hingewiesen (vgl. *R-Motivation*), u.a. in Verbindung mit Verständnisaufgaben in Form der Peer-Instruction-Methode [Ma96].

In den von den Dozenten betreuten Laborpraktika wurden die Studierenden in parallel arbeitende Gruppen eingeteilt (3 Gruppen Informatik, 2 Gruppen Medieninformatik). Diese Gruppen arbeiteten abgestimmt, aber nicht zeitgleich. Durchgeführt wurden die 90-minütigen Praktika in Rechnerlaboren der Fakultät (vgl. *R-Kosten*). Jeder Studierende verfügte über einen Arbeitsplatz bestehend aus einem Laborrechner (Desktop-PC), einer 2D-Maus und einem 22" Monitor. Software-Plattform bildete Microsoft Windows in Verbindung mit der IDE Visual Studio. Alle Teilnehmer einer Arbeitsgruppe arbeiteten parallel (vgl. *R-Umfang*). Störungen und Ablenkungen während der Phase der Aufzeichnung konnten somit nicht ausgeschlossen werden.

Das Lehrkonzept im Rahmen der Laborpraktika umfasst folgende Schritte:

1. Vermittlung der programmtechnischen und empirischen Aufgabenstellung.
2. Implementierung einer Software-Umgebung, in der zweidimensionale Zeigeaufgaben unter verschiedenen Parametern durchgeführt werden können.
3. Aufzeichnen einer Serie von Zeigeaufgaben mittels dieser Software.
4. Auswertung der empirischen Aufzeichnungen.
5. Diskussion der Ergebnisse.

Für die Durchführung dieser Schritte erwies sich der Umfang von zwei Laborpraktika als ausreichend und vertretbar (vgl. *R-Umfang*).

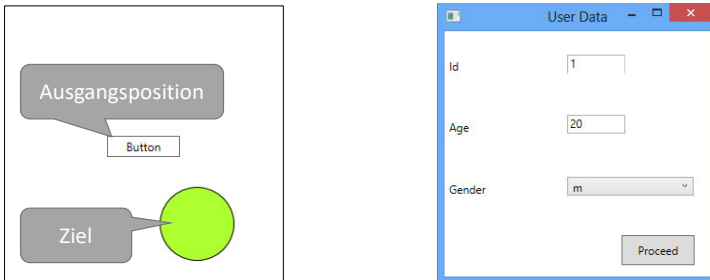


Abb. 1: Benutzerschnittstelle zur Durchführung zweidimensionaler Zeigeaufgaben (Grafik beschnitten und kommentiert, links) und einer der zu implementierenden Dialoge (rechts)

### 3.1 Erster Abschnitt: Programmtechnische Umsetzung

Im ersten Laborpraktikum hatten die Studierenden die Aufgabe ein Werkzeug zu realisieren, mit dem Fitts's Gesetz verifiziert werden sollte. Daneben sollte der Einfluss folgender Variablen auf die Schwierigkeit einer Zeigeaufgabe getestet werden:

- Mauszeigerbeschleunigung (Modifikation des Control-Display Gain [Mac95])
- Vorwissen bei einer Zeigeaufgabe (Precuing [HH13])
- Wechselnde Ruheposition der Hand

Als ersten Schritt führten die Dozenten ein Briefing der Studierenden durch. Damit verbunden war die Ausgabe von schriftlichen Instruktionen. Diese motivierten die Hintergründe der Aufgabe, und spezifizierten die programmtechnische Aufgabe.

Im Anschluss wurde als Grundlage der programmtechnischen Aufgabe ein in C# implementiertes Programmgerüst ausgegeben. Das Gerüst realisiert eine einfache Benutzerschnittstelle zur Durchführung von 2D-Zeigeaufgaben mit zufällig positionierten kreisförmigen Zielen mit variabler Größe (s. Abb. 1). Es misst die Zeit von einem Klick auf eine fest definierte Ausgangsposition bis hin zu einem Klick auf das Ziel. Falls Precuing nicht gewünscht ist, wird das nächste Ziel erst nach Klick auf die Ausgangsposition angezeigt. Eine wechselnde Ruheposition der Hand wird simuliert, in dem der Mauszeiger nach Klicken auf ein Ziel automatisch auf die Ausgangsposition gesetzt wird. Zusätzlich protokolliert die Konfiguration des Experiments und der eigentlichen Messwerte – unter anderem die für die Zeigeaufgabe benötigte Zeit, die Größe und Position des Ziels, sowie einen Fehlerindikator („Ziel verfehlt“).

Die Studierenden wurden angewiesen, das Gerüst punktuell mit grafischen Dialogen zu erweitern; entsprechende Code-Stellen wurden gekennzeichnet. Das für diese Aufgabe erforderliche Code-Verstehen wurde auf in der Lehrveranstaltung behandelte Aspekte

der Dialogprogrammierung (z.B. Datenbindung, Modalität) reduziert. Studierende, die diese Aufgabe nicht bis zum Ende der Lehrveranstaltung fertigstellen konnten, sollten diese Aufgabe bis zum Anfang der folgenden Veranstaltung erledigen. Für die tatsächliche Durchführung der Messungen wurde jedoch eine Musterlösung bereitgestellt, welche von den Studierenden nur noch durch Festlegung mehrerer Variablen konfiguriert werden musste. Diese Musterlösung diente gleichzeitig zur Überprüfung des eigenen Lösungsweges.

#### *Erfahrungen aus vergangenen Semestern*

In der ersten Iteration wurde den Studierenden nur ein Grundgerüst des Werkzeugs zur Verifikation zur Verfügung gestellt. Innerhalb dieses Gerüsts sollten 3 bis 4 Studierende in Gruppenarbeit eine zufällig vom Dozenten zugewiesene Variante der Zeigeaufgabe realisieren (unter anderem eindimensionale oder zweidimensionale Aufgaben, kreisförmige oder ellipsoide Ziele).

Die daraus resultierende Vielfalt an Aufgabenstellungen mit variierender Komplexität erwies sich als ungünstig für die Größe des Korpus der jeweiligen Aufgabe (*R-Korpus*). Zudem führte sie zu sehr unterschiedlichen Umsetzungszeiten, sodass einige Studierendengruppen bereits vor Ende des ersten Abschnittes die programmtechnische Aufgabe beendet hatten, wogegen andere Gruppen noch am Anfang der Umsetzung standen. Somit mussten in der folgenden Veranstaltung diese Studierendengruppen ihre Umsetzung abschließen, während andere Gruppen bereits fertig waren (*R-Planbarkeit* beeinträchtigt). Positive Effekte im Sinne des Konzeptes *Peer-Assisted Learning* (z.B. [Top05]) wurden dem Anschein nach nur begrenzt erreicht. In Konsequenz wurde in der folgenden Iteration die Anzahl der Aufgabenstellungen drastisch reduziert.

Ein weiteres Problem bildeten die teilweise erheblichen Unterschiede in der Umsetzung einer Implementierungsaufgabe. Dies war zum einen trotz individueller Beratungen der einzelnen Gruppen durch die Dozenten auf eine unterschiedliche Interpretation der Aufgabenstellung, zum anderen auf mangelnde Programmierkenntnisse (Code verstehen, Code realisieren) zurückzuführen. Es führte dazu, dass teilweise keine funktionierende Lösung zur Durchführung der Messung bereitstand (*R-Planbarkeit* beeinträchtigt) und Messergebnisse vorhandener Lösungen sich stark voneinander unterschieden (*R-Vergleichbarkeit* beeinträchtigt). Also wurden Punkte zur Erweiterung des Programmgerüsts deutlich markiert und die Erweiterungen inhaltlich auf grafische Dialoge beschränkt. Weiterhin wurden die Studierenden angewiesen, unfertige Umsetzungen bis zum Start der folgenden Veranstaltung selbstständig in Projektarbeit umzusetzen. Diese Maßnahmen führten aber nur zu mäßigem Erfolg, d.h. nach wie vor hatten nicht alle Gruppen die Aufgabe bis zum Start der folgenden Veranstaltung fertiggestellt, sodass nicht gleichzeitig mit den Messungen begonnen werden konnte (*R-Planbarkeit* beeinträchtigt). Als Konsequenz wurde die Ausgabe einer Musterlösung beschlossen, um die Studierenden vor dem zweiten Abschnitt der Aufgabe alle zu einem vorhersagbaren Zeitpunkt in die gleiche technische Ausgangssituation zu versetzen.

### 3.2 Zweiter Abschnitt: Durchführung der empirischen Aufgabe

Im zweiten Abschnitt wurden die Studierenden zunächst in die für die Durchführung der Benutzerstudie relevanten Grundlagen zur Evaluation eingeführt (z.B. Beispiele für den Einfluss von Störvariablen). Anschließend erhielten alle Gruppen die Muster-Implementierung und Informationen zur deren Konfiguration. Es folgte ein Briefing bezüglich der Durchführung des Ablaufs der Messungen.

Zur Untersuchung des Einflusses der Mauszeigerbeschleunigung wurden die Studierenden in zwei Gruppen eingeteilt mit variierender Aufgabenstellung: Studierende im Studiengang Medieninformatik sollten die von Microsoft Windows realisierte Option „Zeigerbeschleunigung verbessern“, welche den Umfang der aus der Bewegung der Maus resultierenden Bewegung des Mauszeigers manipuliert, deaktivieren. Studierende im Studiengang Informatik sollten die Option aktivieren mit den Default-Einstellungen.

Im Folgenden wurde jeder Studierende angewiesen, für jede der vier Konfigurationen (mit/ohne Wechsel der Ruheposition, mit/ohne Precuig) 20 Messungen durchzuführen (d.h. 4x 20 Zeigeaufgaben) und so die Grundlage der späteren Auswertung zu schaffen. Vor der eigentlichen Messung in einer Konfiguration sollte ein Training mit 20 Messungen durchgeführt werden, um den sich selbst messenden Studierenden an den jeweiligen Aufbau zu gewöhnen. Danach wurden die Aufzeichnungen des Experiments von den Dozenten eingesammelt und konsolidiert.

Ergebnis waren für jede der beiden Varianten des Experiments vier Messreihen mit allen Messungen der jeweiligen Konfiguration. Da die Gruppen nicht zeitgleich arbeiteten, war unter Umständen eine Unterbrechung der Bearbeitung erforderlich. Betroffene Gruppen wurden angewiesen, nach Abschluss der Messungen mit der praktischen Bearbeitung des folgenden Abschnitts der Vorlesung zu beginnen.

Nach Abschluss der Konsolidierung sollten die Studierenden die Messreihen ihres Studiengangs mithilfe einfacher Analysefunktionen des Standardprogrammes Microsoft Excel auswerten. Die Studierenden wurden in die Anwendung der linearen Regression eingewiesen und führten diese mit den erhobenen Datensätzen durch. Hierzu wurden die Formeln nach Fitts [Fi54], MacKenzie [Ma92], sowie Accot & Zhai [AZ03] herangezogen. Im Zuge der Analyse wurden die Studierenden mündlich und über ein Handout auf zu untersuchende Fragestellungen hingewiesen. Schließlich wurden die von den Studierenden getätigten Beobachtungen und ermittelten Zusammenhänge diskutiert.

#### *Erfahrungen aus vergangenen Semestern*

In den vorigen Iterationen traten durch die Uneinheitlichkeit der Lösungen seitens der Studierenden verschiedene Schwierigkeiten bei der Auswertung auf. Teilweise wurden Messungen teilweise mittels privater Rechner außerhalb des Labors mit einer Vielfalt an Zeigegeräten durchgeführt, z.B. Touchpads. Zudem konnte die Einhaltung des Ablaufs (insbesondere Ablauf und Umfang des Trainings vor der eigentlichen Messung) nicht sichergestellt werden. Den übermittelten Messwerten waren diese Aspekte aber nicht zu

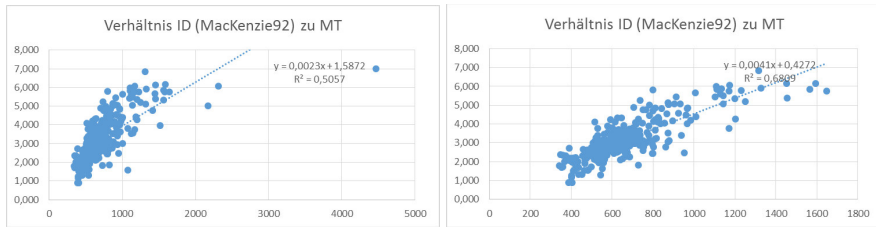


Abb. 2: Beispielhafte Visualisierung der Schwierigkeit (ID) gegenüber der Problemlösungsdauer (MT) nach Fitts' Formel (Erweiterung von MacKenzie); links die Visualisierungen mit Ausreißern durch fehlerhafte Messungen, rechts ohne diese

entnehmen, so dass die Vergleichbarkeit der Werte im Gesamtkorpus nicht gewährleistet war und somit die Aussagekraft der in der Auswertung ermittelten Ergebnisse potentiell beeinträchtigt wurde (vgl. *R-Vergleichbarkeit*). Als Konsequenz wurde die Kontrolle des Ablaufs der Messungen in späteren Iterationen deutlich verstärkt.

Einigen Gruppen wurde aufgrund unzureichender Softwaretests erst nach der Durchführung bewusst, dass das Format der protokollierten Messergebnisse von dem zu erzeugenden Zielformat abwich. Teilweise mussten damit umfangreichere Datenkonvertierungen ausgeführt werden, in Einzelfällen musste die Datenerhebung erneut durchgeführt werden. Zusammengefasst führten diese Verzögerungen dazu, dass ein weiteres Laborpraktikum benötigt wurde (in Konflikt mit *R-Umfang*). Auch diese Problematik wurde später durch die ausgegebene Musterlösung adressiert.

### 3.3 Ergebnisse

An dem Experiment nahmen 21 Informatik-Studierende und 16 Medieninformatik-Studierende teil. Insgesamt standen also 2960 Datensätze zur Auswertung zur Verfügung, welche nach Microsoft Excel zu importieren waren und per Punktdiagramm visualisiert wurden.

Zunächst sollten die Studierenden die Daten bereinigen. Schriftliche Instruktionen luden sie dazu ein, nach Indizien und Ursachen von Ausreißern zu suchen und deren Auswirkung auf den Grad der Annäherung zu prüfen. Dies diente als Basis einer Diskussion, was legitime bzw. nicht legitime Argumentationen zur Eliminierung von Ausreißern sind. Aufgrund der Diskussion mit den Studierenden wurde beschlossen, sämtliche Datensätze zu eliminieren, bei denen ein Fehler protokolliert wurde. Dies führte zu einer deutlichen Verbesserung des Bestimmtheitsmaßes  $R^2$  (s. Abb. 2) und konnte den Studierenden damit als eindruckliches Beispiel dienen.

Danach wurden die Studierenden aufgefordert, die Werte von  $R^2$  in Abhängigkeit von Konfiguration und Formel zu vergleichen. Die Formeln von MacKenzie und Accot & Zhai erreichten eine ähnliche Annäherung und übertrafen Fitts (vgl. Tab. 1). Mit den



Studierenden wurden diese Beobachtungen vor dem Hintergrund des für die Messungen verwendeten Aufbaus diskutiert (z.B. Dimensionalität, kreisförmige Ziele, Maus als Zeigegerät).

Mauszeiger- Beschl.	Veränderliche Ruheposition	Precuing	R <sup>2</sup> Fitts	R <sup>2</sup> MacKenzie	R <sup>2</sup> Accot	MT $\bar{\phi}$
			0,626	0,681	0,690	726,322
		X	0,456	0,478	0,481	542,728
	X		0,447	0,525	0,532	797,306
	X	X	0,491	0,534	0,539	672,925
X			0,553	0,579	0,579	729,798
X		X	0,455	0,466	0,465	556,569
X	X		0,346	0,352	0,352	820,145
X	X	X	0,481	0,502	0,501	648,819

Tab. 1: Zusammenfassung des für die verschiedenen Konfigurationen ermittelten Bestimmtheitsmaßes R<sup>2</sup> sowie der durchschnittlich gemessenen Zeiten (MT  $\bar{\phi}$ )

Zudem konnten die Studierenden den durchschnittlich gemessenen Zeiten Hinweise zum Einfluss der initial genannten Variablen auf die Schwierigkeit einer Zeigeaufgabe entnehmen, was als Stimulus für eine vertiefende Diskussion mit den Dozenten diente.

- **Mauszeigerbeschleunigung:** Die durchschnittlich gemessenen Zeiten vergleichbarer Konfigurationen mit und ohne Mauszeigerbeschleunigung weichen nur gering von einander ab.
- **Vorwissen bei einer Zeigeaufgabe:** Die durchschnittlich gemessenen Zeiten waren unabhängig von der Konfiguration kleiner als einer vergleichbaren Aufgabe ohne Vorwissen – was eine Vereinfachung der Zeigeaufgabe andeutet.
- **Wechselnde Ruheposition der Hand:** Die durchschnittlich gemessenen Zeiten waren unabhängig von der Konfiguration größer als einer vergleichbaren Aufgabe ohne Vorwissen – was eine Erschwerung der Zeigeaufgabe andeutet.

### 3.4 Grenzen der Aussagekraft

Die im Rahmen der Laborpraktika getroffenen Beobachtungen sind in ihrer Aussagekraft begrenzt. Mögliche Auswirkungen des vorgestellten Lehrkonzepts auf den Lernerfolg der Studierenden wurden bislang nicht explizit überprüft. Bezüglich der Beobachtungen zu Fitts' Gesetz ist zu beachten, dass die Durchführung der Messungen auch in der letzten Iteration in einer nur eingeschränkt kontrollierten Umgebung durchgeführt wurde, die zudem in verschiedenen Parametern von Fitts' ursprünglich verwendetem Aufbau abweicht. Die Signifikanz der Beobachtungen wurde dementsprechend bislang nicht weiter überprüft.

## 4 Schlussbemerkungen

Die Untersuchung von Fitts' Gesetz auf Basis des in diesem Artikel vorgestellten Lehrkonzepts ermöglicht die Verbindung praktischer, programmtechnischer Aufgabenstellungen mit der Anwendung empirischer Methoden. Ziel ist die Vermittlung einer ganzheitlichen Sicht auf ein Thema der Mensch-Computer-Interaktion, die für Studierende verschiedene Aspekte eines Softwareentwicklungsprozesses erlebbar macht.

Der über mehrere Iterationen immer weiter entwickelte Ansatz orientiert sich an Rahmenbedingungen der Hochschullehre. Neben organisatorischen Anforderungen (u.a. Planbarkeit, Anteil an der Lehrveranstaltung) erwies sich insbesondere die Beteiligung Studierender am Aufbau eines Korpus aus empirischen Messungen als fordernd, der sich in Umfang und Qualität für darauf aufbauende Analyseaufgaben eignet.

Die derzeit im Einsatz befindliche Variante erwies sich aus Sicht der betreuenden Hochschullehrer als praktikabel aus organisatorischer Sicht. Als günstig aus Sicht der heterogenen Programmierkenntnisse der Studierenden erwiesen sich dabei beispielsweise eine sehr starke Fokussierung der programmtechnischen Aufgaben, sowie im Sinne der Planbarkeit die Ausgabe einer Musterlösung.

Nächste Schritte umfassen unter anderem die empirische Überprüfung des Lernerfolgs seitens der Studierenden im Vergleich zu herkömmlichen Methoden – insbesondere der zuvor praktizierten Verbindung von Vorlesung und Laborpraktika. Kann auch hier ein positiver Effekt nachgewiesen werden, ist geplant das Konzept auf weitere Lehrinhalte der Mensch-Computer Interaktion (z.B., Hicksches Gesetz [Hi52], KLM/GOMS [CMN80]) zu übertragen.

Das Lehrkonzept selbst ist noch nicht abschließend entwickelt und bietet Raum für verschiedene Erweiterungen. Potential im Sinne des forschenden Lernens besteht beispielsweise in der stärkeren Einbindung der Studierenden in die Entwicklung von Hypothesen und entsprechenden Tests. Fordernd bleibt die Integration in die zugrundeliegende Lehrveranstaltung; Auswirkungen auf die Modulkonzeption werden derzeit geprüft.

## Literaturverzeichnis

- [AZ03] Accot, J., Zhai, S.: Refining Fitts' law models for bivariate pointing. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, S. 193–200, 2003.
- [Bu70] Bundesassistentenkonferenz: Forschendes Lernen – Wissenschaftliches Prüfen. Schriften der Bundesassistentenkonferenz 5. Bonn.
- [CMN80] Card, S.K., Moran, T.P., Newell, A.: The keystroke-level model for user performance time with interactive systems. Communications of the ACM. 23, S. 396–410, 1980.

- [Er17] Erlandson, R.: Fitts' Law Game System for Teaching Accessible Design, <http://webpages.eng.wayne.edu/etl/design/fittsgame.html>, Stand: 23.06.2017.
- [Fi54] Fitts, P.M.: The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology* 47(6), S. 381–391, 1954.
- [Gr96] Greenberg, S.: Teaching human computer interaction to programmers. *Interactions*. 3, S. 62–76, 1996.
- [Hi52] Hick, W.E.: On the rate of gain of information. *Quarterly Journal of Experimental Psychology*. 4, S. 11–26, 1952.
- [HHS09] Huber, L., Hellmer, J., Schneider, F. (Hg.): *Forschendes Lernen im Studium*. Bielefeld: Universitätsverlag Webler, 2009.
- [In17] Interaktive Systeme Fakultät Informatik, Technische Hochschule Nürnberg, [https://fwpf.in-tra.th-nuernberg.de/fwpf/Modul\\_Fach.asp?TMod\\_ID=703](https://fwpf.in-tra.th-nuernberg.de/fwpf/Modul_Fach.asp?TMod_ID=703), Stand: 25.04.2017.
- [HH13] Hertzum, M., Hornback, K.: The effect of target precuing on pointing with mouse and touchpad. *International Journal of Human-Computer Interaction*. 29, S. 338–350, 2013.
- [Ma92] MacKenzie, I.S.: Fitts' Law as a Research and Design Tool in Human-Computer Interaction. *Human-Computer Interaction*. 7, S. 91–139, 1992.
- [Ma95] MacKenzie, I.S.: Input devices and interaction techniques for advanced computing. *Virtual environments and advanced interface design*, S. 437–470, 1995.
- [Ma96] Mazur, E.: *Peer Instruction: A User's Manual*. Pearson, 1996.
- [MB92] MacKenzie, I.S., Buxton, W.: Extending Fitts' law to two-dimensional tasks. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, S. 219–226, 1992.
- [Pl06] Plimmer, B.: A computer science HCI course. In: *People and Computers XIX—The Bigger Picture*. Springer, S. 185–199, 2006.
- [SS91] Sears, A., Shneiderman, B.: High precision touchscreens: design strategies and comparisons with a mouse. *International Journal of Man-Machine Studies* 34, S. 593–613, 1991.
- [Si12] Simons, G.J.: Teaching and Visualizing Fitts' Law, 2012. Online: <https://www.semanticscholar.org/paper/Teaching-and-Visualizing-Fitts-Law-Simons-Snodgrass/c122f360c244609500bbb2703be661cc04c0759b>, Stand: 23.06.2017.
- [Top05] Topping, K.J.: Trends in peer learning. *Educational psychology* 25, S. 631–645, 2005.