QualityCrowd

Crowdsourcing für subjektive Videoqualitätstests

Clemens Horch

Bachelorarbeit

QualityCrowd

Crowdsourcing für subjektive Videoqualitätstests

Clemens Horch

15. April 2011

Lehrstuhl für Datenverarbeitung
Technische Universität München

Clemens Horch. *QualityCrowd. Crowdsourcing für subjektive Videoqualitätstests.* Bachelorarbeit, Technische Universität München, München, 2011.

Betreut von Prof. Dr.-Ing. K. Diepold und Dipl.-Ing. Christian Keimel; eingereicht am 15. April 2011 bei der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München.

Lehrstuhl für Datenverarbeitung, Technische Universität München, 80290 München, `http://www.ldv.ei.tum.de`.

Abstract

Despite continuing research on the development of better quality metrics, subjective tests are still indispensable for the assessment of video quality. These tests are both time-consuming and expensive and require installing a suitable laboratory that fulfills the corresponding ITU recommendations. In this thesis the use of crowdsourcing in conjunction with the internet-based performing of such tests shall be examined comparing the results of such a test and the results of conventional laboratory tests.

For performing this test the web-based software *QualityCrowd* was developed, which allows the simple planning and conducting of subjective tests. The software uses Amazon's crowd-sourcing platform *Mechanical Turk* to assign the assessment of the videos to the crowd. Amazon provides the infrastructure for distibuting large numbers of almost any task and paying the workers afterwards.

Another aspect is the evaluation of the technical issues that arise from an internet-based video test. In particular, the problems concerning the compression, delivery and playback of the videos in the participants' browsers are discussed. After considering the various possibilities, a decision in favour of lossless compression using *H.264/AVC* and playback with Adobe's *Flash Player* is taken.

The gathered data show very high correlation with the data from the laboratories they are compared with. Although there are also some significant deviations, the results in general are quite promising and indicate the suitability of the use of crowdsourcing for subjective video tests. Even though the test could not be conducted publicly and the workers be paid, the costs of a test like this one are estimated. It shows that – compared to conventional laboratory tests – a clear cut in costs can be achieved.

Zusammenfassung

Subjektive Tests zur Messung von Videoqualität sind – trotz intensiver Forschung zur Entwicklung besserer Metriken – nach wie vor unabdingbar. Diese Tests sind im Regelfall sowohl zeitaufwendig als auch kostenintensiv und erfordern die Einrichtung eines Testraums nach den entsprechenden ITU-Normen. Eine neue Idee ist der Einsatz von Crowdsourcing in Verbindung mit der internetbasierten Durchführung solcher Tests. In der folgenden Arbeit wird untersucht, inwiefern auf diese Weise durchgeführte Tests mit konventionellen Labortests vergleichbare Ergebnisse liefern können, obwohl die strenge Einhaltung der Testbedingungen und -methodiken im Internet naturgemäß nicht gewährleistet ist.

Für die Durchführung der Qualitätstests wurde die webbasierte Software *QualityCrowd* entwickelt, die eine einfache Planung und Abwicklung solcher Tests ermöglicht. Die Software verwendet für die Durchführung der Videotests Amazons Crowdsourcing-Plattform *Mechanical Turk*. Dieser Dienst stellt eine Infrastruktur zur Verfügung, die es ermöglicht, eine große Zahl von nahezu beliebigen Aufgaben von Menschen im Internet bearbeiten zu lassen. Darüber hinaus ist ein System zur Vergütung der geleisteten Arbeit angeschlossen.

Ein weiterer Aspekt ist die Evaluation der technischen Möglichkeiten für solche Tests, insbesondere im Hinblick auf Videokompression, Auslieferung und Wiedergabe der Testsequenzen im Webbrowser des Teilnehmers. Nach ausführlicher Betrachtung der möglichen Alternativen wird die Entscheidung für die verlustfreie Kompression unter der Verwendung von *H.264/AVC* und die Darstellung der Videos durch den Adobe *Flash Player* getroffen.

Die im Rahmen dieser Arbeit erhobenen Ergebnisse weisen eine sehr hohe Korrelation mit ebenfalls vorliegenden Vergleichsergebnissen aus herkömmlicher Testpraxis auf; es wurden allerdings auch einige signifikante Abweichungen festgestellt. Die trotzdem insgesamt vielversprechenden Ergebnisse lassen auf die grundsätzliche Eignung des Einsatzes von Crowdsourcing für derlei Videotests schließen. Obwohl im Rahmen dieser Arbeit auf eine öffentliche und bezahlte Durchführung des Test verzichtet werden musste, konnten die Kosten des Test abgeschätzt werden. Es zeigt sich dabei, dass die erhoffte deutliche Kosteneinsparung tatsächlich erreicht werden kann.

Inhaltsverzeichnis

1. Einleitung

Die visuelle Qualität von digitalem Video ist für viele Anwendungsbereiche von großem Interesse. Die gesamte Kette der Videoverarbeitung von Aufnahme über Bearbeitung, Speicherung, Übertragung bis hin zur Wiedergabe basieren oftmals auf verlustbehafteter Komprimierung der Videodaten. Andernfalls wäre eine Handhabung der anfallenden Datenmengen auch mit heute zur Verfügung stehender Technologie undenkbar.

Für die Messung von Videoqualität existieren einige mathematische Metriken wie das *PSNR* (Peak signal-to-noise ratio, Spitzen-Signal-Rausch-Verhältnis). Eines der Hauptprobleme dieser Metriken ist allerdings die fehlende oder unzureichende Berücksichtigung der Eigenschaften des menschlichen Sehsystems, weshalb subjektive Videoqualitätstests nach wie vor erforderlich sind.

Für die Durchführung solcher Tests gibt es eine Reihe genormter Bedingungen für den Testraum, die Testmethode und die einzusetzenden Wiedergabegeräte. Zu nennen ist hier insbesondere ITU-R Rec. BT.500 [13] – dort werden genaue Werte für die Raumbeleuchtung, Eigenschaften des verwendeten Displays, die Sitzposition der Versuchspersonen und auch die genaue Testmethodik beschrieben. Mit diesen Maßnahmen wird versucht, die Vergleichbarkeit der erhobenen Ergebnisse zu gewährleisten.

Abbildung 1.1.: Videolabor nach ITU-R Rec. BT.500

Die exakte Einhaltung dieser Normen ist allerdings nicht trivial und erfordert teils nicht unerhebliche Investitionen. Die Anzahl der gleichzeitigen Teilnehmer an einem solchen Test ist oft aufgrund der nicht in beliebiger Größe verfügbaren Referenzdisplays auf zwei beschränkt, was die Durchführung recht zeitaufwendig werden lässt. Darüber hinaus bindet die Betreuung und Einweisung der Teilnehmer sowie die Organisation den Versuchsleiter oft für mehrere Tage.

Eine Idee zur Verbesserung dieser Situation besteht in der Verlagerung eines solchen subjektiven Videotests in das World Wide Web unter dem Einsatz von Crowdsourcing. Damit würde die Einrichtung eines teuren Labors entfallen, und auch Aufgaben wie die Betreuung

und Beaufsichtigung der Teilnehmer könnten in ihrem zeitlichen Umfang deutlich reduziert werden.

Nachdem wichtige Begriffe wie Crowdsourcing erklärt sind, wird zunächst der Markt für Crowdsourcing-Dienstleister betrachtet, um einen passenden Anbieter auszuwählen. Anschließend werden die technische Möglichkeiten für Videotests im Internet beleuchtet, wobei insbesondere die Frage der Datenkompression, aber auch die der einzusetzenden Software von Bedeutung ist. Im dritten Kapitel wird dann die konkrete Durchführung eines Videotests im Internet beschrieben. Dabei wird zunächst auf die für diesen Test entwickelte Software eingegangen und anschließen der genaue Ablauf des Tests erläutert. Abschließend werden die Ergebnisse dieses Tests statistisch ausgewertet und mit vorhandenen konventionell erhobenen Ergebnissen verglichen.

Der offensichtliche Nachteil eines internetbasierten Tests liegt in den nahezu unkontrollierbaren Versuchsumgebung. Deshalb soll vor allem untersucht werden, ob diese Verschlechterung der Bedingungen tatsächlich signifikante Einflüsse auf die erzielten Ergebnisse hat. Weiterhin soll evaluiert werden, ob die erhoffte Kosteneinsparung tatsächlich erreicht werden kann.

2. Voraussetzungen

2.1. Grundbegriffe

2.1.1. Crowdsourcing

Eines der großen Internet-Themen der letzten Zeit ist das sogenannte Crowdsourcing. Dieser Begriff wurde erstmals 2006 von Jeff Howe in einem Artikel des amerikanischen Technologie-Magazins *Wired* [11] verwendet. Er sieht den Begriff als Weiterentwicklung des seit den 1980er Jahren bekannten Outsourcings, also der Auslagerung von Dienstleistungen oder ganzen Unternehmensstrukturen an Drittunternehmen.

Howe sieht nicht zuletzt aufgrund der immer kostengünstiger und in steigender Qualität verfügbaren Produktionsmittel die bisherigen Grenzen zwischen Amateuren und professionellen Anbietern verschwimmen. Technologien wie der Personal Computer oder digitale Foto- und Videokameras sind weitestgehend flächendeckend verfügbar und für einen Großteil der Bevölkerung auch erschwinglich. Die Qualität dieser Consumergeräte und deren Ergebnisse unterscheiden sich oft nicht mehr wesentlich von den mit professionellem Equipment erstellten Ergebnissen. Dies gilt sicherlich heute in noch stärkerem Maße als im Jahr 2006. Zusammen mit der mittlerweile großen Verfügbarkeit von Internetzugängen ermöglicht dies, gewisse Tätigkeiten an die ‚Crowd' – also potentiell die Gesamtheit aller Internetnutzer weltweit – zu delegieren.

Für den Einsatz von Crowdsourcing finden sich viele Beispiele: ein besonders prominentes, das gleichzeitig das enorme Potential von Crowdsourcing demonstriert, ist sicherlich Wikipedia. Seit dem Start der von den Nutzern selbst erstellten Online-Enzyklopädie im Jahr 2001 sind in freiwilliger und weltweit gemeinschaftlicher Arbeit über 18 Millionen Artikel in rund 260 Sprachen entstanden, und ein Ende dieses Wachstums ist derzeit nicht abzusehen. Aber auch das deutlich ältere Modell der Open-Source-Software lässt sich grob dem Begriff des Crowdsourcings unterordnen, auch hier arbeitet eine große Zahl von Menschen an einer Aufgabe, wobei theoretisch jeder nur einen kleinen Teil beiträgt.

2.1.2. *Humans as a Service* (*HuaaS*)

Ein anderes Schlagwort dieser Zeit ist Cloud-Computing. Ein Versuch, diesen schwer zu fassenden Begriff genauer aufzuschlüsseln, führt zum sogenannten *Everything-as-a-Service*-Paradigma (kurz *XaaS*). Man unterscheidet dabei gemeinhin vor allem *Software as a Service* (*SaaS*), *Platform as a Service* (*PaaS*) und *Infrastructure as a Service* (*IaaS*) als die Zurverfügungstellung von IT-Dienstleistungen auf unterschiedlichen Abstraktionsebenen. Überträgt man dieses Konzept auf die menschliche Arbeitskraft und Crowdsourcing kommt man zu *Humans as a Service* oder kurz *HuaaS*. Analog zu den anderen, XaaS-Konzepten wird auch

hier eine Ressource über eine klar definierte Programmierschnittstelle (*API, Application Programming Interface*) zur Verfügung gestellt. Über technische Details der Umsetzung und Fragen der Skalierung und Abrechnung muss sich der Anwender keine Gedanken machen.

HuaaS kann man somit als Kanalisierung der oben beschriebenen, oft auf freiwilliger Basis funktionierenden Crowdsourcing-Erscheinungen gesehen werden. Die in der Masse verfügbare Arbeitskraft wird durch dieses Konzept gebündelt und einfach verfügbar gemacht. Dadurch entsteht eine Dienstleistung, die es ermöglicht, Aufgaben mit Hilfe menschlicher Arbeit zu erfüllen, ohne dabei mit den einzelnen Arbeitern in Kontakt treten zu müssen. Die menschliche Arbeitskraft wird also abstrahiert zur Verfügung gestellt – genau wie beispielsweise *PaaS* Hardware abstrahiert und virtualisiert, um eine wohldefinierte Umgebung für höhere Anwendungen zur Verfügung zu stellen.

An dieser Stelle wird schon deutlich, dass sich dieses Konzept an sich recht gut für subjektive Videotests eignet. Welcher Mensch wo auf der Welt das Video bewertet, ist an sich unerheblich, wichtig ist nur, dass es tatsächlich ein Mensch ist. Die Abstraktion eines *HuaaS*-Anbieters sorgt dabei für die gewünschte organisatorische Vereinfachung. Als wichtigster und wohl auch bekanntester Vertreter auf diesem Markt ist sicherlich Amazon *Mechanical Turk*[1] zu nennen, der im Folgenden näher beschrieben werden soll.

2.2. Auswahl der Crowdsourcing-Plattform

2.2.1. Amazon *Mechanical Turk*

Amazon *Mechanical Turk* wurde im November 2005 von Jeff Bezos, dem Gründer und CEO der Amazon.com, Inc. eröffnet. Der Untertitel „Artificial Artificial Intelligence“ – also in etwa ‚künstliche künstliche Intelligenz‘ – beschreibt gut die Intention des Dienstes. Es sollen vor allem Probleme gelöst werden, die von heutigen Computern und Algorithmen nur sehr schwer oder gar nicht sinnvoll bearbeitet werden können. Ein Mensch kann solche Aufgaben oft mit nur wenig Mühe lösen, und diesen Umstand versucht *Mechanical Turk* auszunutzen. Amazon selbst verwendete laut Pontin [20] die Plattform, um Duplikate in Ihren Produktwebseiten zu finden. Weitere populäre Anwendungsbeispiele sind die Klassifikation von Bildern oder auch kurze Recherchetätigkeiten.

Der Name *Mechanical Turk* bezieht sich auf den 1769 von Wolfgang von Kempelen konstruierten Schachautomaten. Kempelen, ein österreichisch-ungarischer Erfinder und Staatsbeamter, erweckte mit diesem sogenannten „Schachtürken“ beim Zuschauer den Eindruck, der Automat könne selbsttätig Schach spielen. In Wahrheit saß im Inneren des Apparats ein Mensch, der die türkisch gekleidete Figur hinter dem Tisch mechanisch fernsteuerte (Reininger [21]). Mechanical Turk verwendet im Grunde genau dieses Prinzip: es bietet eine ‚Maschine‘ an, die schwierige Probleme löst, im Inneren sitzen allerdings nur Menschen.

Funktionsweise

Die *Mechanical Turk*-Plattform verwaltet die zu bearbeitenden Aufgaben in sogenannten *HIT*s (*Human Intelligence Tasks*). Dies sind kleine und kleinste Aufgaben, die unabhängig

[1] `http://www.mturk.com`

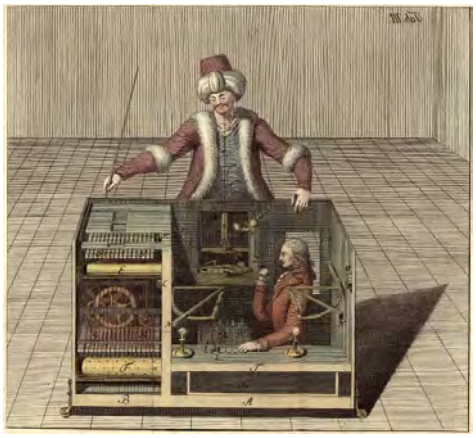

Abbildung 2.1.: Schachtürke, zu Racknitz 1789

voneinander zur bearbeiten sind und meist nur wenige Sekunden oder Minuten Arbeitszeit erfordern. Für jeden dieser HITs legt der Requester einen meist kleinen Geldbetrag, üblicherweise im Bereich einiger Cent, fest, der nach der erfolgreichen Bearbeitung der Aufgabe durch einen Worker ausgeschüttet wird. Amazon selbst behält 10 % dieses Betrags als Gebühr ein. Es besteht die Möglichkeit, einen einmal definierten HIT mit einer großen Zahl an unterschiedlichen Daten durchführen zu lassen. Da es sich bei den meisten Aufgaben nicht um einzelne Fragestellungen handelt, wird davon in der Regel auch Gebrauch gemacht.

Dem Requester steht einerseits eine Weboberfläche zur Definition der *HIT*s zur Verfügung, andererseits ist die Benutzung der Plattform auch vollständig über das angebotene *API* möglich. Es ist sogar so, dass einige Funktionen ausschließlich via API zugänglich sind. Grundsätzlich ist für den Einsatz von *Mechanical Turk* keine weitere eigene technische Infrastruktur nötig, es ist aber auch möglich, externe Webseiten als *HIT*s einzubinden. Je nach Aufgabenstellung eröffnet dies zusätzliche technische Möglichkeiten, wie sie beispielsweise für die hier betrachteten Videotests benötigt werden.

Neben *HIT*s können auch sogenannte *Qualifications* definiert werden, welche ermöglichen, den Kreis der für einen *HIT* zugelassenen Worker einzuschränken. Eine der populärsten Anwendung ist hier wohl eine Altersbeschränkung, die beispielsweise eine Klassifikation von potentiell nicht jugendfreiem Bildmaterial ermöglicht. Es können aber auch eigene Tests definiert werden, die ein Worker absolvieren muss, um eine bestimmte Qualifikation zu erlangen. Ein solcher Test ist sehr ähnlich wie ein *HIT* aufgebaut und könnte beispielsweise vor der Bearbeitung von *HIT*s zur Spracherkennung die Erkennungsrate und damit die sprachlichen Fähigkeiten des Workers überprüfen.

Demographische Daten

Crowdsourcing allgemein und besonders *Mechanical Turk* sind auch Gegenstand derzeitiger Forschung. Vor allem nähere Informationen über die Worker sind bei der Durchführung wissenschaftlicher Experimente von Interesse. Demographische Untersuchungen gibt es bei-

spielsweise von Ipeirotis [12] und Ross et al. [22]. Im Folgenden sollen einige Ergebnisse dieser Studien kurz vorgestellt werden.

Waren bis Ende 2008 noch ca. 80 % der Worker US-Amerikaner, so hat sich bis Anfang 2010 deren Anteil auf ca. 50 % reduziert, dafür sind nun ca. 35 % der Worker indischer Herkunft. Grund hierfür ist vermutlich die in dieser Zeit von Amazon eingeführte Möglichkeit auch Indische Rupien auszuzahlen. Zwei Drittel der amerikanischen Worker sind weiblich, bei den indischen zwei Drittel männlich. Insgesamt sind Männer und Frauen etwa zu gleichen Teilen vertreten.

Das Bildungsniveau der Worker ist in beiden Ländern recht hoch – ungefähr 60 % geben an, mindestens einen Bachelorabschluss zu besitzen. Im Gegensatz dazu ist das Einkommensniveau extrem unterschiedlich. Die Einkommensverteilung der US-Worker ist im großen und ganzen identisch mit der der amerikanischen Bevölkerung, vermutlich etwas geringer. Von den indischen Teilnehmern geben ca. 60 % ein Jahreseinkommen von weniger als 10 000 US-Dollar an, 80 % eines von weniger als $ 20 000. Worker aus beiden Ländern sind eher jung, der Altersdurchschnitt der Amerikaner liegt bei ca. 35, der der Inder bei ca. 25 Jahren.

2.2.2. Mögliche Alternativen

Auf dem Markt für *HuaaS*-Dienstleister gibt es nicht nur Amazons *Mechanical Turk*. Von den Alternativen ist bespielsweise der deutsche Anbieter *Clickworker*[2] zu nennen, der sich vor allem auf die Bearbeitung textbasierter Aufgaben wie Übersetzungen spezialisiert hat. Während Mechanical Turk im Wesentlichen nur seine Plattform zur Verfügung stellt, bietet *Clickworker* auch weitergehende Dienstleistungen wie die gesamte Abwicklung eines Auftrags an. Weitere ähnliche Anbieter sind unter anderem *samasource*[3] oder *mircoWorkers*[4].

Daneben finden gibt es auch Anbieter, die Aufgaben entgegennehmen und an einen oder mehrere der oben genannten Anbieter weiterleiten. Ein Beispeile für einen solchen Meta-Dienst ist der kalifornische Anbieter *CrowdFlower*[5]. Der Vorteil eines solchen Dienstes ist die höhere Reichweite durch gleichzeitige Verwendung mehrerer Plattformen und im Falle von *CrowdFlower* auch ein gegenüber *Mechanical Turk* wesentlich einfacheres *API* und eine benutzerfreundlichere Bedienoberfläche. Der Nachteil liegt vor allem im erheblichen Preisnachteil – *CrowdFlower* verlangt für seine Dienstleistung 33 % der ausgezahlten Beträge.

Von den direkten Anbietern einer Crowdsourcing-Plattform kommt für den hier betrachteten Anwendungszweck eigentlich nur Mechanical Turk in Frage, da sich hier die größte Flexibilität und die vermutlich größte Reichweite findet. Trotz der genaueren Untersuchung von *CrowdFlower* und dessen oben genannten Vorteile erschien der finanzielle Nachteil so groß, dass die Entscheidung schließlich auf *Mechanical Turk* fiel.

[2] `http://www.clickworker.com`
[3] `http://www.samasource.org`
[4] `http://www.microworkers.com`
[5] `http://www.crowdflower.com`

2.3. Auswahl von Playersoftware und Videocodec

2.3.1. Anforderungen

Üblicherweise wird bei Tests zur Beurteilung von Videoqualität unkomprimiertes Videomaterial verwendet. Dies stellt unter Laborbedingungen kein Problem dar, da die erforderliche Hardware problemlos bereitgehalten werden kann und keine bandbreitenbeschränkende Netzwerkverbindung verwendet werden muss. Gerade Letzteres wird bei einer webbasierten Durchführung zum Problem, da die unter Umständen enormen Datenmengen nicht ohne erhebliche Wartezeiten zur Testperson transportiert werden können. Eine verlustbehaftete Komprimierung, wie sie bei der Auslieferung von Video im Internet üblicherweise verwendet wird, kommt hier nicht in Frage, da diese die Testergebnisse verfälschen könnte. Als einziger Ausweg bleibt die verlustlose Komprimierung der Videodaten.

Eine weitere Anforderung ist die möglichst große Verfügbarkeit und Plattformunabhängigkeit der verwendeten Softwarekomponenten. Es soll unter allen Umständen eine möglichst große Basis von Testpersonen erreicht werden. Das schließt ein, dass eine Testperson auch ohne technischen Hürden wie der Installation von Plugins und anderer Software an einem Test teilnehmen können soll. Für die Wiedergabe von in Webseiten eingebetteten Videos gibt es derzeit eine Reihe unterschiedlicher technischer Lösungen, die im Folgenden im Hinblick auf ihre Eignung untersucht werden.

2.3.2. Kandidaten für den Videoplayer

Adobe *Flash Player*

Nach wie vor stellt sicherlich der *Flash Player* von Adobe den de-facto-Standard für die Darstellung von Videos in Webseiten dar. Nicht zuletzt aufgrund von Verbreitungszahlen jenseits der 95 % (laut Adobe Systems Incorporated [4]) ist er nach wie vor nahezu bedenkenlos einsetzbar. Der *Flash Player* unterstützt laut [1, 3] die in Tabelle 2.1 angegebenen Videoformate und -codecs.

Codec	seit Version	Containerformate
Sorenson Spark/H.263	6	SWF, FLV
Screen video	6	SWF, FLV
On2 VP6	8	SWF, FLV
Screen video version 2	8	SWF, FLV
H.264/AVC	9.0.115.0	SWF, F4V, ISO

Tabelle 2.1.: Von Adobe Flash unterstütze Videoformate und -codecs

Die Codecs *Screen Video* und *Screen Video 2* sind primär für die Codierung von Bildschirmaufzeichnungen gedacht. Die einzig verfügbare Dokumentation zu diesen Formaten findet sich offenbar in Adobe Systems Incorporated [2]. Auch scheint die Verbreitung eher gering zu sein und die einzige bekannte aber unvollständige Implementierung abseits von spezieller Software zur Erstellung sogenannter Screencasts, findet sich in FFmpegs *libavcodec*.

Screen Video ist dabei in der Tat ein verlustfreies Verfahren und arbeitet vergleichsweise einfach. Jeder zu codierende Frame wird dabei in eine beliebige Anzahl von Blöcken aufgeteilt. Die Abmessungen eines Blocks müssen dabei jeweils ein Vielfaches von 16 Pixeln, höchstens jedoch 256 Pixel sein. Die Pixel eines Blocks werden mit 24 bit im RGB-Farbraum dargestellt und mit *ZLIB* komprimiert. Key-Frames übertragen jeden Block vollständig, die übrigen Frames nur Blöcke, in denen sich mindestens ein Pixel zum vorhergehenden Frame ändert.

Dieses Verfahren bietet für den gedachten Anwendungsfall, also der Speicherung von Bildschirminhalten, eine recht gute Kompression, da hier typischerweise viele statische Bildinhalte vorhanden sind. Für die Codierung von Videosequenzen ergibt sich im Allgemeinen fast keine Kompression. Erschwerend kommt hinzu, dass die Verwendung von 24 bit RGB zwingend vorgeschrieben ist und daher die Codierung von Material mit Farbunterabtastung (beispielsweise YCbCr 4:2:0) die Datenmenge sogar um einen Faktor von ca. 1,4 vergrößert. Trotz der verlustfreien Codierung ist also Screen Video für den betrachteten Anwendungsfall nicht geeignet.

Screen Video 2 als Verbesserung von *Screen Video* bietet zusätzlich einen weiteren Farbraum jenseits von 24 bit RGB an. Da dieser allerdings nur höchstens 256 verschiedene Farben unterstützt, ist keine verlustfreie Kompression möglich und *Screen Video 2* damit ebenfalls ungeeignet.

Da sowohl *Sorensen Spark/H.263* als auch *VP6* ebenfalls keine verlustfreie Codierung ermöglichen, bleibt als einzige Möglichkeit die Verwendung von *H.264/AVC* und in der Tat ist seit der Einführung des *High 4:4:4 Profile* hier eine verlustfreie Kompression möglich (siehe auch 2.4 auf Seite 18). Aus ungeklärter Ursache konnte allerdings eine fehlerfreie Wiedergabe von verlustfreiem *H.264/AVC* im *Flash Player* erst ab Version 10 bewerkstelligt werden. Da die Verbreitung dieser Version aber ausreichend hoch ist, stellt dies kein größeres Problem dar.

Microsoft *Silverlight*

Einen Konkurrent zu Adobe *Flash* stellt Microsoft *Silverlight* dar. *Silverlight* ist wie auch *Flash* für alle großen Plattformen verfügbar. Für Linux und andere UNIX-basierte Systeme allerdings nur in Form der freien Implementierung *Moonlight*. Die Verbreitung von *Silverlight* liegt derzeit bei ca. 60 %, also deutlich unter der von Adobe *Flash*. Laut Microsoft Corporation [18] unterstützt *Silverlight* in der aktuellen Version ebenfalls *H.264 High Profile* und darüber hinaus auch die Wiedergabe von *YV12 Raw Video*. Aufgrund der deutlich niedrigeren Verbreitung und mangels anderer Vorteile wurde eine Verwendung von *Silverlight* zugunsten des *Flash Players* nicht weiter verfolgt.

HTML5-Video

Im Entwurf der nächsten, fünften, Version der Auszeichnungssprache *HTML* [23] sieht das World Wide Web Consortium (W3C) ein Video-Element vor, das die native Wiedergabe von Videoinhalten in Browser vorsieht, ohne dass diese dazu ein Plugin verwenden müssen. Insbesondere aufgrund der Aussicht, als zukünftiger Standard die derzeit vorherrschende Stellung

von Adobe Flash für das Wiedergeben von Video im WWW abzulösen, wurde diese Technologie ebenfalls ausführlich untersucht. Da der Entwurf des W3C nur das Video-Element an sich spezifiziert, nicht jedoch die zu verwendenen Codecs, müssen die einzelnen Implementierungen der verschiedenen Browserhersteller einzeln betrachtet werden. Tabelle 2.2 gibt eine Übersicht über die Unterstützten Codecs der derzeit größten am Markt verfügbaren Browser.

Produkt	Version	Marktanteil[6]	Codecs
Microsoft Internet Explorer	bis 8	56 %	-
	ab 9	1 %	**H.264**, WebM
Mozilla Firefox	bis 3.0	1 %	-
	ab 3.5	20 %	Theora
	ab 4	1 %	Theora, WebM
Google Chrome	bis 2	0 %	-
	ab 3	11 %	**H.264**, Theora
	ab 6	10 %	**H.264**, Theora, WebM
	später	-	Theora, WebM
Apple Safari	bis 3.0	0 %	-
	ab 3.1	6 %	**H.264**
Opera	bis 10.20	0 %	-
	ab 10.50	2 %	Theora
	ab 10.60	2 %	Theora, WebM

Tabelle 2.2.: HTML5 Video Unterstützung verschiedener Browser

Neben *H.264/AVC* sind hier die Codecs *Theora* und *WebM* vertreten. *Theora* baut als lizenzfreier Codec auf On2 *VP3.2* auf, der Videoteil von *WebM* besteht aus *VP8*. Beide Codecs unterstützen keine verlustfreie Codierung und sind daher nicht geeignet. Es zeigt sich also auch hier nur *H.264/AVC* also einzige mögliche Lösung. Addiert man die Marktanteile der Browser, die derzeit *H.264/AVC* unterstützen, ergibt sich ein Wert von ca. 18 % – ein alleiniger Einsatz dieser Technologie ist also nicht möglich. Anzumerken ist dabei, dass Google am 11.01.2011 bekannt gegeben hat, die Unterstützung von *H.264/AVC* in *Google Chrome* in der Zukunft einzustellen (Jazayeri [16]). Alle bis heute veröffentlichten Versionen dieses Browsers sind davon allerdings noch nicht betroffen.

Standalone Software Client

Ebenfalls eine denkbare Lösung ist die Entwicklung einer speziellen Software, die die Testperson herunterlädt und auf ihrem Rechner installiert. Diese Software kann dann das Laden und Abspielen der Videosequenzen außerhalb des Browsers abwickeln, und mögliche Beschränkungen hinsichtlich der verwendbaren Codecs und Formate fielen damit weg. Diese Lösung

[6] Net Applications Browser Market Share für Februar 2011 `http://marketshare.hitslink.com/browser-market-share.aspx?qprid=0` [aufgerufen am 09.03.2011]

erfordert jedoch die komplizierte Entwicklung eines plattformübergreifend funktionsfähigen Anwendungsprogramms. Darüber hinaus ist aus Gründen der Benutzerfreundlichkeit diese Lösung nicht sinnvoll, und zudem erlaubt Amazon auf seiner Plattform keine *HIT*s, die einen Softwaredownload erfordern [5], so dass diese Lösung insgesamt nicht geeignet ist.

2.3.3. Eingesetzte Lösung

Nach Betrachtung dieser Möglichkeiten fiel die Entscheidung zugunsten einer Mischlösung aus Adobes *Flash Player* und native *HTML5-Video*. Das vereint die fast 100 %-ige Verbreitung des *Flash Players* mit der vermuteten Zukunftssicherheit von HTML5-Video.

Als Codec wird *H.264 High 4:4:4 Profile* in einem *MP4*-Containerformat (MPEG-4 Part 14, ISO/IEC 14496-14) eingesetzt. Dies hat den positiven Nebeneffekt, dass exakt dieselbe Videodatei von beiden Playern problemlos abgespielt werden kann. Die Reduktion der Datenmenge liegt dabei bei den hier verwendeten Videosequenzen zwischen 63 % und 90 % (siehe auch Tabelle B.1 auf Seite 57). Eine ebenfalls technisch denkbare komprimierte Auslieferung der uncodierten YUV-Videorohdaten beispielsweise mittels *gzip* (Kombination aus LZ77 und Huffman-Codierung) bringt im Vergleich dazu eine Datenreduktion von nur 25 % bis 66 %.

Die Auswahl des verwendeten Players geschieht mittels *JavaScript* im Browser der Testperson. Dabei wird die Unterstützung des Browsers für *H.264 High Profile* abgefragt. Fehlt diese, wird der Flash Player geladen. Ist die Version des installierten *Flash* Plugins kleiner der Version 10, wird eine Fehlermeldung angezeigt. Aufgrund der weiter unten erklärten Einschränkungen ist es nicht möglich, verlustfrei kodierte Videos zu erzeugen, die unter Safari abgespielt werden können – sowohl unter Mac OS X als auch unter Windows. Deshalb kommt auch für diesen Browser trotz der theoretischen Unterstützung von HTML5-Video das *Flash*-Plugin zum Einsatz.

Die derzeit immer wichtiger werdenden mobilen Geräte wie diverse Smartphones und Tablets werden von dieser Lösung nicht berücksichtigt. Die teilweise vorhandene Unterstützung für HTML5-Video und *H.264/AVC* scheitert an der Unterstützung des verwendeten *H.264 High 4:4:4 Profile* und den hohen Bitraten, die eine verlustfreie Komprimierung mit sich bringt. Im Test gelang es zwar, ein Video mittels des *Flash Players* auf einem *Android*-basierten Smartphone abzuspielen, diese Option wurde aber nicht weiter verfolgt.

2.4. Videoencodierung

2.4.1. Verlustfreie Videokompression mit *H.264/AVC*

Der internationale Standard *ITU-T Rec. H.264: Advanced video coding for generic audiovisual services* [15] (auch MPEG-4 Part 10, ISO/IEC 14496-10) wurde im Mai 2003 als Nachfolger des ursprünglichen *MPEG-4* Video-Standards veröffentlicht. Die Features des Videocodecs werden zu sogenannten Profilen zusammengefasst, an denen sich schnell die Kompatibilität verschiedener Systeme erkennen lässt. Im ursprünglichen Standard wurden lediglich die Profile *Baseline*, *Main* und *Extended* beschrieben, in Version 3 (März 2005) kamen mit den *Fidelity Range Extensions* (FREXt) diverse *High*-Profile hinzu. Diese *High*-Profile zielen vor allem auf die professionelle Anwendung ab, das *High Profile* wird aber beispielsweise auch bei der

Blu-ray Disk eingesetzt. Das höchste dieser neuen Profile, das *High 4:4:4*-Profil, brachte unter anderem Unterstützung für das 4:4:4-Chromaformat, 12-bit Quantisierung und das hier interessante *Predictive Lossless Coding*. Mit dieser Erweiterung ist also erstmals eine vollständig verlustfreie Kompression möglich.

Mit Version 5 wurde genau dieses Profil im Juni 2006 allerdings wieder aus dem Standard entfernt, um im April 2007 in überarbeiteter Form als *High 4:4:4 Predictive Profile* wieder hinzugefügt zu werden. In der aktuellen Version des *H.264/AVC*-Standards steht also prinzipiell die Option einer verlustfreie Kompression zur Verfügung – allerdings wird diese nur von den wenigsten vorhandenen Implementierungen berücksichtigt.

2.4.2. Implementierung durch *x264*

Für den *H.264/AVC*-Standard gibt es eine Vielzahl an verschiedenen Implementierungen, die meisten davon sind kommerzieller Natur. Daneben existieren mit *x264* auch eine freie Open-Source-Implementierung und mit der *H.264/AVC JM Reference Software* des Fraunhofer Instituts für Nachrichtentechnik eine Referenzimplementierung. Letztere ist aufgrund fehlender Geschwindigkeitsoptimierungen im praktischen Einsatz ohne größere Bedeutung. *x264* dagegen unterstützt einen großen Teil der im Standard beschriebenen Features und damit sogar mehr als die meisten kommerziellen Produkte. Insbesondere das *Predictive Lossless Coding* aus dem *High 4:4:4 Predictive Profile* wird unterstützt.

2.4.3. Implementierungen für die Wiedergabe

Ebenso wie für das Erzeugen von *H.264/AVC*-codierten Videos gibt es auch für deren Wiedergabe eine große Zahl an Softwareimplementierungen. Relevant sind hier die des Adobe *Flash Players* und die der HTML5-Video unterstützenden Browser. Apples *Safari* verwendet für das Dekodieren das *Quicktime*-Framework, welches allerdings höchstens *Main Profile* unterstüzt und somit keine verlustfreien Videos anzeigen kann. *Google Chrome* setzt die quelloffene *libavcodec* aus dem freien FFmpeg-Projekt ein – hier ist die Unterstützung für die verlustfreie Kompression gegeben. Der *Flash Player* unterstützt zwar ebenfalls die Dekodierung von verlustfreiem *H.264/AVC*, allerdings nur solange das mittlerweile wieder aus dem Standard entfernte *High 4:4:4 Profile* verwendet wird.

Ein gut funktionierender Kompromiss ist also die Codierung der Videodaten mittels *x264* in einer alten Version vor der Implementierung des neuen *High 4:4:4 Predictive Profile*. Die so erzeugten Dateien werden sowohl vom *Flash Player* als auch von libavcodec, also *Google Chrome*, fehlerfrei wiedergegeben. Die in A.4 auf Seite 55 beschriebene Lösung verwendet genau diese Vorgehensweise. Sollte Adobe die Unterstützung für das neuere Profil später per Update nachrüsten, kann einfach auf eine aktuelle *x264*-Version umgeschwenkt werden.

2.5. Auswahl des Testmaterials

Das Ziel dieser Arbeit ist nicht, einen Qualitätstest im eigentlichen Sinne duchzuführen, sondern vielmehr die Übereinstimmung der Testergebnisse der webbasierten Test mit denen herkömmlicher Tests zu untersuchen, um den Einsatz von Crowdsourcing zu evaluieren. Daher

werden Testsequenzen benötigt, für die bereits hinreichend viele herkömmliche Testergebnisse für einen Vergleich vorhanden sind.

Ein Pool solcher Testsequenzen wird in De Simone et al. [9, 10] beschrieben und von der *Ecole Politechnique Fédérale de Lausanne, Multimedia Signal Processing Group* und dem *Politecnico di Milano, Dipartimento di Elettronica e Informazione* zur Verfügung gestellt.

Dieser Pool beinhaltet je 78 Videosequenzen in den Auflösungen CIF (352x288 Pixel) und 4CIF (704x576 Pixel). In Frage kommt aufgrund der zu übertragenden Datenmenge ausschließlich das CIF-Format. Die 78 CIF-Sequenzen setzen sich aus den 6 inhaltlich unterschiedlichen Sequenzen *Foreman, Hall, Mobile, Mother, News* und *Paris* zusammen. Es sind jeweils die verlustbehaftet komprimierte Originalsequenz und 12 fehlerhafte Sequenzen mit verschiedenen Fehlerraten (Packet Loss Rate, *PLR*) aus einer simulierten Übertragung über ein mit Paketverlusten behaftetes Netzwerk. Die Sequenzen sind jeweils 300 Frames, also bei 30 Frames je Sekunde jeweils 10 Sekunden lang.

2.6. Zusätzliche Statistikdaten

Es bietet sich an, neben den eigentlichen Daten zur Videoqualität auch weitere Statistikdaten zu erfassen. Viele dieser Daten fallen aufgrund der webbasierten Testdurchführung ohnehin an, andere lassen sich mit geringem Zusatzaufwand erheben. Zu diesen Daten gehören vor allem Bildschirmauflösung und Softwarekonfiguration (Versionen des Betriebssystems, Browsers und *Flash*-Plugins) des Workers, aber auch die pro Videosequenz benötigte Bearbeitungszeit und in begrenztem Maße auch die geografische Position des Workers. Obwohl die Daten nicht zum eigentlichen Testergebnis beitragen, können Sie doch helfen, die Testumgebung besser zu verstehen. Insbesondere die Bearbeitungszeit ist im Hinblick auf die Festlegung der Bezahlung der Worker von Interesse.

2.7. Bisherige Forschungsergebnisse

Die Idee, Crowdsourcing für wissenschaftliche Experimente einzusetzen, ist an sich nicht neu und wurde schon in diversen Veröffentlichungen untersucht. Dabei finden sich Ergebnisse aus unterschiedlichen Bereichen:

Paolacci et al. [19] haben die generelle Eignung von Crowdsourcing mit *Mechanical Turk* für subjektive Experimente untersucht. In einer Studie wurde die Ergebnisse verschiedener Umfragen zu Entscheidungsverfahren im Bereich der Sozialwissenschaften verglichen. Es zeigte sich, dass die Ergebnisse von *Mechanical Turk* durchaus mit denen herkömmlicher Testverfahren vergleichbar waren. Auch wenn diese Erkenntnisee nicht zwingend auf subjektive Videoqualitätstest übertragbar sind, so kann man immerhin die generelle Eignung der Plattform feststellen.

Thematisch näher an subjektiven Videotests ist die Veröffentlichung Marge et al. [17], die den Einsatz von Crowdsourcing für die Transkription gesprochener Sprache untersucht. Dort konnten im Vergleich zu einer herkömmlichen Transkription durch einen einzelnen Menschen quasi identische Fehlerraten erzielt werden. Interessant ist diese Herangehensweise vor allem

aufgrund der auf ca. 10 % reduzierten Kosten. Es wäre durchaus wünschenswert, ein ähnliches Ergebnis auch für subjektive Videotests zu erzielen.

Chen et al. [7, 8] schließlich befassen sich unmittelbar mit dem Einsatz von Crowdsourcing im Bereich subjektiver Tests. Betrachtet werden hier vier unterschiedliche Experimente – zwei akustische und zwei visuelle. Es geht dabei jeweils um den Vergleich unterschiedlicher Encoder und Bitraten einerseits und um die Einflüsse von Paketverlusten andererseits. Die Versuchspersonen wurden hier allerdings nicht zu einer direkten Bewertung eines Stimulus aufgefordert, stattdessen mussten immer zwei Stimuli verglichen und der qualitativ bessere markiert werden. Die erzielten Ergebnisse sind auch hier recht positiv und der Einsatz von Crowdsourcing für derartige Tests wird – je nach Anwendungsfall – empfohlen; es konnten wieder sehr deutliche finanzielle Einsparungen erreicht werden.

3. Durchführung des Videotests

3.1. Testplattform *QualityCrowd*

3.1.1. Idee

Amazon bietet für die *Mechanical Turk*-Plattform ein Webinterface zur Erzeugung und Verwaltung von *HIT*s an. Diese Oberfläche bietet trotz ihrer vergleichsweise hohen Komplexität nicht alle benötigten Funktionen. Daher war die Erstellung einer eigenen Verwaltungssoftware unumgänglich. Da für die Auslieferung der Videosequenzen an die Testpersonen ohnehin ein *HTTP*-Server benötigt wird, bot sich eine Umsetzung als Webapplikation an. Die Umsetzung dieser Idee liegt mit der *QualityCrowd*-Software vor.

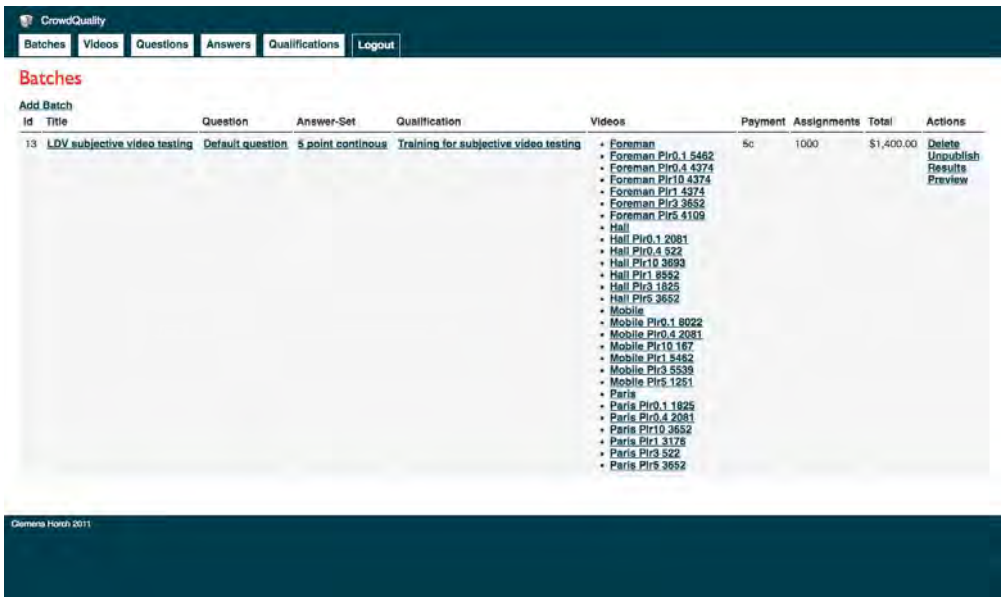

Abbildung 3.1.: QualityCrowd Startseite

Nach der Installation und Konfiguration der Software können die zu testenden Videosequenzen einfach per Browser hochgeladen werden, alle Texte für Fragen, Antwortmöglichkeiten und Hilfestellungen definiert, sowie nach dem Test die Ergebnisse eingesehen werden. Eine Verwendung der Amazon-eigenen Webapplikation ist damit nur noch für das Einzahlen von Guthaben auf das *Mechanical Turk*-Konto notwendig. Die Software ermöglicht so die schnelle und unkomplizierte Durchführung eines subjektiven Videoqualitätstest auf der *Mechanical Turk*-Plattform.

3.1.2. Funktionsweise

Im Folgenden soll kurz auf die Funktionsweise von *QualityCrowd* eingegangen werden. Es wird dabei exemplarisch der Ablauf eines Videoqualitätstests beschrieben, wobei insbesondere verdeutlicht werden soll, wie die unterschiedlichen Komponenten des Systems zusammenarbeiten. In Abbildung 3.2 wird diese Zusammenarbeit vereinfacht dargestellt. Die ausführliche Dokumentation zu *QualityCrowd* befindet sich im Anhang ab Seite 51.

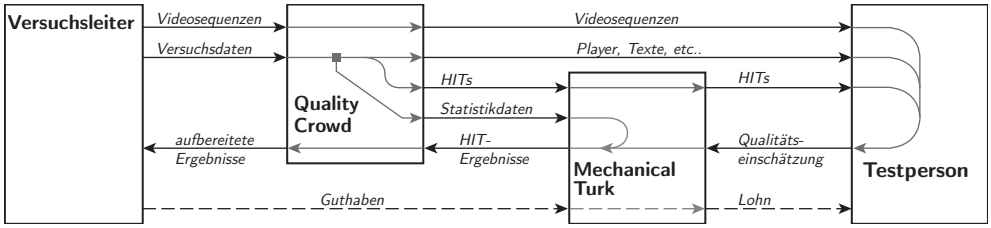

Abbildung 3.2.: QualityCrowd Funktionsweise

Videosequenzen auswählen Der erste Schritt ist stets, die zu testenden Videosequenzen auszuwählen, zu encodieren (siehe auch Anhang A.4 auf Seite 55) und über die Weboberfläche auf den *QualityCrowd*-Server hochzuladen.

Fragen und Antwortmöglichkeiten anlegen Im nächsten Schritt müssen in *QualityCrowd* die diversen Texte für Fragen und Qualifikationstest angelegt werden. Darüber hinaus werden Testmodus und -ablauf konfiguriert.

Test starten Nach Beendigung der Konfiguration werden durch *QualityCrowd* entsprechende *HIT*s und Qualifikationstests in *Mechanical Turk* angelegt.

Test durchführen Ruft ein Worker einen entsprechenden *HIT* in seinem Browser auf, werden die vorher definierten Texte und auch die Videosequenzen samt Player direkt von *QualityCrowd* geladen. Durch diesen Zugriff auf den Server von *QualityCrowd* können zusätzliche Statistikdaten aus den *HTTP*-Headern gewonnen werden. Diese werden nach erfolgter Bearbeitung des HITs zusammen mit der Qualitätseinschätzung der Testperson an *Mechanical Turk* übermittelt.

Ergebnisse abrufen Nach Abschluss des Tests oder auch schon währenddessen kann *QualityCrowd* die Testergebnisse von *Mechanical Tur*k abholen. Anschließend werden sie aufbereitet und dem Versuchsleiter zur Verfügung gestellt.

Bezahlung Für die Bezahlung der Testpersonen muss das Guthabenkonto bei *Mechanical Turk* aufgewertet werden. *Mechanical Turk* zahlt dann automatisch nach dem Einreichen eines HITs den vorher in *QualityCrowd* festgelegten Betrag an den Worker aus.

3.2. Verwendetes Testverfahren

Bei der Durchführung des Test wurde soweit möglich und sinnvoll derselbe Modus verwendet, mit dem auch die Vergleichsergebnisse von De Simone et al. [9] erhoben wurden.

Grundsätzlich wird für diese Untersuchung ein *Single Stimulus*-Verfahren (*SS*) eingesetzt. Dabei sieht die Versuchsperson immer nur eine Sequenz zur gleichen Zeit, ohne dass zusätzlich noch eine Referenzsequenz wie bei einem Double Stimulus-Verfahren gezeigt wird. Das ungestörte Referenzvideo wird allerdings im Test als ein gleichberechtigter Stimulus verwendet. Um die Gesamtdauer des Tests zu beschränken, wurde auf ein Wiederholung der Videosequenzen verzichtet. Vorgesehen ist das einmalige Abspielen und anschließende Bewerten jedes Videos.

Für die Bewertung wird eine fünfstufige, kontinuierliche Skala nach ITU-T Rec. P.910 [14] eingesetzt. Diese ist in *QualityCrowd* durch einen per Maus bedienbaren Schieberegler (siehe Abbildung 3.4) implementiert. Optisch ist dieser der im Vergleichstest verwendeten Skala (siehe Abbildung 3.3) nachempfunden – insbesondere die Beschriftung wurde nicht verändert. Die in Abbildung 3.3 dargestellten Zahlenwerte werden den Testpersonen ebenso wie im Vergleichstest nicht angezeigt und dienen nur der internen numerischen Repräsentation der Bewertungen des Teilnehmers.

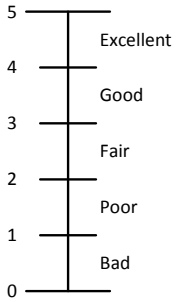

Abbildung 3.3.:
kontinuierliche ITU-Skala

Abbildung 3.4.:
Schieberegler in QualityCrowd

Eine Erweiterung der *QualityCrowd*-Software für andere Testmethodiken wie beispielsweise Double-Stimulus-Verfahren ist denkbar, die Unterstützung für frei definierbare diskrete Bewertungsskalen ist bereits verfügbar.

Insgesamt kann die exakte Einhaltung des genauen Ablaufs und anderer Bestandteile des Testverfahrens nicht garantiert werden. Dies liegt in der Natur der internetgestützten Testdurchführung. Ob diese Ungenauigkeiten im Verfahren allerdings auch zu nachweisbaren Abweichungen bei den Ergebnissen führen, ist gerade der Gegenstand dieser Untersuchung.

3.3. Qualifikationstest

3.3.1. Beschreibung

Vor dem eigentlichen Test wurde ein sogenannter Qualifikationstest durchgeführt. Dabei wurden den Testpersonen fünf Videosequenzen der Serie *News* in zufälliger Reihenfolgen vorge-

führt. Zu jedem Video wurde ein Hinweistext angezeigt, der kurz erläuterte, welche Qualitätsstufe in etwa für dieses Video angemessen sei.

Ziel des Qualifikationstests ist es, den Testpersonen die in etwa zu erwartende Bandbreite an Qualitätsschwankungen und den Einsatz der Wertungsskala zu demonstrieren. Darüber hinaus können die Testpersonen so den Umgang mit Videoplayer und Schieberegler üben. Somit ist dieser Qualifikationstest vergleichbar mit einer Trainingsphase eines herkömmlichen Tests.

Auf die technische Möglichkeit, die Qualitätsbewertung der Videosequenzen im Qualifikationstest auch tatsächlich auszuwerten oder gar zu einer Voraussetzung für die Teilnahme am Test zu machen, wurde hier verzichtet. Für die Teilnahme am Test ist lediglich eine einmalige Teilnahme am Qualifikationstest notwendig. Da die Testpersonen über diese Tatsache nicht informiert wurden, ist trotzdem mit einer gewissenhaften Bearbeitung des Qualifikationstests zu rechnen.

3.3.2. Auswahl der Videosequenzen

Für den Qualifikationstest müssen Videosequenzen verwendet werden, die im eigentlichen Test nicht mehr zum Einsatz kommen, um dessen Ergebnisse nicht zu verfälschen. Aufgrund von technischen Einschränkungen der *Mechanical Turk*-Plattform müssen im Qualifikationstest alle Videos gleichzeitig geladen und untereinander angezeigt werden. Um die Testperson nicht schon vor Beginn des Tests mit extrem langen Ladezeiten abzuschrecken, empfiehlt sich die Auswahl von Sequenzen mit möglichst geringer Dateigröße. Von den zur Verfügung stehenden Videos kommen deshalb allein die Videos der *News*-Reihe (Abbildung 3.3.2 auf der nächsten Seite) in Frage – dort liegen die Dateigrößen alle im Bereich zwischen 4,2 und 4,5 Megabyte. Von den 13 vorhandenen Realisierungen wurden fünf mit unterschiedlichen Fehlerraten ausgewählt, die das Spektrum der auftretenden Bildfehler gut repräsentieren. Insgesamt mussten so für den Qualifikationstest 21,6 MB an Videodaten übertragen werden. Die verwendeten Sequenzen und deren genauen Dateigrößen sind in Tabelle 3.1 aufgeführt.

Sequenz	PLR %	Pattern	Größe MB	Kompressionsfaktor %
News	0	–	4,2	90
	0,4	8253	4,2	90
	1	8297	4,3	90
	3	2081	4,4	90
	10	8297	4,5	90
Summe			21,6	

Tabelle 3.1.: Qualifikationssequenzen mit Dateigrößen und Kompressionsfaktoren

Alle Sequenzen der *News*-Reihe wurden mit *H.264 High Profile* bei einer Bitrate von 283 kbit/s und einem festen Quantisierungsparameter von 31 encodiert. Die übrigen En-

codereinstellungen sind hier nicht weiter von Interesse und können bei De Simone et al. [10] nachvollzogen werden.

Abbildung 3.5.: verwendete Qualifikationssequenz *News*

3.3.3. Testablauf

Der Qualifikationstest wird in *Mechanical Turk* auf einer einzelnen Seite (siehe auch Screenshot in Abbildung B.8 auf Seite 68) durchgeführt. Oberhalb der fünf Videos mit den jeweiligen Hinweistexten und Bewertungsskalen wurde folgender Einleitungstext angezeigt:

> For research purposes the quality of video encoding and transmission shall be evaluated. This qualification test is about showing you what to expect and giving you the chance to get used to the task of rating video quality.
>
> Below you will see some short videos of equal content. Imagine these videos have been transmitted over wireless connections of different quality. Your task is to watch the videos and tell us your opinion of the video quality. As this is a training for the following real tasks, some comments on the videos are presented to show you how we would expect these videos to be rated.
>
> To express your rating, move the red slider to the according position on the scale. Feel free to use the whole scale from top to bottom.
>
> Due to technical reasons the videos might take significantly longer to load than comparable web videos. Unfortunately, this is inevitable for this type of test. If your internet connection is really slow, these tests might not be suitable for you.

Dieser Text soll der Testperson kurz vermitteln, worum es bei dieser Aufgabe geht, ohne zu sehr ins Detail zu gehen, und entspricht etwa der Einweisung des Versuchsleiters vor Beginn eines konventionellen Labortests.

Die gezeigten Sequenzen und die jeweiligen Hinweistexte sind zusammen mit den Mean Opinion Scores des Vergleichstests in Tabelle 3.2 auf der nächsten Seite aufgeführt.

3.4. Durchführung des Tests

3.4.1. Auswahl der Testpersonen

Aus Zeitgründen musste im Rahmen dieser Arbeit auf eine Durchführung des Tests in der realen *Mechanical Turk*-Plattform verzichtet werden. Stattdessen wurde der Versuch in der Entwickler-Sandbox der Plattform durchgeführt. Diese unterscheidet sich weder optisch noch

PLR	Pattern	MOS	Hinweistext
0	–	4,82	This is a video without any transmission errors, so the slider can be pushed to the top end of the scale.
0,4	8253	3,76	This video has some minor transmission errors but "Good" should still be a suitable rating.
1	8297	3,00	Here some transmission errors are clearly visible. As it can get worse put the slider around "Fair".
3	2081	1,67	For this video a rating in the area of "Poor" might be appropriate.
10	8297	1,36	Here you can see a video with heavy errors. We would consider this to be really "Bad" quality.

Tabelle 3.2.: Videosequenzen des Qualifikationstests

in der Bedienung wesentlich von dem Produktivsystem, so dass für den Test dieselben Bedingungen herrschten, wie sie auch bei einer vollständig öffentlichen Durchführung bestanden hätten.

Genaue Informationen über die Zusammensetzung der Versuchspersonen sind nicht vorhanden, da der Test weitgehend anonym ablief. Bei einer öffentlichen Durchführung sind neben der ungefähren geografischen Position auch keine weiteren Daten dieser Art verfügbar. Aufgrund der hier angewandten Rekrutierung durch persönliche Ansprache und der eingeladenen Personen kann von einer großen Vielfalt an Alter, Geschlecht und vorhandenem Fachwissen der Teilnehmer ausgegangen werden. Lediglich die geografische Verteilung der Teilnehmer weicht mit hoher Wahrscheinlichkeit von der real zu erwartenden ab. Auch mögliche Einflüsse der Höhe der Bezahlung der Testpersonen konnten so nicht untersucht werden.

3.4.2. Auswahl der Videosequenzen

Von den insgesamt 78 verfügbaren Videosequenzen wurden insgesamt 28 ausgewählt, um den Test durchzuführen. Dabei wurde von den Reihen *Foreman*, *Mobile*, *Hall* und *Paris* jeweils die ungestörte und sechs Realisierungen mit unterschiedlichen Fehlerraten ausgesucht. Die gewählten Videos und deren finale Dateigrößen sind in Tabelle B.1 auf Seite 57 aufgeführt. Auch hier wurde bei der Auswahl darauf geachtet, dass die zu übertragende Datenmenge ein vernünftiges Maß nicht überschreitet. Aus diesem Grund wurden eben nicht einfach alle Sequenzen getestet, sondern genau diese Auswahl. Einen Überblick über die Inhalte dieser Sequenzen gibt Abbildung 3.4.2 auf Seite 30.

Die originalen Videos wurden zunächst mit *H.264/AVC High Profile* mit der *H.264/AVC JM Reference Software* des Fraunhofer Instituts für Nachrichtentechnik encodiert. Dabei kamen feste Bitraten und ein fester Quantisierungsparameter (*QP*) zum Einsatz. Die jeweils verwendeten Einstellungen zeigt Tabelle 3.3 auf der nächsten Seite. Auch hier sind die exakten Encodereinstellungen nicht weiter interessant und werden bei De Simone et al. [10] genauer erläutert.

Anschließend wurden den *H.264*-Datenströmen mit einer Simulationssoftware Paketfehler

Sequenz	Bitrate kbit/s	QP
Foreman	353	32
Mobile	532	36
Hall	216	32
Paris	480	32

Tabelle 3.3.: Bitraten und Quantisierungsparameter der Testsequenzen

hinzugefügt. Die daraus resultierenden Dateien wurden ohne weitere Bearbeitung zur Verfügung gestellt.

Anschließend mussten die Videos zur weiteren Bearbeitung wieder in YUV-Rohdaten decodiert werden. Dies geschah wieder mit dem JM-Referenz-Decoder. Es folgt schließlich die eigentliche verlustfreie Codierung mit *H.246/AVC High 4:4:4 Profile* unter der Verwendung von *x264*. Nach dem Hinzufügen eines *MPEG4*-Containers sind die Videodaten bereit für die Verwendung in QualityCrowd.

Die gesamte Verarbeitungskette der Videosequenzen ist schematisch in Abbildung 3.6 dargestellt.

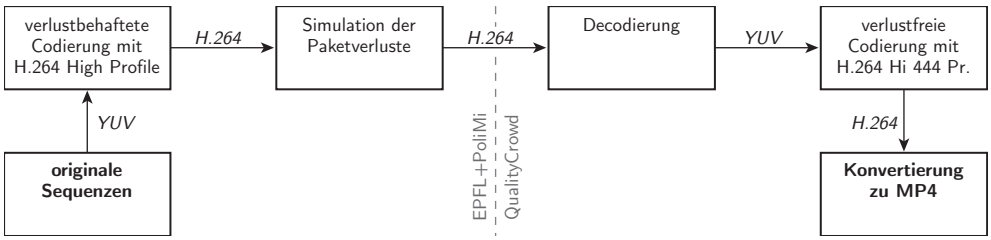

Abbildung 3.6.: Verarbeitungskette der Videosequenzen

3.4.3. Bildschirmansicht

Abbildung B.9 auf Seite 69 zeigt die Bildschirmansicht während der Testdurchführung. Oben steht die schriftliche Testanweisung:

> Your task is to watch the video below and rate its visual quality. Imagine this video has been transmitted over a wireless connection and therefore might suffer from some transmission errors.
>
> Due to the research purposes of this HIT, the video may take quite long to load. Unfortunately, this is inevitable.

Darunter folgt der Videoplayer mit Fortschrittsanzeige und Play-Button. Unterhalb des Players wird die eigentliche Testfrage angezeigt:

> How do you rate the visual quality of the video?

(a) Foreman

(b) Hall

(c) Mobile

(d) Paris

Abbildung 3.7.: verwendete Testsequenzen

Es folgt noch der Schieberegler (siehe Abschnitt 3.2 auf Seite 25) und der Submit-Button, der das Ergebnis abschickt. Um zu verhindern, dass die Testperson ein Ergebnis einreichen kann ohne das Video vorher anzusehen, wird dieser Button erst nach dem ersten vollständigen Abspielen des Videos aktiviert. Vorher trägt dieser die Beschriftung „Watch the video now...“.

3.4.4. Testablauf und Testbedingungen

Nach dem Akzeptieren des *HIT*s wird das Video im Player geladen. Während des Ladevorgangs wird ein Aktivitätsindikator angezeigt, der dem Benutzer den Ladevorgang signalisiert. Nachdem das Video vollständig übertragen wurde, kann die Testperson die Wiedergabe starten. Im Gegensatz zum Vergleichstest kann die Testperson das Video beliebig oft ansehen, das Anspringen bestimmter Stellen im Video („Spulen“) ist allerdings nicht möglich. Der Button für das Absenden der Wertung wird erst nach dem ersten Betrachten aktiviert, damit niemand ohne Ansehen der Sequenz eine sinnlose Wertung abgeben kann. Eine ähnliche Sperre gegen das mehrfache Ansehen wäre zwar technisch ebenfalls möglich, andererseits aber auch leicht durch das erneute Laden der Webseite zu umgehen und wurde daher nicht umgesetzt.

Die Testsequenzen wurden in zufälliger Reihenfolge geladen. Beim Vergleichstest wurde ebenfalls eine zufällige Reihenfolge verwendet, allerdings wurde dort das Aufeinanderfolgen

zweier Sequenzen mit identischem Inhalt ausgeschlossen. Letztere Einschränkung lässt sich mit *Mechanical Turk* nicht realisieren, daher wurde darauf verzichtet.

Die Testperson hatte grundsätzlich unbegrenzt viel Zeit den Test durchzuführen. Im Gegensatz dazu wurde im Vergleichstest nach jeder der zehnsekündigen Videosequenzen ein 3–5 sekündiger Wertungszeitraum eingefügt. Auch diese Bedingung lässt sich in einem Browser nicht sinnvoll herstellen.

Theoretisch denkbar sind eine ganze Reihe von Manipulationen am Testablauf durch den Teilnehmer. Eine Testperson könnte zum Beispiel durch das Anlegen weiterer Konten bei *Mechanical Turk* mehrfach teilnehmen oder gar verschiedene Videosequenzen einem direkten Vergleich unterziehen. Auch die oben beschrieben verzögerte Aktivierung des Submit-Buttons lässt sich technisch leicht umgehen. In der Praxis ist allerdings davon auszugehen, dass derlei Fälle nicht auftreten. Es ist nicht zu erwarten, dass eine Testperson den dafür erforderlichen Aufwand freiwillig und ohne zu erwartenden persönlichen Vorteil auf sich nimmt.

Nicht auszuschließen ist dagegen die Möglichkeit, dass nicht alle Testpersonen auch alle Videos ansehen – und selbst wenn sie das tun, kann eine beliebige Zeitspanne zwischen der Betrachtung der einzelnen Sequenzen liegen. Auch diese Einschränkungen muss man hier in Kauf nehmen. Man steht daher bei der Durchführung eines solchen Test höchstwahrscheinlich vor der Entscheidung, nur vollständige Ergebnissätze zu verwenden oder auch die unvollständigen mit in die Auswertung einzubeziehen. Das ist aber letztlich bei einem herkömmlichen Videotest nicht anders, auch hier kann eine Testperson die Bewertung einiger Sequenzen beispielsweise aufgrund von Unaufmerksamkeit überspringen.

Gänzlich unklar sind die äußeren Umstände und Umweltbedingungen beim Teilnehmer. Das verwendete Display, der Betrachtungsabstand und die Beleuchtungssituation können frei variiert werden, und eine eventuelle Ablenkung durch Musik oder Gespräche sind ebenfalls nicht auszuschließen. Es wird zwar angestrebt, die Videos pixelgenau und ohne Skalierung auf dem Bildschirm anzuzeigen – verwendet der Teilnehmer jedoch die mittlerweile in allen Browsern anzutreffende Zoom-Funktion, wird in den meisten Fällen eine skalierte Version wiedergegeben. Auch hier kann man nur hoffen, dass die wenigsten Personen die entsprechende Funktion tatsächlich nutzen.

4. Testergebnisse und Analyse

4.1. Statistische Analyse der Ergebnisse

4.1.1. Beschreibung der Datensätze

In einem Zeitraum von rund drei Wochen konnten für die $n = 28$ Videosequenzen Qualitätsbewertungen von $N_{MTurk} = 19$ Versuchspersonen erhoben werden. Weitere sieben Personen haben am Test teilgenommen, aber nicht alle Videos bewertet. Die genauen Gründe hierfür sind naturgemäß nicht bekannt, bei drei dieser sieben Teilnehmern liegt allerdings die mittlere Bearbeitungszeit für eine Videosequenz bei über drei Minuten. Eine recht wahrscheinliche Begründung für den Abbruch des Tests könnte also eine zu langsame Internetverbindung und die dadurch bedingten sehr langen Wartezeiten sein. Bei einer Testperson ist dieser Abbruchgrund durch entsprechende E-Mail-Kommunikation belegt. Die übrigen vier Abbrecher zeigen keine Auffälligkeiten bei der mittleren Bearbeitungszeit, hier kann der Grund nur in nachlassendem Interesse vermutet werden.

Alle hier folgenden Analysen werden nur mit dem kleineren Satz der vollständigen Ergebnisse durchgeführt, da sich zeigte, dass durch das Einbeziehen der unvollständigen Ergebnisse keine weiteren Verbesserungen erzielt werden können. Insgesamt werden vier verschiedene Datensätze parallel verarbeitet und anschließend verglichen. Es handelt sich hierbei um

- die hier mittels *Mechanical Turk* erhobenen Ergebnisse (kurz *MTurk*),
- die Ergebnisse der *Ecole Politechnique Fédérale de Lausanne* (kurz *EPFL*),
- die Ergebnisse der *Politecnico di Milano* (kurz *PoliMi*) und
- der Kombination von *EPFL* und *PoliMi* (kurz *EPFL+PoliMi*).

Die Anzahlen der Versuchsteilnehmer bei den Vergleichsergebnissen sind $N_{EPFL} = 17$ und $N_{PoliMi} = 23$.

4.1.2. Datenaufbereitung

Die Aufbereitung und Analyse der Ergebnisdaten orientierte sich eng an der Vorgehensweise von De Simone et al. [10], um eine optimale Vergleichbarkeit der Resultate zu gewährleisten. Im ersten Schritt wurden die Ergebnisdatensätze aus *QualityCrowd* einer Plausibilitätsprüfung unterzogen. Dabei wurden die Wertungen eines Teilnehmers aus den Daten entfernt, da dieser – möglicherweise aufgrund eines technischen Problems – alle Sequenzen mit „Bad“ bewertet hat. Die verbliebenen Rohdaten (siehe Tabelle B.2 auf Seite 58) wurden anschließend in *MATLAB* weiterverarbeitet. Teilweise wurden hierfür die den Vergleichsergebnissen beiliegenden *MATLAB*-Funktionen direkt verwendet.

Abbildung 4.1.: Datenaufbereitung

Zunächst wurden die Daten einer Normalisierung unterzogen. Dazu werden die Ergebnisse je Testperson so verschoben, dass der Mittelwert einer Person und der Mittelwert des gesamten Datensatzes identisch sind. Die korrigierte Bewertung m'_{ij} des j-ten Videos der i-ten Person wird also wie folgt aus dem gemessenen Wert m'_{ij} berechnet:

$$m'_{ij} = m_{ij} - (\overline{m_i} - \overline{m})$$

Anschließend werden alle m'_{ij} kleiner null auf null und alle größer fünf auf fünf gesetzt. Diese Prozedur korrigiert zumindest teilweise die verschiedenen Interpretationen der Bewertungsskala der unterschiedlichen Testpersonen.

Nach der Normalisierung wurden mittels eines Screenings nach ITU-R [13], Rec. BT.500, Annex 2, Abschnitt 2.3.1, mögliche Ausreißer gesucht und aus dem Datensatz entfernt. Dabei geht N in N' über. Tabelle 4.1 zeigt die gefundenen und entfernten Ausreißer.

	N	Ausreißer	N'
MTurk	19	10	18
EPFL	17	2, 10, 11	14
PoliMi	23	2, 20	21
EPFL+PoliMi	40	10, 19, 37	37

Tabelle 4.1.: gefundene Ausreißer

4.1.3. Mean Opinion Score

Nach dieser Vorverarbeitung können weitere Untersuchungen mit diesen Daten vorgenommen werden. Insbesondere ist hier der *Mean Opinion Score* einer Videosequenz von Interesse. Dieser Mittelwert einer Sequenz j berechnet sich folgendermaßen:

$$MOS_j = \overline{m_j} = \frac{\sum_{i=1}^{N} m'_{ij}}{N'}$$

Die Werte für alle vier Datensätze sind in Tabelle 4.2 auf der nächsten Seite aufgeführt, grafisch dargestellt sind sie für *MTurk* in Abbildung 4.2 auf Seite 37 und für *PoliMi* in Abbildung 4.3 auf Seite 38. Entsprechende Abbildungen für die weiteren Messreihen finden sich im Anhang auf den Seiten 61 und 62.

Insbesondere aus den Abbildungen erkennt man die Plausibilität der erhobenen Ergebnisse. Wie auch bei den Vergleichsdaten können die untersuchten Sequenzen anhand der Ergebnisse in eine eindeutige Reihenfolge gebracht werden, die in allen Fällen mit der Fehlerrate der

Sequenz	PLR	Pattern	MTurk	EPFL	PoliMi	PoliMi+EPFL
	0	–	4,2057	4,8232	4,5954	4,6078
	0,1	5462	4,0379	4,3923	4,3080	4,3018
	0,4	4374	3,0012	3,2004	3,5761	3,4424
Foreman	1	4374	2,3101	2,8654	2,7879	2,8494
	3	3652	1,1734	1,7082	1,9128	1,7691
	5	4109	0,7687	0,7005	1,1193	0,9255
	10	4374	0,4836	0,2958	0,6966	0,5152
	0	–	4,2193	4,7387	4,6955	4,6770
	0,1	2081	4,0321	3,9198	4,3224	4,1695
	0,4	522	2,8495	3,2404	3,4455	3,3069
Hall	1	8552	1,9887	2,3876	2,7130	2,6366
	3	1825	0,9504	1,1559	1,4036	1,3405
	5	3652	0,7586	0,9160	1,3336	1,1698
	10	3693	0,6810	0,6692	0,8385	0,7621
	0	–	4,0749	4,8656	4,6759	4,7199
	0,1	8022	3,9751	4,4432	4,4264	4,4099
	0,4	2081	3,2326	3,8079	3,9475	3,9143
Mobile	1	5462	2,6348	2,9434	3,4738	3,2550
	3	5539	1,4917	1,8757	2,0081	1,9594
	5	1251	1,0665	1,2847	1,4149	1,3746
	10	167	0,4941	0,4032	0,9410	0,7038
	0	–	4,1176	4,8619	4,6490	4,7033
	0,1	1825	3,9343	4,3143	4,2681	4,3071
	0,4	2081	3,6581	3,9355	4,1476	4,0526
Paris	1	3176	2,6951	2,9329	3,0452	3,0895
	3	522	0,9437	1,1238	1,7408	1,5280
	5	3652	0,8495	1,1411	1,0359	1,1142
	10	3652	0,5176	0,5520	0,8567	0,6520
Mittelwert			2,3266	2,6249	2,7993	2,7235
Offset zu MTurk				0,2983	0,4726	0,3968

Tabelle 4.2.: Mean Opinion Scores

Videodaten übereinstimmt. Es wurden also auch die teilweise nur schwer sichtbaren Unterschiede zwischen Videos sehr ähnlicher Fehlerrate (beispielsweise mit einer *PLR* von 0,1 und 0,4) offenbar richtig bewertet.

Aus den *Mean Opinion Scores* lässt sich außerdem ein möglicher konstanter Offset zwischen den Datensätzen berechnen. In der Tat tritt hier eine Verschiebung um ca. 0,4 zugunsten der konventionell erhobenen Datensätze zu Tage – mit anderen Worten wurden in *Mechanical Turk* die Videosequenzen durchschnittlich um 0,4 Punkte oder knapp 8 % schlechter bewertet. Ob dieser Offset statistisch signifikant ist, wird im weiteren Verlauf der Analyse untersucht werden.

4.1.4. Streuung der Messwerte

Die Streuung der Messwerte liefert ebenfalls ein wichtiges Indiz für die Zuverlässigkeit der Testmethode. Diese wurde insbesondere unter Verwendung der empirischen Standardabweichung S_j untersucht.

$$S_j = \sqrt{\frac{1}{N'} \sum_{i=1}^{N'} \left(m'_{ij} - \overline{m'_j} \right)}$$

In Tabelle 4.3 sind die arithmetischen Mittel der entsprechenden Werte für alle Datensätze aufgeführt, und man erkennt schnell, dass es keine wesentlichen Unterschiede in den Streuungen der einzelnen Versuchsergebnisse gibt. Vielmehr reihen sich die *MTurk*-Ergebnisse zwischen den Werten der beiden Labors ein.

	MTurk	EPFL	PoliMi	EPFL+PoliMi
Foreman	0,501	0,633	0,460	0,562
Hall	0,490	0,521	0,505	0,507
Mobile	0,583	0,565	0,487	0,547
Paris	0,603	0,526	0,435	0,476
alle	0,544	0,561	0,472	0,523

Tabelle 4.3.: empirische Standardabweichungen der Testergebnisse

In den Abbildungen 4.2 auf der nächsten Seite und 4.3 auf Seite 38 (bzw. B.1 auf Seite 61 und B.2 auf Seite 62) sind die Standardabweichungen als symmetrische Intervalle um den *Mean Opinion Score* aufgetragen und auch hier zeigen sich bei allen Messreihen sehr ähnliche Bilder.

Ebenfalls dargestellt sind die jeweiligen Minima und Maxima. Obwohl diese aufgrund ihrer hohen Empfindlichkeit für Ausreißer statistisch nicht sehr relevant sind, wiederholt sich auch hier dieses Bild. Auffällig ist hier höchstens der extreme Ausreißer bei der Sequenz Paris (0 % *PLR*, *MTurk*). Die Bewertung mit 0,055 ist wohl nur durch eine Fehlbedienung oder ein technisches Problem zu erklären und ist auch der einzige auffällige Wert dieses Teilnehmers.

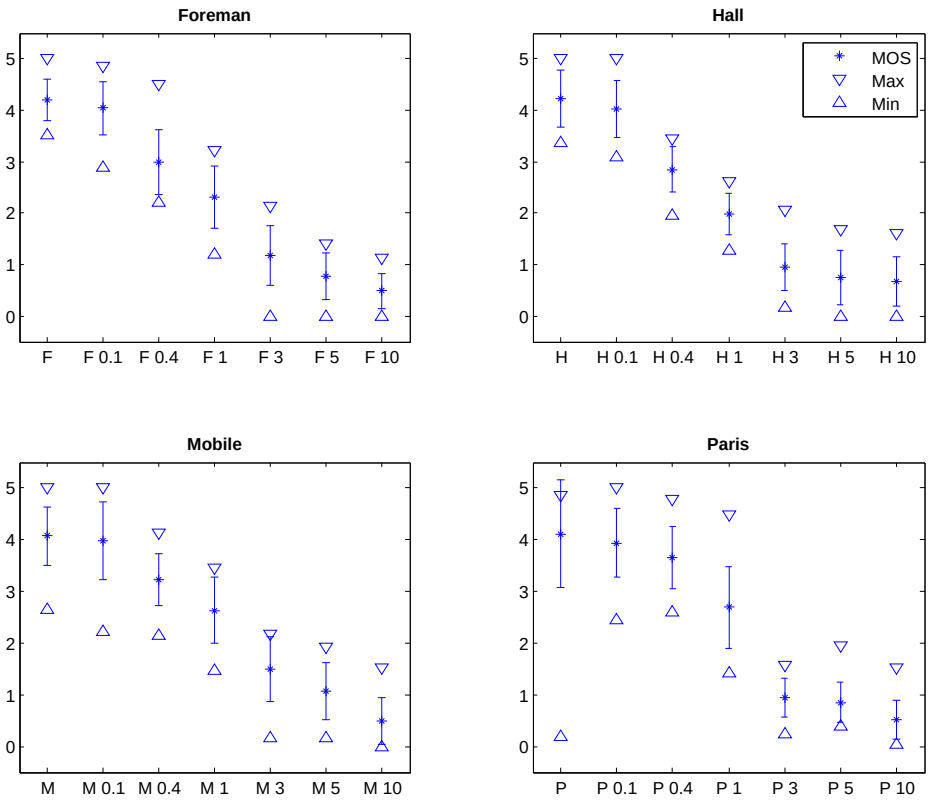

Abbildung 4.2.: *Mean Opinion Scores* und empirische Standardabweichung für *MTurk*

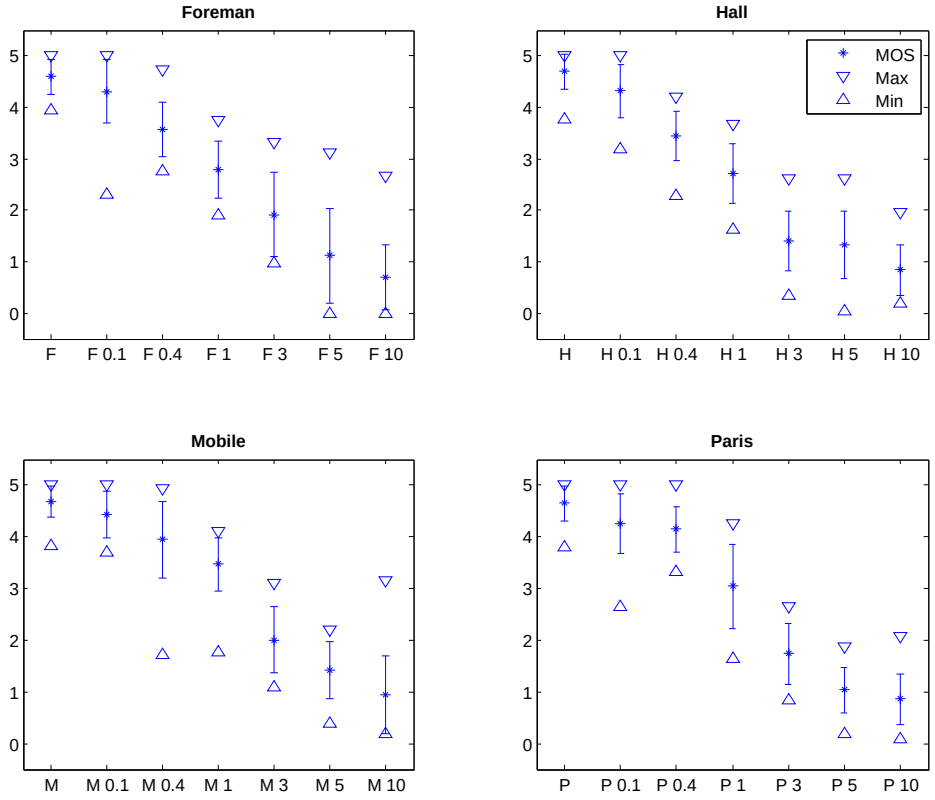

Abbildung 4.3.: *Mean Opinion Scores* und empirische Standardabweichung für *PoliMi*

4.1.5. Korrelation

Schon augenscheinlich erkennt man die recht gute Übereinstimmung der *MOS*-Werte der *MTurk*-Testreihe mit denen der Vergleichsergebnisse. Zur genaueren Untersuchung dieser Übereinstimmung wurden die Pearson-Korrelationskoeffizienten berechnet. Dies geschah jeweils getrennt für die vier verschiedenen Videosequenzen und auch insgesamt für alle Videoinhalte gemeinsam. Der Korrelationskoeffizient r_{12} zwischen den *Mean Opinion Scores* eines Datensatzes (1) mit den Ergebnissen $m_{ij}^{(1)}$ und den *Mean Opinion Scores* eines Datensatzes (2) mit den Ergebnissen $m_{ij}^{(2)}$ berechnet sich im Allgemeinen wie folgt:

$$r_{12} = \frac{\frac{1}{n}\sum_{j=1}^{n}\left(\overline{m_j}^{(1)} - \overline{m}^{(1)}\right)\left(\overline{m_j}^{(2)} - \overline{m}^{(2)}\right)}{\sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(\overline{m_j}^{(1)} - \overline{m}^{(1)}\right)^2} \cdot \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(\overline{m_j}^{(2)} - \overline{m}^{(2)}\right)^2}}$$

Die Werte in Tabelle 4.4 auf der nächsten Seite sind alle größer oder gleich 0,99, somit liegt stets sehr hohe Korrelation vor. Zum Vergleich sind auch die Korrelationskoeffizienten der beiden Vergleichsdatensätze untereinander in der letzten Spalte aufgeführt. Man sieht, dass die Koeffizienten der Ergebnisse *MTurk* betreffend in der selben Größenordnung liegen wie die der beiden herkömmlichen Labortests.

	MTurk EPFL	MTurk PoliMi	MTurk EPFL+PoliMi	EPFL PoliMi
Foreman	0,9899	0,9929	0,9927	0,9949
Hall	0,9901	0,9922	0,9919	0,9955
Mobile	0,9966	0,9948	0,9972	0,9913
Paris	0,9963	0,9925	0,9963	0,9896
alle	0,9920	0,9922	0,9937	0,9918

Tabelle 4.4.: Korrelationskoeffizienten der Testergebnisse

Der Zusammenhang der Ergebnisse zwischen den beiden konventionell durchgeführten Testreihen ist also genauso stark wie der jeder dieser Testreihen mit den online erhobenen Ergebnissen. Diese Tatsache ist wohl der stärkste Hinweis auf eine sinnvolle Einsetzbarkeit des erprobten Crowdsourcing-Verfahrens für derlei Videoqualitätstests.

Auch optisch ist diese hohe Korrelation schnell zu erkennen. Abbildung 4.4 auf der nächsten Seite zeigt in vier Streudiagrammen den Zusammenhang der Mean Opinion Scores von *MTurk* und *EPFL+PoliMi* zusammen mit den entsprechenden Pearson-Korrelationskoeffizienten. Die blaue Winkelhalbierende dient der Orientierung – hier lägen die Punkte kompletter Übereinstimmung.

In dieser Abbildung erkennt man zusätzlich den in Abschnitt 4.1.3 auf Seite 34 festgestellten Offset daran, dass alle roten Punkte oberhalb der Winkelhalbierenden liegen. Es zeigt sich deutlich, dass dieser Offset nicht etwa von zufälligen Schwankungen herrührt. Vielmehr ist ersichtlich, dass sich die roten Punkte um eine parallel um den Offset verschobene Gerade gruppieren. Die übrigen Streudiagramme mit den Zusammenhängen der anderen Datensätze finden sich in den Abbildungen auf den Seiten 63 bis 65.

4.1.6. Konfidenzintervalle

Zur weiteren Bestätigung dieser Übereinstimmung wurden auch noch die Konfidenzintervalle mittels der Student-t-Verteilung auf dem 95 %-Niveau berechnet. Auch hier wurde die Vorgehensweise eng an die von De Simone et al. [10] angelehnt. In Abbildung 4.5 auf Seite 41 sind diese Konfidenzintervalle für die Datensätze *MTurk* und *PoliMi* dargestellt. Man erkennt schnell an der Größe der Intervalle die gute Verlässlichkeit der ermittelten *MOS*-Werte aufgrund der ausreichenden Größe der Stichprobe – sprich, Anzahl der Testteilnehmer.

Anhand der Überlappung der Konfidenzintervalle beider Testreihen lässt sich feststellen, ob die oben errechnete sehr hohe Korrelation auch signifikant ist. Man sieht recht deutlich, besonders gut am Video *Mobile*, dass dies hier nicht überall gegeben ist. Es zeigt sich also recht deutlich, dass der oben ermittelte konstante Offset der beiden Datensätze von 0,47 Punkten statistisch durchaus relevant ist. Andererseits wird auch deutlich, dass bei einer gedachten Verschiebung der blau gezeichneten Werte um 0,47 nach oben durchwegs eine hohe Signifikanz gegeben ist. Geht es also nur um der Vergleich der Qualität der getesteten Sequen-

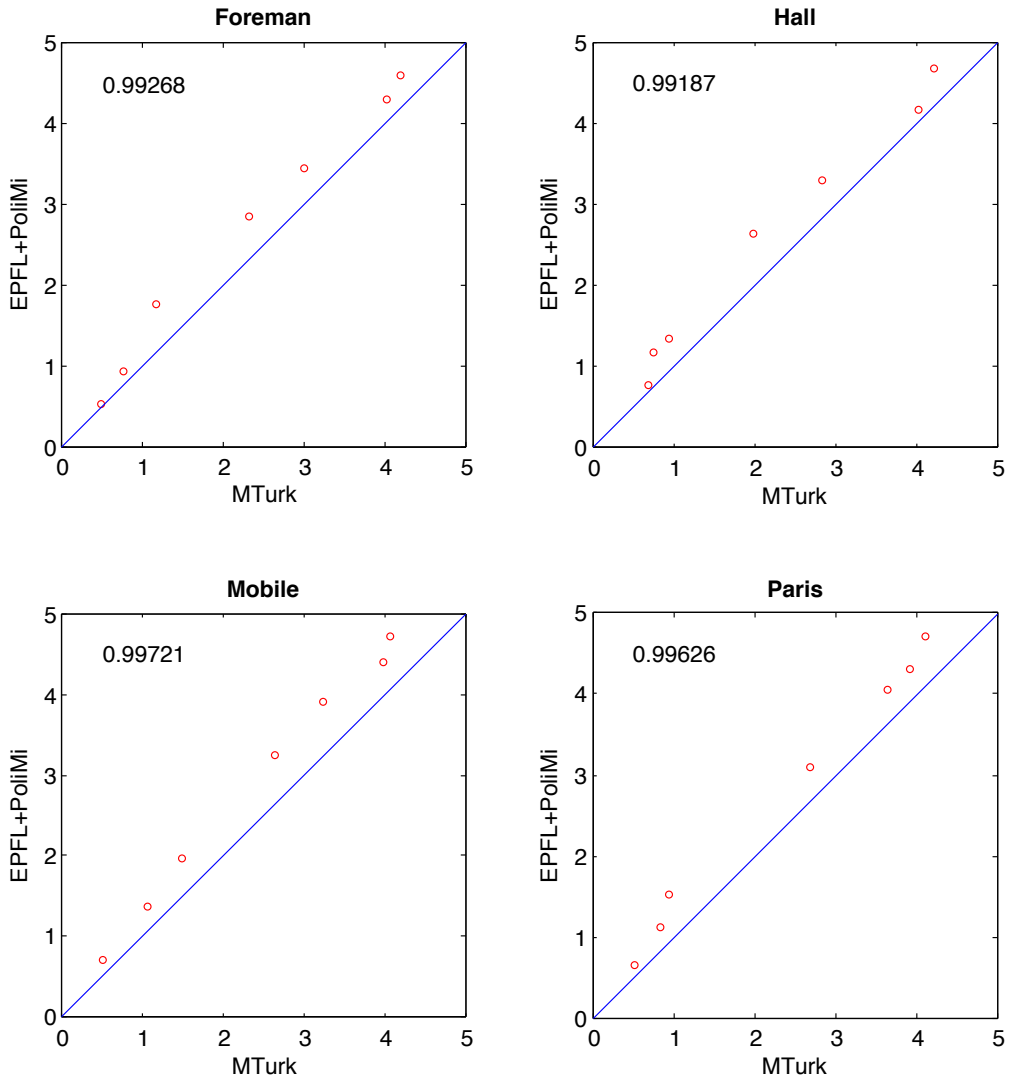

Abbildung 4.4.: Streudiagramme und Korrelationskoeffizienten der *MOS* von *EPFL+PoliMi* und *MTurk*

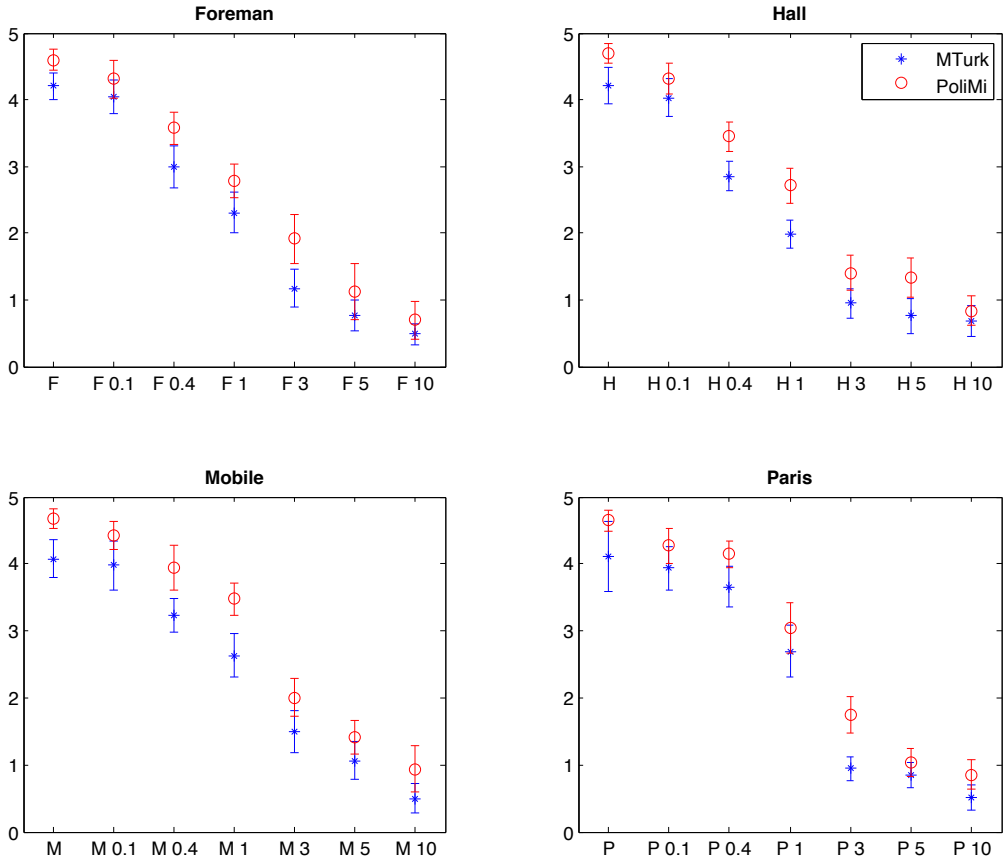

Abbildung 4.5.: Konfidenzintervalle der *MOS* von *PoliMi* und *MTurk*

zen untereinander und nicht um eine absolute Qualitätsbewertung, ist die Übereinstimmung durchaus signifikant.

Vergleicht man diese Ergebnisse mit denen der Reihen *EPFL* und *PoliMi* in Abbildung 4.6, stellt man fest, dass dort beinahe alle Konfidenzintervalle überlappen. Die Übereinstimmung zwischen den Ergebnissen der konventionellen Labors ist also deutlich besser als die Übereinstimmung der *MTurk*-Ergebnisse mit denen der Vergleichsdaten. Hier zeigen sich also offenbar doch Schwächen des Crowdsourcing-Tests.

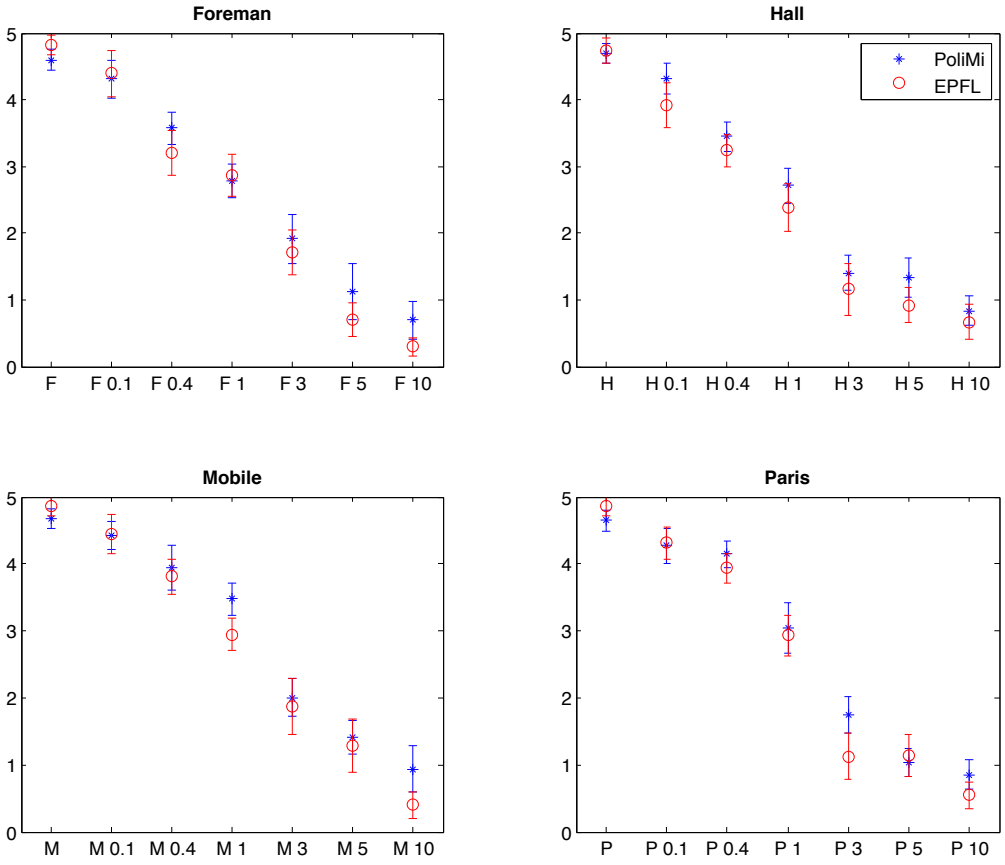

Abbildung 4.6.: Konfidenzintervalle der MOS von *PoliMi* und *EPFL*

4.2. Auswertung der zusätzlichen Statistikdaten

4.2.1. Bearbeitungsdauer

Modellierung

Gemessen wurde die Bearbeitungszeit eines *HIT*s ab dem Zeitpunkt, zu dem der Browser das Laden der Seite beendet hat, bis zum Drücken des Submit-Buttons. Diese gemessene Zeit t_m setzt sich aus der Übertragungszeit der Videodatei, bestimmt durch die Dateigröße s und die

Sequenz	PLR %	Pattern	t_m s	Sequenz	PLR %	Pattern	t_m s
	0	–	42		0	–	63
	0,1	5462	51		0,1	8022	57
	0,4	4374	43		0,4	2081	63
Foreman	1	4374	46	Mobile	1	5462	63
	3	3652	46		3	5539	69
	5	4109	45		5	1251	60
	10	4374	39		10	167	61
	0	–	40		–	–	63
	0,1	2081	37		0,1	1825	41
	0,4	522	39		0,4	2081	42
Hall	1	8552	46	Paris	1	3176	44
	3	1825	35		3	522	36
	5	3652	33		5	3652	40
	10	3693	32		10	3652	37
				Mittelwert			47

Tabelle 4.5.: mittlere Bearbeitungszeiten

Übertragungsrate r, einer kurzen Reaktionspause t_b, sowie der Laufzeit des Videos t_v und der Zeit für die Bewertung t_r zusammen. Nimmt man an, dass die Versuchsperson das Video nur einmal abspielt und direkt nach dem Einstellen der Wertung den *HIT* abschickt, lässt sich t_m wie folgt modellieren:

$$t_m = \frac{s}{r} + t_b + t_v + t_r$$

Berechnet man für jede Videosequenz die mittlere Bearbeitungsdauer (Werte siehe Tabelle 4.5) und setzt diese mit der entsprechenden Dateigröße ins Verhältnis, lassen sich mittels linearer Regression r und t_r bestimmen. Der Wert $t_v = 10$ s ist für die hier verwendeten Videosequenzen konstant, die Pause wird mit $t_b = 2$ s geschätzt. Grafisch ist diese Analyse in Abbildung 4.7 auf der nächsten Seite dargestellt. Man erhält die folgenden Werte:

$$r = 3,70 \text{ MBit/s}$$
$$t_r = 15,6 \ s$$

Analyse

Unter Berücksichtigung dieser Werte lässt sich der typische zeitliche Verlauf des Tests einer einzelnen Videosequenz rekonstruieren. Die Werte in Abbildung 4.8 auf der nächsten Seite beziehen sich auf eine durchschnittliche Dateigröße von 9,3 MB, welche ziemlich genau der der Dateien der Foreman-Sequenzen (9,6 MB) entspricht.

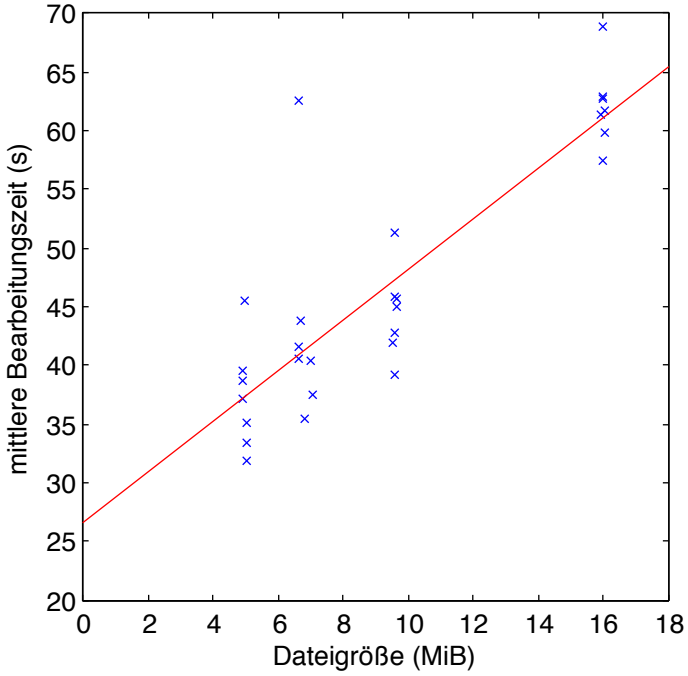

Abbildung 4.7.: mittlere Bearbeitungszeit in Abhängigkeit der Dateigröße mit Regressionsgerade

Man erkennt, dass ca. 43 % der Gesamtdauer auf das Laden der Videodatei entfällt; für die jeweils kleinste (*Hall*) und größte (*Mobile*) vorkommende Dateigröße verschiebt sich dieser Wert auf 29 % bzw. 57 %.

Für eine Durchführung in der öffentlichen *Mechanical Turk*-Plattform muss möglicherweise mit einer niedrigeren Übertragungsrate r gerechnet werden, da die Breitbandversorgung im Raum München im Vergleich zur weltweiten Verfügbarkeit überdurchschnittlich ist. Insbesondere gilt dies bei einem zu erwartenden hohen Anteil an Testteilnehmern aus Indien. An den übermittelten IP-Adressen erkennt man außerdem, dass unter den Teilnehmern auch vier mit direktem Anschluss an das Münchner Wissenschaftsnetz waren, bei denen deshalb Ladezeiten im Bereich von nur 1–2 Sekunden wahrscheinlich sind. Ohne diese 4 Datensätze sinkt r auf 3,1 Mbit/s. Genauere Erkenntnisse kann in diesem Punkt aber nur ein entsprechender Versuch liefern.

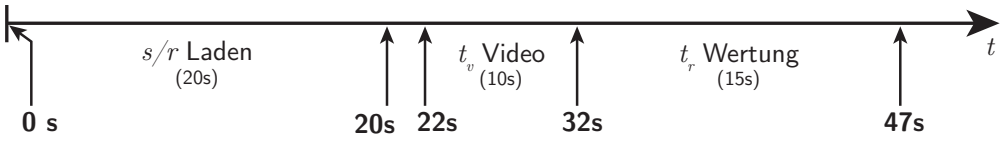

Abbildung 4.8.: typischer Zeitverlauf der Bearbeitung eines *HIT*s

Anwendung

Die zu erwartende Bearbeitungszeit ist insbesondere für die Festlegung des pro HITs auszuzahlenden Geldbetrags P_{HIT} von Interesse. Unter Berücksichtigung der obigen Ergebnisse und eines Stundenlohns w lässt sich P_{HIT} folgendermaßen abschätzen:

$$P_{HIT} = w \cdot \frac{\frac{s}{r} + t_b + t_v + t_r}{3600 \text{ s}}$$

Die Gesamtkosten P_{tot} eines Videotests mit n Videosequenzen und N Teilnehmern belaufen sich somit auf

$$P_{tot} = P_{HIT} \cdot N \cdot n$$

Laut Paolacci et al. [19] ist ein extrem niedriger Stundenlohn von unter $2,00 nicht unüblich. Geht man von $w = \$3,00$ und den obigen Werten für s, r und t_x aus, erhält man einen Betrag $P_{HIT} = \$0,039$. Rund 5 Cent kämen also als Bezahlung pro *HIT* in Frage. Die Gesamtkosten für diesen Test lägen also bei $P_{tot} = \$26,60$ – verglichen mit den Kosten eines Labortests ein recht geringer Betrag.

4.2.2. Softwarekonfiguration

Neben der Bearbeitungszeit wurden auch die Betriebssystem-, Browser- und *Flash Player*-Versionen sowie die Bildschirmauflösung der Versuchspersonen erfasst. Insbesondere die Statistik der verwendeten Browser und Betriebssysteme zeigt die selbst bei dieser vergleichsweise kleinen Zahl an Teilnehmern sehr hohe Diversität der Versuchsbedingungen und belegt gleichzeitig die Funktionsfähigkeit der *QualityCrowd*-Software unter den verschiedenen Plattformen. Die anhand des vom Browser gesendeten *HTTP-User-Agent* gemessene Verteilung zeigen die Abbildungen 4.9 und 4.10, die genauen Zahlen werden in den Tabellen B.4 und B.5 auf Seite 60 aufgeführt. Dort sind die Zahlen auch nach den einzelnen Versionsnummern aufgeschlüsselt.

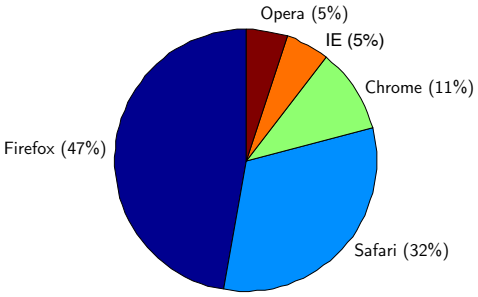

Abbildung 4.9.:
Verwendete Browser

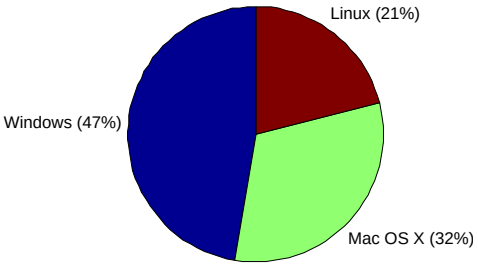

Abbildung 4.10.:
Verwendete Betriebssysteme

Die Daten über die installierten *Flash Player*-Version und über die verwendeten Bildschirmauflösungen bringen als solche keine wesentlichen Zusatzerkenntnisse, sind aber vollständidigkeitshalber in den Tabellen B.6 und B.7 auf Seite 60 dargestellt. Die ungefähre geografische

Position der Teilnehmer wurde zwar ebenfalls über die IP des Rechners des Teilnehmers ermittelt, aufgrund der nur halböffentlichen Testdurchführung sind die Ergebnisse allerdings nicht sinnvoll verwendbar – fast alle Teilnehmer stammten aus dem Großraum München.

4.3. Schlussfolgerungen

Insgesamt zeigt sich eine erstaunlich hohe Übereinstimmung der Testergebnisse mit den konventionellen Vergleichsergebnissen. Diese legt den Schluss nahe, dass die Durchführung von subjektiven Videoqualitätstests im Internet eine sinnvolle Option ist. Auch wenn eine signifikante Abweichung in Form des oben diskutierten Offsets auftrat, sind die Ergebnisse bezüglich der relativen Qualitätseinschätzung der getesteten Sequenzen untereinander einwandfrei. Für eine absolute Qualitätsmetrik scheint die internetbasierte Durchführung also offenbar nicht geeignet, allerdings sind solche absoluten Ergebnisse in den seltensten Fällen wirklich gefragt.

4.3.1. Einflüsse des Testmaterials

Einschränkend muss man jedoch feststellen, dass sich die hier erhobenen Ergebnisse möglicherweise auf die spezielle Natur der evaluierten Videosequenzen beschränken. Es wurden Videos mit Übertragungsfehlern aus Paketverlusten untersucht, die insbesondere bei höheren Fehlerraten (*PLR*) sehr ausgeprägte und eindeutig wahrnehmbare Artefakte aufweisen. Geht es in einem Test beispielsweise um die Untersuchung der Eigenschaften unterschiedlicher Videocodecs beziehungsweise derer Parameter, sind die auftretenden Artefakte oft sehr viel subtiler. Man kann vermuten, dass diese Natur der Bildfehler für die Evaluierung in dieser Testumgebung besonders geeignet ist. Artefakte wie Ringing oder geringfügige Farbverfälschungen sind möglicherweise in einer derart unkontrollierten Testumgebung schwieriger zu bewerten. Klarheit können hier allerdings nur weitere Experimente bringen.

4.3.2. Einflüsse der Testumgebung

Für die schon mehrfach erwähnte Verschiebung der Ergebnisse zu schlechteren Wertungen kann es verschiedenste Gründe geben. So ist es nicht auszuschließen, dass sich diese Abweichung aus der gesamten Testsituation ergibt. Für die Testpersonen unterschied sich diese Situation nicht wesentlich von einer gewohnten Arbeits- oder Freizeitsituation. Man kann annehmen, dass die meisten der Teilnehmer die Tätigkeit, Webvideos in ihrem Browser zu betrachten, als alltäglich empfinden und auch schon erhebliche Erfahrung damit gesammelt haben. Die Videos, die sie allerdings aus diesem Umfeld, also beispielsweise von Plattformen wie *Youtube* oder *Vimeo* oder auch der Mediatheken der großen Fernsehsender kennen, unterscheiden sich deutlich von den verwendeten Testsequenzen. Einerseits sind diese oft deutlich höher aufgelöst (z.B. *Youtube* derzeit standardmäßig 640x360), andererseits von erheblich sympathischerem Inhalt als die im Jahr 2011 recht altmodisch wirkenden Testsequenzen. All diese Faktoren haben möglicherweise zu dieser Abweichung beigetragen.

4.3.3. Einflüsse der veränderten Testmethodik

Qualifikationstest

Ein weiterer, relativ großer Unterschied in der Durchführung des Tests gegenüber den Vergleichstests liegt im durchgeführten Qualifikationstest. Dort wurde dieser Testbestandteil mit der doppelten Anzahl – also zehn – verschiedenen Sequenzen durchgeführt und mit ausführlichen Bemerkungen und Erläuterungen des Testleiters begleitet. Die Gründe für diese Abweichung wurden oben schon genauer erörtert. Auch die Möglichkeit der Teilnehmer mündliche Rückfragen zu stellen stand verständlicherweise nicht zur Verfügung.

Reihenfolge

Obwohl die Reihenfolge der Videosequenzen, wie beim Vergleichstest, durch den Zufall bestimmt wurde, konnte – wie oben erwähnt – das Aufeinanderfolgen zweier Videos mit gleichem Inhalt nicht vermieden werden. In der Folge ist es nicht auszuschließen, dass die Ergebnisse durch den Kontexteffekt beeinflusst wurden. Folgt beispielsweise eine Sequenz mittlerer Qualität auf eine desselben Inhalts aber sehr guter Qualität, ist der wahrgenommene Qualitätsunterschied oft größer und die Bewertung des zweiten Videos oft schlechter, als sie es bei Folge auf eine schlechte Sequenz wäre. Da nur vier inhaltlich verschiedene Sequenzen verwendet wurden, war die Wahrscheinlichkeit für eine solche ungünstige Reihenfolge recht hoch.

Außerdem muss man anmerken, dass es sich bei den Ergebnissen *EPFL* und *PoliMi* nur um Teilmengen eines größeren Tests mit einer größeren Zahl untersuchter Videosequenzen handelt. Auch dies kann die Testergebnisse beeinflusst haben, da dadurch das Umfeld der gezeigten Videos verändert wurde.

5. Schlussbemerkung

Im Verlauf dieser Arbeit zeigt sich, dass der Einsatz von Crowdsourcing für subjektive Videoqualitätstests durchaus möglich ist. Von technischer Seite stehen dem keine großen Hindernisse im Weg. Die verlustfreie Kompression mit *H.264/AVC* ermöglicht es, auch ohne übermäßige Wartezeiten für den Teilnehmer die erforderlichen Videodaten über das Internet zu übertragen. Für den Einsatz mit hochaufgelösten HD-Videos sind zwar sowohl die heutigen Internetverbindungen als auch die im privaten Umfeld verbreiteten Computer noch deutlich zu langsam, die hier untersuchte CIF-Auflösung stellt jedoch einen guten Kompromiss dar.

Durch den Einsatz der *QualityCrowd*-Software und *Mechanical Turk* ist es sogar wahrscheinlich erstmals möglich, einen subjektiven Videotest gänzlich ohne Labor, Betreuungsarbeit und großen Organisationsaufwand durchzuführen. Es reicht aus, das Amazon-Konto mit einem entsprechenden Geldbetrag aufzuwerten, die Videos auszusuchen, zu codieren und den Testablauf in der Software zu definieren. Nach einiger Zeit können dann die fertigen Ergebnisse aus *QualityCrowd* für die weitere Analyse exportiert werden.

Dadurch eröffnet sich auch die Möglichkeit einer neue Art und Weise der Anwendung von subjektiven Tests. Durch den sinkenden Aufwand ist es mit diesem System eher möglich, auch Fragestellungen zu untersuchen, für die sich ein herkömmlicher Test nicht lohnen würde. Dabei sind kürzere Tests vermutliche sogar besser für Crowdsourcing geeignet als längere. So ist eine Ergänzung der vorhandenen und erprobten Testmethodiken durch Crowdsourcing durchaus vorstellbar.

Den Ergebnissen des Test kann eine durchweg hohe Qualität bescheinigt werden. Es lassen sich aus ihnen weitgehend dieselben Schlüsse ziehen wie aus denen des Labortests, die statistische Streuung ist im wesentlichen genauso groß und die Korrelation mit den Vergleichsergebnissen ist außerordentlich hoch. Trotz des nicht vollständig geklärten, signifikanten kleinen Offsets eignen sich die Ergebnisse also durchaus für die Untersuchung von Videoqualität.

Die hier erzielten guten Ergebnisse gelten jedoch nur vorbehaltlich der oben erwähnten Einschränkungen. Vor einem alleinigen Einsatz von Crowdsourcing für einen Videotest sollten in jedem Fall weitere Tests wie dieser durchgeführt werden. Vor allem die Verwendung mit anderem, schwieriger zu bewertendem Testmaterial, aber auch der wirklich vollständig öffentliche Einsatz der *Mechanical Turk*-Plattform müssen noch genauer untersucht werden.

A. Dokumentation QualityCrowd

A.1. Installation

QualityCrowd ist eine Webapplikation und basiert auf dem MVC-Framework *CakePHP*. Als solche benötigt sie zur Ausführung eine Webserverumgebung und eine Datenbankanbindung. Die Anforderungen für die Installation sind laut *CakePHP*-Dokumentation [6] folgende:

- ein *HTTP*-Server, vorzugsweise Apache *httpd* mit *mod_rewrite*
- *PHP* ab Version 4.3.2
- ein Datenbankserver, z.B. *MySQL* (ab Version 4), *PostgreSQL*, *Microsoft SQL Server*, *Oracle*, *SQLite*

Für diese Arbeit wurde konkret folgende Konfiguration verwendet:

- Apache *httpd* 2.2.14 (mit *mod_rewrite*)
- *PHP* 5.3.2
- *MySQL* 5.1.41 (mit *InnoDB*)

Des Weiteren muss der Webserver für die Verwendung von *HTTPS* konfiguriert und ein entsprechendes Zertifikat vorhanden sein. Dies ist notwendig, da die *Mechanical Turk*-Webseite ebenfalls über *HTTPS* angesprochen wird und es bei einer Einbettung von Inhalten von *QualityCrowd* sonst zu Fehlermeldungen kommen kann, die im Sinne der Benutzerfreundlichkeit zu vermeiden sind.

Darüber hinaus müssen für die Anbindung an *Mechanical Turk* die *Amazon Mechanical Turk Command Line Tools* installiert sein[1]. Voraussetzung für die Installation ist eine funktionsfähige *JAVA*-Laufzeitumgebung. Im Test kam hier das *OpenJDK Runtime Environment* zum Einsatz.

A.2. Konfiguration

Für die Konfiguration der Software stehen die Konfigurationsdateien in `app/config/` zur Verfügung. Die zwei wichtigsten werden im Folgenden kurz beschrieben, für weitere Optionen ist die *CakePHP*-Dokumentation heranzuziehen.

[1] erhältlich unter `http://aws.amazon.com/developertools/Amazon-Mechanical-Turk/694`

Datenbank

Hier müssen die Zugangsdaten zur Datenbank eingetragen werden.

```
<?php
class DATABASE_CONFIG {
  var $default = array(
    'driver'     => 'mysql',
    'persistent' => false,
    'host'       => '127.0.0.1',
    'login'      => '*username*',
    'password'   => '*password*',
    'database'   => '*database*',
    'prefix'     => '',
  );
}
?>
```

Listing A.1: app/config/database.php

Um die Datenbank zu initialisieren, führen Sie im Wurzelverzeichnis der Installation den Befehl `./cake/console/cake schema create` aus und bestätigen die Nachfragen mit `y`. Wenn die Datenbankkonfiguration korrekt ist, sollten jetzt die benötigten Tabellen in der Datenbank angelegt worden sein.

Anschließend muss noch ein erster Benutzer in der Datenbank angelegt werden. Der Benutzername lautet `admin`, das Passwort `admin`. Wenden Sie dazu folgenden *SQL*-Befehl auf die Datenbank an:

```
INSERT INTO `users` (`id`, `username`, `password`) VALUES
(1, 'admin', 'ec08d2cec79f3bb56ac8b8b0ba95541c205c2841');
```

Anschließend sollte *QualityCrowd* per Webbrowser erreichbar sein und der Login funktionieren.

Mechanical Turk

Für die Anbindung an *Mechanical Turk* müssen einige Parameter konfiguriert werden. `mturk.clipath` bezeichnet den lokalen Pfad der *Amazon Mechanical Turk Command Line Tools*, die Werte für `mturk.accesskey` und `mturk.secretkey` erhält man im *Amazon Web Services*-Portal[2]. Der Parameter `baseurl` muss mit der *URL* belegt werden, unter der die gesamte Installation per *HTTPS* erreichbar ist. Soll statt des Produktivsystems die *Developer Sandbox* von *Mechanical Turk* verwendet werden, muss im Parameter `mturk.serviceurl` der Hostname gegen `mechanicalturk.sandbox.amazonaws.com` ausgetauscht werden.

```
<?php
Configure::write('mturk.javahome', '/usr');
Configure::write('mturk.clipath', '*pfad*/aws-cli/bin');
```

[2] https://aws-portal.amazon.com/gp/aws/developer/account/index.html?action=access-key

```
Configure::write('mturk.accesskey', '*access-key*');
Configure::write('mturk.secretkey', '*secret-key');
Configure::write('mturk.serviceurl',
  'http://mechanicalturk.amazonaws.com?
   Service=AWSMechanicalTurkRequester');

Configure::write('baseurl', 'https://quality.ldv.ei.tum.de/qc');
?>
```

Listing A.2: `app/config/mturk.php`

A.3. Verwendung

Im Folgenden sollen kurz die einzelnen Schritte, die für die Durchführung eines Qualitätstest mit *QualityCrowd* nötig sind, beschrieben werden.

Videos hochladen

Nach dem Login wechselt man im Menü am oberen Seitenrand zu *Videos*. Dort kann man mit *Add Video* die zu testenden Videodateien hochladen. Jedes Video hat einen Titel, der nur für interne Zwecke verwendet wird. Im Feld *Qualification hint* kann ein Hinweistext eingegeben werden, der angezeigt wird, wenn das Video in einem Qualifikationstest verwendet wird.

Die Videodateien müssen in einem *MP4*-Container (nach ISO/IEC 14496-14) vorliegen, *H.264/AVC* soll als Codec verwendet werden. Die Playersoftware ist aktuell nur für Videos mit CIF-Auflösung (352x288 Pixel) ausgelegt.

Fragen und Antwortmöglichkeiten anlegen

Auf der Unterseite *Questions* lassen sich die Texte, die jeweils ober- und unterhalb des zu bewertenden Videos angezeigt werden, festlegen. Ein einmal angelegter Satz von Texten kann auch in mehreren Tests wiederverwendet werden.

Unter dem Menüpunkt *Answers* werden die Bewertungsskalen definiert. Es können immer Paare, bestehend aus angezeigter Antwortmöglichkeit und einem übermittelten Zahlenwert, definiert werden. Die Option *Continous scale* bewirkt, dass statt einer Auswahlmöglichkeit mit Radio-Buttons ein stufenloser Schieberegler (siehe Abbildung 3.4 auf Seite 25) angezeigt wird. Die Werte der *Value*-Spalte werden in diesem Fall ignoriert und stattdessen ein Wert im Bereich zwischen 0 und 1000 gespeichert, der die Einstellung des Schiebereglers repräsentiert.

Qualifikationstest anlegen

Vor der Teilnahme an einem Videotest muss jede Testperson einen Qualifikationstest absolvieren (siehe auch Abschnitt 3.3 auf Seite 25). Ein solcher Test kann unter „Qualifications“ zusammengestellt werden. Im Formular zur Erstellung müssen folgende Paramter angegeben werden:

Title Der Titel, der auch in Mechanical Turk als Titel des Qualifikationstests angezeigt wird.

Keywords Hier können durch Kommata getrennte Schlagworte eingegeben werden, die die Auffindbarkeit des Qualifikationstests auf der Mechanical Turk Webseite verbessern.

Description Die hier einzutragende kurze, prägnante Beschreibung, die die zur erlangende Qualifikation erläutert, wird wie der Titel öffentlich angezeigt.

Question Der hier eingegebene Text wird oberhalb des Qualifikationstests angezeigt. Er sollte kurz umreißen, worum es bei diesem Test geht und konkrete Arbeitsanweisungen enthalten. Dies ist möglicherweise auch der richtige Ort, um generelle Informationen auch zur anschließend folgenden Testdurchführung zu geben.

Testduration Zeitraum in Sekunden, die ein Worker Zeit hat, diesen Test durchzuführen. Standardmäßig ist eine Stunde (3600 Sekunden) eingestellt. Dieser Wert kann großzügig bemessen werden, und es sollte im Normalfall keinen Grund geben, von diesem Wert abzuweichen.

Weiterhin muss ein Antwort-Set für den Qualifikationstest angegeben werden und die verwendeten Videos ausgewählt werden. Diese Videos werden, wie oben erwähnt, gemeinsam mit ihrem *Qualification Hint* und alle auf einer Seite untereinander angezeigt. Es empfiehlt sich daher bei der Auswahl der Videos die für diese Seite zu übertragende Datenmenge zu berücksichtigen, um überlange Wartezeiten zu vermeiden. Ein Beispiel für einen Qualifikationstest zeigt Abbildung B.8 auf Seite 68.

Batch anlegen

Nach einem Klick auf *Add Batch* erscheint das Formular zum Anlegen eines neuen Batches, also dem eigentlichen Set der zu testenden Videos. Im Folgenden werden die einzelnen Optionen näher beschrieben:

Title Der Titel, der auch in *Mechanical Turk* als Titel des *HIT*s angezeigt wird.

Description Wird wie der Titel öffentlich angezeigt und sollte eine kurze, prägnante Beschreibung enthalten, die dem Worker verrät, worum es in diesem *HIT* geht.

Keywords Hier können durch Kommata getrennte Schlagworte eingegeben werden, die die Auffindbarkeit des *HIT*s verbessern. Die Keywords werden ebenfalls öffentlich angezeigt.

Payment Definiert den pro Videosequenz (!) an den Worker ausgezahlten Betrag in US-Dollar.

Assignments Definiert die Anzahl an Workern, die höchstens pro Videosequenz zugelassen werden. Hilfreich, wenn die Gesamtkosten begrenzt werden sollen.

Assignmentduration Zeitraum in Sekunden, die ein Worker Zeit hat, einen *HIT*, sprich eine Videosequenz, zu bearbeiten. Standardmäßig ist eine Stunde (3600 Sekunden) eingestellt. Dieser Wert kann großzügig bemessen werden – er soll nur verhindern, dass Worker, die einen *HIT* akzeptieren und dann nicht abschließen, die Testdurchführung blockieren können.

HIT Lifetime Hier kann der Zeitraum in Sekunden, in dem der Batch in Mechanical zur Bearbeitung freigeschaltet ist, definiert werden. Der Standardwert ist 604 800 Sekunden, was 7 Tagen entspricht.

Es folgen noch die Auswahlmöglichkeiten zur Verknüpfung mit Frage, Qualifikationstest, Antwortmöglichkeiten und Videos.

Nach dem Speichern ist der Batch angelegt, und mit einem Klick auf *Preview* in der Übersichtsliste der Batches kann eine Vorschau des Batches angezeigt werden, so wie ihn der Worker später sehen wird.

Batch veröffentlichen

Durch einen Klick auf *Publish* in der Batch-Übersichtsliste werden die entsprechenden *HIT*s in *Mechanical Turk* angelegt. Es ist sicherlich sinnvoll, sich von der korrekten Funktion anschließend in Amazons Webinterface zu überzeugen.

Die Funktion *Unpublish* ermöglicht es, bereits veröffentlichte Batches wieder zurückzuziehen. **Achtung**, dabei gehen alle bisher erzeugten Testergebnisse verloren.

Ergebnisse herunterladen

Die Aktion *Results* in der Batch-Übersicht lädt die aktuellen Ergebnisse von *Mechanical Turk* herunter und zeigt eine einfache Auswertung pro Videosequenz an. *download as CSV-file* ermöglicht es, die ausführlichen Testergebnisse für eine weitergehende Analyse beispielsweise in *Excel* oder *MATLAB*, als CSV-Datei herunterzuladen.

A.4. Videoencodierung

Wie in Abschnitt 2.4 auf Seite 18 beschrieben, ist die Erzeugung von verlustfrei mit *H.264/AVC* encodierten Videosequenzen ist nicht unproblematisch. Für das Encodieren wird die freie Implementierung *x264* verwendet, das wie oben erwähnt in einer bestimmten älteren Version vorliegen muss. Aus diesem Grund wurden einige Skripte entwickelt, die in der Lage sind, automatisiert eine entsprechende Softwareumgebung herzustellen und entsprechend kompatible Videos zu erzeugen. Dabei wird stets nach dem Encodieren einer Sequenz die tatsächliche Verlustfreiheit anhand einer Prüfsumme überprüft.

Die Skripte wurden unter Ubuntu 10.10, Debian 6 (Squeeze) und MacOS X 10.6 getestet. Aufgrund der vergleichsweise alten Version von *x264* kommt es auf 64Bit-Systemen möglicherweise zu Installationsproblemen.

Installation

Folgende Programme müssen auf dem System installiert sein:

- *automake*
- *git*

- *gcc*
- *yasm*

Sind diese Voraussetzungen erfüllt, kann mit `./install.sh` die Umgebung installiert werden. Dabei werden die entsprechenden notwendigen Versionen von *x264* und *ffmpeg* heruntergeladen und kompiliert. Beide Tools stehen anschließend in `./bin` zur Verfügung.

Encodierung

Das Skript `./ll264.sh` erwartet als Parameter eine Datei mit unkomprimiertem YUV-Rawvideo und der Endung `.yuv`. Es erzeugt eine verlustfrei komprimierte *MPEG4*-Datei mit der Endung `.mp4`. Diese Datei kann direkt in *QualityCrowd* verwendet werden.

Sollen mehrere Videos encodiert werden, kann das Skript `./ll264dir.sh` eingesetzt werden. Dieses erwartet als einzigen Parameter den Pfad zu einem Verzeichnis mit `.yuv`-Dateien.

B. Weitere Tabellen und Abbildungen

B.1. Tabellen

Sequenz	PLR %	Pattern	Größe MB	Kompressionsfaktor %
Foreman	0	–	9,6	78
	0,1	5462	9,6	78
	0,4	4374	9,6	78
	1	4374	9,6	78
	3	3652	9,6	78
	5	4109	9,7	78
	10	4374	9,6	78
Hall	0	–	4,9	89
	0,1	2081	4,9	89
	0,4	522	4,9	89
	1	8552	5,0	89
	3	1825	5,0	88
	5	3652	5,0	88
	10	3693	5,0	88
Mobile	0	–	16,0	63
	0,1	8022	16,0	63
	0,4	2081	16,0	63
	1	5462	16,0	63
	3	5539	16,0	63
	5	1251	16,1	63
	10	167	15,9	63
Paris	0	–	6,6	85
	0,1	1825	6,7	85
	0,4	2081	6,6	85
	1	3176	6,7	85
	3	522	6,8	84
	5	3652	7,0	84
	10	3652	7,1	84
Summe			261,3	

Tabelle B.1.: Testsequenzen mit Dateigrößen und Kompressionsfaktoren

Sequenz	PLR	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	–	4,250	3,085	3,805	3,945	3,585	4,165	4,000	4,500	4,890	4,470	3,915	3,915	4,500	4,665	5,000	4,780	4,695	4,795	3,530
	0,1	4,470	2,665	3,780	2,695	3,945	3,695	3,640	4,500	4,195	3,915	3,695	4,695	4,415	4,195	4,695	4,360	4,780	4,235	4,165
	0,4	4,055	1,445	3,280	2,415	3,000	2,780	2,500	3,610	3,750	3,335	2,500	2,445	3,280	3,500	2,030	4,390	3,470	3,765	2,055
Foreman	1	1,720	1,305	2,250	2,085	2,500	3,140	2,030	2,500	1,780	3,530	2,305	3,000	3,305	3,030	1,030	1,165	3,280	2,910	1,445
	3	1,195	1,140	1,665	1,110	1,555	1,030	1,530	2,500	1,000	0,220	1,030	0,640	2,280	1,530	0,000	0,555	1,305	1,295	0,555
	5	0,640	0,530	1,250	1,220	1,110	1,055	0,530	1,530	0,195	0,000	0,000	1,470	1,415	0,720	0,000	0,250	0,890	0,940	0,530
	10	0,390	0,220	0,470	0,500	0,445	0,470	0,555	1,500	0,280	0,750	0,000	0,720	0,500	0,890	0,000	0,000	0,165	0,000	0,585
	–	4,470	2,500	3,555	3,805	3,445	4,055	4,415	4,555	4,805	4,720	3,640	3,585	4,360	4,720	5,000	5,000	4,860	5,000	5,000
	0,1	4,110	2,220	3,500	3,750	3,500	3,805	3,555	4,470	4,860	4,750	4,250	4,220	4,305	5,000	4,195	5,000	3,835	3,620	3,195
	0,4	3,220	1,445	3,000	2,970	2,610	3,000	3,445	3,445	1,945	3,750	2,335	3,220	3,305	3,140	2,250	1,835	3,695	2,880	2,500
Hall	1	1,470	1,445	1,835	1,415	2,140	1,835	2,500	2,500	2,500	2,945	1,720	2,250	2,500	2,500	1,280	1,165	2,585	3,355	1,030
	3	1,360	1,195	1,415	0,695	1,030	0,640	1,445	1,445	0,695	0,780	0,000	1,140	1,195	1,140	0,000	0,500	1,530	1,440	0,720
	5	0,695	0,805	1,640	0,530	0,360	0,335	1,110	0,445	0,305	1,195	0,000	1,030	1,970	1,055	0,915	0,030	0,000	0,265	0,805
	10	1,110	0,720	0,915	0,360	0,000	0,835	0,860	1,500	0,555	0,640	0,000	0,780	0,530	1,750	1,000	0,000	0,110	0,000	0,220
	–	3,500	1,780	4,030	3,555	3,695	4,500	4,165	4,695	4,780	4,555	4,030	4,000	4,415	4,640	4,835	4,750	4,195	4,590	3,055
	0,1	4,470	1,360	3,945	3,860	4,055	4,555	3,470	3,500	3,915	4,665	4,220	4,500	4,500	4,360	4,835	4,780	4,360	3,910	2,030
	0,4	3,335	1,665	3,500	2,500	3,500	3,085	3,530	2,500	4,085	3,530	3,250	2,970	4,470	3,695	3,000	3,445	3,445	3,355	2,500
Mobile	1	1,915	1,415	2,030	3,220	3,030	3,360	3,445	2,500	1,415	2,500	2,280	2,780	3,555	1,970	2,360	2,970	3,470	2,500	3,030
	3	1,530	1,055	2,220	1,805	1,390	1,970	1,805	1,500	0,250	2,390	1,500	1,890	1,695	1,780	0,000	1,945	0,610	1,440	1,335
	5	0,805	1,055	1,250	1,585	1,030	1,470	1,470	1,500	0,610	1,665	0,470	1,915	0,970	0,665	0,390	0,335	0,415	1,325	1,415
	10	0,000	0,640	0,890	0,970	0,555	0,585	0,585	1,470	0,305	0,720	0,000	0,390	0,390	0,085	0,000	0,110	0,030	0,410	0,500
	–	4,220	3,415	3,305	4,415	4,470	4,555	4,470	4,610	4,665	4,890	3,445	4,055	5,000	4,530	4,695	0,055	5,000	5,000	4,140
	0,1	3,470	2,805	2,500	3,555	5,000	4,195	3,835	3,530	4,250	3,445	3,805	4,500	4,250	3,945	5,000	4,750	4,445	4,590	3,530
	0,4	2,915	1,720	3,665	3,030	3,835	3,970	3,470	3,585	4,640	4,500	2,835	3,585	4,360	3,970	4,030	4,665	4,390	4,410	2,500
Paris	1	3,555	1,445	2,970	1,890	3,470	3,220	2,500	2,500	1,915	4,280	0,970	2,500	3,500	2,500	2,335	4,360	3,030	3,060	1,390
	3	0,640	0,695	1,220	0,695	1,165	1,500	1,530	1,500	0,915	0,750	0,445	0,470	1,110	1,335	0,695	0,110	1,280	1,120	0,750
	5	0,970	0,250	2,000	0,695	1,360	0,750	1,030	1,445	0,335	0,780	0,000	0,555	1,110	0,970	0,695	0,360	1,305	0,000	0,500
	10	0,390	0,640	0,695	0,415	0,860	0,500	0,555	1,445	0,000	0,445	0,000	0,470	0,640	0,970	0,000	0,000	0,915	0,000	0,195
Mittelwert		2,317	1,452	2,378	2,132	2,38	2,466	2,428	2,689	2,280	2,647	1,880	2,418	2,779	2,616	2,152	2,202	2,575	2,508	1,900
Minimum		0,000	0,220	0,470	0,360	0,000	0,335	0,530	0,445	0,000	0,000	0,000	0,390	0,390	0,085	0,000	0,000	0,000	0,000	0,195
Maximum		4,470	3,415	4,030	4,415	5,000	4,555	4,470	4,695	4,890	4,890	4,250	4,695	5,000	5,000	5,000	5,000	5,000	5,000	5,000

Tabelle B.2.: vollständiger Ergebnisdatensatz – Teil 1, vollständige Teilnehmer

Sequenz	PLR	20	21	22	23	24	25	26
	–	3,835	4,585		4,610		5,000	
	0,1	3,555	4,085					4,585
	0,4	2,780	3,695	3,280	2,500	4,470		4,665
Foreman	1	2,305		3,555		3,585		
	3	1,000	1,695	1,000	1,470	2,500	1,585	
	5	0,585	0,750	1,415	1,000		1,055	
	10	0,000	0,220			1,390		0,220
	–	3,780	4,165	4,110	4,665			4,470
	0,1		3,500	4,000		4,585		
	0,4	2,555	3,140	2,835	2,500	4,415	3,055	
Hall	1	2,000	1,280	2,165	1,970		1,360	
	3	0,000	0,470	1,305				
	5	0,530	0,945	0,780	0,780			1,220
	10	0,000	0,470	0,415		1,360		
	–	3,780		3,945		4,750		3,665
	0,1	3,280	3,890	4,305	3,470		4,140	
	0,4	3,000		3,445				
Mobile	1	1,915	2,030			3,250		
	3	1,000	0,890	2,500	1,250	2,500		
	5	0,555	1,360			1,665		
	10	0,000	0,000	0,720			0,165	
	–	4,000	5,000		4,470		5,000	
	0,1		4,085	4,195		4,500		4,640
	0,4	3,165	3,140		4,720			
Paris	1	2,140	2,665	2,250	2,030	4,390	1,970	
	3	0,280	0,585				1,890	
	5	0,530	0,220	1,335	0,640	1,530	0,945	
	10	0,280	0,000					0,335
Anzahl		26	25	19	14	14	11	8
Mittelwert		1,802	2,115	2,503	2,577	3,206	2,379	2,975
Minimum		0,000	0,000	0,415	0,640	1,360	0,165	0,220
Maxmimum		4,000	5,000	4,305	4,720	4,750	5,000	4,665

Tabelle B.3.: vollständiger Ergebnisdatensatz – Teil 2, abgebrochene Teilnehmer

Betriebssystem	Version	Anzahl		Anteil
Windows	7	6	9	47 %
	XP	2		
	Vista	1		
Mac OS X	10.6.7	5	6	32 %
	10.6.6	1		
Linux	Ubuntu	2	4	21 %
	SUSE	1		
	sonstige	1		

Tabelle B.4.: Verwendete Betriebssysteme und -versionen

Browser	Version	Anzahl		Anteil
Mozilla Firefox	3.6.16	6	9	47 %
	4.0	3		
Apple Safari	5.04	4	6	32 %
	5.05	2		
Google Chrome	10.0.648.204	1	2	11 %
	10.0.648.205	1		
Microsoft Internet Explorer	9.0	1	1	5 %
Opera	11.00	1	1	5 %

Tabelle B.5.: Verwendete Browser und -versionen

Version	Anzahl
10.0.45	1
10.1.102	1
10.2.152	5
10.2.153	8
10.2.154	2
10.2.159	2

Tabelle B.6.: Verwendete *Flash Player* Versionen

Auflösung	Anzahl
1920×1200	5
1680×1050	4
1440×900	3
1280×1024	2
1366×768	1
1280×800	1
1152×864	1
1024×768	2

Tabelle B.7.: Verwendete Bildschirmauflösungen

B.2. Abbildungen

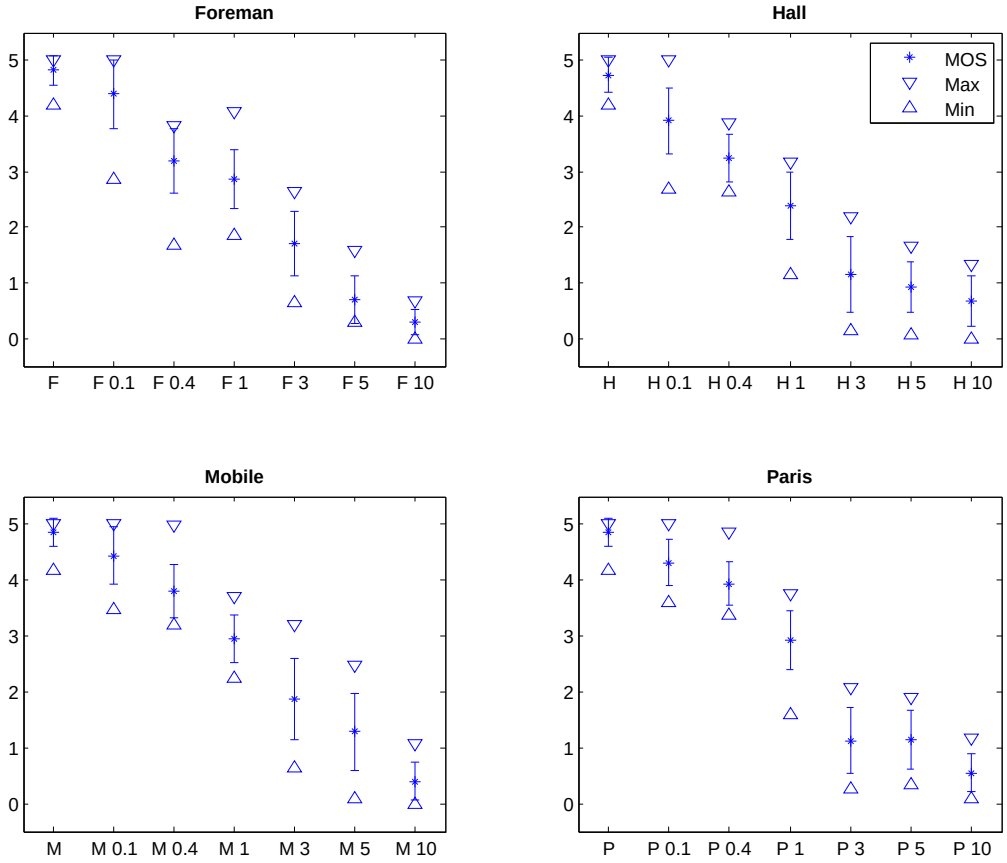

Abbildung B.1.: *Mean Opinion Scores* und empirische Standardabweichung für *EPFL*

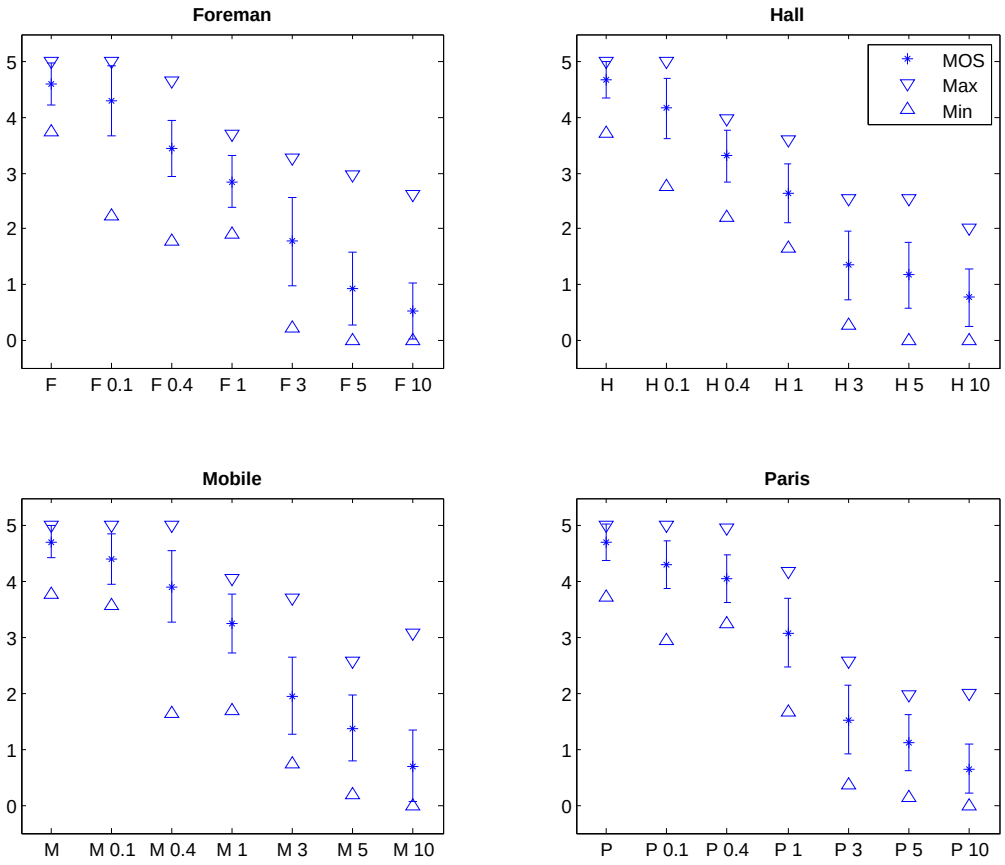

Abbildung B.2.: *Mean Opinion Scores* und empirische Standardabweichung für *EPFL+PoliMi*

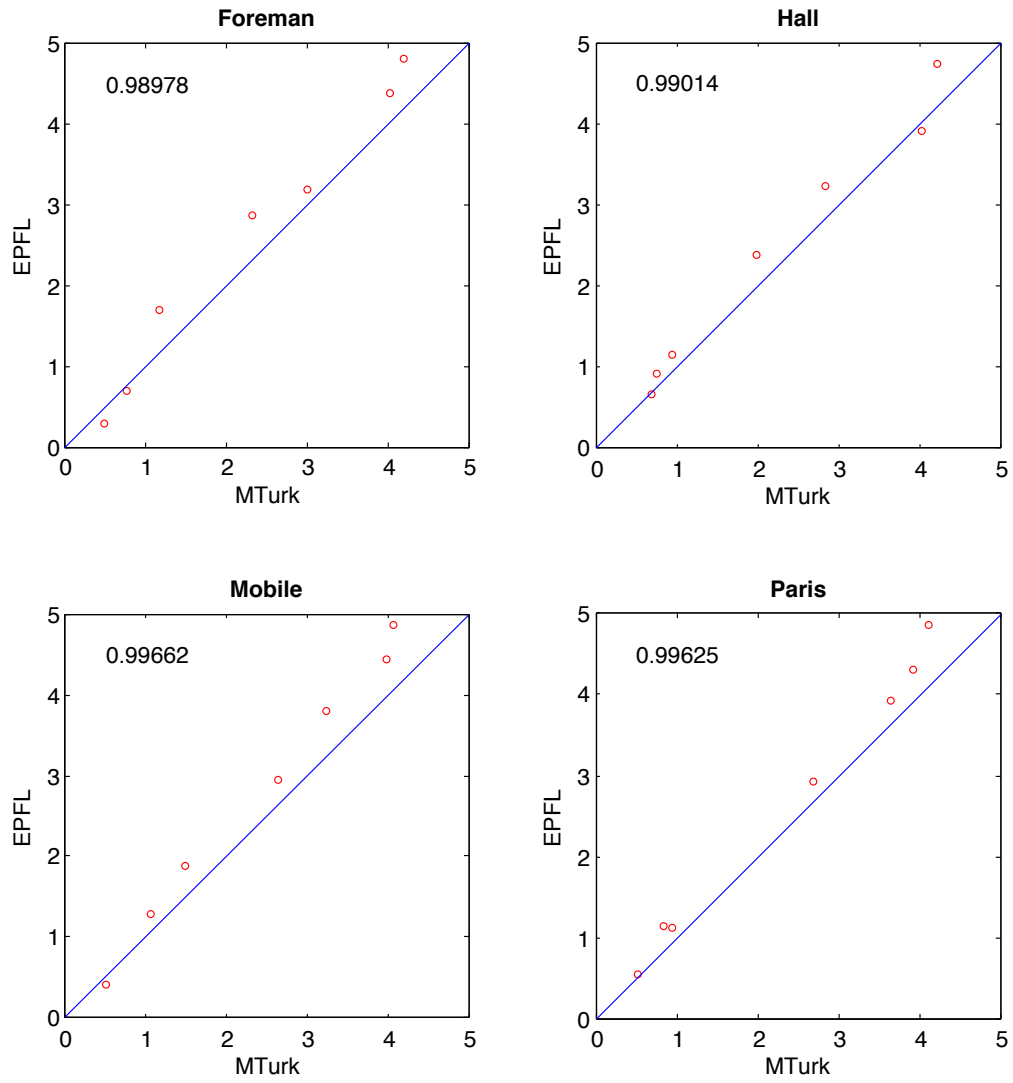

Abbildung B.3.: Streudiagramme und Korrelationskoeffizienten der *MOS* von *EPFL* und *MTurk*

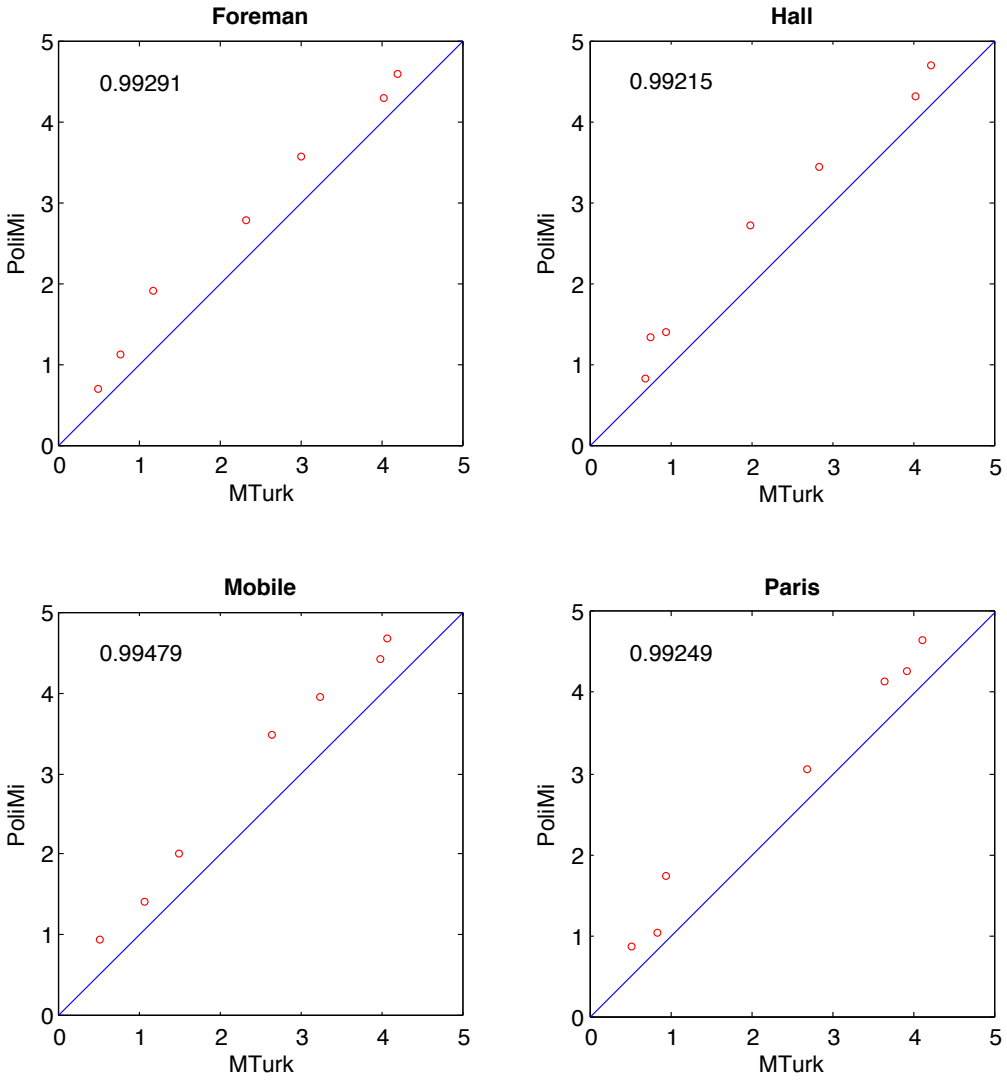

Abbildung B.4.: Streudiagramme und Korrelationskoeffizienten der *MOS* von *PoliMi* und *MTurk*

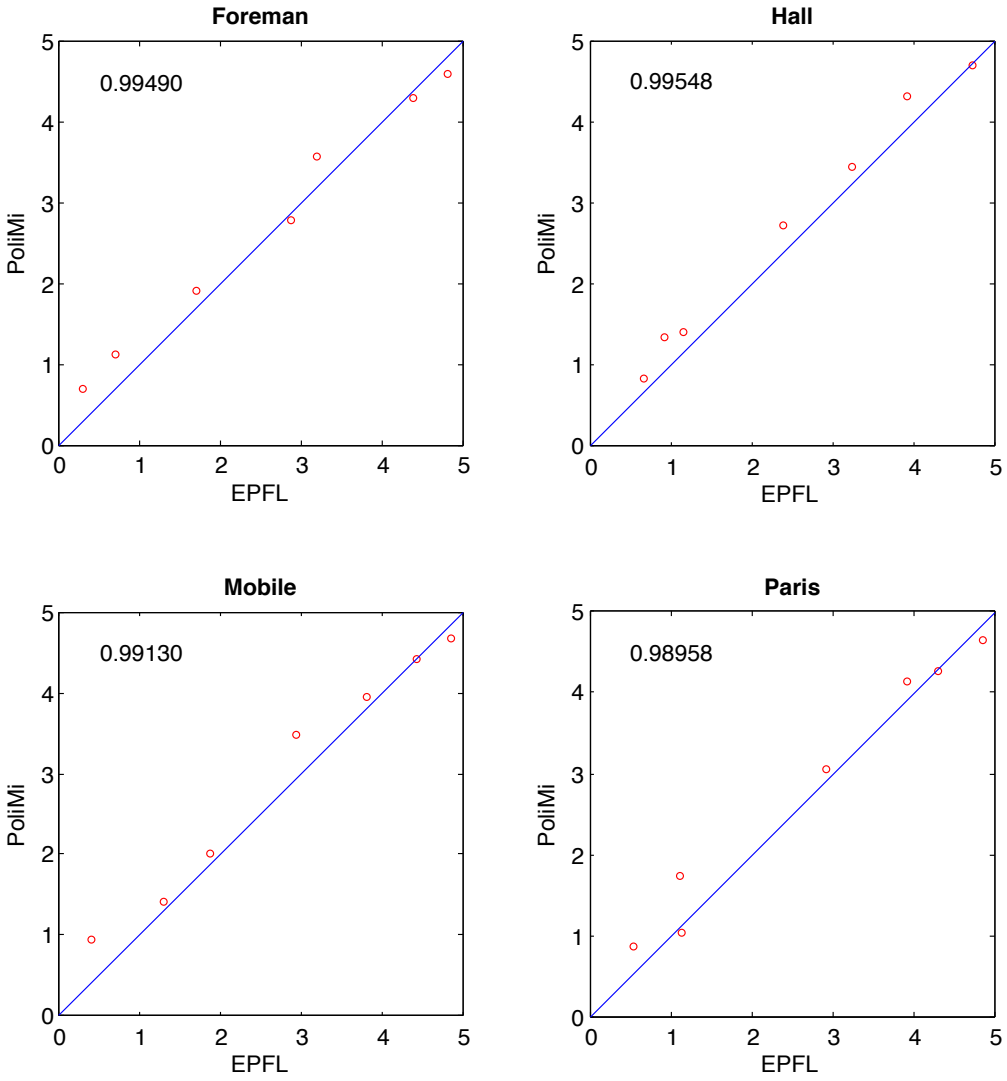

Abbildung B.5.: Streudiagramme und Korrelationskoeffizienten der MOS von *EPFL* und *PoliMi*

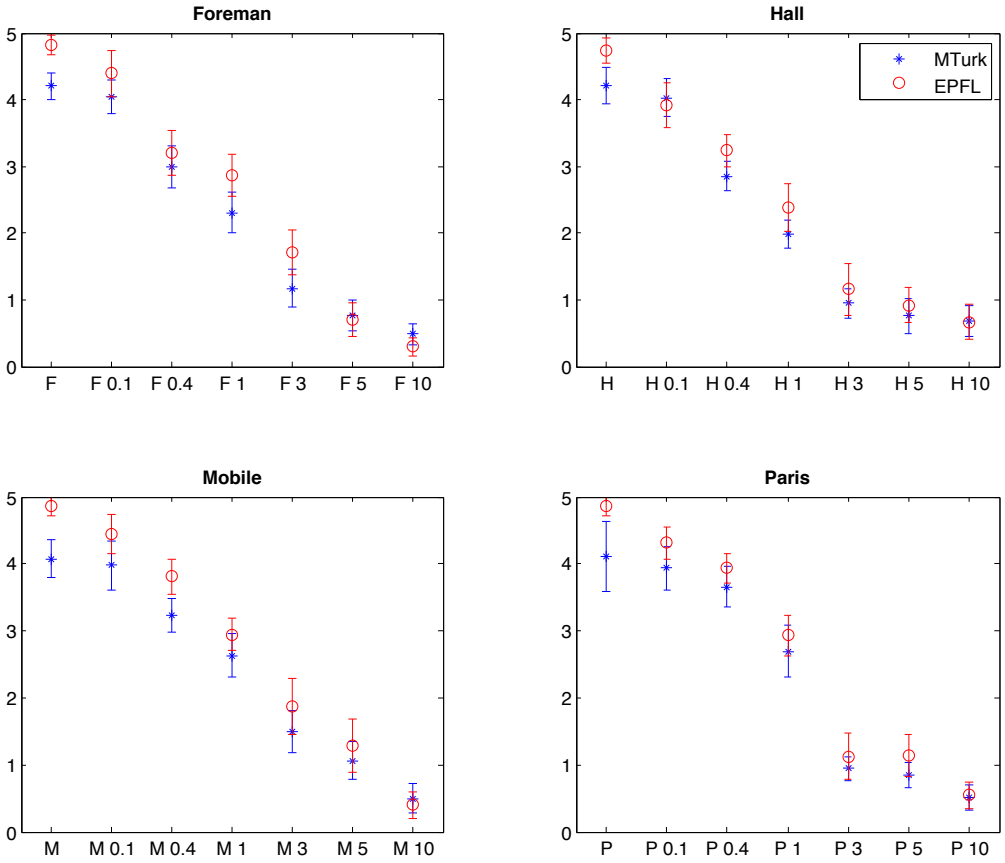

Abbildung B.6.: Konfidenzintervalle der *MOS* von *EPFL* und *MTurk*

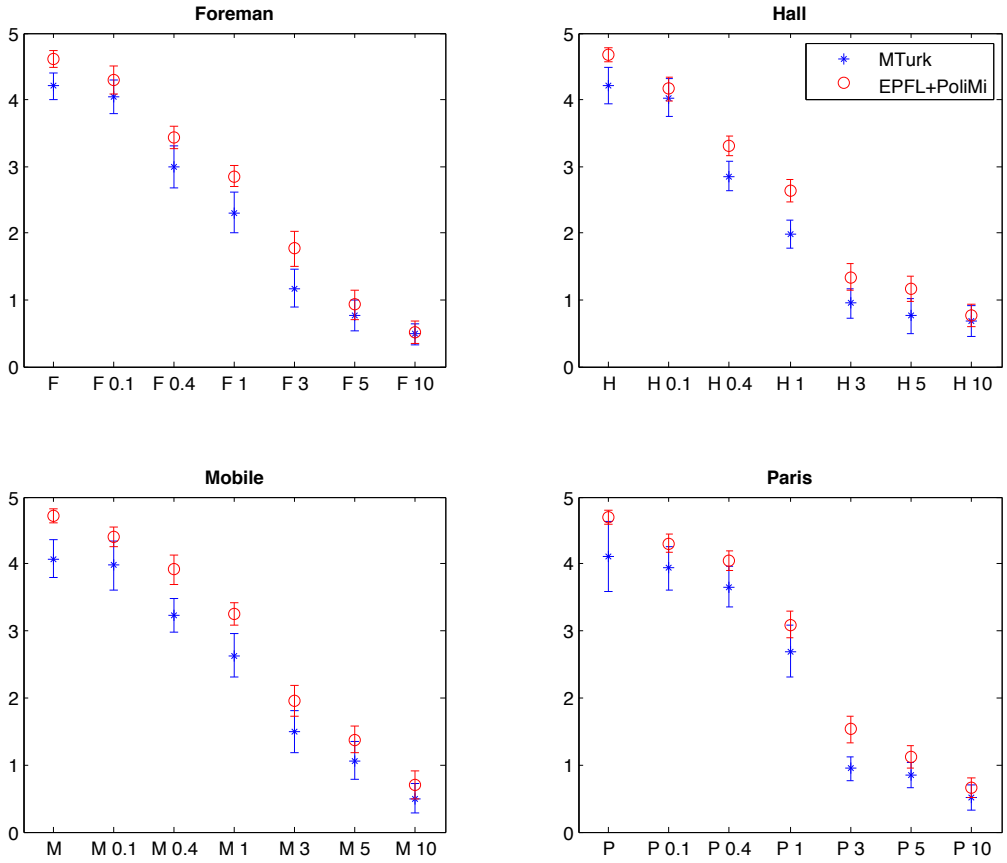

Abbildung B.7.: Konfidenzintervalle der *MOS* von *EPFL+PoliMi* und *MTurk*

Abbildung B.8.: Qualifikationstest in *Mechanical Turk*

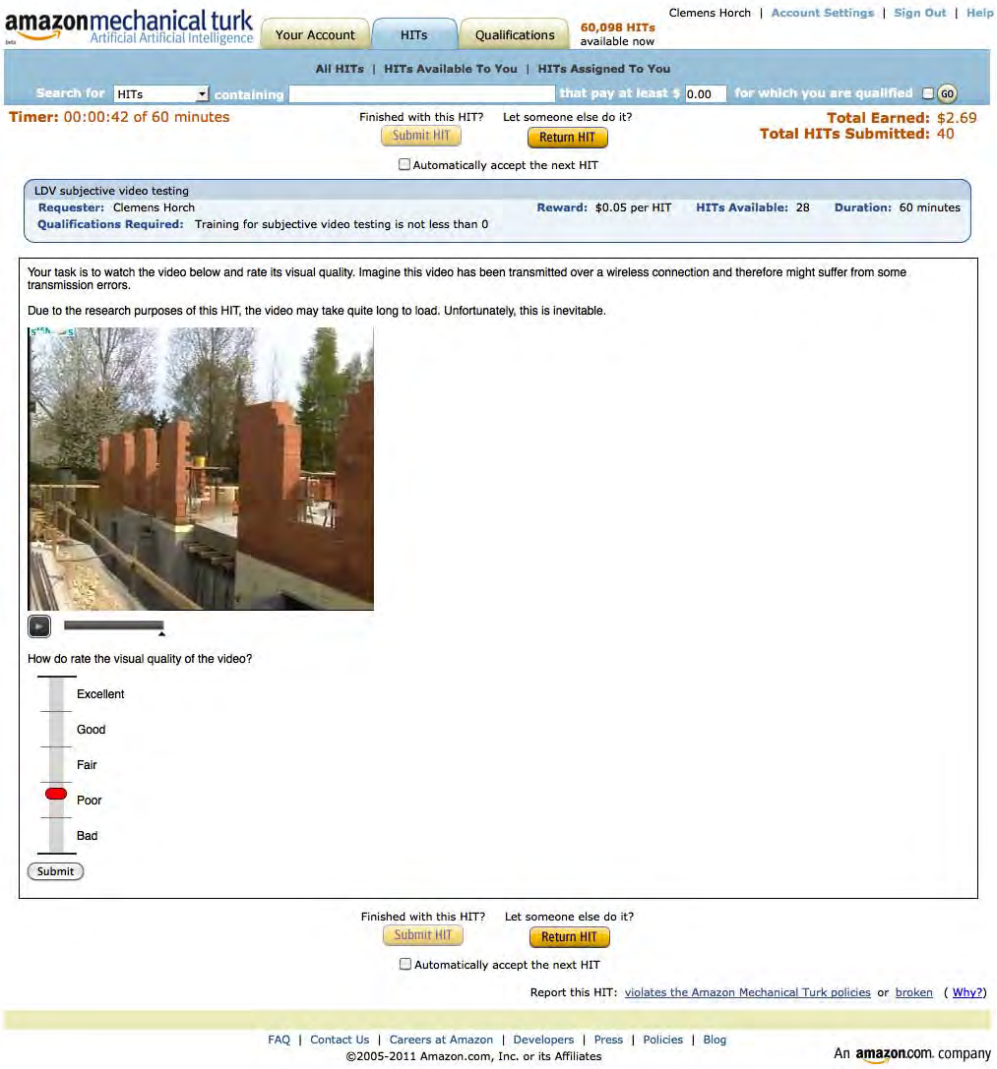

Abbildung B.9.: *HIT* mit Videotest in *Mechanical Turk*

Literaturverzeichnis

1. Adobe Systems Incorporated. *List of codecs supported by Adobe Flash Player*, Dezember 2007. URL `http://kb2.adobe.com/cps/402/kb402866.html`.

2. Adobe Systems Incorporated. *SWF File Format Specification (Version 10)*, November 2008.

3. Adobe Systems Incorporated. *Adobe Flash Video File Format Specification (Version 10.1)*, August 2010.

4. Adobe Systems Incorporated. *Flash Player Version Penetration*. Adobe Systems Incorporated, Dezember 2010. URL `http://www.adobe.com/products/player_census/flashplayer/version_penetration.html`.

5. Amazon.com, Inc. *Amazon Mechanical Turk Requester FAQ*. URL `https://requester.mturk.com/mturk/help?helpPage=policies`.

6. Cake Software Foundation, Inc. *The CakePHP 1.3 Book - 3.1 Requirements*, Februar 2010.

7. K.T. Chen, C.J. Chang, C.C. Wu, Y.C. Chang und C.L. Lei. Quadrant of euphoria: a crowdsourcing platform for qoe assessment. In *Network, IEEE*, 24(2), S. 28 –35, März-April 2010.

8. K.T. Chen, C.C. Wu, Y.C. Chang und C.L. Lei. A crowdsourceable qoe evaluation framework for multimedia content. In *Proceedings of the 17th ACM international conference on Multimedia*, MM '09, S. 491–500. ACM, 2009.

9. F. De Simone, M. Naccari, M. Tagliasacchi, F. Dufaux, S. Tubaro und T. Ebrahimi. Subjective assessment of H.264/AVC video sequences transmitted over a noisy channel. In *Proceedings of the First International Workshop on Quality of Multimedia Experience (QoMEX 2009)*. Juli 2009.

10. F. De Simone, M. Tagliasacchi, M. Naccari, S. Tubaro und T. Ebrahimi. H.264/AVC video database for the evaluation of quality metrics. In *Proceedings of the 35th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2010)*. März 2010.

11. J. Howe. The rise of crowdsourcing the rise of crowdsourcing the rise of crowdsourcing the rise of crowdsourcing. In *Wired*, Juni 2006. URL `http://www.wired.com/wired/archive/14.06/crowds.html`.

12. P. Ipeirotis. Demographics of mechanical turk. In *New York University, CeDER Working Papers*, März 2010.

13. ITU-R. *BT.500: Methodology for the subjective assessment of the quality of television pictures (ITU-R Recommendation BT.500-12)*. International Telcommunications Union, November 2009.

14. ITU-T. *P.910: Subjective video quality assessment methods for multimedia applications (ITU-T Recommendation P.910)*. International Telcommunications Union, April 2008.

15. ITU-T. *H.264: Advanced video coding for generic audiovisual services*. International Telcommunications Union, Oktober 2010.

16. M. Jazayeri. Html video codec support in chrome. In *The Chromium Blog*, Januar 2011. URL `http://blog.chromium.org/2011/01/html-video-codec-support-in-chrome.html`.

17. M. Marge, S. Banerjee und A. Rudnicky. Using the amazon mechanical turk for transcription of spoken language. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, S. 5270 –5273. März 2010.

18. Microsoft Corporation. *Supported Media Formats, Protocols, and Log Fields*. URL `http://msdn.microsoft.com/en-us/library/cc189080(v=vs.95).aspx`.

19. G. Paolacci, J. Chandler und P. Ipeirotis. Running experiments on amazon mechanical turk. In *Judgment and Decision Making*, Vol. 5(No. 5), S. 411–419, Juni 2010.

20. J. Pontin. Artificial intelligence, with help from the human. In *New York Times*, März 2007. URL `http://www.nytimes.com/2007/03/25/business/yourmoney/25Stream.html`.

21. A. Reininger. *Wolfgang von Kempelen, eine Biografie*. Praesens-Verlag Wien, 2007.

22. J. Ross, L. Irani, M.S. Silberman, B. Tomlinson und A. Zaldivar. Who are the crowdworkers? shifting demographics in mechanical turk. In *alt.CHI session of CHI 2010 extended abstracts on human factors in computing systems*, S. 2863–2872, April 2010.

23. World Wide Web Consortium. *HTML5 (W3C Working Draft 13 January 2011)*, Januar 2011. URL `http://www.w3.org/TR/html5/video.html`.