

# Innovative Interaktionstechniken für einen Multi-Display-Arbeitsplatz zur Bildauswertung

Bachelorarbeit  
von

Tim Reiter

An der Fakultät für Informatik  
Fraunhofer IOSB (IAD)

Erstgutachter:	Prof. Dr.-Ing. habil. Jürgen Beyerer
Zweitgutachter:	Prof. Dr. rer. nat. Hartwig Steusloff
Betreuender Mitarbeiter:	MSc. Sebastian Maier

Bearbeitungszeit: 01.10.2014 – 31.01.2015



# Zusammenfassung

Computersysteme mit mehreren Bildschirmen sind inzwischen weit verbreitet. Multimonitorsysteme werden oft in speziellen Arbeitsplätzen eingesetzt, bei denen es wichtig ist, viele Informationen und Daten gleichzeitig darstellen zu können. Dies können beispielsweise Arbeitsplätze zur Analyse von Finanzmärkten oder zur Auswertung von Luftbildern sein. Die traditionelle Steuerung durch Maus und Tastatur weist jedoch Probleme auf. So müssen etwa oft lange Wege mit der Maus zurückgelegt werden um bestimmte Aktionen durchzuführen. Auch die üblichen Betriebssysteme bieten wenige für Multimonitorsysteme konzipierte Steuerungsmöglichkeiten.

Gleichzeitig werden viele alternative Sensoren zur Computerinteraktion erforscht und sind teilweise schon marktreif. Diese Arbeit beschäftigt sich mit der Entwicklung innovativer Interaktionstechniken unter Einsatz von Sensoren zur Erkennung von Kopfdrehung und Zeigegesten um die Bedienung eines Multi-Display-Arbeitsplatzes zu beschleunigen und einfacher zu gestalten. Die untersuchten Interaktionstechniken lassen sich auf beliebigen Multimonitorsystemen einsetzen und wurden am Beispiel eines Systems zur Bildauswertung implementiert. Außerdem werden aktuelle Abläufe und Techniken zur Bildauswertung erklärt und weitere Forschungsgebiete zur Verbesserung dieser erläutert.



# Abstract

Computer systems with multiple monitors are now widespread. Multiple-monitor systems offer the advantage of being able to display a lot of information and data simultaneously. These systems are often used in specific jobs, such as for analysis of financial markets or for image analysis. However, conventional human-computer-interaction by mouse and keyboard can decrease efficiency when dealing with multiple monitors. For example, users often have to traverse long distances with the mouse to be able to perform certain actions. In addition, most operating systems provide little alternative interaction methods.

On the other hand, the recent progress in sensor technology offers new alternatives. In this thesis, we investigate innovative interaction techniques using sensors detecting head rotation and pointing gestures to accelerate and simplify the operation of a multi-display workstation. The investigated interaction techniques can be used on any multi-monitor systems and have been implemented on a system for image analysis. Furthermore, we explain current techniques for image analysis and show several possible areas of research to improve efficiency.



# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>9</b>
1.1. Bildauswerteplatz der Zukunft . . . . .	9
<b>2. Ansätze zur Weiterentwicklung</b>	<b>13</b>
2.1. Ablauf der Bildauswertung . . . . .	13
2.2. Ergebnisse der Diskussion über den Bildauswerteplatz der Zukunft . . . . .	15
2.3. Erweiterung des Bildauswerteplatzes . . . . .	16
2.3.1. Vergleich zwischen Stereobildauswerter und aktueller Software zur Annotation . . . . .	16
2.3.2. Evaluation des Fragebogens zur Bildauswertung . . . . .	16
<b>3. Design und Entwicklung der Interaktionstechniken</b>	<b>19</b>
3.1. Problemstellung . . . . .	19
3.2. Verwandte Arbeiten . . . . .	20
3.3. Lösungsansätze . . . . .	22
3.3.1. Multimonitormenü zum Wechseln von Programmen . . . . .	22
3.3.2. Positionierung und Verschieben von Fenstern . . . . .	23
3.3.3. Automatische Mauspositionierung . . . . .	24
3.4. Implementation . . . . .	26
3.4.1. DisplayManager . . . . .	27
3.4.1.1. Erkennen der Monitor-Konfiguration . . . . .	27
3.4.1.2. DisplayMapper . . . . .	28
3.4.1.3. MousePositionOnDisplayManager . . . . .	28
3.4.2. GestureManager . . . . .	29
3.4.2.1. Listener . . . . .	29
3.4.2.2. Vertauschen von Fenstern . . . . .	29
3.4.2.3. Task-Switch-Menü . . . . .	30
3.4.3. HeadposeManager . . . . .	31
3.4.3.1. IntersectionManager . . . . .	31
3.4.3.2. Betrachten der Tastatur . . . . .	33
<b>4. Verbesserungsmöglichkeiten für zukünftige Arbeiten</b>	<b>35</b>
<b>Anhang A. Fragebogen zur Erweiterung des Bildauswerteplatzes</b>	<b>39</b>
<b>Literaturverzeichnis</b>	<b>47</b>





# 1. Einleitung

In diesem Kapitel wird zunächst das Multimonitorsystem zur Bildauswertung vorgestellt. Kapitel 2 erläutert den Ablauf eines Auftrags zur Bildauswertung und nennt mehrere mögliche Forschungsbereiche, mit denen ein Bildauswertesystem verbessert werden könnte. Schließlich beschreibt Kapitel 3 die hier entwickelten Implementationstechniken und Kapitel 4 zeigt Verbesserungsmöglichkeiten für zukünftige Arbeiten.

## 1.1. Bildauswerteplatz der Zukunft

Der Bildauswerteplatz der Zukunft ist ein Projekt zur Entwicklung eines Arbeitsplatzes für Bildauswerter, in dem alle zur Aufgabenbearbeitung erforderlichen Werkzeuge bündig zur Verfügung gestellt werden. Hierbei handelt es sich um militärische Bildauswertung, bei der vor allem Luftbildaufnahmen ausgewertet werden. Durch neue Interaktionstechniken soll die Bedienung am Bildauswerteplatz erleichtert und die Arbeitsgeschwindigkeit erhöht werden.

Die Werkzeuge stellen dabei folgende Funktionen bereit:

- **Bilddarstellung und -annotation:**  
Auszuwertende Bilder in verschiedenen Formaten lassen sich darstellen und mit Annotationen versehen. Annotationen umfassen dabei verschiedene grafische Elemente wie Linien, Pfeile, Polygone oder Freihandzeichnungen sowie Text und aufnahmeppezifische Informationen wie einen Kompass oder einen Maßstab.
- **Kartendarstellung:**  
Ein Werkzeug zur Kartendarstellung kann den Aufnahmeort und dessen Umgebung visualisieren und so zusätzliche Informationen zur Bildauswertung bereitstellen.
- **Berichterstellung:**  
In einem Programm zur Berichterstellung dokumentiert der Bildauswerter die Analyse der auszuwertenden Bilder. Der Bericht stellt das Ergebnis der Bildauswertung dar.
- **Erkennungsassistentz:**  
Eine Erkennungsassistentz hilft dem Bildauswerter bei der Identifikation bestimmter Objekte des Bildes. Beispielsweise kann der Bildauswerter die genaue Modellnummer eines Schiffes im Bild identifizieren indem er spezielle Informationen wie die Anzahl der Schlotte oder die Form des Schiffes angibt.

Des Weiteren kommen allgemeine Programme wie Webbrowser zur Darstellung von zusätzlichen Informationen zum Einsatz.

Um alle benötigten Informationen für eine effektive und effiziente Bildauswertung darstellen zu können benutzt der Bildauswerteplatz der Zukunft eine Multimonitorkonfiguration, welche aus vier "Full HD"-Bildschirmen besteht. Eine Besonderheit sind der mittlere Monitor, welcher mithilfe einer Polarisationsbrille stereoskopische 3D-Darstellungen ermöglicht, sowie der untere Bildschirm, der touch-fähig ist. Die Bildannotationssoftware Stereobildauswerter (SBA) unterstützt die Anzeige von stereoskopischen Bildern. Stereoskopische Bilder sind insbesondere bei Aufnahmen aus der Luft für den Bildauswerter nützlich, da so Informationen über die Höhe von einzelnen Objekten wie z.B. Gebäuden besser sichtbar sind. Fingereingaben auf dem Touchscreen können zum Bedienen des Programms zur Kartendarstellung genutzt werden. So kann die Karte mit einem Finger bewegt und mit Zwei-Finger-Gesten vergrößert bzw. verkleinert werden.

Da die drei Programme zur Kartendarstellung, Bildannotation und Erkennungsassistentz eine sehr hohe Anzahl an Informationen visualisieren müssen und Bildauswerter diese Tools simultan benutzen, werden diese, wie Abbildung 1.1 zeigt, jeweils auf einem eigenen Bildschirm im Vollbildmodus angezeigt. Der vierte Monitor wird zur Darstellung von Webbrowsern und Erstellung von Berichten verwendet.



Abbildung 1.1.: Der Bildauswerteplatz der Zukunft

Die Bedienung des Bildauswertepplatzes der Zukunft erfolgt hauptsächlich durch Maus und Tastatur. Zusätzlich werden neue Bedienungskonzepte eingesetzt um die Schwächen, die eine traditionelle Bedienung in einem großen Multimonitorsystem aufweist, zu kompensieren und den Arbeitsablauf zu vereinfachen. Wie in Abbildung 1.2 dargestellt werden dazu zwei Microsoft Kinect-Kameras eingesetzt, welche übereinander über dem mittleren Bildschirm platziert werden.



Abbildung 1.2.: Der Bildauswertepplatz mit zwei Kinect-Kameras zur Erkennung von Zeigegesten sowie zum Erfassen der Kopfdrehung

Die untere Kinect-Kamera betrachtet die Kopfdrehung des Nutzers und berechnet, welchen Monitor der Nutzer betrachtet. Die obere Kamera ist nach unten gerichtet, um Zeigegesten auf die Bildschirme erkennen zu können. Zeigt der Nutzer mit seinem Finger auf einen Bildschirm, wird mithilfe der oberen Kamera berechnet auf welchen Monitor und welche grobe Position auf diesem der Nutzer zeigt. Ein separater Computer führt diese Berechnungen durch und sendet die Ergebnisse an den Bildauswertepplatz.

Hauptbestandteil dieser Arbeit ist der Entwurf und die Implementation eines Systems, das diese Daten zur intuitiven und effizienten Bedienung des Bildauswertepplatzes der Zukunft einsetzt. Um weitere relevante Forschungs- und Weiterentwicklungsthemen zu erfassen wurde zunächst eine Diskussion mit Endanwendern des Systems durchgeführt. Als Grundlage dafür wurde der zu Beginn dieser Arbeit aktuelle Stand des Projekts präsentiert und ein Fragebogen über die Relevanz möglicher zukünftiger Funktionen erstellt.



## 2. Ansätze zur Weiterentwicklung

Um mögliche Erweiterungen des Bildauswerteplatzes der Zukunft zu erfassen, wurde eine Diskussion mit Bildauswertern durchgeführt. Hierbei wurde zuerst anhand einer Demonstration der aktuelle Stand des Bildauswerteplatzes vorgestellt und anschließend über mögliche Erweiterungen diskutiert. Ein Fragebogen half dabei, die Relevanz zukünftiger Funktionen des Stereobildauswerters (SBA) zu erfassen. Im Folgenden werden zunächst der grobe Ablauf eines Auftrags zur Bildauswertung skizziert, dann die Ergebnisse der Diskussion vorgestellt und abschließend die Möglichkeiten zur Verbesserung der Bildauswertesoftware evaluiert.

### 2.1. Ablauf der Bildauswertung

Zunächst wird der Arbeitsablauf eines von Bildauswertern durchgeführten Auftrags beschrieben. Das Team besteht dabei aus mehreren Bildauswertern und einem „Mission Manager“ (auch „Cutter“ genannt). Dessen Aufgaben sind sowohl die Bildauswertung als auch die Führung und Qualitätssicherung des Auftrags.

1. Der Mission Manager erstellt aus dem Aufklärungsauftrag (Air Tasking Order ATO) einen Flug- und Aufnahmeplan für RecceLite ([rl]), ein System zur Aufnahme von Luftbildern. Die ATO enthält unter anderem Informationen zu Einstufung, Flugplan, Missionsnummer, Zielnummer, zu verwendenden Sensor und „time over target“ (Zeit, wann ein Ziel erreicht werden soll). Die Software der Ground Exploitation Station (GES) zeigt eine Karte an auf welcher ein Korridor eingezeichnet wird, der den Flugpfad des Flugzeugs angibt. Hier wird anhand einer aus der ATO erstellten Prioritized Target List (PTL) markiert, an welchen Bereichen und in welcher Reihenfolge eine Aufnahme erstellt werden soll.
2. In einem Briefing vor der Durchführung des Flugs werden Details wie unter anderem die aktuelle Wetterlage besprochen und anschließend die Mission durchgeführt. Wie Abbildung 2.1 zeigt werden die Bilder während des Fluges automatisch durch RecceLite aufgenommen, wenn der Pilot in die vordefinierten Korridore einfliegt. Hierbei wird unter anderem der Fokus der Kamera automatisch aufgrund der aktuellen Höhe und der bekannten Höhe des Grundes gesetzt. Eine manuelle Aufnahme ist zusätzlich durch einen sogenannten Waffensystemoffizier (WSO) möglich.

Die aufgenommenen Bilder lassen sich bei bestehender Sichtverbindung in einem Radius von 250km via Downlink-Richtfunk fast in Echtzeit übertragen. Außerdem werden die Bilder auf einem SSD-Speicher gespeichert, der sich nach dem Flug auslesen lässt. Per Uplink kann die Mission vom Mission Manager aktualisiert werden. RecceLite kann zusätzlich per Mosaikerstellung mehrere aufgenommene Bilder zu einem großen zusammensetzen. Dadurch lassen sich große Gebiete in einem Bild darstellen.



Abbildung 2.1.: Aufnahme von Bildern inklusive automatischer Mosaikerstellung.

Quelle: [http://www.rafael.co.il/marketing/SIP\\_STORAGE/FILES/5/955.pdf](http://www.rafael.co.il/marketing/SIP_STORAGE/FILES/5/955.pdf), Stand: 24.10.2014

3. Die beim Flug erstellten Bilder werden vom Mission Manager gesichtet. Auszuwertende Bilder und Teilbereiche werden ausgewählt und verschiedenen Bildauswertern zugeteilt.
4. Die Bildauswerter werten die ihnen zugewiesenen Bilder aus, indem sie die Bilder annotieren und einen Bericht erstellen. Zur Bildannotation werden unter anderem Photoshop ([ps]) und AbleTiff ([at]) verwendet. Vor allem Photoshop bietet eine Vielzahl an Bildkorrekturfunktionen, mit denen sich Details aus Bildern herausstellen lassen, die vorher nicht sichtbar waren. Verschiedene Einzeichnungen, wie beispielsweise Linien, Rechtecke, Ellipsen, Text und Abstände, lassen sich in das Bild einfügen um so wichtige Stelle hervorzuheben. Problematisch ist, dass einige zur Annotation eingesetzte Programme keine Geoinformationen, also Informationen, die mit einer auf die Erde bezogenen Position verbunden sind, unterstützen. Diese wichtigen Geoinformationen eines Bildes, welche die genaue Größe und Position der Aufnahme beschreiben, gehen damit verloren. Ein weiteres Manko ist die fehlende Unterstützung von Stereobildern. Diese können von RecceLite zwar erstellt werden, werden aber zurzeit selten verwendet. Zur Berichterstellung wird meist das Format RECCEXRep ([xre]) oder PowerPoint ([pp]) bzw. Word ([wrđ]) genutzt.
5. Die Berichte inklusiver annotierter Bilder werden zum Schluss vom Mission Manager auf Konsistenz und Auftragserfüllung geprüft und dann an den Auftraggeber weitergereicht.

## 2.2. Ergebnisse der Diskussion über den Bildauswerteplatz der Zukunft

Teil der Diskussion war nicht nur die Erweiterung von bestehenden Programmen des Bildauswerteplatzes, sondern auch die Entwicklung und Erforschung neuer Programme, welche den Bildauswerter in verschiedenen Schritten des Ablaufs unterstützen. Diskutiert wurde über Systeme zu Flugplanung, Mission Management und Überwachung.

- **Flugplanung:**

Bei der Diskussion über den Bildauswerteplatz der Zukunft stellte sich heraus, dass man für ein System zur Flugplanung neue Funktionen erforschen könnte. Gewünscht wurden vor allem folgende Funktionen:

- **Anzeige einer Wolkenuntergrenze:**

Vollständige Wetterinformationen werden bei der Flugplanung nicht benötigt, da diese beim Briefing mit dem Piloten vor dem Flug besprochen werden. Eine Anzeige der Wolkenuntergrenze auf der Karte wäre jedoch schon bei der Flugplanung nützlich, da die Qualität der aufgenommenen Bilder drastisch sinkt, wenn die Aufnahmen oberhalb der Wolken erfolgen.

- **Service Orders of Battle:**

Ein Warnsystem, welches Positionen bekannter Luftabwehrstationen inklusive Reichweiten darstellt, wäre eine sinnvolle Funktion um die Gefahr eines Flugzeugabschlusses zu minimieren.

- **Anzeige von Ländergrenzen:**

Eine deutliche Anzeige von Ländergrenzen würde das Risiko einen Flugpfad über Ländergrenzen hinweg zu planen minimieren.

Solche Features ließen sich sehr gut als Erweiterung einer Software zur Kartendarstellung implementieren und somit in den Bildauswerteplatz der Zukunft integrieren.

- **Mission-Management-Tool:**

Die aktuelle Version des Bildauswerteplatzes der Zukunft richtet sich an die Bildauswertung. Während der Diskussion wurde die Idee zu einem System für den Mission Manager besprochen. Ziel dieses Tools wäre die effiziente und übersichtliche Führung von Aufklärern, mit dem sich Aufgaben verteilen, Ergebnisse kontrollieren und der aktuelle Stand der Auswertung überprüfen ließen. Das Tool sollte alle wichtigen Informationen einer Mission anzeigen. Diese beinhalten unter anderem, welche Bilder zu einer Mission vorhanden sind, von welchem Piloten wann mit welchem System die Aufnahmen erfolgt sind, Prioritätsstufen verschiedener Bilder und die PTL inklusive IDs der einzelnen Ziele und Missionen. Zur Verteilung der Aufgaben an verschiedene Bildauswerter wäre zudem ein gesten- und/oder touchscreenbasiertes System denkbar, mit dem man Aufgaben per Wisch den Bildauswertern zuteilen könnte. Da der Mission Manager meistens auch Bildauswertung betreibt, wäre es sinnvoll, das Mission-Management-Tool in den Bildauswerteplatz der Zukunft zu integrieren, sodass schnell zwischen Mission-Management und Bildauswertung umgeschaltet werden kann.

- **”Patterns of life”- Szenario:**

Ein weiterer möglicher Teil der Bildauswertung ist das sogenannte „patterns of life“-Szenario. Dies beschreibt die mehrere Stunden andauernde Überwachung eines Gebietes durch ein unbemanntes Luftfahrzeug (engl. unmanned aerial vehicle, kurz UAV), beispielsweise vor einem Eingriff durch Bodentruppen. Aufgabe der Bildauswerter ist es, alle wichtigen Informationen aus der Luftbilddaufnahme, in diesem Fall ein Full Motion Video (FMV), zu extrahieren. Dies beinhaltet unter anderem die Anzahl der Personen in dem Gebiet, die Bewegung von Personen und ob möglicherweise bestimmte Stellen des Gebietes von Personen vermieden werden. Zurzeit werden diese Informationen in Programmen zur Tabellenkalkulation wie Microsoft Excel ([xcl]) hinterlegt. Die Entwicklung eines speziellen Systems für ein solches Szenario könnte die Auswertung beschleunigen und vereinfachen.

## 2.3. Erweiterung des Bildauswerteplatzes

### 2.3.1. Vergleich zwischen Stereobildauswerter und aktueller Software zur Annotation

Wie in 2.1 beschrieben unterstützen nur wenige aktuelle Softwareprodukte, die zur Bildannotation verwendet werden, Stereobilder. Zudem fehlt bei vielen Programmen, wie z.B. Photoshop, die Unterstützung von Geodaten. Die aktuelle Version des Annotationstools am Bildauswerteplatz der Zukunft, Stereobildauswerter (SBA) genannt, unterstützt sowohl das Arbeiten mit Geoinformationen als auch die Anzeige von Stereobildern. Vorhandene Geodaten eines Bildes werden stets angezeigt und auch für Annotationsmöglichkeiten wie Kompass und Maßstab verwendet. Genaue Längen- und Breitengrade einer Position lassen sich zudem direkt herauskopieren und müssen nicht, wie zum Beispiel bei AbleTiff, mühsam abgetippt werden. Die Möglichkeit, Längen- und Breitengrade direkt in einen Bericht übernehmen zu können und nicht mit der Gefahr eines Fehlers abschreiben zu müssen, wurde als Feature gewünscht. Für Bilder, die keine Georeferenzierung besitzen, bietet SBA die Möglichkeit manuell Geodaten hinzuzufügen. Während die Anzeige von Stereobildern und die Unterstützung von Geodaten große Vorteile des SBA sind, die in aktueller Software (teilweise) nicht vorhanden sind, gibt es auch eine Reihe an fehlenden Features. Diese umfassen unter anderem Funktionen zur Bildkorrektur, wie z.B. die Einstellung von Kontrast, Helligkeit, FFT (Fast Fourier Transformationen), Sättigung, Schärfe und Histogrammfunktionen wie Spreizung oder Dehnung. Dazu kommen weitere Annotationstools, wie etwa das Einzeichnen von Winkeln, taktischen Zeichen, Funktionen zum Messen von Flächen und Bildfilter, wie z.B. Rauschentfernung oder Kantenerkennung.

### 2.3.2. Evaluation des Fragebogens zur Bildauswertung

Um zu evaluieren, wie der Bildauswerteplatz weiterentwickelt werden kann wurde ein Fragebogen entwickelt (siehe Anhang A). Dieser enthält eine Reihe an möglichen zukünftigen Features. Die Bildauswerter wurden gebeten, die vorgestellten Features nach Nützlichkeit auf einer Skala von 1 (sehr hilfreich) bis 5 (irrelevant) zu bewerten. Insgesamt wurden drei Fragebögen ausgefüllt, wobei ein Fragebogen von einer Person, die kein Bildauswerter ist, beantwortet wurde. Da nur drei Fragebögen vorliegen, können keine statistischen Aussagen getroffen werden. Die Antworten des Nicht-Bildauswerter wurden bei der Evaluation schwächer gewichtet, da sie teilweise stark von den anderen Antworten abweichen. Dennoch wurden fast alle Fragen mehrheitlich mit „sehr hilfreich“ bewertet. Somit scheinen fast alle vorgestellten Features sehr hilfreiche Verbesserungen zu sein, die entweder eine intuitivere und schnellere Bedienung ermöglichen oder zusätzliche Möglichkeiten zur Bildannotation zur Verfügung stellen. Zusätzlich wurde auf Grundlage einer Diskussion mit den Bildauswertern die Forschungsrelevanz der einzelnen Funktionen abgeschätzt.



Nicht alle Features sind für das Forschungsprojekt des Bildauswerteplatzes der Zukunft wichtig. Beispielsweise ist die Unterstützung von allen gängigen Koordinatensystemen und Exportformaten zwar für ein kommerzielles Programm Pflicht, für die Forschung jedoch irrelevant. Als besonders forschungsrelevant oder nützlich wurden folgende Funktionen erkannt:

- **Für Touchscreens optimierte Bedienung der Bildauswertung:**

Viele Nutzer bevorzugen Touchscreens anstelle von Mauseingaben um Zeichnungen am Computer zu erstellen, da die Eingabe durch Finger oder Stift auf einem Touchscreen sehr ähnlich ist wie das Zeichnen mit Stift und Papier. Freihandzeichnungen könnten so schneller und intuitiver angefertigt werden.

- **Verbesserung der Interaktion**

Traditionelle Steuerung eines Computers mit Maus und Tastatur weist verschiedene Probleme bei Systemen mit vielen Monitoren auf. Viele alternative Eingabemöglichkeiten durch Einsatz verschiedener Sensorik werden erforscht und sind teilweise schon marktreif. So gibt es etwa verschiedene Produkte zur Erkennung von Personen inklusive Positionen von Kopf, Beinen, Händen und Fingern. Zur Verbesserung der Interaktion in Multimonitorsystemen könnten solche Eingabemethoden genutzt werden.

- **Gestengesteuerte Funktionen:**

Gestenbasierte Eingabemethoden könnten nicht nur zur Steuerung des Betriebssystems verwendet werden, sondern auch für die Steuerung und Kommunikation zwischen speziellen Programmen. Da Bildauswerter mehrere verschiedene Programme gleichzeitig nutzen, wäre die Entwicklung von Gesten, welche die Datenübertragung zwischen diesen Programmen erleichtern, interessant. So ließe sich beispielsweise durch eine Handgeste ein mit Annotationen und Einzeichnungen versehenes Bild direkt in das Programm zur Erstellung des Berichts einfügen.

- **Anbindung an Coalition Shared Data Server ([CSD])**

Ergebnisse und Daten zur Bildauswertung werden aktuell häufig auf getrennten Systemen gespeichert und lassen sich somit nur umständlich von mehreren Nutzern verarbeiten. Die CSD ist eine Datenbanklösung, mit welcher Aufklärungsergebnisse von mehreren Nutzern gleichzeitig genutzt werden können. Aufnahmen ließen sich beispielsweise direkt aus der CSD laden und Ergebnisse könnten dort gespeichert werden. Dadurch würde die Komplexität der Koordination zwischen mehreren Bildauswertern verringert werden.

- **Algorithmen zur Veränderungserkennung mehrerer Bilder:**

Wenn mehrere Bilder des gleichen Ortes zu unterschiedlichen Zeitpunkten aufgenommen wurden, ist es für Bildauswerter vor allem wichtig, Veränderungen zwischen den Bildern zu erkennen. Um diesen Prozess zu automatisieren benötigt man einen Algorithmus, welcher eine Menge von Pixeln liefert, die in zwei oder mehr Bildern signifikant unterschiedlich sind. Hauptschwierigkeit ist dabei, wichtige von unwichtigen Veränderungen zu trennen. Unwichtige Veränderungen sind Pixel, die beispielsweise aufgrund von Rauschen, Veränderung der Beleuchtung oder Kamerabewegung verändert wurden. Verfahren zur automatischen Veränderungserkennung (engl. „image change detection“) werden unter anderem in [RAAKR05] beschrieben.

- **Schnelle Bearbeitung mehrerer Bilder:**

Bei manueller Erkennung von Veränderungen und Unterschieden in Bildern muss der Nutzer zwischen Bildern schnell und ohne Störung umschalten können. Wenn bei Aufnahmen desselben Ortes zu verschiedenen Zeitpunkten die Aufnahmezeiten bekannt sind, könnte dem Nutzer beispielsweise eine Zeitleiste angeboten werden, mit welcher er zwischen den Bildern schnell umschalten kann.

- **Darstellung zusätzlicher Meta-Informationen:**

Öffentliche zugängliche Informationen aus dem Internet könnten dem Bildauswerter zur Verfügung gestellt werden. Dies könnten zum Beispiel Daten über das Wetter oder die Verkehrslage des Aufnahmeorts sein.

- **3D-Rekonstruktion aus Stereobildern:**

Stereobilder enthalten Tiefeninformationen, welche jedoch nur auf einem 3D-fähigen Monitor vom Nutzer erfasst werden können. Außerdem können Stereobilder nur aus dem Aufnahmewinkel betrachtet werden und lassen sich nicht wie ein 3D-Objekt frei drehen. Deswegen wäre es nützlich 3D-Daten aus dem Stereobild zu lesen und zur Erstellung eines 3D-Objekts zu nutzen. Verschiedene Verfahren dazu werden in [SCD<sup>+</sup>06] vorgestellt und verglichen. Bei der Rekonstruktion können jedoch nie alle 3D-Informationen des Aufnahmeorts rekonstruiert werden, da pro Pixel nur eine Höheninformation extrahiert werden kann. Deshalb können beispielsweise Überdeckungen nicht erkannt werden.

- **3D-Daten zur Auswertung:**

Aufnahmen, die Geodaten enthalten, könnten mit zusätzlichen 3D-Daten kombiniert werden. Aus einer Datenbank könnte ein 3D-Modell des Terrains geladen werden, auf welches die Aufnahme wie in Abbildung 2.2 projiziert wird. Falls vorhanden, könnten auch 3D-Objekte wie Gebäude hinzugefügt werden. Mit dieser Technik könnten Bildern, die keine Stereoinformationen enthalten, Höheninformationen hinzugefügt werden. Zudem ließe sich das Modell aus allen Winkeln frei betrachten und könnte so dem Bildauswerter zusätzliche Informationen über die Umgebung liefern.

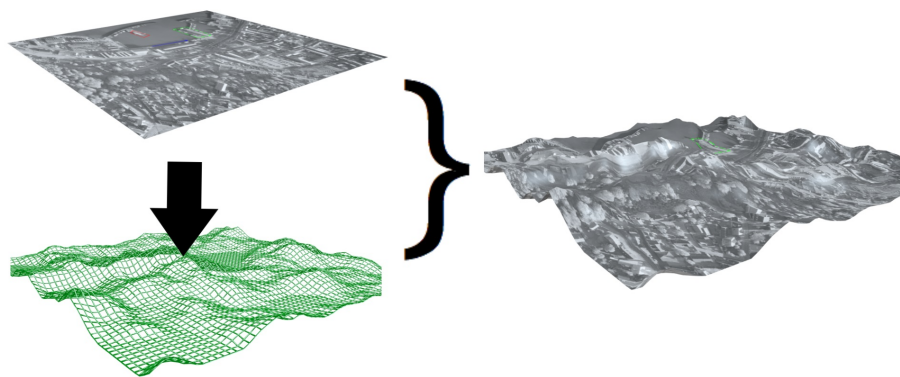


Abbildung 2.2.: Das Bild wird auf ein 3D-Modell des entsprechenden Terrains projiziert.

In dieser Arbeit wurde sich auf die Verbesserung der Interaktion konzentriert, welche in den folgenden Kapiteln erläutert wird. In 3.1 wird die Problemstellung genauer erklärt, 3.2 zeigt verwandte Arbeiten und 3.3 stellt das eigene System der Interaktionstechniken vor, dessen Umsetzung in 3.4 erläutert wird. Zuletzt wird in 4 ein Ausblick über mögliche Verbesserungen in zukünftigen Arbeiten gegeben.

## 3. Design und Entwicklung der Interaktionstechniken

### 3.1. Problemstellung

Der Bildauswerteplatz der Zukunft stellt mit seinen vier Monitoren und der Bedienung von Vollbildprogrammen neue Herausforderungen an die Interaktion. Weit verbreitete Betriebssysteme wie Windows sind nicht für die Bedienung solcher Arbeitsplätze ausgelegt und traditionelle Steuerung durch Maus und Tastatur weist Probleme auf. Folgende Probleme, die sich durch Einsatz alternativer Interaktionstechniken verbessern ließen, wurden identifiziert:

- Multimonitorsysteme haben im Vergleich zu Konfigurationen mit nur einem Bildschirm einen vielfach größeren Raum, in dem die Maus bewegt werden kann. Dadurch müssen wesentlich größere Distanzen mit der Maus zurückgelegt werden, was beeinträchtigte Produktivität und Komfort zur Folge hat. Erhöhung der Zeigergeschwindigkeit und -beschleunigung zur Kompensation der größeren Distanzen sind für viele Nutzer nicht praktikabel, da dann sehr viel Feingefühl notwendig ist, um gezielt bestimmte Bereiche auf dem Bildschirm auswählen zu können.
- Multimonitorsysteme kommen meistens in Situationen zum Einsatz, in denen viele Informationen gleichzeitig sichtbar sein müssen und somit viele Programme gleichzeitig ausgeführt werden. Dies führt zu einer oftmals hohen Anzahl an geöffneten Fenstern. Einige Betriebssysteme bieten jedoch nur Taskswitch-Programme, also Programme, mit denen das aktive Fenster gewechselt werden kann, die nicht für mehrere Monitore optimiert sind. Unter Windows 7 wird zum Beispiel nur ein Menü auf dem Hauptbildschirm angezeigt, welches sämtliche geöffneten Fenster anzeigt. Eine bessere Multimonitorunterstützung bietet in der Hinsicht OS X Maverick, welches im "Mission Control" genannten Programm auf jedem Monitor ein eigenes Menü anzeigt. Wie in Abbildung 3.1 zu sehen werden in jedem Menü jeweils nur diejenigen Programme angezeigt, die auf diesem Bildschirm geöffnet sind. Jedoch müssen auch in "Mission Control" teilweise lange Wege mit der Maus zurückgelegt werden um das gewünschte Fenster zu aktivieren. Eine gestenbasierte Steuerung könnte die Zeit zum Wechseln von Programmen minimieren.

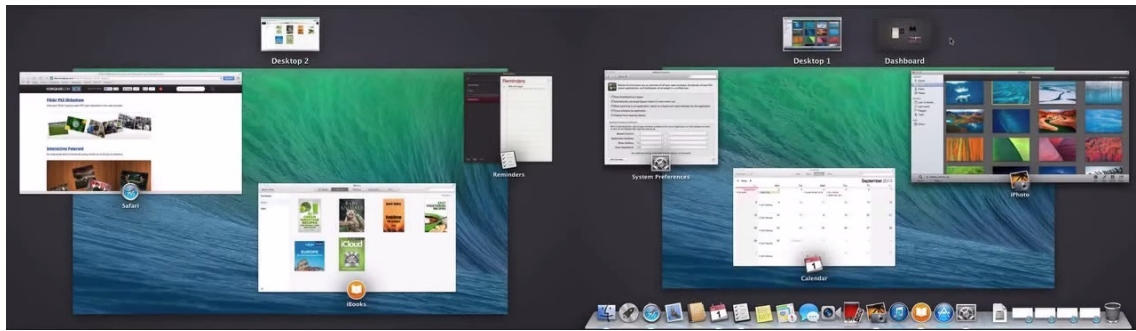


Abbildung 3.1.: "Mission Control" unter OS X Mavericks.

Quelle: <http://youtu.be/2iSU9B1aSBg?t=5m45s>, Stand: 22.01.2015

- Der Bildauswerteplatz der Zukunft ist vorrangig für die Bedienung von Programmen im Vollbildmodus ausgelegt. Auf einem Monitor ist somit meist nur ein Programm sichtbar. Die Neuordnung von Vollbildprogrammen auf mehreren Bildschirmen wird von aktuellen Betriebssystemen nur rudimentär unterstützt. Manuelles Tauschen von zwei Vollbildprogrammen ist relativ zeitaufwändig, da zunächst bei traditioneller Steuerung jeweils manuell der Vollbildmodus deaktiviert, die Positionen der Programme neu angeordnet und der Vollbildmodus zuletzt wieder aktiviert werden muss.

### 3.2. Verwandte Arbeiten

Techniken, um die Zeigerposition von einem Bildschirm zu einem anderen zu setzen werden in [BF05], [BF07] und [AOS05] erläutert. In allen wird die Drehung des Kopfes als Eingabemöglichkeit benutzt. Während [BF05] einen am Kopf befestigten Sensor benötigt, wird in [AOS05] ein Stereokamerasystem genutzt. [BF05] stellt zudem Steuerungsalternativen zum Wechseln des Bildschirms vor, wie die Eingabe durch Maus- oder Tastaturtasten. Zudem werden in [BF05] drei verschiedene Arten zum Platzieren der Maus auf dem Monitor genannt: "Fixed-Location Placement Strategy", bei der die Maus immer in der Mitte des Monitors platziert wird, "Frame-Relative Placement Strategy", bei der die Position auf dem neuen Bildschirm relativ die gleiche wie auf dem alten ist und "Frame-Dependent Placement Strategy", welche der hier implementierten Technik entspricht und in 3.3.3 erläutert wird.

In [BF05] muss zum Setzen der Maus durch Kopfdrehung zusätzlich eine Maustaste gedrückt werden, da bei Steuerung durch reines Kopfdrehen das "Midas touch"-Problem (siehe [Jac91]) auftrat, sodass Nutzer die Maus auf einen neuen Bildschirm positionierten ohne dies zu wollen. Dies kann zum Beispiel auftreten, wenn Nutzer nur kurz eine Information auf einem anderen Monitor lesen wollen. Dieses Problem besteht bei dem hier implementierten System ebenfalls, wurde jedoch nicht als störend bewertet. Deshalb wurde auf das zusätzliche Betätigen einer Taste zum Positionieren der Maus verzichtet.

In [AOS05] wird das manuelle Bewegen der Maus eingeschränkt: Die Maus kann manuell nur auf dem aktuellen Bildschirm bewegt werden und nicht auf einen anderen Monitor. Grund dafür sind Konflikte zwischen manueller Mausbewegung auf einen anderen Monitor und Positionieren der Maus durch Kopfdrehung. Das hier implementierte System schränkt die Mausbewegung nicht ein, da Nutzer häufig manuell die Maus auf einen anderen Bildschirm bewegen möchten. Konflikte traten nur auf, wenn die Maus manuell auf einen Bildschirm bewegt wird und der Nutzer gleichzeitig auf einen anderen sieht. In diesem Fall springt die Maus auf den betrachteten Bildschirm, da davon ausgegangen wird, dass Nutzer den Bereich fokussieren, in dem sie interagieren wollen.

[HSM<sup>+</sup>04] analysiert Verhaltensunterschiede zwischen Konfigurationen mit einem einzelnen Monitor und solchen mit mehreren. Ein Resultat dieser Analyse ist, dass Nutzer bei Multimonitorsystemen andere Verhaltensweisen beim Wechseln von Programmen haben als bei Systemen mit nur einem Bildschirm. Daraus wird schlussgefolgert, dass spezielle Taskswitch-Mechanismen für Multimonitorsysteme untersucht werden sollten.

Auch in [CRM<sup>+</sup>06] werden Probleme aufgewiesen, die bei der Benutzung von Systemen mit großen bzw. mehreren Bildschirmen auftreten. Unter anderem wird erläutert, dass Informationen weit voneinander entfernt sein können und die Interaktion dadurch schwieriger und zeitaufwändiger wird. Zudem zeigt [CRM<sup>+</sup>06], dass Probleme auftreten können, wenn im laufenden Betrieb die Monitorkonfiguration geändert wird, zum Beispiel indem ein Bildschirm ausgesteckt wird. Die in dieser Arbeit implementierten Interaktionssysteme wurden so entwickelt, dass sie zur Laufzeit Veränderungen in der Monitorkonfiguration erkennen und sich daran anpassen können.

[TCH<sup>+</sup>09] analysiert das Nutzerverhalten beim Wechseln von Programmen und stellt Grundsätze zur Entwicklung von Taskswitch-Programmen vor, die in [TSGC11] für die Entwicklung eines optimierten Taskswitch-Systems verwendet werden. Dieses ist in Abbildung 3.2 dargestellt. Die wichtigsten Prinzipien dieses Systems sind die räumliche Konsistenz und die Größenveränderung von Fenstern des Taskswitch-Menüs. Räumliche Konsistenz bedeutet, dass Programme immer an der gleichen Stelle im Taskswitch-Menü bleiben. Dadurch können Fenster schneller ausgewählt werden, da der Nutzer weiß, wo sich welche Einträge im Menü befinden. Die Größenveränderung sorgt dafür, dass häufig genutzte Programme einen größeren Eintrag im Taskswitch-Menü bekommen. Das in dieser Arbeit entwickelte System zum Wechseln von Programmen implementiert diese Prinzipien nicht, da der Fokus auf der Minimierung von Mauswegen lag. Jedoch würde die Integration dieser Prinzipien eine sinnvolle Verbesserung darstellen.

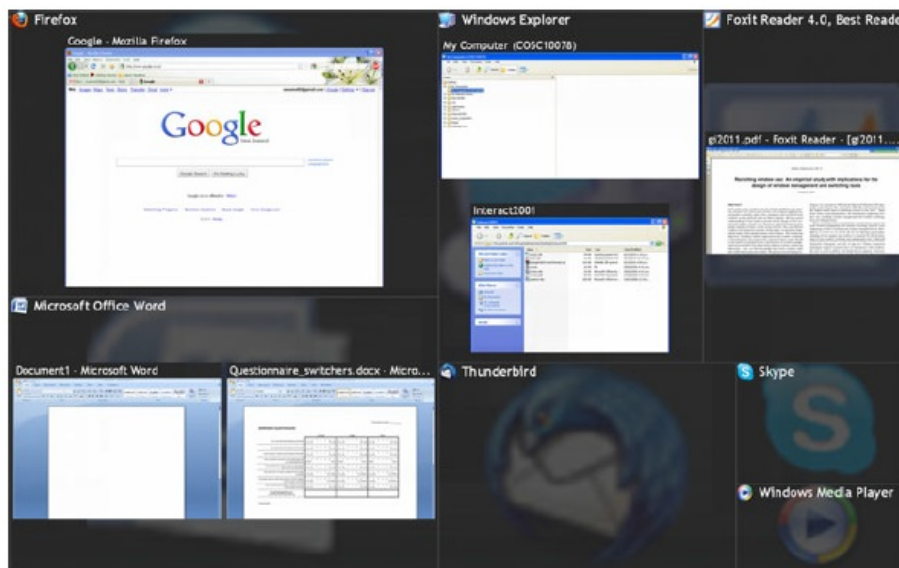


Abbildung 3.2.: Das in [TCH<sup>+</sup>09] implementierte Menü zum Wechseln von Programmen. Zu beachten ist die unterschiedliche Größe von Einträgen aufgrund deren Nutzungshäufigkeit.

In dieser Arbeit wird lediglich die Kopfdrehung des Nutzers verfolgt und nicht dessen Augenbewegungen. Verschiedene Algorithmen zum Erkennen von Augenbewegungen werden unter anderem in [SG00] erläutert. Eine Möglichkeit, wie diese Erkennung zur Steuerung der Maus benutzt werden kann, wird in [BO09] gezeigt.

### 3.3. Lösungsansätze

Um eine hohe Akzeptanz des Bildauswerteplatzes bei Endanwendern zu finden wurde kein neues oder gering verbreitetes Betriebssystem gewählt und keine vollständig neue Steuerung entwickelt. Stattdessen wird die Steuerung durch Maus und Tastatur beibehalten und um neue Interaktionsmöglichkeiten ergänzt. Als Betriebssystem wird das weit verbreitete Windows 7 verwendet, welches um zusätzliche Funktionalität erweitert wird. Dadurch benötigen Endanwender nur geringe Einarbeitungszeiten in das System.

Folgende Funktionen wurden zur Verbesserung der Interaktion entwickelt:

- Multimonitormenü zum Wechseln von Programmen
- Positionierung und Verschieben von Fenstern
- Automatische Mauspositionierung

Zu Beginn dieser Arbeit waren schon rudimentäre Grundlagen des Interaktionssystems vorhanden. Ein System zur automatischen Mauspositionierung sowie ein Ansatz zum Wechseln von Programmen existierten bereits und wurden innerhalb dieser Arbeit um zusätzliche Funktionen erweitert und in den Bereichen Robustheit und Zuverlässigkeit verbessert. Die einzelnen Funktionen werden in den folgenden Abschnitten genauer vorgestellt.

#### 3.3.1. Multimonitormenü zum Wechseln von Programmen

Um das Wechseln von Programmen schneller und übersichtlicher zu gestalten wurde ein eigenes Taskswitch-System implementiert, welches wie "Mission Control" auf jedem Monitor ein eigenes Menü anzeigt. Die Anordnung der Fenster erfolgt jedoch in einem Kreismenü (siehe Abbildung 3.3).

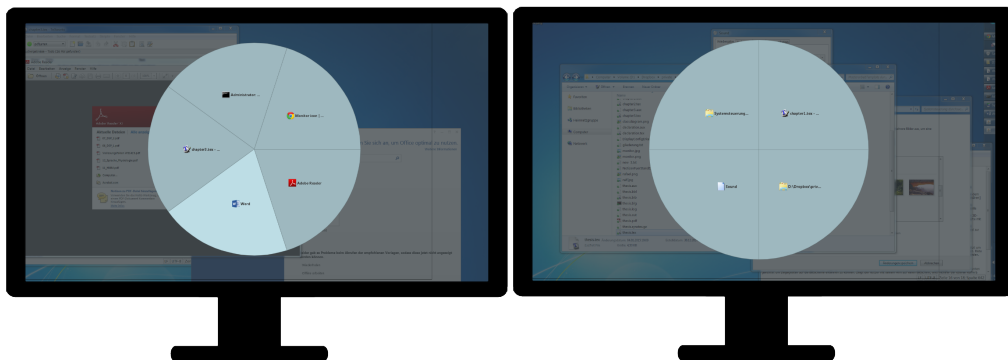


Abbildung 3.3.: Kreismenü des Taskswitch-Systems. Auf jedem Bildschirm werden nur die Fenster zur Auswahl gestellt, die sich auf dem jeweiligen Monitor befinden.

Das Taskswitch-System lässt sich entweder konventionell über eine Tastenkombination starten oder durch eine Zeigegeste auf einen bestimmten Monitor. Das Menü öffnet sich in letzterem Fall nur auf dem Bildschirm, auf den der Nutzer zeigt. Vorteile des Kreismenüs sind, dass vor allem in Kombination mit dem automatischen Setzen der Maus durch Kopfdrehung nur sehr geringe Wege mit der Maus zurückgelegt werden müssen um das gewünschte Fenster zu aktivieren. Zudem erlaubt ein Kreismenü bei Steuerung durch Zeigegesten eine geringe Präzision bei der Erkennung von Position und Richtung der Hand beziehungsweise Finger. Um etwa die komplette Mauspositionierung durch Zeigegesten zu ermöglichen, müsste das Erkennungssystem deutlich genauer sein. Im Menü des Taskswitch-Systems müssen hingegen nur so viele Positionen unterschieden werden wie es Einträge im Menü gibt. Zur Steuerung des Taskswitch-Systems wurden mehrere Möglichkeiten implementiert und getestet.



### 3.3.2. Positionierung und Verschieben von Fenstern

Um das Neuankordnen von Vollbildprogrammen schneller und einfacher zu gestalten, wurde ein System entwickelt, welches das Neupositionieren automatisiert und sich intuitiv steuern lässt. Indem zuerst auf einen Monitor und anschließend auf einen zweiten Monitor gezeigt wird, werden die Fenster auf den entsprechenden Bildschirmen vertauscht. Durch einen Pfeil wird dem Nutzer visualisiert, auf welchen Monitoren die Programme vertauscht werden (siehe Abbildung 3.4). Sobald der Nutzer den Arm zurücknimmt werden die entsprechenden Programme vertauscht.

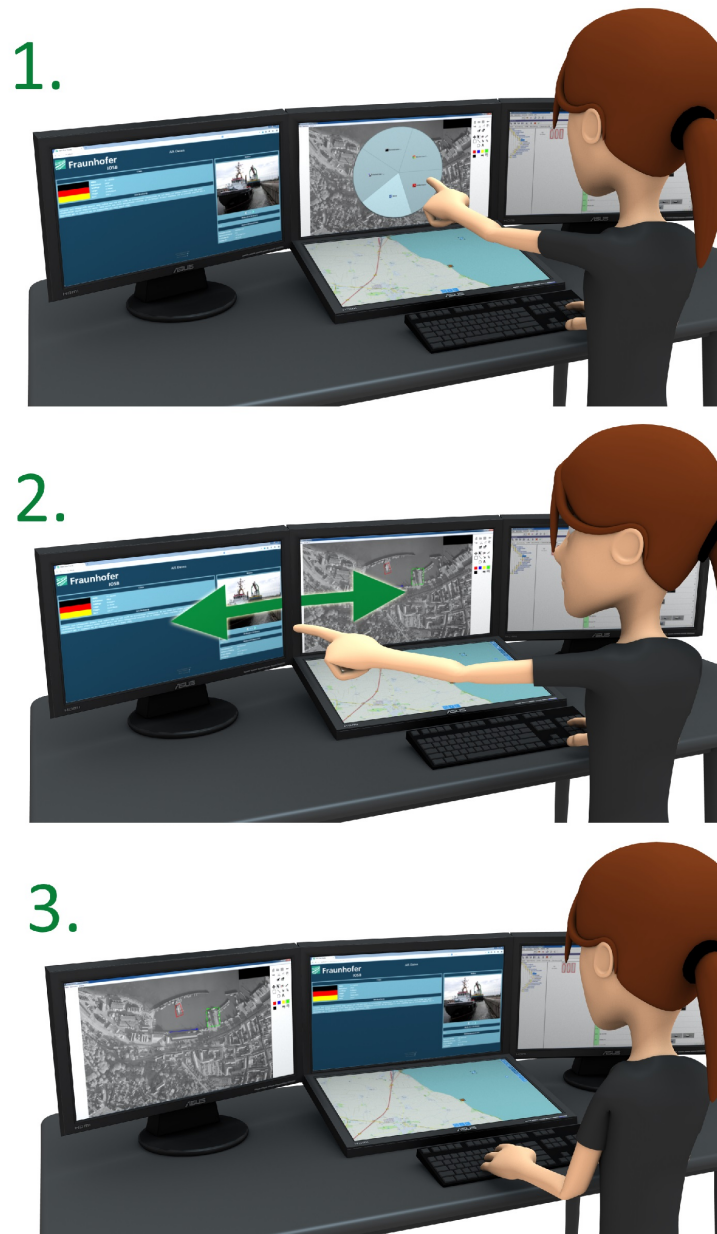


Abbildung 3.4.: 1. Der Nutzer zeigt auf einen Bildschirm und aktiviert dadurch das Kreis-  
menü zum Wechseln von Programmen.  
2. Der Nutzer zeigt auf einen zweiten Bildschirm, ein grüner Pfeil visuali-  
siert die Möglichkeit zum Vertauschen der beiden Vollbildprogramme  
3. Die Fenster wurden vertauscht.

Da die Kinect, welche zur Erkennung von Zeigegesten eingesetzt wird, keine Gesten auf den unteren Bildschirm erkennen kann wurde für diesen ein alternatives Bedienungskonzept implementiert. Durch eine Fünf-Finger-Wischgeste auf dem Touchscreen können Vollbildfenster mit dem unteren Bildschirm vertauscht werden. Die Richtung der Wischgeste (links, oben, rechts) spezifiziert den gewünschten Bildschirm.

Ein sinnvoller Einsatzzweck dieses Systems ist zum Beispiel das Vertauschen von Programmen auf den Touchscreen. So kann etwa das Programm zur Kartendarstellung, das standardmäßig auf dem Touchscreen sichtbar ist um die Karte mit den Fingern verschieben zu können, mit dem Stereobildauswerteprogramm vertauscht werden um Bildannotationen mit der Hand einfügen zu können.

Zusätzlich wurde eine alternative Funktion zum Positionieren von Fenstern entwickelt. Diese benutzt die Kinect, welche die Kopfdrehung verfolgt, um Fenster auf andere Bildschirme zu bringen. Dazu muss die Maustaste innerhalb des gewünschten Fensters gedrückt sein, während der Kopf zu einem anderen Bildschirm gedreht wird. Das System setzt dann Fenster und Maus auf diesen Bildschirm. Als Pixelposition wird jedoch nicht die zuletzt gespeicherte gewählt, sondern die relative Position von Fenster und Maus auf dem "alten" Bildschirm (siehe Abbildung 3.5).

$$newPosition = newDisplay.origin + currentWindowPosition - currentDisplay.origin$$

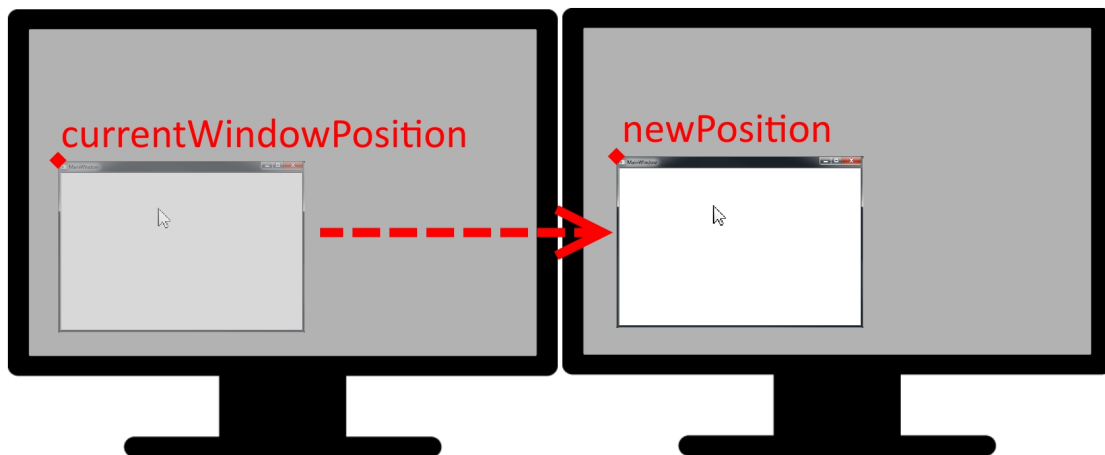


Abbildung 3.5.: Berechnung der neuen Position bei Verschieben eines Fensters durch Kopfdrehung bei gedrückter Maustaste.

### 3.3.3. Automatische Mauspositionierung

Um Mauswege zu verkürzen und die Nutzergeschwindigkeit zu beschleunigen benutzt der Bildauswertepplatz der Zukunft eine Kinect, welche Position und Drehung des Kopfes verfolgt. Diese Information wird benutzt, um automatisch die Mausposition auf denjenigen Bildschirm zu setzen, auf den der Nutzer sieht. Dadurch verringert sich der Arbeitsbereich der Maus auf die Größe eines einzelnen Monitors. Für jeden Monitor wird die letzte Pixelposition der Maus auf diesem gespeichert. Beim Setzen der Maus auf einen neuen Bildschirm wird die Maus auf die zuletzt gespeicherte Position gesetzt. Ein Vorteil dieser Technik ist, dass so zwischen Programmen gewechselt werden kann ohne die Maus zu bewegen oder eine Taste zu drücken. Dadurch kann zwischen tastaturgesteuerten Programmen wie Texteditoren oder Kommandozeilenprogrammen gewechselt werden ohne die Finger von der Tastatur nehmen zu müssen.



Falls noch keine Position gespeichert ist, wird die Maus auf die Mitte des Bildschirms gesetzt, da die größte Strecke die von dort zurückgelegt werden muss die Hälfte der Bildschirmdiagonale ist. Bei Verlassen des Monitors durch Mausbewegung per Hand wird ebenfalls die Mitte des Bildschirms als letzte Pixelposition gespeichert und nicht die Stelle, an der die Maus zuletzt auf dem Monitor war. Der Grund dafür ist, dass die letzte Position auf dem Bildschirm am Bildschirmrand liegt und dies keine Stelle ist an welcher der Nutzer später eine Interaktion durchführen möchte.

Wie in Abbildung 3.6 dargestellt wird eine kurze Animation an der Position der Maus bei jedem Setzen der Maus abgespielt, um die neue Position dem Nutzer zu visualisieren.

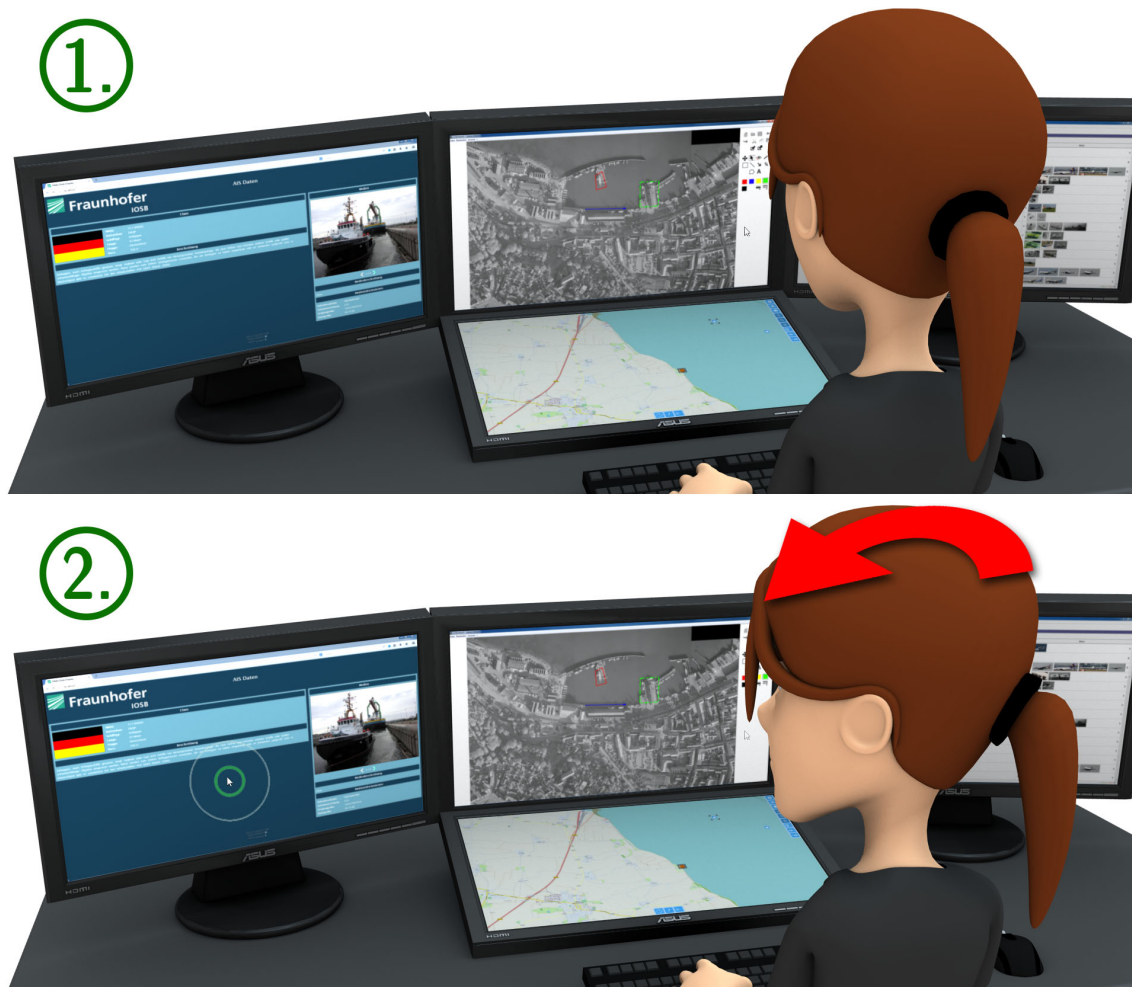


Abbildung 3.6.: 1.) Der Nutzer betrachtet den mittleren Bildschirm, auf dem sich auch die Maus befindet.

2.) Durch eine Kopfdrehung wird die Maus auf den linken Monitor gebracht, ein animierter Kreis signalisiert dem Nutzer die neue Position.

### 3.4. Implementation

Wie im (vereinfachten) Klassendiagramm in Abbildung 3.7 zu erkennen, besteht das Interaktionsprogramm besteht aus drei Subsystemen:

- **DisplayManager**: erkennt die aktuelle Monitor-Konfiguration und speichert Informationen über einzelne Bildschirme.
- **GestureManager**: kümmert sich um gestenbasierte Interaktion zum Verschieben oder Wechseln von Programmen.
- **HeadposeManager**: benutzt die Kopfdrehung zur Steuerung von Maus und Fenstern.

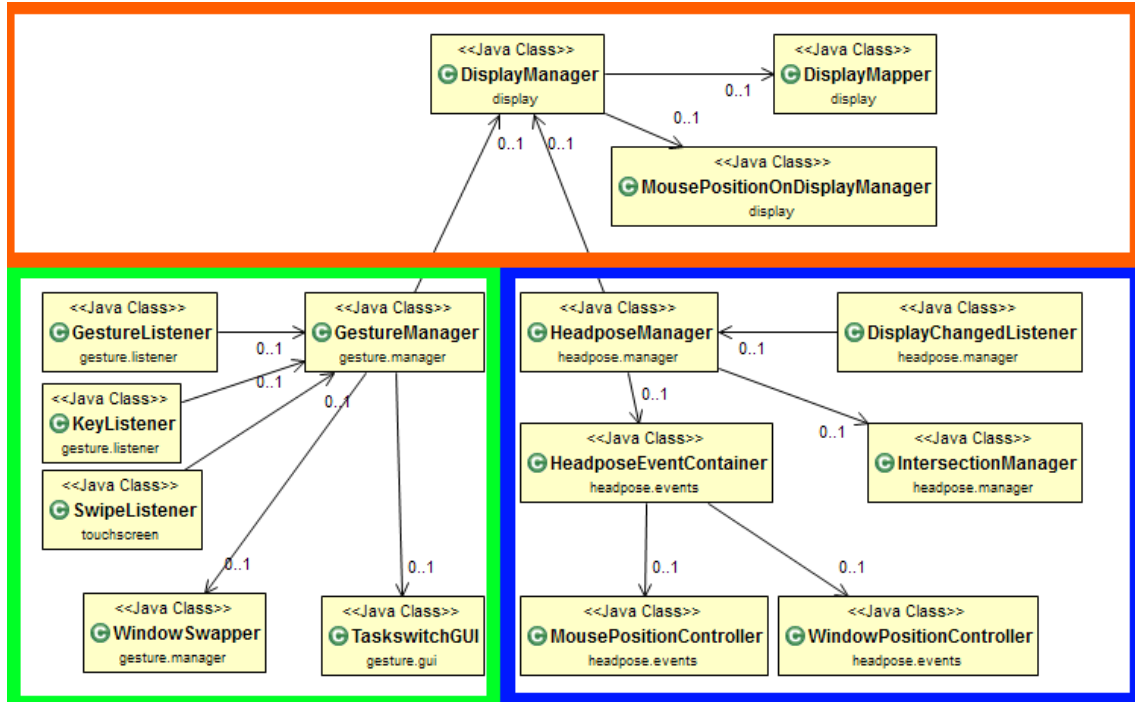


Abbildung 3.7.: Vereinfachtes Klassendiagramm des Interaktions-Systems.

**GestureManager** und **HeadposeManager** weisen eine ähnliche Struktur auf. Beide erhalten Eingabedaten durch verschiedene Listener-Klassen und benutzen diese Informationen für spezielle Aktionen wie das Anzeigen eines Menüs zum Programmwechsel durch die **TaskswitchGUI** oder das Setzen der Positionen von Maus oder Fenstern durch den **MousePositionController** oder den **WindowPositionController**. Daten über die einzelnen Bildschirme, wie deren Auflösung oder relative Position, werden vom **DisplayManager** berechnet und dem **GestureManager** und **HeadposeManager** zur Verfügung gestellt. Die Implementation erfolgte in Java auf Windows 7 unter Verwendung plattformspezifischer Bibliotheken. Die einzelnen Subsysteme werden im Folgenden genauer vorgestellt.

### 3.4.1. DisplayManager

Der `DisplayManager` ist vor allem für das Erkennen der Monitorkonfiguration zuständig. Zudem instanziiert er die Klassen `DisplayMapper` und `MousePositionOnDisplayManager`, die weitere bildschirmspezifische Informationen beinhalten.

#### 3.4.1.1. Erkennen der Monitor-Konfiguration

Hauptaufgabe des Displaymanagers ist die Erkennung der angeschlossenen Monitore inklusive ihrer relativen Positionen, was Voraussetzung für das gesamte Programm ist. Für den Einsatz am Bildauswertepplatz der Zukunft wurde ein Algorithmus implementiert, der versucht, die Konfiguration der Bildschirme zu identifizieren. Das Betriebssystem erkennt die angeschlossenen Monitore, deren Auflösung und Ursprungspositionen. Der Ursprung eines Monitors liegt in der oberen, linken Ecke (siehe Abbildung 3.8). Aufgrund dieser

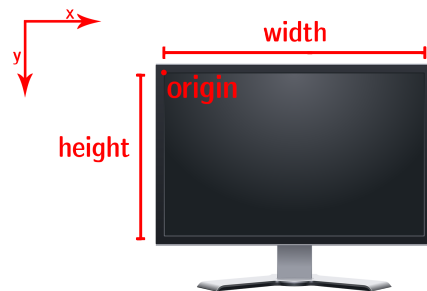


Abbildung 3.8.: Ursprung, Auflösung und Koordinatensystem eines Bildschirms

Information wird versucht zu erkennen, welcher Bildschirm der linke, obere, rechte und untere ist. Windows weist zwar jedem Monitor eine Identifikationsnummer zu, diese kann sich jedoch ändern und eignet sich somit nicht zur Konfigurationserkennung.

```

1 List<Display> displaysDown = findDisplaysDown(displays);
2 if (displaysDown.size() == 1) {
3     // Der untere Monitor ist angeschlossen
4     displayDown = displaysDown.get(0);
5     for (Display display : displays) {
6         if (display.x < displayDown.x) {
7             displayLeft = display;
8         } else if (display.x >= displayDown.x + displayDown.
              width) {
9             displayRight = display;
10        } else {
11            displayUp = display;
12        }
13 } else { // Der untere Monitor ist nicht angeschlossen
14     if (monitors.size() > 2) {
15         displays.remove(displayDown);
16         displayLeft = findDisplayLeft(displays);
17         displays.remove(displayLeft);
18         displayRight = findDisplayRight(displays);
19         displays.remove(displayRight);
20         displayUp = displays.get(0); //der mittlere Bildschirm
              bleibt übrig.
21     }
22 }
```

Die Funktionen `findDisplaysDown()`, `findDisplayLeft()` und `findDisplayRight()` identifizieren die jeweiligen Bildschirme durch Vergleichen der Ursprungskoordinaten der Monitore. Die spezifischen Ursprungskoordinaten des Bildauswerteplatzes sind in Abbildung 3.9 visualisiert.



Abbildung 3.9.: Ursprungskoordinaten des Bildauswerteplatzes.

Monitorkonfigurationen können jedoch in Anzahl und Position der Bildschirme in anderen Systemen abweichen.

#### 3.4.1.2. DisplayMapper

Wie in Kapitel 1.1 beschrieben finden alle Kinect-basierten Gestenerkennungsberechnungen auf einem separaten Computer statt, der die Ergebnisse der Berechnungen an das Interaktionssystem sendet. In diesen Ergebnissen werden positive Integerzahlen zur Identifikation der einzelnen Bildschirme benutzt (siehe Abbildung 3.10). In der Klasse `DisplayMapper` erfolgt die Zuweisung welche Integerzahl welchem Bildschirm entspricht.



Abbildung 3.10.: Identifikationsnummern der Monitore.

#### 3.4.1.3. MousePositionOnDisplayManager

In dieser Klasse wird für jeden Bildschirm die letzte aktuelle Mausposition gespeichert. Diese Daten werden vor allem im `HeadposeManager` benötigt um beim Setzen der Maus auf einen neuen Monitor die richtige Position zu wählen. Mithilfe der Bibliothek `JNativeHook` ([jnh]) wird bei jeder Bewegung der Maus die Mausposition auf dem entsprechenden Bildschirm aktualisiert.

### 3.4.2. GestureManager

Aufgabe des Gesture-Systems ist es, dem Nutzer intuitive, zuverlässige und schnell nutzbare Möglichkeiten zu geben durch Handgesten die Arbeit am Computer zu erleichtern.

#### 3.4.2.1. Listener

Die Klasse `GestureListener` empfängt Daten über ausgeführte Gesten, welche vom separaten System zur Gestenerkennung gesendet werden. Diese Daten enthalten die Information, auf welchen Bildschirm gezeigt wird (z.B. "left" für den linken Monitor) und eine Position auf welche Stelle auf diesem Bildschirm gezeigt wird. Bei einer Zeigegeste werden Informationen über diese Geste ca. alle 40 Millisekunden empfangen. Um bei dieser hohen Informationsdichte eine flüssige Bedienung zu ermöglichen wurde der Code auf Parallelisierung optimiert und die Bearbeitung einer Nachricht erfolgt in einem eigenen Thread. Ein Problem ist die Erkennung, wann ein Nutzer absichtlich eine Geste ausführen möchte und wann nicht. Es kann vorkommen, dass Nutzer eine Bewegung mit ihren Armen durchführen ohne dabei eine Interaktion mit dem System auslösen zu wollen. Dieses Problem wurde softwaretechnisch so gelöst, dass ein zeitbasierter Filter alle erkannten Gesten verwirft die eine zu geringe Dauer besitzen. Gesten werden somit nur nach einer kurzen Wartezeit ausgelöst. Diese Zeit ist jedoch standardmäßig auf 200 Millisekunden eingestellt und deshalb für Nutzer kaum wahrnehmbar. Das Problem konnte jedoch nicht vollständig behoben werden. Schnelle Bewegungen lösen bei Einsatz des Filters keine Gesten aus, falls Nutzer jedoch beispielsweise zu Kommunikationszwecken auf einen Bereich eines Bildschirms zeigen ohne eine Interaktion mit dem System auslösen zu wollen lässt sich diese Intention nicht erkennen.

Weitere Listener-Klassen wurden eingebaut um alternative Steuerungseingaben zu ermöglichen. `KeyListener` empfängt Informationen über gedrückte Tasten auf der Tastatur um eine traditionelle Steuerung für das Menü zum Wechseln von Programmen anzubieten. Standardmäßig lässt sich damit das Task-Switch-Menü über die Feststelltaste öffnen, welche von vielen Nutzern sonst nie verwendet wird. Alternativ kann auch die von Windows benutzte Tastenkombination Alt+Tab zum Wechseln von Programmen überschrieben werden.

`SwipeListener` registriert Wischbewegungen auf dem unteren touchfähigen Bildschirm. Diese werden benutzt um Fenster mit dem unteren Bildschirm tauschen zu können. Das in dieser Arbeit verwendete Betriebssystem Windows 7 bietet leider nur eine rudimentäre Unterstützung von Touchscreens. Das aktuelle System kann Wischgesten nur auf einem speziellen Fenster, welches auf dem Touchscreen platziert ist, erkennen. Wenn dieses Fenster aktiv ist können andere Programme auf demselben Bildschirm nicht verwendet werden. Deshalb muss derzeit der Nutzer eine Taste drücken, um das Fenster zum Erkennen von Touch-Eingaben und -Gesten zu aktivieren und dann eine Geste auf diesem ausführen. Erneutes Drücken dieser Taste minimiert das Erkennungsfenster, damit der Nutzer mit den eigentlichen Programmen zur Bildauswertung interagieren kann. Eine Lösung, bei der kein spezielles Fenster zur Erkennung von Wischgesten benötigt wird, wäre sehr sinnvoll, da es die Bedienung erheblich vereinfachen würde.

#### 3.4.2.2. Vertauschen von Fenstern

`WindowSwapper` ermöglicht das Vertauschen von Fenstern durch Gesten. Beginnt der Nutzer eine Geste, indem er auf einen Bildschirm zeigt, wird zunächst das Kreismenü zum Wechseln von Programmen angezeigt. Falls nun jedoch auf einen anderen Monitor gezeigt wird, wird das Kreismenü geschlossen und ein Pfeil angezeigt, um anzudeuten, dass der Nutzer zwei Vollbildprogramme vertauschen möchte. Beim Beenden der Geste führt dann der `WindowSwapper` diese Vertauschung durch. Die Fenster werden dabei nicht sofort versetzt, sondern in einer Animation zu der neuen Position bewegt.

Dadurch wird dem Nutzer die Neuordnung klar visualisiert und Verwirrung vorgebeugt. Da sich unter Windows 7 wie auch bei vielen anderen Betriebssystemen die Fenster im Vollbildmodus nicht problemlos verschieben lassen, wird zuerst der Vollbildmodus beendet, dann die Animation durchgeführt und schließlich der Vollbildmodus wieder gestartet (siehe Abbildung 3.11). Diese Steuerung der Fenster durch Code erfolgt dabei mithilfe der WinAPI und der Skriptsprache Autohotkey ([ahk]).

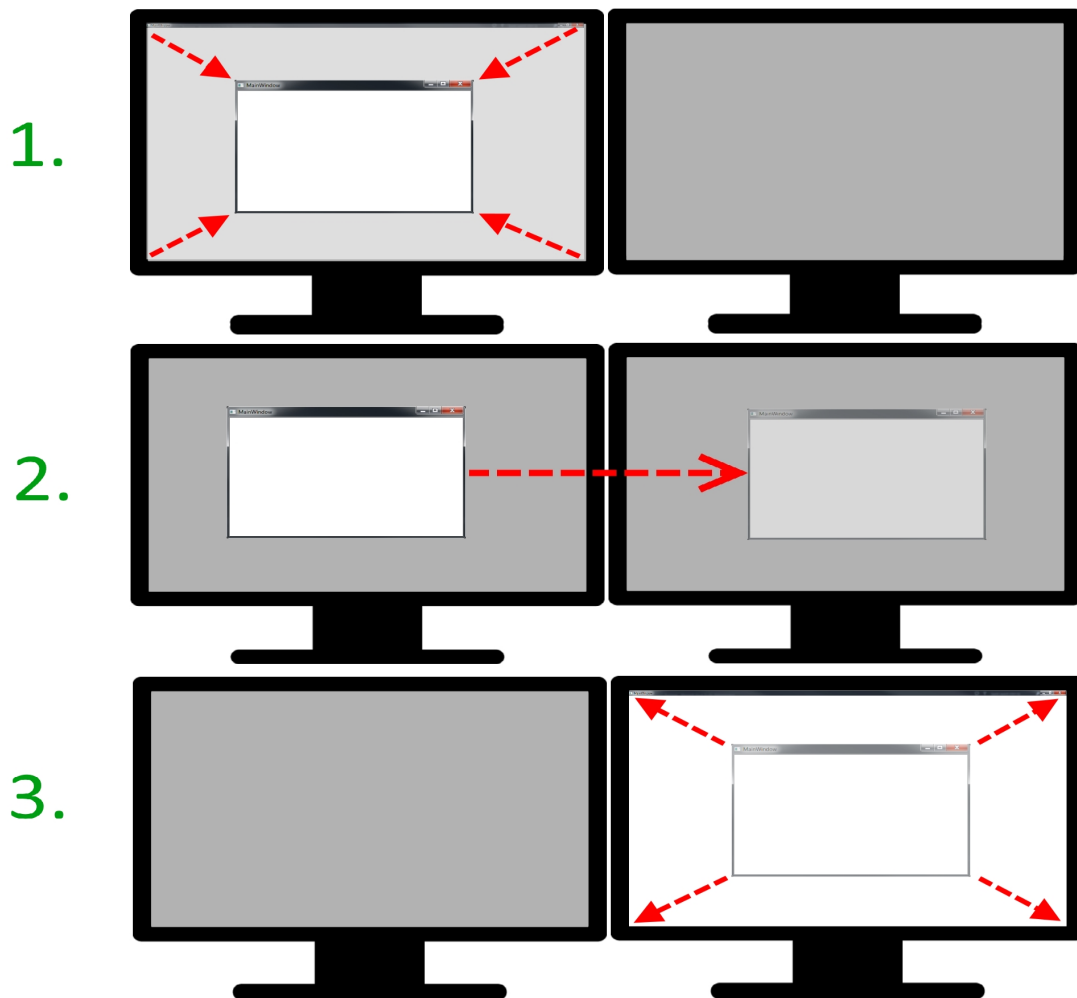


Abbildung 3.11.: Neupositionierung eines Fensters: 1. Beenden des Vollbildmodus', 2. Animiertes Verschieben, 3. Aktivieren des Vollbildmodus'

#### 3.4.2.3. Task-Switch-Menü

Mithilfe der Klasse `TaskswitchGUI` wird das Kreismenü zum Wechseln von Programmen erstellt. Wird auf einen Monitor gezeigt, so erzeugt `TaskswitchGUI` ein Kreismenü auf diesem Bildschirm. Falls das Menü durch Tastendruck erzeugt wird, so erscheint auf jedem Bildschirm ein Kreismenü. In beiden Fällen beinhaltet das Kreismenü all diejenigen Fenster, welche auf dem entsprechenden Monitor sichtbar sind. Um Informationen über die Programme und Fenster des Systems erhalten zu können wird das Windows Application Programming Interface (WinAPI) verwendet. Um die dynamischen Programmbibliotheken (Dynamic Linked Libraries, kurz DLL) der WinAPI unter Java nutzen zu können wurde die Java-Bibliothek Java Native Access ([jna]) eingesetzt.



Zur Erkennung, auf welchem Monitor ein Fenster platziert ist, wird geprüft, ob das vom Monitor durch dessen Ursprungsposition und Auflösung definierte Rechteck die Position des Fensters, welche der Position dessen oberer linken Ecke entspricht, enthält.

Die Auswahl des gewünschten Programms erfolgt entweder durch die Maus oder durch eine Zeigegeste. Für letztere wurden verschiedene Methoden entwickelt und getestet. Zunächst wurde die Präzision der Handpositionserkennung untersucht. Die Positionen, die angeben, auf welchen Bereich des Bildschirms gezeigt wird, werden in Positionen auf einem Kreis umgewandelt, um ein bestimmtes Fenster des Kreismenüs auswählen zu können. Da die von der Kinect erkannten Daten jedoch oftmals zu ungenau sind, um Fenster zuverlässig auswählen zu können, wurde eine alternative zeitbasierte Steuerungsmöglichkeit implementiert. Bei dieser werden die relativen Positionen vernachlässigt und lediglich die Information, auf welches Display gezeigt wird, verwendet. Das Programm wählt automatisch ein Fenster für einen kurzen Zeitraum aus und schaltet danach das nächste aktiv. Bei Beenden der Zeigegeste wird das zuletzt aktive Fenster in den Fokus gebracht.

### 3.4.3. HeadposeManager

Der **HeadposeManager** behandelt sämtliche Daten, welche zur Steuerung des Systems durch Kopfdrehung relevant sind. Eine wesentliche Herausforderung war es, die Robustheit und Zuverlässigkeit des Systems zu maximieren, da Nutzer bei automatischem Setzen der Mausposition nur eine extrem geringe Fehlerrate akzeptieren. Besonders schwierig gestaltete sich die Behandlung in den Fällen, in denen der Nutzer auf Bereiche zwischen zwei Monitoren sieht. Um diese Fälle kümmert sich der **IntersectionManager**.

Die durch den **DisplayChangeListener** empfangenen Daten enthalten für jeden Monitor die Wahrscheinlichkeit, dass dieser vom Nutzer betrachtet wird. Zunächst werden die höchste und zweithöchste Wahrscheinlichkeit gesucht. Unterschreitet die Differenz dieser beiden Wahrscheinlichkeitswerte einen bestimmten Wert wird das Headpose-Ereignis vom **IntersectionManager** behandelt. Andernfalls wird die Mausposition auf den Bildschirm mit der höchsten Wahrscheinlichkeit gesetzt.

```

if highestPropability - secondHighestPropability < THRESHOLD then
    setMouseToDisplayWithHighestPropability();
else
    IntersectionManager.handleHeadposeEvent();
end if

```

#### 3.4.3.1. IntersectionManager

Als Überschneidung werden die Randbereiche definiert, die zwischen zwei Bildschirmen liegen. Abbildung 3.12 zeigt die Überschneidungen des Bildauswerteplatzes.



Abbildung 3.12.: Überschneidungsbereiche des Bildauswerteplatzes.

Bei Betrachtung einer solchen Überschneidung sind die Wahrscheinlichkeiten für die jeweiligen Monitore fast gleich. Würde die Maus immer auf den Monitor mit der höchsten Wahrscheinlichkeit gesetzt werden würde in Überschneidungsbereichen ein "Zittern" auftreten, da die Wahrscheinlichkeitswerte aufgrund von Ungenauigkeiten in Konfiguration und Erkennung der Kinect und minimalen Kopfbewegungen geringen Schwankungen unterliegen. Dies hat zur Folge, dass der Monitor mit der höchsten Wahrscheinlichkeit häufig wechselt.

Im ersten Versuch, dieses Problem zu beheben, wurde angenommen, dass bei Betrachtung eines Überschneidungsbereiches die Maus an einer Position in diesem Bereich gewünscht wird. Deshalb wurde in diesem Versuch in Überschneidungsfällen eine Mausposition am Rand eines betroffenen Bildschirms gewählt. Somit müsste, auch wenn der Nutzer die Maus eigentlich am Rand des anderen Monitors positioniert wünschte, nur ein geringer Weg mit der Maus zurückgelegt werden um die beabsichtigte Aktion auszuführen. In der Praxis erwies sich dieser Ansatz jedoch als nicht praktikabel, da er erstens die Interaktion komplexer und schwieriger zu erlernen machte und, da die Augen unabhängig vom Kopf bewegt werden können, Nutzer teilweise keinen Überschneidungsbereich betrachteten obwohl ihre Kopfdrehung dies implizierte (siehe Abbildung 3.13).

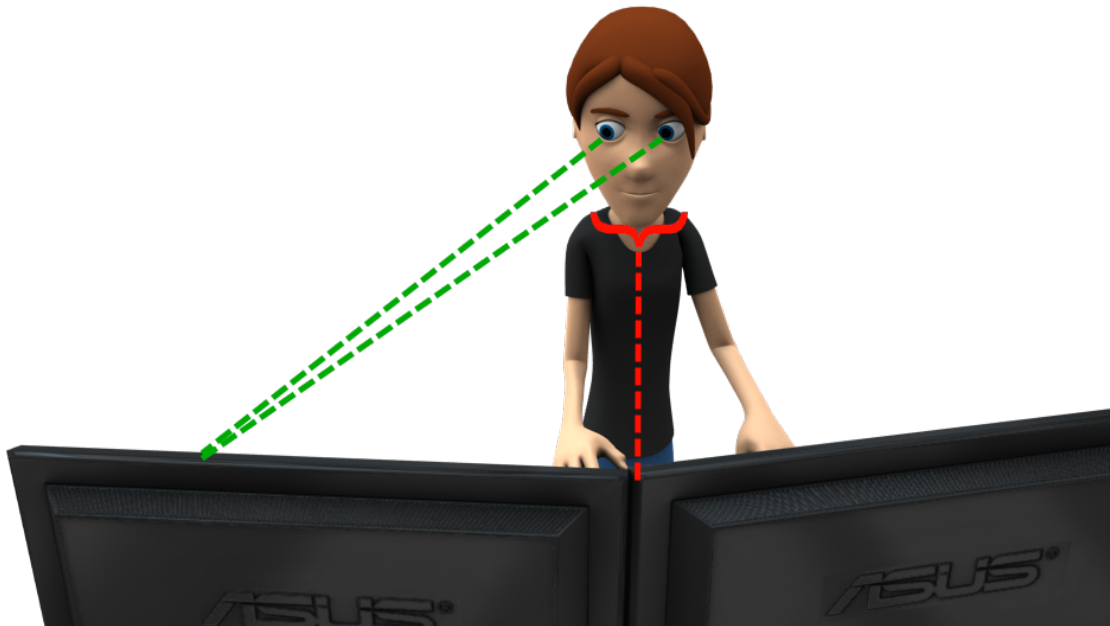


Abbildung 3.13.: Der Nutzer betrachtet eine andere Stelle als die Kopfdrehung impliziert.

Wie von [AOS05] erläutert wird der Kopf im Allgemeinen so bewegt, dass die Augen nahe ihrer Ruheposition und entfernt von extremen Rotationen bleiben. Im Einsatz des Interaktionssystems ist jedoch aufgefallen, dass Nutzer bei Informationen, die am äußeren Rand des linken bzw. rechten Bildschirms positioniert sind, meist ihren Kopf nicht vollständig zu diesem Rand drehen.

Der zweite Versuch macht sich zu Nutze, dass beim Bildauswertepplatz der Zukunft jeder Überschneidungsbereich den mittleren Monitor beinhaltet. Beim Arbeiten auf dem mittleren Bildschirm haben Nutzer eine Kopfdrehung die ungefähr der Ruheposition entspricht. Wird ein Überschneidungsbereich betrachtet möchte der Nutzer deshalb meist nicht den mittleren Monitor betrachten, da die Kopfdrehung bei Betrachtung einer Überschneidung deutlich von der Ruheposition abweicht. Deshalb wird bei diesem Ansatz in einem Überschneidungsfall immer der Bildschirm gewählt, der nicht der mittlere ist.



### 3.4.3.2. Betrachten der Tastatur

Viele Nutzer tippen nicht blind, sondern neigen bei Tastatureingaben ihren Kopf nach unten, um die Tastatur sehen zu können. Beim Testen des Systems zur automatischen Mauspositionierung durch Kopfdrehung stellte sich heraus, dass eine Kopfneigung, um die Tastatur zu sehen, oftmals fast die gleiche ist wie eine Kopfneigung, um den unteren Bildschirm des Bildauswerteplatzes zu betrachten. Somit kann nicht unterschieden werden, ob der Nutzer die Maus auf den unteren Bildschirm platzieren oder lediglich die Tastatur sehen möchte. Aus diesem Grund wurde das automatische Positionieren der Maus auf den unteren Monitor deaktiviert. Die Maus kann somit nur durch manuelles Bewegen oder durch Berühren des Touchscreens auf diesen gebracht werden. Dennoch bereitet eine Kopfdrehung zur Betrachtung der Tastatur dem Interaktionssystem Probleme. Wenn nämlich während dieser Kopfdrehung die Maus automatisch auf einen anderen Bildschirm gebracht wird kann es passieren, dass die Tastatureingaben in einem anderen Programm erfolgen, als beabsichtigt. Bei Tastatureingaben in einem Programm auf dem linken oder rechten Monitor passierte es in Tests öfters, dass bei einer Kopfdrehung in Richtung Tastatur die Maus auf den mittleren Bildschirm positioniert wurde. Um das unbeabsichtigte Neupositionieren der Maus bei einer Kopfdrehung zur Tastatur zu verhindern, wurde ein zeitbasierter Filter implementiert. Dieser bewirkt, dass die Maus erst auf den mittleren Monitor gebracht wird, wenn der Nutzer für 0,5 Sekunden diesen betrachtet. Nachteil dieses Filters ist jedoch, dass die Interaktion dadurch spürbar verlangsamt wird. Eine Erkennung, ob Nutzer bei einer Kopfdrehung einen Bildschirm oder die Tastatur betrachten möchten, würde diesen Filter obsolet machen und die Interaktion verbessern.



## 4. Verbesserungsmöglichkeiten für zukünftige Arbeiten

In diesem Kapitel werden bestehende Probleme der in dieser Arbeit implementierten Interaktionstechniken beschrieben und verschiedene Möglichkeiten skizziert, mit denen diese Probleme in zukünftigen Arbeiten verbessert werden könnten. Es werden sowohl design- als auch implementationstechnische Probleme erläutert.

- **Gestenerkennung:**

Ein großes Problem bei der Verwendung von gestenbasierten Interaktionstechniken ist die Erkennung, ob der Nutzer tatsächlich eine Interaktionsgeste ausführen möchte oder nicht. Wie in 3.4.2.1 beschrieben, reicht die hier implementierte zeitbasierte Gestenfilterung nicht aus, um dieses Problem vollständig zu beheben. Da eine automatische Erkennung der Intention der Nutzer sehr kompliziert ist, könnte dem Nutzer die Möglichkeit gegeben werden, das Gestenerkennungssystem manuell zu deaktivieren. Dies könnte zum Beispiel durch eine Tastenkombination realisiert werden, mit der der Nutzer die Gestenerkennung aus- und anschalten kann.

Hinzu kommt, dass oftmals bei Eingaben auf dem Touchscreen eine Zeigegeste auf den mittleren Monitor erkannt wird, da der Nutzer seinen Arm ausstrecken muss um mit dem Touchscreen interagieren zu können. Diese Fehlerkennung könnte möglicherweise durch Verwendung genauerer Sensoren verhindert werden.

- **Mehrere Nutzer:**

Derzeit ist das Interaktionssystem auf die Benutzung durch eine Person zugeschnitten. Viele Sensoren, wie auch die Kinect-Kamera, sind auch imstande mehrere Nutzer gleichzeitig zu erkennen. Wenn momentan zwei oder mehr Personen von den Kinect-Kameras des Bildauswerteplatzes erkannt werden, ist das Interaktionssystem nicht sinnvoll zu verwenden, da in zufälliger Reihenfolge die Daten beider Nutzer nacheinander verarbeitet werden. Bei gleichzeitiger Betrachtung verschiedener Bildschirme kann es passieren, dass die Mausposition ständig zwischen beiden Bildschirmen hin- und herwechselt. Deshalb wäre es sinnvoll, das Interaktionssystem so zu erweitern, dass die Bedienung auch bei mehreren Personen zuverlässig funktioniert.

- **Management der Vollbild-Modi:**

Verschiedene Programme behandeln die Darstellung der Fenster in einem Vollbild-Modus unterschiedlich. Manche Programme besitzen gar keinen Vollbild-Modus. Erschwerend kommt hinzu, dass unter Windows zwischen einem maximierten Fenster und einem im Vollbild-Modus unterschieden wird. Somit kann sich zwar ein Programm über den gesamten Bildschirm erstrecken, aber trotzdem nicht im Vollbild-Modus sondern lediglich maximiert sein. Des weiteren gibt es Programme, die im Vollbildmodus bewirken, dass auf mehreren Bildschirmen nur eine schwarze Farbe angezeigt wird, und nicht die Programme die sich eigentlich dort befinden. Diese und weitere Probleme bei der Verwaltung von Fenstern im Vollbild-Modus haben dazu geführt, dass Interaktionstechniken, wie das Vertauschen von Fenstern durch eine Zeigegeste, bei manchen Programmen nicht zufriedenstellend funktionieren. Um diese Problematik zu verbessern müsste das Interaktionssystem in der Verwaltung von Vollbild-Modi verbessert werden, um die Vielzahl an unterschiedlichen Arten des Vollbild-Modus zu erkennen. Jedoch müssen auch die verschiedenen Programme des Bildauswerteplatzes der Zukunft angepasst werden, da diese die Vollbild-Darstellung verweigern und somit das Interaktionssystem stören können.

- **Verbesserung des Taskswitch-Menüs:**

Das Kreismenü zum Wechseln von Programmen könnte um zusätzliche Funktionalität erweitert werden, um die Bedienung zu beschleunigen und die Übersicht zu erhöhen. Beispielsweise ließen sich die in [TSGC11] vorgestellten und in Kapitel 3.2 erklärten Prinzipien der räumlichen Konsistenz und der Größenveränderungen auf Basis der Nutzungshäufigkeiten in das bestehende Kreismenü integriert werden. Des weiteren könnten in das Menü integrierte Miniaturbilder der einzelnen Programme dabei helfen, das gewünschte Fenster schneller zu finden. Abbildung 4.1 zeigt eine Skizze, wie diese Funktionen in das bestehende Menü eingebaut werden könnten.

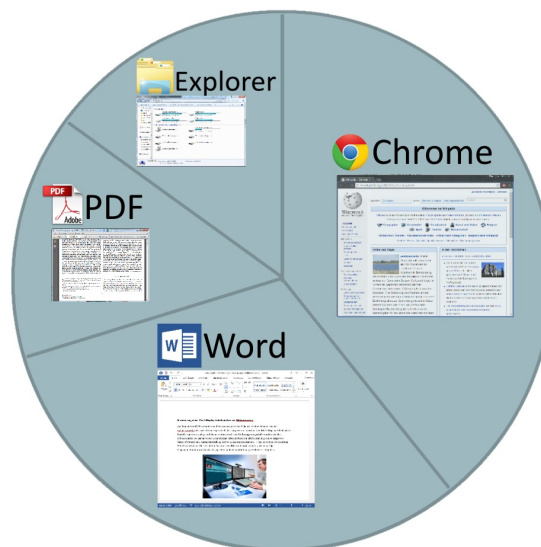


Abbildung 4.1.: Mögliche Verbesserungen des Kreismenüs. Zu beachten sind die Miniaturbilder der einzelnen Fenster und die unterschiedlichen Größen der Menüeinträge auf Basis der Nutzungshäufigkeiten.

- **Entwicklung zusätzlicher Interaktionstechniken:**

Weitere Interaktionstechniken zur Verbesserung der Bedienung könnten implementiert werden. Beispielsweise ließen sich weitere Gesten auf dem Touchscreen zur Steuerung des Systems einsetzen. Hierbei könnten auf Touchscreens optimierte Betriebssysteme wie Windows 8, iOS und Android als Inspirationsquellen dienen. Außerdem könnten zusätzliche Gesten entwickelt werden, nicht nur zur Steuerung des Betriebssystems sondern auch, wie in 2.3.2 vorgestellt, zur Steuerung und Kommunikation zwischen speziellen Programmen. Weitere Gesten könnten zum Beispiel verschiedene Fingergesten beinhalten.

- **Einsatz zusätzlicher Sensoren:**

Der Bildauswertepplatz der Zukunft könnte um zusätzliche Sensoren erweitert werden um weitere Eingabeoptionen zu ermöglichen. Mögliche Sensoren sind beispielsweise ein Mikrofon zur Steuerung des Systems durch Sprachbefehle oder spezielle Sensoren zur Verfolgung der Augen.

- **Erweiterung auf allgemeine Arbeitsplätze:**

Aktuell ist das Interaktionssystem auf die Bedienung des Bildauswertepplatzes der Zukunft zugeschnitten. Die in dieser Arbeit entwickelten Interaktionstechniken ließen sich jedoch auch auf allgemeine Multimonitorsysteme anwenden und würden in Systemen mit mehr als vier Monitoren noch größere Geschwindigkeitsvorteile in der Bedienung bewirken. Zur Benutzung auf beliebigen Multimonitorsystemen wäre vor allem eine allgemeine Repräsentation der Bildschirmkonfiguration im System notwendig, welche eine beliebige Anzahl an Bildschirmen mit beliebigen Positionen und Auflösungen erlaubt. Dafür wäre auch eine Erweiterung des Systems zur Erkennung von Gesten und Kopfdrehungen durch die Kinect-Kameras nötig.

Weil außerdem plattformspezifische Bibliotheken zum Management der Fenster verwendet wurden, müsste die Software für andere Betriebssysteme gegebenenfalls portiert werden.

- **Verbesserung der Sensoren:**

Wie in Kapitel 3.4.2.1 beschrieben, ist die Positions- und Richtungserkennung der Finger durch eine Kinect-Kamera zu ungenau, um zuverlässig verwendet werden zu können. Außerdem lassen sich Kopfdrehungen auf den unteren Bildschirm und auf die Tastatur nicht unterscheiden. Deshalb könnte man das System durch Verwendung genauerer Sensoren verbessern. Beispielhaft wurde dies mit einem "Leap Motion"-Sensor ([lm]) getestet (siehe Abbildung 4.2). Dieser Sensor wird auf dem Tisch platziert und nicht über den Bildschirmen wie die Kinect. Im Test stellte sich heraus, dass mithilfe des "Leap Motion"-Sensors die Gestenerkennung deutlich genauer ist und sich sogar theoretisch zur kompletten Steuerung der Maus nutzen lässt. Da der Sensor nicht auf große Multi-Monitorsysteme ausgelegt ist und der Bereich, in dem Finger erkannt werden relativ klein ist, müssten, wie Abbildung 4.3 zeigt, mindestens zwei "Leap Motion"-Geräte verwendet werden.



Abbildung 4.2.: "Leap Motion"-Sensor.

Quelle: <http://www.mjdinteractive.com/wp-content/uploads/2013/09/Leap-Motion-Technology.png>,  
Stand: 12.01.2015



Abbildung 4.3.: Positionierung der Sensoren am Bildauswertepplatz der Zukunft. Die grünen Kreise visualisieren den Bereich, in dem Finger und Hände erkannt werden.

# A. Fragebogen zur Erweiterung des Bildauswerteplatzes

Fragebogen zum Stereobildauswerter

## Zukünftige Features

Um den Bildauswerteplatz zu verbessern sind eine Reihe an neuen Features geplant. Bitte schätzen sie die Nützlichkeit dieser Features ab.

## Verbesserte Bedienung

---

### 1. Bedienung der Bildauswertung auf Touchscreens

Stereobildauswertung für Touchscreens optimieren für intuitives und schnelles Zeichnen. Durch einen einfachen Knopfdruck lassen sich die Positionen von GeoViewer und Stereobildauswerter vertauschen.

*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant

## Fragebogen zum Stereobildauswerter

## Unterstützung mehrerer Bilder

### 5. Bearbeitung mehrerer Bilder

Durch eine Zeitleiste oder einen Knopf kann zwischen mehreren Bildern schnell umgeschaltet werden. Zur Anwendung bei Aufnahmen des selben Orts zu verschiedenen Zeitpunkten um Veränderungen schnell zu erkennen.  
*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant

### 6. Algorithmen zur Erkennung von Veränderungen mehrerer Bilder

Algorithmen, die automatisch Unterschiede zwischen zwei oder mehr Bildern anzeigen. Dadurch lassen sich Veränderungen bei Aufnahmen des selben Orts zu verschiedenen Zeitpunkten schneller finden.  
*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant

### 2. Einfügen des annotierten Bilds in den Bericht

Das im Bildauswerter aktuell geöffnete Bild direkt mitsamt Einzeichnungen in den Bericht (z.B. Word-Dokument oder RecceExRep) einfügen. Dies kann durch eine einfache Menü-Option oder eine schnelle Handgeste erfolgen.  
*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant

### 3. Verbesserte Footprints des GeoViewers

Geoviewer zeigt nicht nur das aktuell geöffnete Bild, sondern mehrere Bilder aus der CSD an. Ein Klick auf ein Bild der Karte öffnet dieses im Stereobildauswerter.  
*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant

### 4. Drag & Drop

Über Drag & Drop Bilder direkt importieren oder Projekte sofort öffnen.  
*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant



## Features zur Bildbearbeitung

### 7. Bildkorrekturen

Einstellen von Kontrast, Helligkeit, Sättigung und Bearbeitung einzelner Farbkanäle (Rot, Grün, Blau) um Details des Bildes besser erkennen zu können. Zusätzlich eine Histogramm-Anzeige mit Funktionen wie Histogrammspreizung und Histogrammbegrenzung.

*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant

### 8. Bildkorrekturen auf einzelnen Bereichen

Bildkorrekturen nur auf einen bestimmten Bereich des Bildes anwenden, um beispielsweise einen Teil des Bildes aufzuhellen.

*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant

### 9. Bildfilter

Anwendung verschiedener Filter (z.B. Kantenerkennung, Rauschentfernung, Invertieren) auf das Bild um weitere Details des Bildes erkennen zu können.

*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant

### 10. Bild Zuschneiden

Zuschneiden des Bildes auf einen kleineren Ausschnitt. Geodaten bleiben dabei erhalten.

*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant

## Fragebogen zum Stereobildauswerter

## Neue Tools

### 11. Zusätzliche Webinformationen

Über die Position des Bildes zusätzliche Informationen über Web-Services abfragen, wie z.B. Daten über Wetter, Verkehrslage, etc.  
*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant

### 12. 3D-Daten zur Auswertung

Das auszuwertende Bild wird auf ein dreidimensionales Terrain abgebildet, welches sich aus allen Richtungen betrachten lässt. Dadurch sind Höheninformationen des Geländes besser sichtbar.  
*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant

### 13. 3D-Rekonstruktion

Aus dem Stereobild durch Markierung von Merkmalen 3D-Informationen rekonstruieren, die sich frei betrachten lassen.  
*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant

### 14. Messen von Flächen

Tool, um nicht nur die Länge einer Linie, sondern auch die Größe einer Fläche (z.B. in Quadratmeter) messen zu können  
*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant

## Fragebogen zum Stereobildauswerter

15. **Einzeichnen von Längen oder Flächen**

Längen oder Flächen nicht nur messen können, sondern direkt in das Bild einzeichnen (z.B. um die Länge eines Schiffs zu markieren).

*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant

16. **Drucken**

Möglichkeit, annotierte Bilder direkt auszudrucken.

*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant

## Zusätzliche Formate

17. **Unterstützung von HDR**

Darstellung und Bearbeitung von High Dynamic Range Bildern (= Hochkontrastbilder).

*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant

18. **Unterstützung von Hyperspektralbildern**

Darstellung und Bearbeitung von Hyperspektralbildern (= Bilder, die Spektren außerhalb des sichtbaren Bereichs enthalten).

*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant

19. **Zusätzliche Koordinatensysteme**

Unterstützung und Anzeige von mehreren Koordinatensystemen (z.B. UTM).

*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant

20. **Zusätzliche Exportformate**

Bilder nicht nur als JPG oder PNG exportieren, sondern weitere Dateiformate unterstützen (z.B. NSIF und NTIF).

*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr hilfreich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	irrelevant

Fragebogen zum Stereobildauswerter

## Allgemeine Fragen

### 21. Welche Programme verwenden sie zur Bildauswertung?

Name + Einsatzzweck

---

---

---

---

---

---

---

---

---

---

### 22. Existiert ein genauer Ablauf, wann welches Tool eingesetzt wird?

Wenn ja, wie sieht dieser Ablauf aus?

---

---

---

---

---

---

---

---

---

---

23. **Was sind ihrer Meinung nach die größten Vorteile des Bildauswerteplatz' der Zukunft?**

---

---

---

---

---

---

---

---

---

---

24. **Welche wichtigen Programme / Funktionen vermissen sie am Bildauswerteplatz der Zukunft?**

Name und kurze Begründung

---

---

---

---

---

---

---

---

---

---

## Fragebogen zum Stereobildauswerter

**25. In welchen Formaten erhalten sie die auszuwertenden Bilder?**

zum Beispiel: GeoTiff, PNG, NSIF, NTIF, etc..

---

---

---

---

---

---

---

---

---

---

**26. Wie werden die ausgewerteten Bilder weiterverarbeitet?**

Werden die ausgewerteten Bilder nur für Berichte verwendet oder zusätzlich in anderen Bereichen?

---

---

---

---

---

---

---

---

---

---

# Literaturverzeichnis

- [ahk] *Autohotkey*. <http://www.autohotkey.com/>. Stand: 2015-01-11.
- [AOS05] Mark Ashdown, Kenji Oka, and Yoichi Sato: *Combining head tracking and mouse input for a gui on multiple monitors*. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, pages 1188–1191, New York, NY, USA, 2005. ACM, ISBN 1-59593-002-7. <http://doi.acm.org/10.1145/1056808.1056873>.
- [at] *Able tiff*. <http://www.graphicregion.com/abletiffannot.htm>. Stand: 2015-01-13.
- [BF05] Hrvoje Benko and Steven Feiner: *Multi-monitor mouse*. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, pages 1208–1211, New York, NY, USA, 2005. ACM, ISBN 1-59593-002-7. <http://doi.acm.org/10.1145/1056808.1056878>.
- [BF07] Hrvoje Benko and Steven Feiner: *Pointer warping in heterogeneous multi-monitor environments*. In *Proceedings of Graphics Interface 2007*, GI '07, pages 111–117, New York, NY, USA, 2007. ACM, ISBN 978-1-56881-337-0. <http://doi.acm.org/10.1145/1268517.1268537>.
- [BO09] Renaud Blanch and Michaël Ortega: *Rake cursor: Improving pointing performance with concurrent input channels*. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1415–1418, New York, NY, USA, 2009. ACM, ISBN 978-1-60558-246-7. <http://doi.acm.org/10.1145/1518701.1518914>.
- [CRM<sup>+</sup>06] Mary Czerwinski, George Robertson, Brian Meyers, Greg Smith, Daniel Robbins, and Desney Tan: *Large display research overview*. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 69–74, New York, NY, USA, 2006. ACM, ISBN 1-59593-298-4. <http://doi.acm.org/10.1145/1125451.1125471>.
- [CSD] *Coalition shared data server*. [http://www.iosb.fraunhofer.de/servlet/is/4637/Produktflyer\\_CSD-Server\\_english.pdf?command=downloadContent&filename=Produktflyer\\_CSD-Server\\_english.pdf](http://www.iosb.fraunhofer.de/servlet/is/4637/Produktflyer_CSD-Server_english.pdf?command=downloadContent&filename=Produktflyer_CSD-Server_english.pdf). Stand: 2015-01-21.
- [HSM<sup>+</sup>04] Dugald Ralph Hutchings, Greg Smith, Brian Meyers, Mary Czerwinski, and George Robertson: *Display space usage and window management operation comparisons between single monitor and multiple monitor users*. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '04, pages 32–39, New York, NY, USA, 2004. ACM, ISBN 1-58113-867-9. <http://doi.acm.org/10.1145/989863.989867>.
- [Jac91] Robert J. K. Jacob: *The use of eye movements in human-computer interaction techniques: What you look at is what you get*. ACM Trans. Inf. Syst.,

- 9(2):152–169, April 1991, ISSN 1046-8188. <http://doi.acm.org/10.1145/123078.128728>.
- [jna] *Java native access*. <https://github.com/twall/jna>. Stand: 2015-01-11.
- [jnh] *Jnativehook*. <https://github.com/kwhat/jnativehook>. Stand: 2015-01-12.
- [lm] *Leap motion*. <https://www.leapmotion.com/>. Stand: 2015-01-12.
- [pp] *Powerpoint*. <http://products.office.com/de-DE/powerpoint>. Stand: 2015-01-13.
- [ps] *Photoshop*. <http://www.adobe.com/de/products/photoshop.html>. Stand: 2015-01-13.
- [RAAKR05] R.J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam: *Image change detection algorithms: a systematic survey*. Image Processing, IEEE Transactions on, 14(3):294–307, March 2005, ISSN 1057-7149.
- [rl] *Reccelite*. <http://www.rafael.co.il/Marketing/334-915-en/Marketing.aspx1>. Stand: 2015-01-20.
- [SCD<sup>+</sup>06] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski: *A comparison and evaluation of multi-view stereo reconstruction algorithms*. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 519–528, June 2006.
- [SG00] Dario D. Salvucci and Joseph H. Goldberg: *Identifying fixations and saccades in eye-tracking protocols*. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, ETRA '00, pages 71–78, New York, NY, USA, 2000. ACM, ISBN 1-58113-280-8. <http://doi.acm.org/10.1145/355017.355028>.
- [TCH<sup>+</sup>09] Susanne Tak, Andy Cockburn, Keith Humm, David Ahlström, Carl Gutwin, and Joey Scarr: *Improving window switching interfaces*. In *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part II*, INTERACT '09, pages 187–200, Berlin, Heidelberg, 2009. Springer-Verlag, ISBN 978-3-642-03657-6. [http://dx.doi.org/10.1007/978-3-642-03658-3\\_25](http://dx.doi.org/10.1007/978-3-642-03658-3_25).
- [TSGC11] Susanne Tak, Joey Scarr, Carl Gutwin, and Andy Cockburn: *Supporting window switching with spatially consistent thumbnail zones: Design and evaluation*. In *Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part I*, INTERACT'11, pages 331–347, Berlin, Heidelberg, 2011. Springer-Verlag, ISBN 978-3-642-23773-7. <http://dl.acm.org/citation.cfm?id=2042053.2042088>.
- [wrđ] *Word*. <http://products.office.com/de-DE/word>. Stand: 2015-01-13.
- [xcl] *Excel*. <http://products.office.com/de-DE/excel>. Stand: 2015-01-13.
- [xre] *Reccexrep*. siehe <http://www.globalsecurity.org/intell/library/policy/army/fm/34-43/ch2.htm>, Stand: 2015-01-20.



---

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

**Karlsruhe, 28.01.2015**

.....  
(**Tim Reiter**)