

Stitch-to-Tile: Eine Gestensteuerung zur Auslösung von Bildschirmenerweiterung

Dominick Madden

Bachelor-Thesis

Studiengang Informatik

Fakultät für Informatik

Hochschule Mannheim

22.09.2015

Durchgeführt an der Hochschule Mannheim, Fakultät für Informatik

Betreuer: Prof. Kirstin Kohler, Hochschule Mannheim

Zweitkorrektor: Horst Schneider, B. Sc, Hochschule Mannheim

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Mannheim, 22.09.2015

Dominick Madden

Inhalt

1	Einleitung	1
1.1	Ziele der Arbeit	2
1.2	Aufbau der Arbeit	2
2	State of the Art	3
2.1	Das „Spontaneous Device Sharing Problem“	3
2.2	„Synchronous Gestures“	4
2.3	Display Tiling	7
3	Analyse der Stitch-to-Tile Interaktion.....	9
3.1	Bildschirmerweiterung auf mobilen Endgeräten	9
3.1.1	Visuelles Tiling.....	9
3.1.2	Logisches Tiling	15
3.2	Analyse der Stitching Geste	18
3.3	Design der Stitch-to-Tile Geste	19
3.4	Gegenüberstellung von Stitch-to-Tile und Hinkleys Pen-Stitching.....	26
4	Prototypische Umsetzung der Interaktion	28
4.1	Wahl der Kommunikationstechnologie	29
4.2	Implementierung der Interaktion	31
4.2.1	Verbindung	32
4.2.2	Erkennung eines Swipes	35
4.2.3	Erkennung eines Stitches	36
4.2.4	Tiling.....	41
5	Ausblick.....	44
	Tabellenverzeichnis	iv
	Abbildungsverzeichnis	v
	Literaturverzeichnis	vi

1 Einleitung

Die Benutzung von mehreren Monitoren ist bei Nutzern von Desktop PCs und Laptops weit verbreitet. Mehrere Studien belegen, dass ein Multi-Monitor-Setup Multitasking erleichtert und somit die Produktivität von Nutzern steigert. [GRU01, HUT04, TRU08]

Dies ist laut **Grudin (2001)** nicht nur auf die größere Arbeitsfläche zurückzuführen. Er schreibt, die Verwendung mehrerer Monitore sei vorteilhaft, selbst wenn man dadurch nicht an Bildschirmfläche gewinnt.

In diesem Zusammenhang liefert Grudin eine Analogie, die sich auf die Wohnfläche in einem Haus bezieht.

„Ein Haus mit einem großen Schlafzimmer ist nicht das gleiche wie ein Haus mit zwei mittelgroßen Schlafzimmern. In dem Haus mit zwei Schlafzimmern wird der zweite Raum für einen anderen Zweck verwendet – Vielleicht als Gästezimmer und Büro. Man *könnte* das große Schlafzimmer für diese Zwecke nutzen, tun wir normalerweise jedoch nicht, selbst wenn es groß ist. Die Wand macht einen Unterschied.“ [GRU01]

Die Wände stehen hier für die Ränder eines Monitors; also die Lücke, die zwischen den Bildschirmoberflächen entsteht, wenn man mehrere Monitore nebeneinander stellt. Die Verwendung mehrerer Bildschirme erlaubt es dem Benutzer zu entscheiden, was er in seinem Sichtfeld haben will, statt nur die Hauptanwendung vor sich zu haben.

Multi-Screen Nutzung hat Vorteile und ist für die Benutzung von PCs etabliert. Für mobile Endgeräte hingegen ist dies noch nicht der Fall. Laut einer Studie von **Google (2012)** werden von Amerikanern zwar täglich eine Vielzahl von Geräten genutzt, jedoch meist in einer zeitlichen Abfolge. Eine Tätigkeit, wie eine Internetrecherche, beginnt beispielsweise auf dem Smartphone und wird später auf dem größeren Bildschirm des Computers fortgesetzt.

Diese Arbeit befasst sich mit der Bildschirmerweiterung bei mobilen Endgeräten, wie Tablets und Smartphones, und stellt eine intuitive Lösung vor diese auszulösen.

1.1 Ziele der Arbeit

Das Hauptziel dieser Arbeit ist die Konzeption und Ausarbeitung einer Gestensteuerung zur Auslösung von Bildschirmerweiterung auf mobilen Endgeräten. Die entwickelte Geste erhält den Namen „Stitch-to-Tile“. Es handelt sich dabei um eine „Synchronous Gesture“¹. Diese machen vor allem im mobilen Bereich Sinn, da die Geräte über viele Sensoren verfügen und dadurch eine Vielzahl von solchen Gesten ermöglichen. Die von **Hinkley et al. (2004)** vorgestellte Stitching Geste ist eine vielversprechende Synchronous Gesture, die jedoch auch einige auf Hardware basierende Schwächen aufweist. Die Benutzung von modernen Smartphones und Tablets erlaubt es, die Geste etwas anzupassen und zu einem validen Framework auszubauen, sodass diese von anderen Projekten verwendet werden kann.

Ein weiteres Ziel der Arbeit ist die Erörterung von Bildschirmerweiterung auf mobilen Endgeräten. Die Fälle, in denen heute mehrere Bildschirme gleichzeitig genutzt werden, stehen meist in Zusammenhang mit Arbeiten, die viel Bildschirmfläche erfordern. Solche Aufgaben werden nicht im selben Ausmaß auf Smartphones oder Tablets durchgeführt, wie es bei Desktop PCs der Fall ist. Es müssen daher sinnvolle Anwendungsfälle für Bildschirmerweiterung bei mobilen Endgeräten erarbeitet werden.

1.2 Aufbau der Arbeit

Kapitel 2 beschreibt zugrundeliegende Arbeiten aus den Bereichen der Gestensteuerung und Bildschirmerweiterung auf mobilen Endgeräten.

In Kapitel 3 geht auf die Analyse und Konzeption der hier entwickelten Interaktion ein. Dabei wird die Bildschirmerweiterung separat von der Geste betrachtet.

Kapitel 4 zeigt die Prototypische Implementierung der Interaktion, wobei einzelne wichtige Designentscheidungen und ihre technische Umsetzung beschrieben werden.

Zuletzt gibt Kapitel 5 einen Ausblick auf mögliche zukünftige Verbesserungen am Konzept und der Implementierung der hier entwickelten Geste.

¹ Definiert in Kapitel 2.2

2 State of the Art

Das folgende Kapitel behandelt den aktuellen Stand der Technik in den für diese Arbeit relevanten Feldern. Dabei handelt es sich um die gestenbasierte Interaktion zwischen Geräten und das Konzept der Bildschirmenerweiterung. Dazu wird auf verwandte Arbeiten aus diesen Bereichen verwiesen.

2.1 Das „Spontaneous Device Sharing Problem“

“Das ‘Spontaneous Device Sharing Problem’ stellt die folgende Frage: Wie kann ein Benutzer eine zweckvolle Verbindung herstellen zwischen zwei oder mehr Zielgeräten ohne dass sie vorher gegenseitig ihre Netzwerkadressen kennen?“ [HIN04]

Ob man sich nun einen Link von seinem Smartphone auf sein Tablet übertragen oder einer anderen Person im selben Raum ein Bild schicken möchte, geschieht dies meist über externe Mittel wie Email oder Chat-Applikationen. Die Verbindung von mobilen Endgeräten kann viele solcher Aufgaben erleichtern.

Das Problem ist, dass die Verbindung und Bildung von Gruppen zwischen Geräten einen komplizierten Prozess erfordert. Man hat eine Auswahl von verschiedenen Technologien zur Verfügung, die jeweils ihre Vor- und Nachteile haben. Unter anderem sind NFC (Near-Field-Communication), Bluetooth und Wi-Fi Direct Möglichkeiten, moderne Mobilgeräte miteinander zu verbinden. Zusätzlich erfordert jede dieser Technologien vor der Verbindung einen Namen oder eine Art Adresse der Geräte, die bekannt sein müssen.

Wenn dieser Verbindungsprozess zu kompliziert oder mühsam ist, verlieren Benutzer unter Umständen komplett das Interesse an Applikationen, die mehrere Geräte verbinden. [JOK15]

Hinkley (2003) sieht die im folgenden Kapitel beschriebenen „Synchronous Gestures“ als einen vielversprechenden Kandidaten, das „Spontaneous Device Sharing Problem“ zu lösen.

2.2 „Synchronous Gestures“

„Synchronous Gestures“ werden als Aktivitätsmuster definiert, die entweder von mehreren Nutzern oder von einem Nutzer mit mehreren Geräten verwendet werden können und eine neue Bedeutung annehmen, wenn sie gleichzeitig oder in einer bestimmten zeitlichen Abfolge geschehen. [HIN03]

In seinem Paper beschreibt er eine „Synchronous Gesture“, die von ihm prototypisch umgesetzt wurde. Das sogenannte „Bumping“, also Anstoßen von zwei mit Sensoren ausgestatteten Tablet Computern, stellt zwischen ihnen eine Verbindung her (Siehe Abbildung 1). Die Metapher, die für diese Geste verwendet wird, ist das Anstoßen zweier Trinkgläser.

Zur technischen Umsetzung werden dabei die durch den Bump erzeugten Accelerometerdaten der beiden Geräte miteinander verglichen. Wenn zum selben Zeitpunkt Ausschläge in entgegengesetzte Richtungen erfolgen, erkennen die Tablets sowohl, dass sie aneinander angestoßen wurden, als auch ihre räumliche Orientierung zueinander (Siehe Abbildung 2).

Sobald die sensorische Erkennung der „Bump-Geste“ erfolgt ist, wird der Bildschirm des einen Tablets von dem anderen erweitert und bildet so die doppelte Arbeitsfläche.



Abbildung 1: Bump Geste

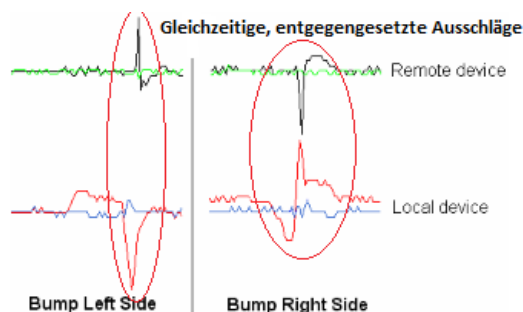


Abbildung 2: Bump Erkennung

Hinkley et al. (2004) definieren eine weitere „Synchronous Gesture“, die unter anderem eine Verbindung zwischen Tablet Computern erstellen kann. „Stitching“ lässt den Benutzer einen Stylus über den Bildschirmrand des einen Gerätes hinaus auf den Bildschirm des zweiten führen. Die Metapher für diese Geste ist das Nähen von Stoffen, wobei der Stylus als Nadel dient, um die beiden Bildschirme miteinander zu verbinden (Siehe Abbildung 3).

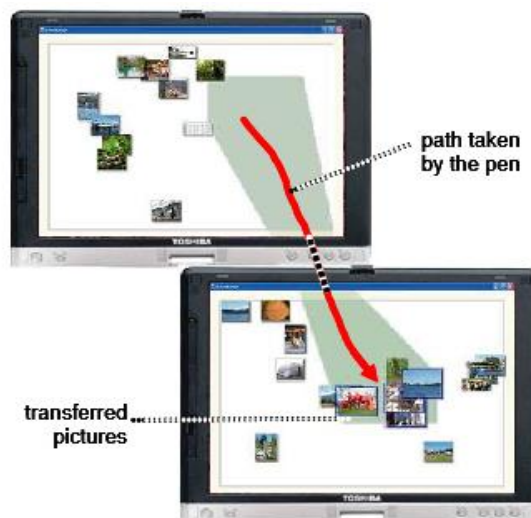


Abbildung 3: Pen-Stitching

Als Prototyp für diese Geste wurde eine Foto-Sharing Anwendung (Genannt Stitch-master) entwickelt, in der man Bilder sowohl von einem Gerät auf das andere verschieben, als auch auf beiden Bildschirmen erweitert anzeigen kann.

Um einen „Stitch“ zu erkennen, werden auch hier zeitnah erkannte Sensordaten miteinander verglichen. Legt man den Stift auf einem der Tablets auf und führt ihn über den Rand des Bildschirms, wird diese Information, samt des Winkels der Stiftführung, an den „Stitching-Server“ geschickt. Ebenso gehen Informationen beim Server ein, wenn von einem der Bildschirmränder eine Linie mit dem Stift gezogen wird. Geschehen diese Aktionen innerhalb einer kurzen Zeitspanne, werden Ausgangs- und Eingangswinkel verglichen.

Der Server registriert einen „Stitch“, wenn die Winkel ähnlich genug sind. Diese Methode garantiert, dass keine versehentlichen „Stitches“ erkannt werden. Außerdem im-

plizieren die Informationen im „Stitching-Server“ die räumliche Orientierung der Geräte zueinander. Man weiß, dass die Geräte nebeneinander liegen müssen, wenn der Stift das eine Tablet am rechten Rand verlässt und dann am linken Bildschirmrand des anderen eingeht.

2.3 Display Tiling

Ming Li und Leif Kobbelt (2012) stellen eine neue Methode des Display Tilings vor, die es erlaubt, dass mehrere mobile Geräte in willkürlicher Anordnung ein großes Bild darstellen können. Die Smartphones und Tablets dienen hier als bewegliche „Fenster“, durch die man Bildausschnitte sehen kann. Das Bild selbst befindet sich an einem festen Ort auf der Oberfläche, auf der die Geräte aufliegen (Siehe Abbildung 4).



Abbildung 4: Dynamic Display Tiling

Dies wird erreicht, indem eine Referenzmarkierung an der Decke über dem Tisch angebracht wird, welche von den Frontkameras der Smartgeräte erkannt wird. Durch diese visuellen Informationen kann die Position und Orientierung relativ zu den anderen Geräten errechnet werden. Sind diese Daten erst bekannt, kann ein Bild angezeigt werden. Zusätzlich zum Anzeigen kann man auf jedem der Bildschirme Smartphone-typische Gesten wie Pinch und Swipe durchführen und so mit dem Bild interagieren, als hätte man nur ein großes Display. **[LI12]**

Das Projekt „iPodWall“ von der Designfirma „Welikesmall“ benutzt 20 iPods, die an einem Holzkasten befestigt wurden, sodass sie ein 4x5 Raster ergeben. Mithilfe einer App, die auf jedem der Geräte installiert ist, läuft eine Slideshow von Fotos. Zusätzlich jedoch kann auch eines der Bilder auf allen Bildschirmen erweitert angezeigt werden (Siehe Abbildung 5). Wie man sieht, wurde das Foto in 20 gleich große Teile geteilt, wobei jeder der iPods eines anzeigt. Dies hat den Nachteil, dass der Abstand zwischen den Geräten und die Ränder nicht berücksichtigt werden, wie das bei **Ming Li und Leif Kobbelt** der Fall ist. **[WLS]**



Abbildung 5: iPodWall

Ein etwas anderes Konzept von Display Tiling wurde von **Seb Lee-Delisle (2011)** mit seinem Projekt „PixelPhones“ vorgestellt. Hierbei steht er auf einer Bühne vor einem Saal mit hunderten von Personen, die jeweils ihr Smartphone mit dem Display in Richtung Bühne in die Höhe halten. Zuvor wurde das Publikum dazu aufgefordert, eine Webseite damit aufzurufen, die jedes der Geräte ein eindeutiges Blinksignal durchführen lässt. Auf der Bühne steht eine Kamera, die dieses Blinken filmt, sodass jedem Smartphone eine ID zugeordnet werden kann. **[LEE11]**

Da nun die relativen Positionen der Geräte bekannt sind, kann ein simples Bild, wie ein Farbverlauf, auf über 200 Bildschirmen angezeigt werden, wobei jeder ein Pixel darstellt.

3 Analyse der Stitch-to-Tile Interaktion

Im folgenden Kapitel werden die beiden Bestandteile der Stitch-to-Tile Interaktion, nämlich die Bildschirmerweiterung und die Stitching Geste, separat analysiert. Diese Bestandteile stehen in einer Aktion-Reaktion-Beziehung; das heißt der Tiling Prozess wird durch die Stitching Geste ausgelöst.

Zunächst wird auf den gewünschten Effekt, also das Tiling, eingegangen.

3.1 Bildschirmerweiterung auf mobilen Endgeräten

Wie in den vorherigen Kapiteln bereits erwähnt wurde, ist die Bildschirmerweiterung bei der Nutzung von Desktop-PCs bereits seit geraumer Zeit etabliert. Auf mobilen Geräten ist dies bisher noch nicht der Fall. Um die Gründe für diese Tatsache zu verstehen, muss man den allgemeinen Begriff der Bildschirmerweiterung genauer betrachten.

Grundlegend ist die Erkenntnis, dass mehrere Formen des Screen Tilings existieren, die vollkommen unterschiedliche Anwendungen und somit auch voneinander unterschiedliche Anforderungen haben.

In dieser Arbeit wurden zwei Formen der Bildschirmerweiterung entwickelt und als Visuelles- und Logisches Tiling benannt. Im Folgenden werden diese definiert und erörtert.

3.1.1 Visuelles Tiling

Als „Visuelles Tiling“ wird hier eine Art der Bildschirmerweiterung bezeichnet.

Hierbei handelt es sich um die reine Erweiterung der Bildschirmfläche: Man betrachtet den - sonst nur auf einem Display angezeigten - Inhalt aufgeteilt auf mehreren Bildschirmen. Jedes Display zeigt dabei einen Teil des originalen Bildes an. Dies hat zur Folge, dass der dargestellte Inhalt nun größer ist.

Visuelles Tiling an mobilen Endgeräten unterscheidet sich von der klassischen Bildschirmerweiterung an Desktop-PCs vor allem in einem Punkt: Benutzt man am PC zwei Monitore, so hat man zunächst zwei separate Arbeitsflächen. Da Desktop-Betriebssysteme Fenster benutzen um Anwendungen anzuzeigen, kann man verschiedene

Fenster auf verschiedenen Monitoren anzeigen lassen, oder aber ein Fenster über mehrere Monitore anzeigen. Dieser zweite Anwendungsfall kommt dem Visuellen Tiling sehr nahe.

Anwendungen

Sinnvolle Anwendungen für Visuelles Tiling sind all jene, bei denen man von einer größeren Bildschirmfläche profitiert. Beispiele dafür wären Foto-, Video- oder Spieleanwendungen. Je mehr Fläche zur Verfügung steht, desto größer sind sowohl die Gesamtheit als auch die einzelnen Elemente des dargestellten Inhalts.

Somit könnte man die dargestellten Medien auch erkennen, wenn man weiter von den Geräten entfernt ist. Dies würde es auch erleichtern, solche Medien mit mehreren Personen zu konsumieren, da man nicht zusammen über einem kleinen Bildschirm kauern muss, um etwas zu erkennen (Siehe Abbildung 6).

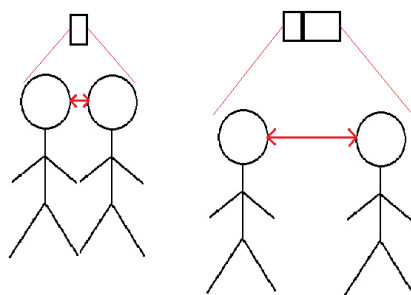


Abbildung 6: Entfernungen zwischen Bildschirmen und Personen

Die identifizierten Anwendungsfälle haben die Gemeinsamkeit, dass man mit Geräten nach dem Tiling-Vorgang nicht mehr direkt interagieren muss. Ist der Inhalt erst auf den Bildschirmen angezeigt, so konsumiert man ihn.

Eine weitere Anwendung für Visuelles Tiling ist, dass man Medien anzeigen kann, die nicht für gängige Bildschirmformate bestimmt sind. Ein Beispiel dafür wäre ein Panorama Foto. Dadurch, dass die Gesamtfläche des nutzbaren Bildschirms nicht mehr die Standardmaße hat, können solche Inhalte Visuell Getiled angezeigt werden und dadurch wieder größer erscheinen. (Siehe Abbildungen 7 und 8)

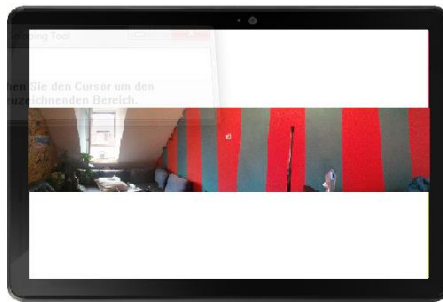


Abbildung 7: Panorama Foto auf Tablet

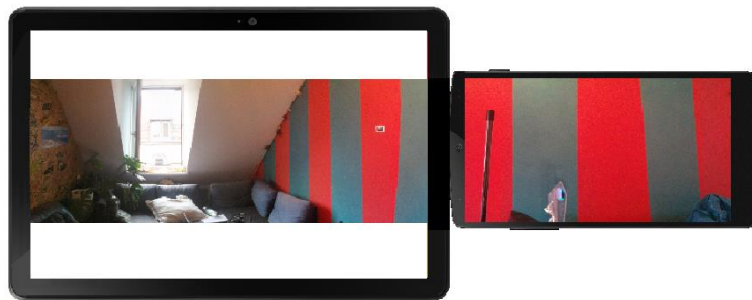


Abbildung 8: Panorama Foto von Tablet auf Smartphone erweitert

Die bisher genannten Anwendungen haben den Fokus, ein einzelnes Bild größer darzustellen, um die Betrachtung zu vereinfachen. Jedoch könnte man durch Visuelles Tiling auch zwei Hälften eines Bildes separat auf zwei Bildschirmen anzeigen. Dies wäre beispielsweise bei gescannten Büchern oder Magazinen hilfreich, da man die originalen Bedingungen simuliert. So hätte man jeweils eine physische Buchseite auf einem Bildschirm statt einer Bilddatei, die zwei Buchseiten enthält.

Anforderungen

Visuelles Tiling hat spezifische Anforderungen an Nutzer, Geräte und genutzte Applikationen.

Zunächst werden geeignete Bildschirmgrößen betrachtet. Hierzu werden zuerst die gängigsten Größen definiert:

- Smartphone: ~ 3 – 6 Zoll Bildschirmdiagonale
- Tablet: ~ 7 – 10 Zoll Bildschirmdiagonale
- Tabletop: ~ 30 – 40 Zoll Bildschirmdiagonale

Wie bereits erwähnt, ist das Ziel des Visuellen Tilings, mehr Fläche zu bekommen, auf der etwas angezeigt werden kann. Bei genauerer Betrachtung fällt jedoch auf, dass dies nicht mit allen Größenkombinationen gegeben ist. Ist der Unterschied zwischen den Bildschirmdimensionen zu groß, ist sogar das Gegenteil der Fall.

Da sowohl die Bildschirme der benutzten Geräte, als auch die angezeigten Inhalte im Normalfall rechteckig sind, entsteht bei Geräten mit sehr unterschiedlichen Bildschirmgrößen viel ungenutzte Fläche. (Siehe Abbildung 9, rot) Wie man sieht, ist die durch Visuelles Tiling entstandene Fläche in etwa so groß wie die Fläche des Tabletscreens allein. Wird der Größenunterschied noch größer, hat man nach dem Tiling sogar weniger Bildschirmfläche als davor.

Der größte Mehrwert (100%) entsteht, wenn man Geräte mit identischen Displaygrößen verwendet (Siehe Abbildung 10).

Jedoch gibt es auch bestimmte Medien, mit denen Visuelles Tiling auch mit unterschiedlich großen Geräten Sinn ergibt, wie zum Beispiel Panorama Fotos.



Abbildung 9: Display Tiling von Tablet auf Smartphone

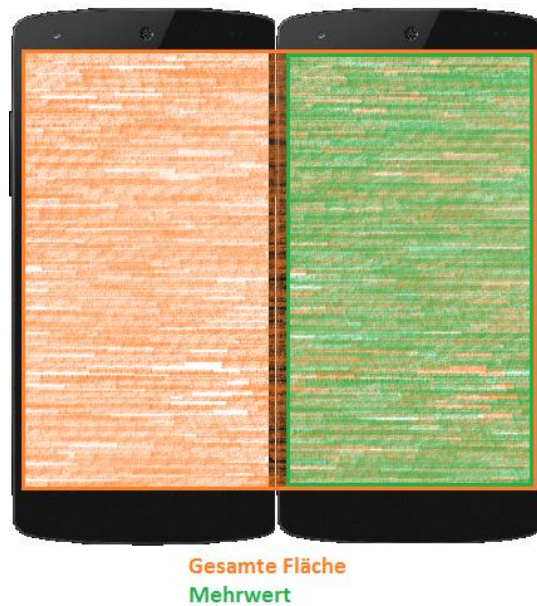


Abbildung 10: Display Tiling von Smartphone auf Smartphone

Eine weitere Anforderung ist, dass die benutzten Geräte direkt aneinander liegen müssen, d.h. parallel zueinander, mit null Millimeter Abstand und so, dass möglichst viel Bildschirmfläche gewonnen wird.

In jedem Fall muss den Geräten vor dem Tiling-Vorgang ihre genaue räumliche Orientierung zueinander bekannt sein.

Probleme

Ein Problem des Visuellen Tilings ist die bereits erwähnte, verschwendete Bildschirmfläche bei unterschiedlich großen Geräten. Dieses Problem ließe sich höchstens lösen, indem eine Applikation die Bildschirmgrößen der Geräte in Betracht zieht und Inhalt generiert, der genau die Fläche der Displays ausfüllt. Für traditionelle Medien, wie Fotos oder Videos, wäre dies jedoch auch nicht sehr hilfreich, da dann ein Teil des Bildes fehlen würde. Solch eine Verwendung von Visuellem Tiling könnte jedoch in Spielen, wie beispielsweise dynamischen Puzzles, Anwendung finden.

Ein weiteres Problem, welches auch bei der Bildschirmerweiterung an Desktop-Computern auftritt, ist die Lücke zwischen den Displays. Jeder Bildschirm hat einen Rand, was dazu führt, dass durch Nebeneinanderlegen zweier Geräte eine Lücke zwischen den Displays entsteht. Dadurch, dass mobile Endgeräte viel kleiner sind als Computermonitore, ist die Lücke zwischen den Geräten relativ zu ihren Dimensionen größer. Man hat zwei Möglichkeiten damit umzugehen.

Die leichter umzusetzende Variante ist es, die Lücke zu ignorieren. Will man zum Beispiel ein Bild erweitert anzeigen, würde der Teil des Bildes, der sich am Rand zwischen den Geräten befindet, breiter erscheinen. Es entsteht also ein Versatz. (Siehe Abbildung 11, links) Der Vorteil dieses Vorgehens ist, dass man nichts von dem Bild verliert, also jeder Pixel angezeigt wird. Dies wird auch so bei der Bildschirmvergrößerung an PCs gehandhabt.

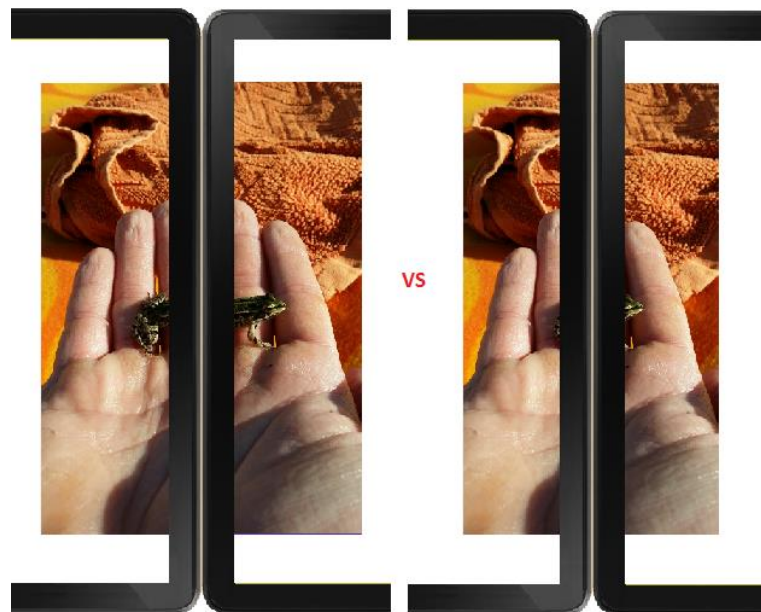


Abbildung 11: Display Tiling und Umgang mit Bildschirmrändern

Die zweite Möglichkeit ist, die Lücke miteinzubeziehen. In diesem Fall gäbe es keinen Versatz, jedoch würde ein Teil des Bildes fehlen, da dieser „hinter den Rändern versteckt“ wäre. (Siehe Abbildung 11, rechts) Diese Methode ist technisch schwieriger umzusetzen, da den Geräten ihre eigene Randdicke bekannt sein müsste, um berechnen zu können, wie viele Pixel aus dem Bild herauszuschneiden sind.

3.1.2 Logisches Tiling

Als „Logisches Tiling“ wird hier die zweite Form der Bildschirmvergrößerung an mobilen Endgeräten bezeichnet. Im Gegensatz zu Visuellem Tiling geht es nicht nur um reinen Bildschirmflächenzuwachs. Stattdessen wird bei dieser Art von Tiling ein Teil der Programmlogik ausgelagert- sowohl um Funktionalitäten zu erweitern, als auch um die User Experience zu verbessern.

Dieser Ansatz hat viele Parallelen zu „Second Screen“ Anwendungen. Das Konzept von Second Screen ist, dass man als Fernsehzuschauer parallel zu der ausgestrahlten Sendung sein mobiles Endgerät benutzt, um zum Kontext relevante Informationen angezeigt zu bekommen. Zusätzlich gibt es in vielen Second Screen Applikationen Interaktionsmöglichkeiten wie Umfragen, Chat oder Quiz. Den Fernsehbildschirm nennt man hier First Screen, den Mobilgerätbildschirm den namensgebenden Second Screen.

[WIK15a]

Logisches Tiling hat sowohl als First-, als auch als Second Screen ein mobiles Endgerät. Dieser Unterschied scheint zunächst nicht allzu groß zu sein. Bei näherer Betrachtung rechtfertigt er jedoch die Benutzung einer neuen Bezeichnung: Logisches Tiling.

Man kann Logisches Tiling zwar in der gleichen Form benutzen wie Second Screen, wobei das Hauptgerät ein an sich non-interaktives Medium, wie einen Film oder Stream, anzeigt und das Zweitgerät Informationen und Interaktionsmöglichkeiten anbietet. Was Logisches Tiling jedoch von Second Screen abhebt, ist, dass auf beiden Geräten vollständig interaktive Medien dargeboten werden können.

Anwendungen

Die offensichtlichste Anwendung für Logisches Tiling ist jene, welche auch am nächsten an Second Screen Anwendungen liegt.

Man könnte sich auf seinem Hauptgerät einen Stream anschauen, das heißt ein Video, welches live aufgenommen und gesendet wird. Streamer interagieren oftmals mit ihren Zuschauern über den zum Stream gehörigen live-Chat, man kann die Sendung also direkt beeinflussen. Dies macht einen großen Teil des Reizes aus, Live-Streamern zuzuschauen.

Schaut man sich solch einen Stream jedoch auf einem kleinen Bildschirm wie dem eines Tablets an, so nimmt das Chatfenster einen großen Teil der Bildschirmfläche in Anspruch.

Eine Streaming Applikation, die Logisches Tiling benutzt, könnte dem Nutzer die Möglichkeit geben, den Chat auf ein weiteres Mobilgerät auszulagern. Dies würde es erlauben, dass das gesamte Tabletdisplay den Stream anzeigt, während man den Chat-Teil auf seinem Smartphone benutzt.

Eine weitere mögliche Anwendung würde sich beispielsweise bei Lernapplikationen anbieten. So hätte man auf dem Hauptgerät Informationen und Fragen, die es zu beantworten gilt. Auf dem Zweitgerät hätte man dann die Möglichkeit, die Antworten einzugeben, ohne dass der auf dem Hauptbildschirm angezeigte Inhalt sich verändern würde.

Auch für Mehrspieler-Spiele ist Logisches Tiling ein interessantes Feature. Ein Spiel könnte jedem Spieler eigene, nur von ihm ersichtliche, Informationen und Spielmechaniken anbieten. Auf einem anderen Gerät würde dann der Gesamtfortschritt, beziehungsweise das Spielziel, angezeigt werden.

Es gibt bereits Spiele, die ähnliche Spielmechaniken verwenden. Ein bekannter Vertreter ist das kooperative Mehrspieler-Gesellschaftsspiel „Spaceteam“ von „Sleeping Beast Games“, welches es für Android Geräte gibt.

Das Spielziel von Spaceteam ist es, gemeinsam ein Raumschiff zu fliegen und zu navigieren. Dabei wird jedem Spieler seine eigene Rolle, wie Techniker oder Navigator, zugewiesen. Der Spieler in der Rolle des Kapitäns bekommt alle Informationen auf seinem Gerät angezeigt und muss dem jeweils dafür zuständigen Spieler verbal Befehle erteilen. Dieses Spiel basiert auf dem Prinzip des Logischen Tilings, wobei das „Kapitänsgerät“ als Hauptgerät dient, welches die Informationen vereint, während die „Crewgeräte“ die Interaktionsmöglichkeiten anbieten. [SBG12]

Anforderungen

Anders als beim Visuellen Tiling spielt die relative Größe der mobilen Endgeräte keine wichtige Rolle, da nicht ein Bildelement „zerschnitten“ und auf verschiedenen Bildschirmen angezeigt werden muss. So kann man theoretisch jegliche Kombination von Geräten verwenden; Smartphone zu Tabletop ist ebenso möglich wie Tablet zu Tablet.

In den vorher genannten Anwendungsbeispielen ist es jedoch am sinnvollsten, das größte vorhandene Gerät als Hauptgerät zu verwenden.

Dies ermöglicht es, die relevantesten Inhalte auf dem größten Bildschirm anzuzeigen, während die kleineren Geräte meist als Eingabegeräte dienen.

Auch ist die Position der Geräte zueinander nicht festgelegt, solange sich diese innerhalb des Verbindungsradius befinden. Man hat im Gegensatz zu Visuellem Tiling keine grafischen Komponenten, die möglichst lücken- und versatzlos über mehrere Geräte hinweg angezeigt werden sollen. Die einzigen Anforderungen an die Position der Geräte ist, dass der Nutzer den Hauptbildschirm sieht und sich nahe genug an den sekundären Geräten befindet, um diese benutzen zu können.

Die wichtigsten Anforderungen müssen die Applikationen erfüllen, die diese Art von Tiling implementieren. Bei Visuellem Tiling könnte man theoretisch jeden beliebigen Bildschirm zerteilen und auf mehreren Geräten anzeigen. Will man in einer Applikation Logisches Tiling verwenden, so braucht man von Anfang an sinnvoll modularisierte Teilanwendungen, die auf andere Geräte ausgelagert werden können. In dem vorher genannten Streaming-App Beispiel wären die beiden Module das eigentliche Video und der Chat-Teil.

3.2 Analyse der Stitching Geste

Will der Nutzer eine der Tiling-Funktionalitäten nutzen, so braucht er eine Funktion, welche das Tiling auslöst. Da allen Geräten, die an der Bildschirmerweiterung beteiligt sein sollen, der Befehl zum Tiling erteilt werden und dafür eine Verbindung zwischen diesen Geräten geschaffen werden muss, handelt es sich hierbei um eine Art des Spontaneous Device Sharing Problems. Wie bereits in Kapitel 2.1 beschrieben, können Synchronous Gestures eine gute Lösung für solche Probleme liefern.

Stitch-to-Tile erfüllt alle Kriterien einer Synchronous Gesture. Der Benutzer möchte eine gemeinsame Funktionalität zweier Geräte nutzen, indem er seinen Finger von dem Bildschirm eines der Geräte auf den des anderen führt. Beide Teile der Geste, nämlich der ausgehende Swipe auf dem ersten Gerät und der eingehende auf dem zweiten, geschehen in kurzer zeitlicher Abfolge. Zusätzlich werden die Swipes durch die Sensoren der Geräte registriert.

Eine positive Eigenschaft der Geste ist, dass der Benutzer alle dahinterstehenden Konzepte bereits versteht. Mit Stitch-to-Tile legt der Nutzer den Finger, wie bei einem Swipe, auf den Bildschirm des Gerätes, welches er erweitern möchte und fährt über den Bildschirmrand hinaus auf den Bildschirm des zweiten Gerätes.

„Drag & Drop“ kennen die meisten Menschen, die schon mit Computern gearbeitet haben. Möchte man den Bildschirm des einen Smartphones nun auf dem anderen angezeigt bekommen, ist es einfach sich vorzustellen, dass man diesen „aufhebt“, indem man ihn berührt und auf den Bildschirm des zweiten Gerätes zieht.

Zusätzlich haben die meisten mobilen Endgeräte heute eine sehr ähnliche Tiefe zueinander und eine Frontseite, die meist nur aus einer durchgängigen glatten Oberfläche besteht. Hat man nun zum Beispiel zwei Smartphones nebeneinander auf einem Tisch liegen, sind diese in den meisten Fällen bündig. Dies erleichtert die Ausführung der Stitch-to-Tile Geste.

Hinkley und Kollegen (2004) haben sechs, als Designfragen formulierte, Anforderungen an Lösungen des Spontaneous Device Sharing Problems gestellt:

- **Connection:** Wie wird eine Verbindung hergestellt?
- **Command:** Was für eine Art Verbindung ist benötigt?
- **Operands:** Welche Informationen werden geteilt?
- **Geometry:** Was ist das räumliche Verhältnis zwischen den Geräten?
- **Coexistence:** Wie Koexistieren die Verbindungsgesten mit traditionellen Interaktionen oder natürlich auftretendem Benutzerverhalten?
- **Proxemics:** Wie teilen sich die Benutzer den physischen Raum?

Diese Fragen werden im folgenden Kapitel beantwortet.

3.3 Design der Stitch-to-Tile Geste

Connection: Aufbau der Verbindung zwischen den Geräten

Die eigentliche Stitching-Geste funktioniert mithilfe eines Floating Action Buttons mit einem Kreuz aus zwei Pfeilen als Symbol (Siehe Abbildungen 12 & 13). Der Nutzer kann diesen mit seinem Finger willkürlich auf dem Bildschirm bewegen. Lässt er ihn los, begibt sich der Button wieder in seine ursprüngliche Position in der Mitte des Bildschirms zurück. Nur wenn der Button an den Bildschirmrand geführt wird und diesen berührt, verbleibt er in dieser Position. Führt der Nutzer seinen Finger weiter auf den Bildschirm des zweiten Gerätes, so erscheint der Button wieder unter seinem Finger. Dies soll, schon bevor die Bildschirme getiled werden, den Eindruck vermitteln, dass man ein gemeinsames Display hat - nicht eines pro Gerät.

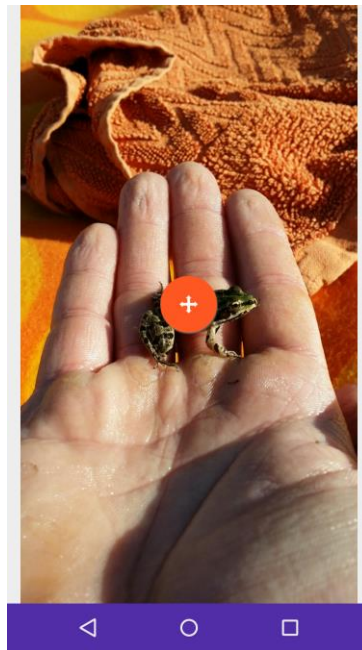


Abbildung 12: FAB zu Beginn der Interaktion

Dadurch, dass man den Stitch nicht an einem willkürlichen Ort auf dem Display beginnen kann, sondern ein Floating Action Button zugeschaltet wird, kann man Stitch-to-Tile leichter in andere Applikationen einbauen, ohne mit bestehenden Funktionen zu kollidieren. Die funktionalen Elemente der Applikation würden dann „unter“ der durch Swipe-to-Tile benutzten Ebene liegen.

Ein weiterer Grund dafür, dass ein Floating Action Button für die Interaktion genutzt wird, hat mit der Benutzerfreundlichkeit zu tun. Ohne müsste das Konzept des Stitching zunächst erklärt werden. Das Symbol auf dem Button soll dem Nutzer signalisieren, dass dieser beweglich ist und macht die Geste verständlicher.

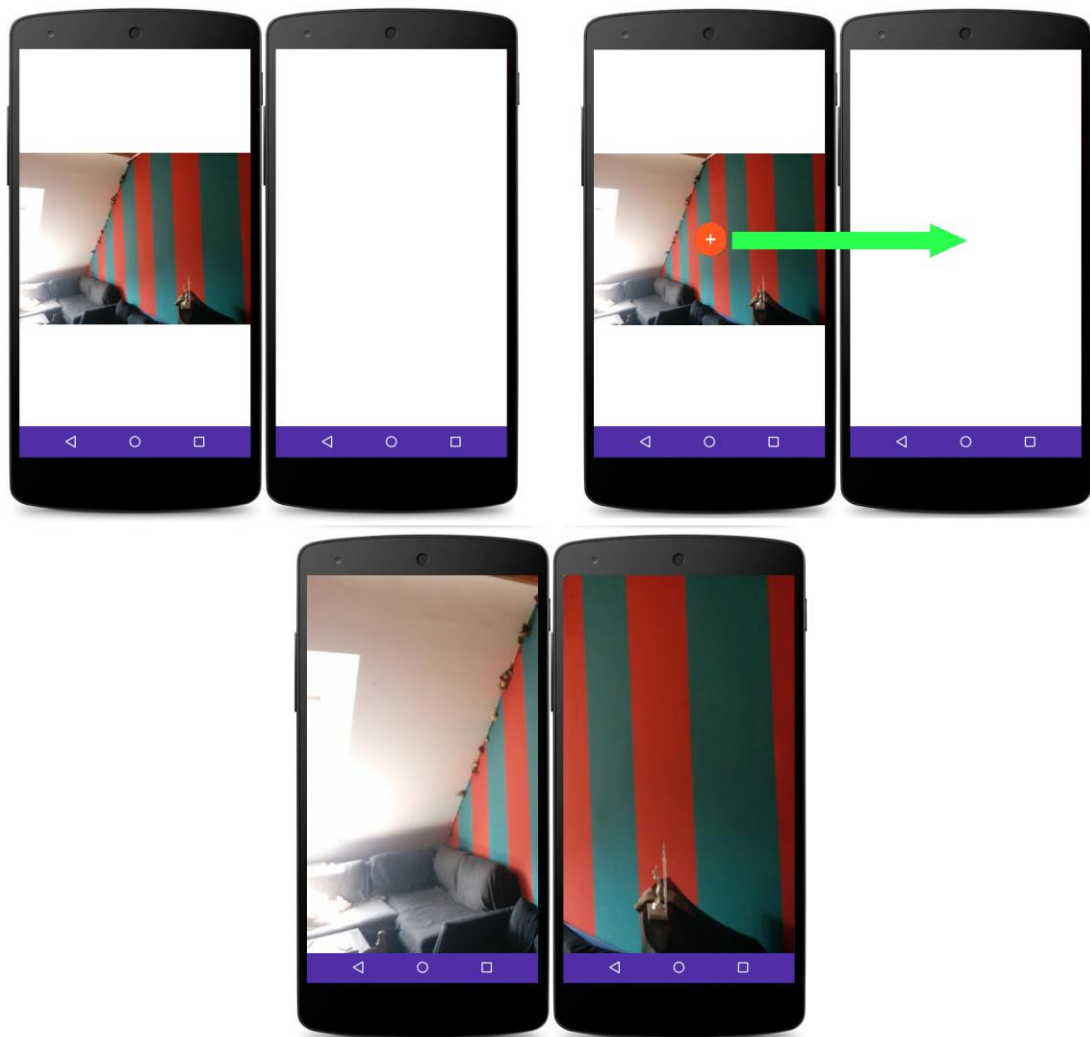


Abbildung 13: Ablauf der Interaktion

Außerdem gibt der Button dem Benutzer ein direktes visuelles Feedback darüber, ob er den Stitch korrekt ausgeführt hat. Verbleibt der Floating Action Button am Rand des ersten Gerätes und taucht am anderen Rand des zweiten auf, wurde alles richtig ausgeführt, sodass die beiden Swipes miteinander verglichen werden können um zu bestimmen, ob sie zusammen einen Stitch ergeben.

Stitch-Erkennung

Technisch besteht ein Stitch aus zwei Swipes: Einem aus- und einem eingehenden, die in direkter zeitlicher Abfolge ausgeführt werden. Damit ein Stitch als solcher erkannt wird, müssen drei Bedingungen erfüllt werden.

1. Die Grundrichtungen der beiden Swipes müssen gleich sein.
2. Der zeitliche Abstand zwischen den beiden Swipes darf nicht zu groß sein.
3. Die Winkel, in denen die Swipes die Ränder schneiden, dürfen nicht zu unterschiedlich sein (± 10 Grad).

Eine positive Nebenwirkung, die durch das Benutzen eines Floating Action Buttons entsteht, ist, dass damit der Startpunkt eines jeden Stitches gegeben ist, da sich der Button immer in der Mitte des Bildschirms befindet. Der tatsächliche Pfad des Swipes, den der Nutzer mit dem Finger ausführt, ist für die Erkennung eines Stitches nicht relevant. Der Pfad, der von dem Gerät erkannt wird, ist die Verbindung zwischen Start- und Endpunkt eines Swipes (Siehe Abbildung 14).

Auf dem Gerät, auf dem sich der zu erweiternde Inhalt befindet, sind diese beiden Punkte der Mittelpunkt des Bildschirms und die Stelle an dem der Swipe den Bildschirmrand schneidet. Auf dem anderen Gerät ist es der Schnittpunkt des Bildschirmrandes und der Punkt an dem der Nutzer den Finger vom Display nimmt. Wenn zwischen den beiden Geräten noch weitere sind, dann sind die beiden erkannten Punkte jeweils die Schnittpunkte an den beiden Bildschirmrändern.

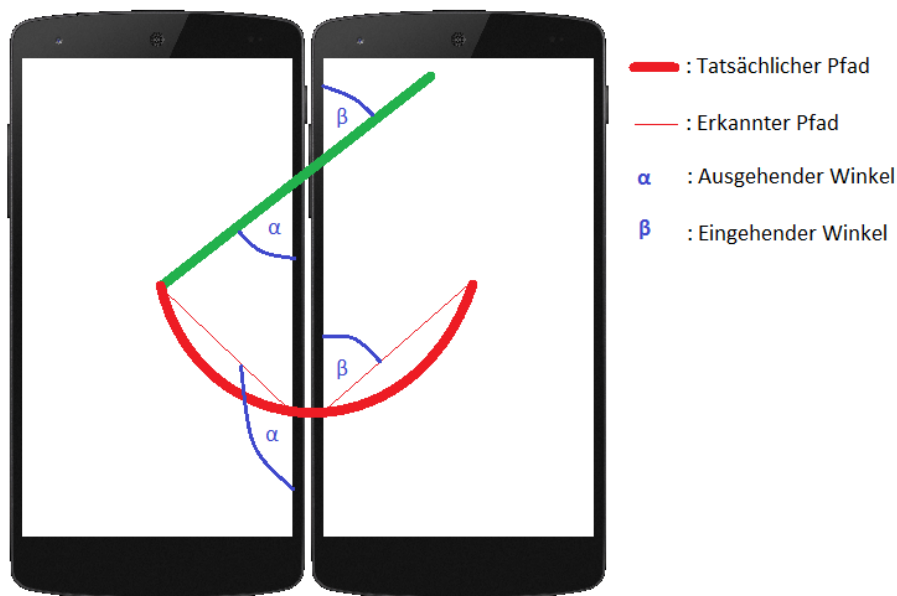


Abbildung 14: Stitch-Erkennung

Command: Was für Verbindungen können hergestellt werden?

Der hier entwickelte Prototyp beschränkt sich explizit auf die rein visuelle Erweiterung eines Bildschirms auf den nächsten. Dies ist jedoch keine Einschränkung der Stitching Geste an sich, sondern wurde gewählt um den Scope des Projektes klein zu halten und die Validität der Geste zu verdeutlichen. Die Geste stellt lediglich einen Auslöser für eine Interaktion zwischen Geräten dar. In diesem Fall ist der ausgelöste Effekt Visuelles Tiling.

Will der Nutzer etwas visuell tilen, so erscheint der Button mit den gekreuzten Pfeilen. Führt der Benutzer - ausgehend von dem Gerät mit dem angezeigten Bild auf ein anderes, das die Stitch-to-Tile Applikation geöffnet hat- einen Stitch durch, wird das Bild erweitert angezeigt.

Dies bedeutet, dass abhängig von den Bildschirmflächen und deren Seitenverhältnissen, das Bild digital zerschnitten wird. Beide Geräte zeigen nun jeweils einen Teil des Bildes auf ihrem Bildschirm an.

Aufbauend auf diesem Prototyp könnte man in Zukunft weitere Funktionalitäten, wie das reine Senden von Daten oder eine Art Logisches Tiling (Siehe Kapitel 3.1.2), hinzufügen. So könnte man beispielsweise einen Video Stream auf seinem Tablet anschauen und durch einen Stitch auf sein Smartphone dort im Chat mit anderen Zuschauern interagieren.

Operands: Auswahl der zu teilenden Daten

Für das Stitch-to-Tile Pattern gibt es nur implizit einen Operanden. Der aktuell angezeigte Bildschirm ist die Datei, welche sich die Geräte teilen.

Geometry: Wo befinden sich die Geräte relativ zueinander?

Um die Stitch-to-Tile Geste sinnvoll nutzen zu können, sollten sich die Geräte parallel und mit möglichst wenig Abstand zueinander befinden. Im Idealfall liegen die Geräte so, dass sie sich berühren. Das heißt das Zweitgerät sollte entweder links, rechts, über oder unter dem Hauptgerät liegen. Auf einer flachen Oberfläche, wie einem Tisch, ist es am einfachsten die Tiling-Funktionalität zu nutzen. Jedoch könnte der Nutzer auch beide Geräte in der Hand halten und mit der zweiten den Stitch durchführen.

Sollen mehr als zwei Geräte für einen Stitch-to-Tile benutzt werden, so müssen die Geräte so liegen, dass der Nutzer mit dem Wisch eines Fingers über alle Bildschirme fahren kann. Läge das Hauptgerät mit dem Bild, das man erweitern möchte, in der Mitte zwischen zwei anderen Geräten, müsste man zwei Stitches durchführen, was den Nutzer verwirren könnte. (Siehe Abbildung 15)

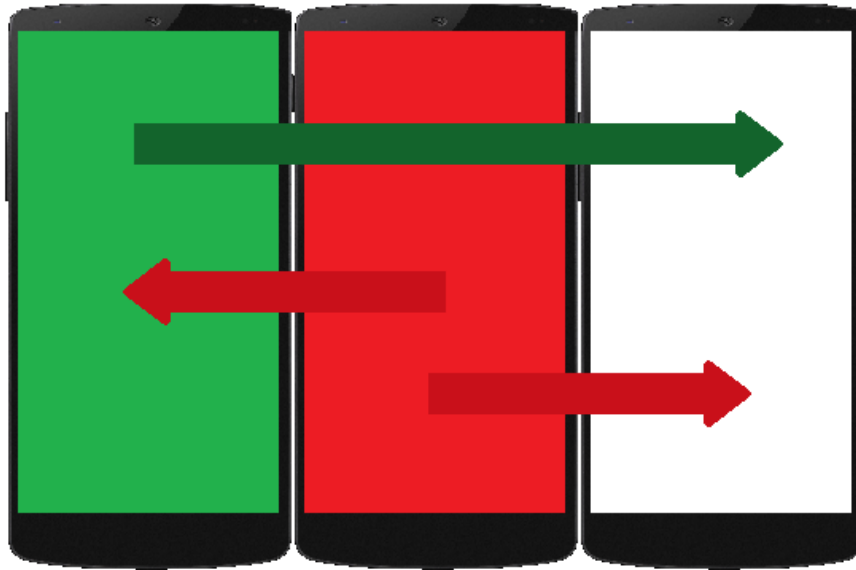


Abbildung 15: Stichrichtung bei mehr als 2 Geräten

Theoretisch könnte die Stitch-to-Tile Funktion auch von zwei Personen genutzt werden, wobei die Geräte nicht nebeneinander liegen müssten. Die Voraussetzung wäre, dass sie sich innerhalb der Signalreichweite befänden und die Nutzer ihre Eingaben gut aufeinander abstimmten. Wenn beispielsweise die Möglichkeit bestehen würde Daten auszutauschen, könnte dies eine sinnvolle Anwendung darstellen.

Coexistence: Verhalten der Geste im Zusammenspiel mit anderen Funktionen

Auf einem mobilen Smartgerät ist in fast jede Interaktion eine Berührung des Displays involviert. Dies muss man in Betracht ziehen, will man eine neue „Synchronous Gesture“ entwickeln, welche auch die Berührungssensoren der Bildschirme verwendet. Die erste Hälfte eines Stitches ist ein Swipe von der Mitte des Bildschirms, wo sich der

Floating Action Button befindet, zum Rand. Dies macht es einerseits leicht verständlich, da jeder Nutzer eines Smartgerätes diesen Bewegungsablauf versteht und gewohnt ist.

Andererseits muss man in der Applikation, die die Stitch-to-Tile Interaktion implementieren möchte darauf achten, dass die Geste von der App nicht missverstanden wird. Viele Apps, wie zum Beispiel diverse Foto Galerien, nutzen einen Swipe zur Seite des Bildschirms zum Blättern auf das nächste Bild. Solch eine Geste könnte leicht als der Anfang eines Stitches erkannt werden, wenn die App nicht richtig damit umgeht. Theoretisch könnte jede Art von Touch-Event in einer Applikation mit Stitch-to-Tile in Konflikt treten, weshalb an dieser Stelle eine mögliche Lösung dieses Problems vorgeschlagen wird.

Viele Apps haben eine Share-Funktion integriert, die durch Antippen eines Buttons eine Liste von Möglichkeiten anzeigt, wie man die Daten teilen möchte. Zum Beispiel kann man Fotos aus der Galerie App über diese Funktionalität direkt auf ein soziales Netzwerk posten oder per Email an Freunde verschicken. Stitch-to-Tile könnte als eine dieser Methoden hinzugefügt werden, was den Stitch-to-Tile Floating-Action-Button erscheinen lassen würde.

Durch Zuschalten des Buttons verhindert man, dass ein Stitch an einem willkürlichen Ort begonnen wird und somit mit anderen Funktionen verwechselt werden könnte.

Ein weiteres Problem könnte entstehen, wenn der Benutzer auf seinem Gerät andere Apps installiert hat, die die Funktionalität des „eingehenden Swipes“ nutzen. Viele Custom-Launcher-Apps erlauben dem Nutzer sich selbst Funktionen zu definieren, wenn er beispielsweise vom unteren oder linken Rand hereinswiped. Der zweite Teil eines Stitches würde dann diese Funktion ausführen und nicht den gewünschten Stitch-to-Tile Vorgang starten.

Proxemics: Wie teilen sich die Benutzer den physischen Raum?

Proxemics bezeichnet eine Subkategorie des Fachgebiets der nonverbalen Kommunikation zwischen Menschen. **[WIK15b]** Geht man davon aus, dass nur ein Benutzer mehrere Geräte nutzt um den Bildschirm eines dieser Geräte zu erweitern, spielen Proxemics keine Rolle bei der Stitch-to-Tile Interaktion.

3.4 Gegenüberstellung von Stitch-to-Tile und Hinkleys Pen-Stitching

Die Stitching-Geste wurde bereits in Kapitel 2.2 als Methode zum Verbindungsaufbau zwischen Geräten genannt. In jenem Fall wurde ein Stylus benutzt um zwei Tablet-Computer miteinander zu koppeln [HIN04]. Diese Methode wird nachfolgend Pen-Stitching genannt, um eine Abgrenzung zu der hier entwickelten Stitch-to-Tile Geste zu schaffen.

Im Gegensatz zu Stitch-to-Tile, wobei kein zusätzliches Eingabegerät nötig ist, benutzt Pen-Stitching Geräte, die nur über einen speziellen Stylus bedienbar sind. Dies wird in erster Linie als Nachteil gesehen, da die Zugänglichkeit der Geste dadurch sinkt. Nicht jeder besitzt ein Gerät, das mit einem Stylus bedienbar ist.

Jedoch bestärkt der Stylus die Metapher des Nähens weiter: Es handelt sich dabei um ein Werkzeug zum Verbinden zweier Dinge vergleichbar mit einer Nadel, die man benutzt, um zwei Stücke Stoff zusammenzunähen. Diese Metapher wird auch in Stitch-to-Tile übernommen, wirkt jedoch etwas schwächer aufgrund des fehlenden Werkzeuges.

Außerdem benutzt Pen-Stitching eine weitere Metapher, die mit dem benutzten Stift zusammenhängt. Als visuelles Feedback wird eine Malspur - wie bei einem echten Stift auf einem Blatt - auf dem Bildschirm gezeichnet, sobald man den Stylus aufsetzt. Auf diese Malspur wird bei Stitch-to-Tile verzichtet, zum einen, weil man keinen Stift für diese Interaktion benutzt und zum anderen, weil der Floating Action Button für die Rolle des visuellen Feedbacks gewählt wurde.

Vergleicht man die für die beiden Gesten verwendeten Geräte weiter, so fällt auf, dass die Bildschirme beim Pen-Stitching einige Millimeter in dem Gehäuse des Tablet-Computers eingebettet sind. Dies ist bei den modernen mobilen Endgeräten, welche bei Stitch-to-Tile verwendet werden, nicht der Fall.

In Tabelle 1 werden die wichtigsten Unterschiede zwischen der, hier entwickelten, Stitch-to-Tile Geste und **Hinkley's (2004)** Pen-Stitching tabellarisch gegenübergestellt.

Merkmal	Stitch-to-Tile	Pen-Stitching
Benutzte Geräte	Smartphones/Tablets	Tablet-Computer
Betriebssystem	Mobiles Betriebssystem (Android, iOS, etc.)	Desktop Betriebssystem (Windows, Linux, etc.)
Größe der Geräte	Ca. 3 – 12 Zoll Bild- schirmdiagonale	Ca. 15 Zoll Bildschirmdia- gonale
Eingabegerät	Finger	Spezieller Stylus
Anzahl Benutzer	Min. 1	Min. 2
Haupt-Anwendungsfall	Erweiterung der Bild- schirme (Visuelles- oder Logisches Tiling)	Kollaboration zwischen Personen und ihren Gerä- ten. Daten kopieren, schi- cken, zusammen bearbei- ten.
Visuelles Feedback	Floating Action Button	„Malspur“ des Stylus
Haptisches Feedback	Vibration bei erfolgrei- chem Stitch	---

Tabelle 1: Gegenüberstellung der Gesten

4 Prototypische Umsetzung der Interaktion

In Kapitel 3 wurde die gesamte Stitch-to-Tile Interaktion analysiert und konzipiert. Sie setzt sich aus zwei Teilen zusammen, nämlich der Bildschirmerweiterung bei mobilen Endgeräten und der Geste, die diese auslöst. Für die Bildschirmerweiterung wurden zwei Möglichkeiten definiert, Visuelles- und Logisches Tiling.

Will man Stitch-to-Tile nun in Form einer Applikation umsetzen, muss man zunächst entscheiden, welche Form des Tilings man implementieren möchte.

Der hier entwickelte Prototyp ermöglicht Visuelles Tiling auf mobilen Endgeräten, die das Android-Betriebssystem benutzen. Es wird also ein Bild, welches sich auf einem Smartphone oder Tablet befindet, auf ein zweites erweitert, sodass das Bild nun größer zu sehen ist.

Während der Implementierung der Interaktion kamen sowohl vom Betriebssystem, als auch von den Geräten Einschränkungen auf, mit denen es umzugehen galt. Diese werden an den passenden Stellen erwähnt.

Einfachheitshalber beschränkt sich der Prototyp auf das Tiling von nur zwei Geräten. Dies könnte man in zukünftigen Versionen erweitern.

Im Folgenden wird die Umsetzung dieses Prototyps beschrieben, wobei zuerst auf die gewählte Technologie zur Datenübertragung eingegangen wird. Im Anschluss darauf wird der gesamte Ablauf der Interaktion dargestellt. Zuletzt werden die einzelnen interessanten Bestandteile genauer betrachtet.

4.1 Wahl der Kommunikationstechnologie

Da Stitch-to-Tile einen Lösungsansatz für das Spontaneous Device Sharing Problem liefern soll, braucht man ein technisches Mittel, um die beteiligten Geräte miteinander zu verbinden.

Es gibt eine Vielzahl an Technologien, die für die Kommunikation zwischen Geräten in Frage käme, wie zum Beispiel Bluetooth, NFC oder Wi-Fi Direct. Hat man die direkten Adressen der Geräte oder schaltet einen Server zwischen sie, könnte man Daten auch über das Internet versenden.

Zunächst muss man also entscheiden, ob man die Geräte direkt miteinander verbinden möchte oder aber die Verbindung über einen Server laufen soll. Mit dieser Entscheidung schränkt man die Auswahl an nützlichen Technologien ein.

Die eine Möglichkeit wäre, dass alle Daten, wie die zur Stitckerkennung nötigen Swipe Informationen und das zu tilende Bild, an einen Server geschickt, dort verarbeitet und dann weiter an das Zweitgerät geleitet werden.

Für diese Herangehensweise gäbe es zwei valide Methoden:

1. Der Server befindet sich in unmittelbarer Nähe (ist im selben WLAN Netzwerk oder bietet Bluetooth-, NFC-, Wi-Fi Direct Funktionalität) zu den benutzten mobilen Endgeräten und bietet die Möglichkeit sich so mit ihm zu verbinden.
2. Der Server ist nicht unmittelbar erreichbar, sondern empfängt und sendet Daten über das Internet.

Beide Methoden hätten den Vorteil, dass sich die beteiligten Geräte nicht gegenseitig kennen müssen. Ist ein Gerät mit dem Server verbunden, so sendet es einfach alle relevanten Daten an diesen und die Logik im Server entscheidet, was damit geschehen soll. Die nötigen Berechnungen zur Überprüfung, ob ein Stitch ausgeführt wurde, sowie das Zerteilen des Bildes könnte der Server übernehmen und somit die Smartgeräte entlasten.

Die erste Methode wurde verworfen, da das Aufsetzen eines lokalen Servers ein zusätzliches Gerät erfordern und dies die Einstiegshürde für die potentiellen Benutzer der Stitch-to-Tile Funktionalität höher setzen würde.

Der Endnutzer sollte möglichst keine Eigenleistung erbringen müssen, um das Produkt gebrauchen zu können.

Bei der zweiten der beiden Methoden wäre die anfängliche Verwendung zwar benutzerfreundlicher, jedoch wirft sie ihre eigenen Probleme auf. Zum einen müssten App-Entwickler einen Stitching-Server aufsetzen, um die Stitch-to-Tile Funktionalität in ihre Applikation zu integrieren. Dies würde die Betriebskosten erhöhen, sofern die App nicht schon eine existierende Serverinfrastruktur benutzt. Selbst wenn Server bereitstünden, würde die Implementierung von Stitch-to-Tile zusätzliche Backend-Programmierung erfordern.

Zum anderen wäre das Stitching dann von weiteren Faktoren abhängig: Signalstärke des mobilen Internets, Kosten für die mobile Datennutzung, Serverausfälle etc.

Aus diesen Gründen wurde entschieden, dass die Server Herangehensweise für Stitch-to-Tile ungeeignet wäre.

Also würden sich die Geräte direkt miteinander verbinden und könnten ohne „Mittelsmann“ Daten miteinander austauschen. Dies bietet den entscheidenden Vorteil, dass für die Verwendung keine weiteren Ressourcen von Nöten sind. Der Endnutzer hat keine zusätzliche Arbeit und die App-Entwickler keine zusätzlichen Kosten. Dies erhöht die Wahrscheinlichkeit dafür, dass sich eine Form von Stitch-to-Tile etabliert.

Für diesen ersten Prototyp wurde die Verbindung und Kommunikation mit Wi-Fi Direct umgesetzt. Die beiden offensichtlichen Alternativen wären Bluetooth und NFC. Da potentiell Bilder mit sehr hohen Auflösungen und somit einer großen Dateigröße für die Bildschirmerweiterung genutzt werden sollen und Wi-Fi Direct die schnellste Datenübertragungsrate dieser drei Technologien ermöglicht, wurde diese gewählt.

Wi-Fi Direct

Mit Wi-Fi Direct können Geräte ein Ad-hoc Netz miteinander eröffnen und so, ähnlich wie über Bluetooth, direkt miteinander kommunizieren. Für Entfernungen von bis zu Zehn Metern, die man mit Stitch-to-Tile niemals erreicht, liegt die Datenübertragungsrate bei 54 Megabit pro Sekunde. Eine hohe Übertragungsgeschwindigkeit ist sehr vorteilhaft, da dies die Wartezeit zwischen ausgeführtem Stitch und erfolgreichem Tiling stark verkürzt und sich lange Reaktionszeiten negativ auf die Zufriedenheit der Nutzer auswirken.

4.2 Implementierung der Interaktion

Die Stitch-to-Tile Interaktion lässt sich grob in drei Phasen teilen, was anhand des high-level-Aktivitätsdiagramms in Abbildung 16 erläutert wird. Zuerst muss zwischen den beteiligten Geräten eine funktionierende Wi-Fi Direct Verbindung bestehen. Diese Verbindung wird zu Beginn hergestellt (Blauer Kasten).

Nachdem die Verbindung hergestellt wurde, kann der Nutzer entweder ein Bild wählen, welches er visuell tilen möchte, oder aber das bereits auf dem Bildschirm angezeigte Bild benutzen.

Daraufhin kann der Nutzer einen Stitch durchführen, um das Tiling zu starten. Die Prototyp Applikation muss nun prüfen, ob es sich um einen korrekten Stitch gehandelt hat (Grüner Kasten).

Wurde die ausgeführte Geste als ein korrekter Stitch erkannt, so beginnt die Bildverarbeitung, die das originale Bild zerschneidet und skaliert, sodass es auf den beiden Bildschirmen angezeigt werden kann (Roter Kasten).

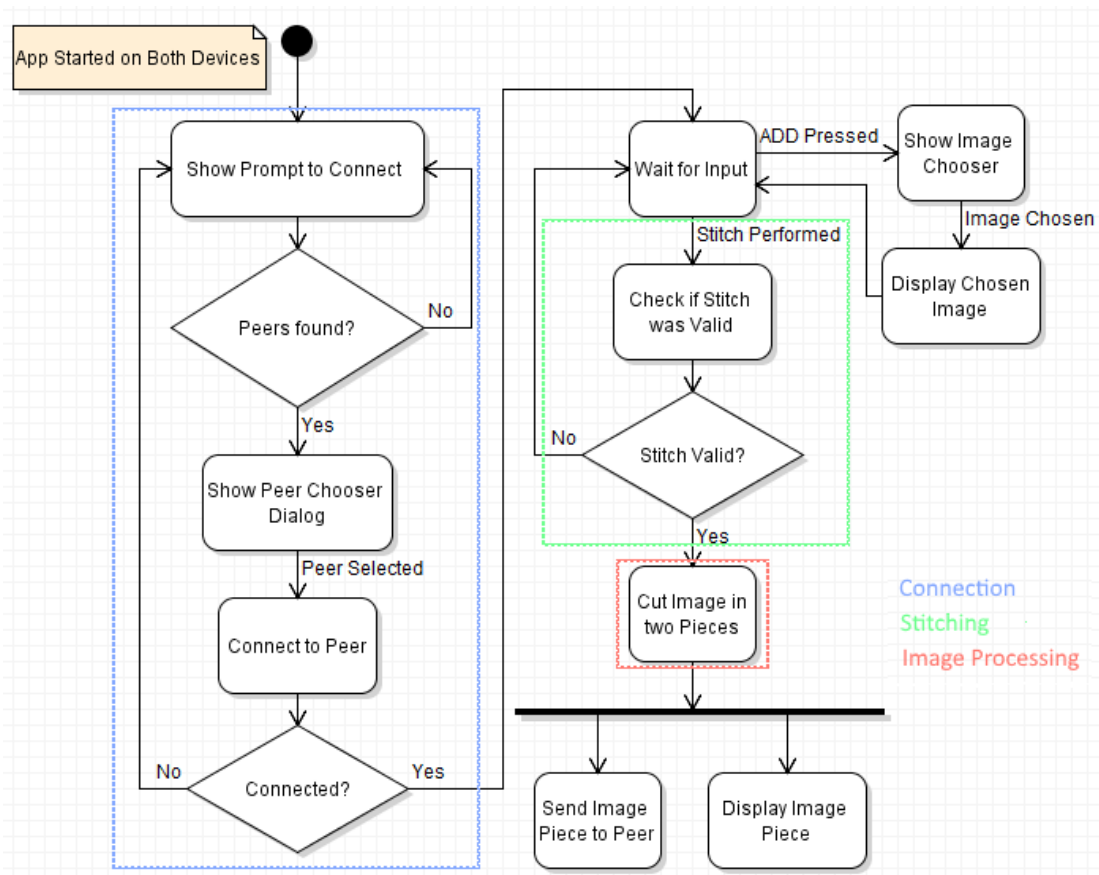


Abbildung 16: Aktivitätsdiagramm Stitch-to-Tile

4.2.1 Verbindung

Die tatsächliche Verbindung der beteiligten Geräte steht nicht in direktem Zusammenhang mit der Stitching-Geste, sondern wird im Vorherein hergestellt. Startet man die Stitch-to-Tile App auf mindestens zwei Geräten, so können sich diese miteinander verbinden. Erst mit einer funktionierenden Verbindung kann der User einen Stitch durchführen.

Diese Designentscheidung wurde aufgrund einer technischen Limitation getroffen, die während des Entwicklungsprozesses erkannt wurde. Es besteht keine Möglichkeit zwei Geräte miteinander zu koppeln, ohne dass der Nutzer über einen Popup-Dialog bestätigt, dass er diese Verbindung auch möchte. Dies macht aus dem sicherheitstechnischen Aspekt Sinn, da eine automatische Verbindung zwischen Geräten zu leicht zu missbrauchen wäre, um beispielsweise Malware zu übertragen oder Daten zu stehlen. Die User Experience hat unter dieser Einschränkung jedoch gelitten.

Der erste Prototyp der Stitch-to-Tile App hat sowohl die Verbindung, als auch das Tiling mit nur einer Nutzerinteraktion, dem Stitch, zu lösen versucht. Der Fluss des Ablaufes wurde jedoch durch den Popup für die Bestätigung der Verbindung unterbrochen, was als negativ empfunden wurde. So wurde in der aktuellen Version der Verbindungsprozess ausgelagert, sodass die Geste nun nur noch einen Zweck erfüllt: Das Auslösen der Bildschirmvergrößerung. Die Verbindung wird beim Start der Applikation auf beiden Geräten mithilfe von Dialogfenstern initialisiert. (Siehe Abbildung 17)



Abbildung 17: Verbindungsfenster

Startet der Nutzer die Stitch-to-Tile Applikation auf zwei seiner Android Smartgeräte, so öffnet sich auf beiden Bildschirmen ein Dialogfenster, welches den Nutzer darauf hinweist, dass keine Wi-Fi Direct Verbindung besteht (Siehe Abbildung 17, links). Bestätigt der Nutzer die Suche nach einer Verbindung, sucht das Smartgerät nach anderen

Geräten, die sich in seiner WLAN Reichweite zu einer Wi-Fi Direkt Verbindung anbieten. Eine Liste aller gefundenen Geräte wird in einem weiteren Dialogfenster auf dem Bildschirm angezeigt (Siehe Abbildung 17, mitte).

Der Nutzer kann nun sein Zweitgerät aus dieser Liste auswählen und wird daraufhin mit diesem verbunden. Das Gerät, auf dem die User Inputs durchgeführt werden, ist bei dieser Verbindung automatisch der Group Owner der Wi-Fi Direct Verbindung.

Abbildung 18 zeigt den Verbindungsprozess nochmals genauer in einem Sequenzdiagramm. Die Hauptaktivität, die für den User sichtbar ist, bekommt zu Anfang mitgeteilt, dass keine Verbindung besteht und öffnet das erste Fenster. Nachdem das Netzwerk nach möglichen Verbindungspartnern durchsucht wurde, öffnet sich das zweite. Nach der Auswahl des gewünschten Peers wird eine Verbindung zu diesem hergestellt und die für die Kommunikation nötigen Schritte gestartet.

Dem Nutzer wird bekanntgegeben, dass der Verbindungsprozess erfolgreich war (Siehe Abbildung 17, rechts).

Hier beginnt die eigentliche Stitch-to-Tile Funktionalität des Prototyps.

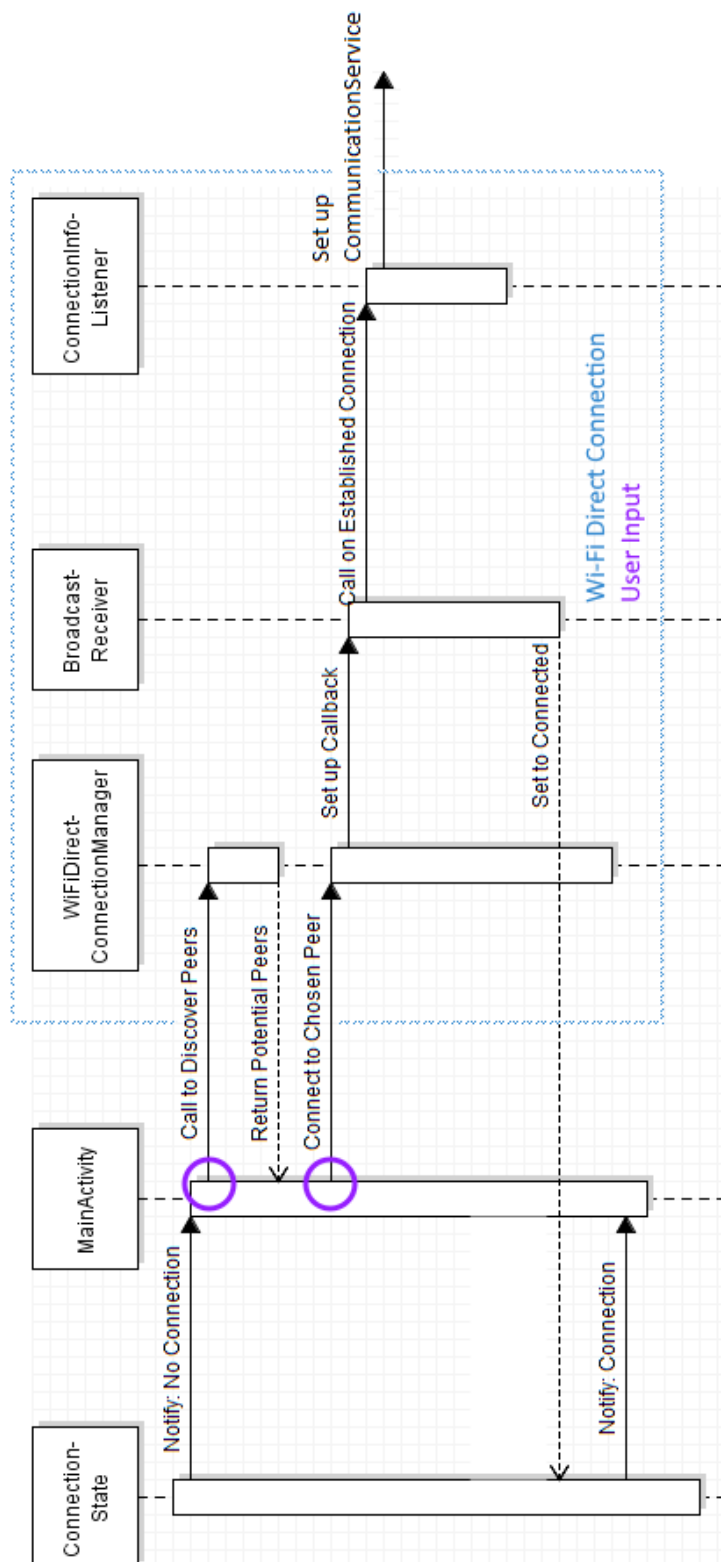


Abbildung 18: Sequenzdiagramm zur Herstellung der Verbindung

4.2.2 Erkennung eines Swipes

Wie schon in Kapitel 3.3 beschrieben, besteht ein Stitch technisch gesehen aus zwei Swipes. Erst wenn diese beiden Swipes die richtigen Bedingungen erfüllen, werden diese als ein erfolgreicher Stitch anerkannt.

In diesem Prototyp wird zwischen zwei Swipes unterschieden:

1. Der ausgehende Swipe
2. Der eingehende Swipe

Der ausgehende Swipe geht von dem Hauptgerät aus, also von dem Gerät, das das Bild anzeigt, welches man auf den Bildschirm des Zweitgeräts erweitern möchte. Er beginnt immer in der Mitte des Bildschirms und endet am Rand. Der eingehende Swipe kommt vom Bildschirmrand des Gerätes und endet an der Stelle, an der der Nutzer seinen Finger vom Display nimmt.

Der ausgehende Swipe wird erstellt, sobald der Nutzer den Floating-Action-Button von der Mitte an den Bildschirmrand führt.

Die Daten, welche nötig für die Stitcherkennung sind, werden innerhalb des Swipes gespeichert. Ein Swipe besteht aus einem Startpunkt, einem Endpunkt, der Bewegungsrichtung (Links, rechts, oben, unten) und der Information, ob es sich um einen ausgehenden oder eingehenden Swipe handelt.

Um einen Swipe konsistent erkennen zu können, musste an den Bildschirmrändern eine Kulanz von 100 Pixeln miteinbezogen werden. Ohne diese müsste der Nutzer den Finger bis zum letzten Pixel am Rand des Gerätes bewegen, was sich als unzuverlässig erwiesen hat.

Wird ein Swipe erfolgreich erstellt, informiert dieser den Kommunikationsservice. Der Service kümmert sich daraufhin um die Erkennung des Stitches.

4.2.3 Erkennung eines Stitches

Damit ein Stitch erkannt werden kann, müssen Swipes miteinander verglichen werden. Dafür wird eine Kommunikation zwischen den beiden, durch Wi-Fi Direct verbundenen, Geräten benötigt, da die Daten dieser Swipes ausgetauscht werden müssen. Diese Aufgabe übernehmen die beiden Kommunikationsservices. Diese Services werden direkt nach der Erstellung der Wi-Fi Direkt Verbindung erstellt (Siehe Abbildung 18, rechts).

Auf dem Hauptgerät heißt der Kommunikationsservice Server Service und auf dem Zweitgerät Client Service. Diese unterscheiden sich nur durch das anfängliche Aufsetzen der Sockets, der Kommunikationsgrundlage, durch die Daten ausgetauscht werden.

Ein Kommunikationsservice empfängt also sowohl die Information, dass ein Swipe erstellt wurde, als auch Nachrichten von dem jeweils anderen Gerät. Der Service besitzt eine Zustandsmaschine, an die diese Informationen weitergeleitet werden. Abhängig vom jeweiligen Zustand und der eingegangenen Daten werden die zum Stitching nötigen Funktionen ausgeführt.

Die Zustandsmaschine

Im Grundzustand befindet sich die State Machine in einem „Leerlauf“. Sie wartet auf eine Aktivierung in Form eines Swipes oder einer Nachricht. Dieser Leerlauf-Zustand nennt sich Idle-State. Abhängig von der jeweiligen Aktion begibt sich die Zustandsmaschine dann in einen von drei Modi (Siehe Abbildung 19):

1. Ausgehender Swipe (links)
2. Warten auf eingehenden Swipe (mitte)
3. Eingehender Swipe (rechts)

Diese verhalten sich unterschiedlich und ermöglichen so eine Kommunikation zwischen ihnen, sodass zuletzt ein Stitch erkannt werden und das Tiling eingeleitet werden kann. Programmiertechnisch macht die Nutzung einer State Machine für diesen Anwendungsfall Sinn, da man den gesamten Informationsaustausch und die Entscheidungen, was mit diesen Informationen angefangen werden soll, in den Maschinen abwickeln kann. Die Services wissen nicht, in welchem Zustand sich die State Machines

befinden. Das müssen sie auch nicht, denn sie geben die Informationen (Swipes oder Nachrichten) einfach weiter und die Maschinen kümmern sich um die Stitcherkennung und die Einleitung des Tilings.

Nach dem ersten Event, das den Zustand der Maschine setzt, wird fünf Sekunden gewartet und auf weitere Events gewartet. Passiert in dieser Zeit nichts, so begibt sich die Zustandsmaschine wieder in den Idle-State. Dies hat den positiven Nebeneffekt, dass man die Nachrichten nicht mit einem Timestamp versehen muss, um sicher zu gehen, dass die Swipes in kurzer zeitlicher Abfolge geschehen.

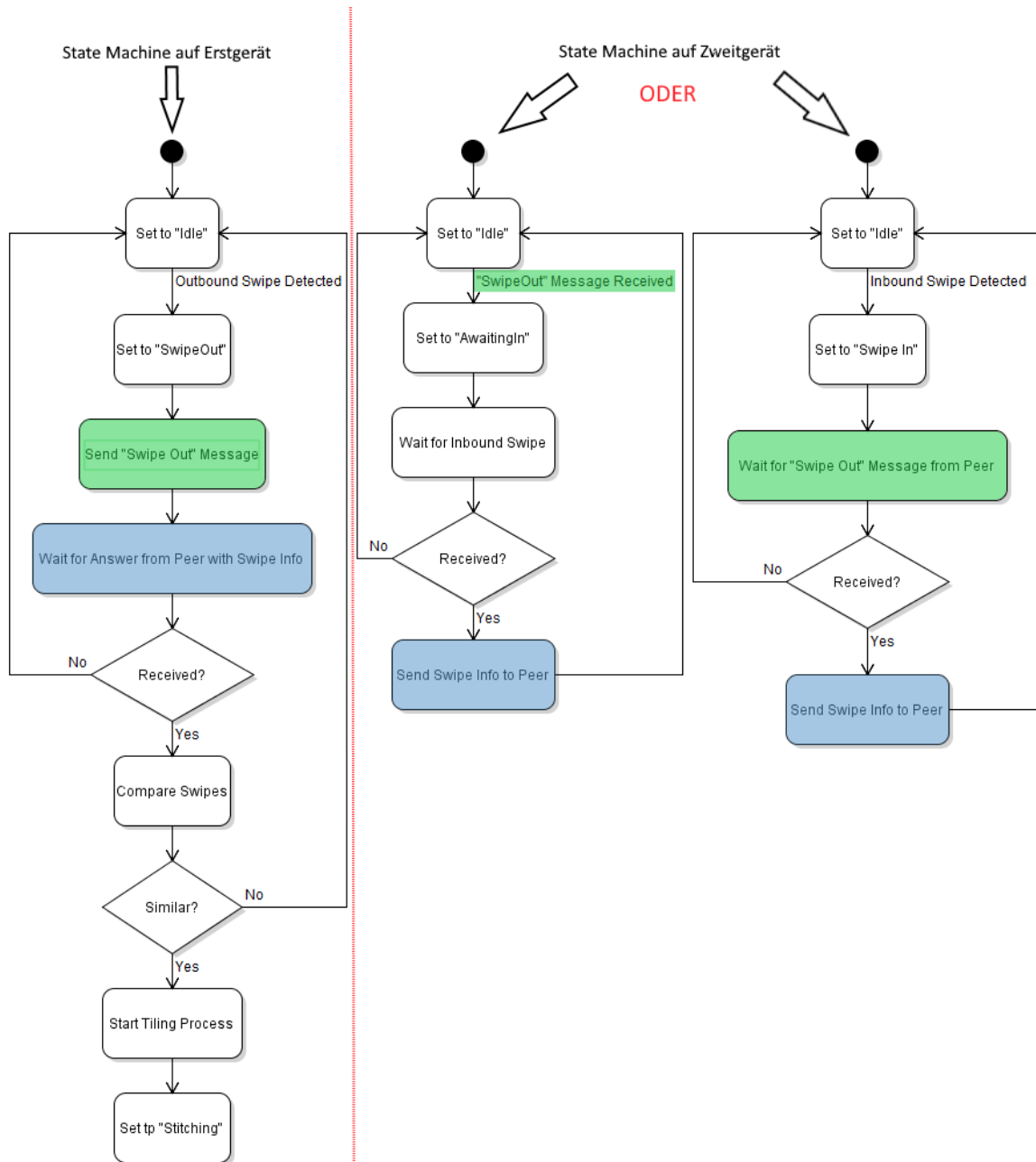


Abbildung 19: Zustandsmaschinen zur Kommunikation zwischen den Geräten

An dieser Stelle werden anhand eines typischen Beispielablaufs und mithilfe von Aktivitätsdiagrammen für jeden der Zustände (Abbildung 19) die Interaktionen zwischen den State Machines erläutert.

Anfangs befinden sich die Zustandsmaschinen auf beiden Geräten im Idle-State. Führt der Nutzer nun einen Stitch durch, bewegt also den Floating-Action-Button vom Bildschirm des Hauptgeräts auf den des Zweitgeräts, bekommt zunächst die State Machine des Hauptgeräts (SM1) einen ausgehenden Swipe gemeldet und versetzt sich in den „SwipeOut“ Zustand (Abbildung 19, links). Daraufhin sendet SM1 eine Nachricht an das Zweitgerät und dessen State Machine (SM2), um die Information über den ausgehenden Swipe weiterzugeben (Grün eingefärbt).

Typischerweise empfängt SM2 die Nachricht, bevor der eingehende Swipe auf dem Zweitgerät beendet worden ist. Dies versetzt SM2 in den „AwaitingIn“ Zustand (Abbildung 19, mitte), das heißt es wird fünf Sekunden auf einen eingehenden Swipe gewartet. Passiert dieser eingehende Swipe, so werden die Informationen über diesen (Richtung und Winkel) an das Erstgerät und somit SM1 weitergegeben (Blau eingefärbt).

Sollte eine Verzögerung der gesendeten „SwipeOut“ Nachricht entstehen, wodurch der eingehende Swipe vor dieser registriert wird, wird SM2 in den „SwipeIn“ Zustand versetzt (Abbildung 19, rechts). In diesem Zustand befindet sie sich für fünf Sekunden oder bis die Nachricht vom Hauptgerät empfangen wurde. Wird sie empfangen, sendet SM2 wie im vorherigen Fall wieder die Informationen des eingehenden Swipes (Blau eingefärbt).

SM1 überprüft dann, ob der empfangene eingehende- und der eigene ausgehende Swipe ähnlich genug sind, um als ein Stitch anerkannt zu werden. Ist dies der Fall, so wird der Tiling Prozess eingeleitet, indem das Zweitgerät die Daten über seine Bildschirmdimensionen an das Erstgerät sendet.

Die gesamte Interaktion zwischen den Geräten und den einzelnen Klassen der Applikation ab dem Geschehen eines Stitches lässt sich in den Sequenzdiagrammen in Abbildungen 20 und 21 genauer erkennen. Der gesamte Stickerkennungsteil ist mit grünen Kästen gekennzeichnet.

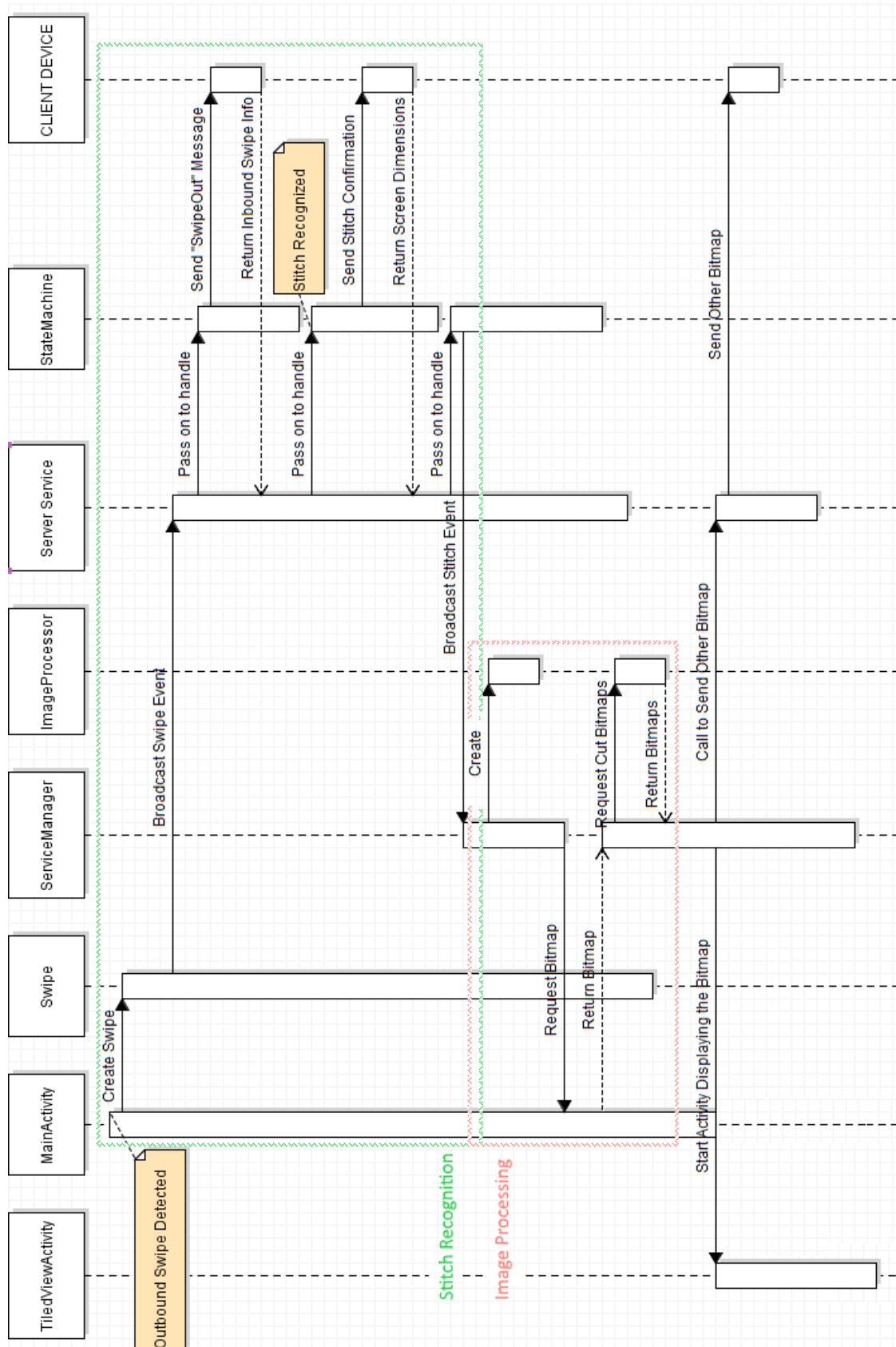


Abbildung 12: Sequenzdiagramm Serverseitig

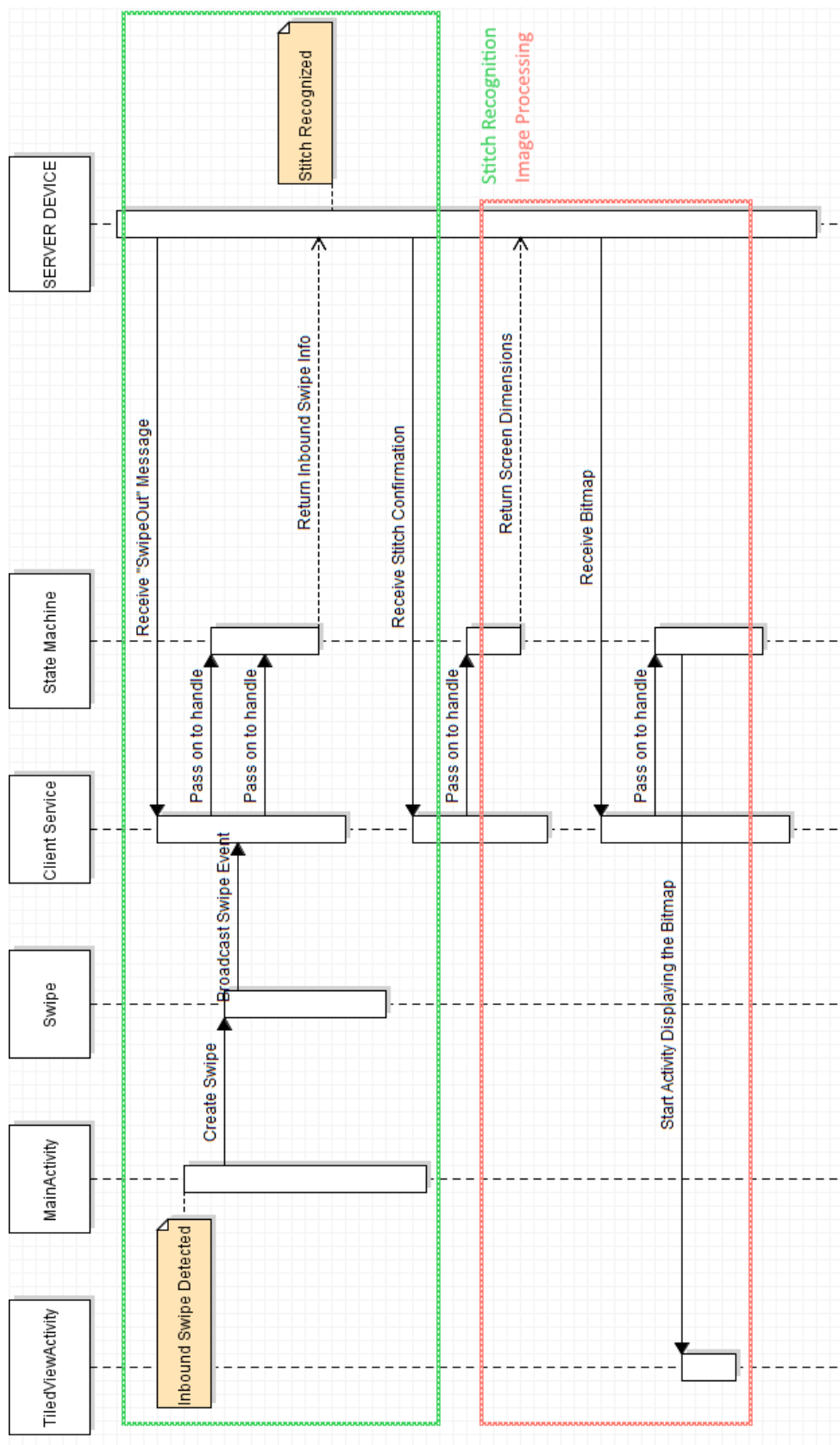


Abbildung 21: Sequenzdiagramm Clientseitig

4.2.4 Tiling

Bei der im Prototyp verwendeten Form von Tiling handelt es sich um reines Visuelles Tiling, wie es in Kapitel 3.1.1 definiert wurde. Das bedeutet, dass der gewünschte Effekt ist, die Bildschirmfläche zu vergrößern, um sich ein Bild, das sich auf einem Gerät befindet, getiled anschauen zu können. Vereinfacht ausgedrückt wird das ausgewählte Bild zerschnitten, skaliert und daraufhin jeweils eine Hälfte auf den beiden Geräten angezeigt.

Für diesen Prozess benötigt man einige Informationen über die benutzten Geräte. Diese wären zum einen die realweltlichen Größen der beiden Bildschirme und zum anderen die Pixeldichte dieser. Wurde, wie in dem vorherigen Kapitel beschrieben, ein Stitch erkannt, so sendet die State Machine auf dem Zweitgerät diese Informationen an das Erstgerät (Siehe Abbildung 21, roter Kasten). Dort werden sie dann in einem separaten Programmteil verarbeitet (Siehe Abbildung 20, ImageProcessor). An dieser Stelle wird nun genauer auf die Bildverarbeitung eingegangen und nachfolgend an einem Beispiel verdeutlicht.

Image Processing

Die Bildverarbeitung findet ausschließlich auf dem Gerät statt, von dem aus der Nutzer den Stitch gestartet hat. Das Zweitgerät stellt seinen fertig bearbeiteten Bildausschnitt nur dar.

Der Prozess läuft in zwei Schritten ab:

1. Zerschneiden des Bildes in zwei Teile
2. Separates Skalieren der Teile

Für den ersten Schritt benötigt man nur die realweltlichen Maße der beiden Displays. Zunächst wird ein Prozentsatz errechnet, der bestimmt, in welchem Verhältnis das Bild zerschnitten werden muss. Dafür teilt man die Breite des eigenen Displays durch die Summe aus den Breiten beider Displays. Der resultierende Bruch bestimmt den Anteil des originalen Bildes, welcher später auf dem Hauptgerät angezeigt wird.

Der Zweite Schritt hat sich als etwas komplizierter und fehleranfälliger herausgestellt. Würde man die Bildteile ohne weitere Skalierung auf den Geräten anzeigen, so würde dies Probleme aufwerfen. Zum einen würde die Bildschirmgröße des Hauptgerätes nicht effizient genutzt werden, da das Bild vor dem Zerschneiden die maximale Fläche genutzt hat und nun nur noch einen Bruchteil der Breite besitzt.

Ein weiteres, größeres Problem entsteht, wenn die benutzten Bildschirme der Geräte nicht die gleiche Pixeldichte aufweisen. In diesem Fall würden die Bildteile nicht gleichhoch sein, wodurch der gewünschte Zweck, nämlich die Illusion eines gemeinsamen Displays, nicht erfüllt würde.

Aus diesen Gründen müssen Berechnungen angestellt werden, um die Bildteile ordnungsgemäß skalieren zu können.

Da es sich hier um einen Prototyp handelt und das Ziel ist, ein Bild möglichst groß darzustellen, wird unabhängig von Auflösung und Seitenverhältnis des Bildes, die gesamte Breite beider Geräte benutzt. Das resultiert bei Bildern mit geringer Auflösung in einer gewissen Unschärfe, wie man sie von zu starkem Zoomen kennt. Außerdem können mit dieser Methode abhängig von den Seitenverhältnissen Bilder gestreckt oder gestaucht werden. Da hier nur die Validität von Stitch-to-Tile aufgezeigt werden soll, werden solche Probleme in Kauf genommen.

Das gesamte Bild nimmt also am Ende immer die Breite der Summe beider Displays an. Die Höhe des Bildes wird von der Höhe des kleineren der beiden Geräte bestimmt. Dies ermöglicht die bestmögliche Ausnutzung der Displays wie auch in Abbildungen 9 und 10 zu erkennen ist.

Für das kleinere der beiden Geräte ist die Aufgabe dadurch einfach, da man das ihm zugehörige Bildteil auf die gesamte Pixelanzahl des Displays skaliert. Auf dem größeren Gerät hingegen darf sich die realweltliche Höhe (z.B. in cm) des Bildteiles nicht von der auf dem kleineren Gerät unterscheiden, auch wenn dessen Display eine andere Pixeldichte besitzt. Das heißt, dass die Höhe in Pixeln für das Bild auf dem größeren Gerät durchaus eine andere sein kann.

Anhand eines realen Beispiels wird nun die gesamte Bildverarbeitung genauer erläutert:

Der Nutzer möchte ein Bild von seinem Nexus 5 Smartphone auf sein Nexus 7 Tablet Visuell Tilen. Das Nexus 5 liegt links neben dem Tablet. Das heißt, dass das Handy den linken Bildausschnitt erhalten wird.

Die realen Breiten der Displays in Zoll betragen 2,25 für das Nexus 5 und 3,75 für das Nexus 7. Das ergibt eine Gesamtbreite von 6 Zoll, die das Bild ausfüllen muss. Teilt man die 2,25 Zoll des Hauptgeräts durch die gesamten 6 Zoll, erhält man einen Prozentsatz von 37,5%.

Das originale Bild wird nun in zwei Teile geschnitten, wobei der erste vom linken Rand des Bildes 37,5% der Pixel zählt und der zweite den Rest des Bildes ergibt.

Da das Nexus 5 das kleinere der beiden Geräte ist, soll die gesamte Bildschirmfläche genutzt werden. Das bedeutet, dass der linke Bildteil auf 1080 Pixel Breite und 1920 Pixel Höhe skaliert wird, also 1080x1920.

Der rechte Bildteil, also jener welcher auf dem Nexus 7 angezeigt werden soll, muss die gleiche Höhe haben wie der linke. Also muss das Bild in der Höhe auf so viele Pixel skaliert werden, dass es den 4 Zoll des Nexus 5 entspricht.

Das Tablet hat eine Pixeldichte von 213 Pixel pro Zoll. Multipliziert man diese Zahl mit der gewünschten Höhe von 4 Zoll, so erhält man als Ergebnis 852 Pixel. Der rechte Teil des Bildes muss also auf eine Höhe von 852 Pixel und die Pixelanzahl in der Breite des Tablets skaliert werden. Insgesamt ergibt das eine Bildauflösung von 852x800.

Problem

In der Theorie müsste dieses Konzept für jede Gerätekombination einwandfrei funktionieren. Leider ist dies in der Praxis nicht der Fall. Manche Geräte geben nämlich ungenaue oder gar direkt falsche Werte zurück, wenn man Bildschirmdimensionen oder Pixeldichten anfordert. Dies hat dann die Auswirkung, dass die beiden Bildteile in so einem Fall nicht die gewünschten Dimensionen einnehmen und dadurch eines größer als das andere oder anderweitig verzerrt angezeigt wird.

Ein Beispiel für ein Gerät, welches falsche Daten liefert, ist das Samsung Galaxy Note 2. Um den Stitch-to-Tile Prototyp zu testen, wurde versucht, von einem Nexus 5 auf ein Note 2 zu tilen. Das Bild auf dem Note 2 wurde in der Breite gestreckt. Nach Überprüfung der Zahlen ist aufgefallen, dass es als Bildschirmbreite 2,25 Zoll angegeben hatte, also genau wie das Nexus Gerät. Durch Nachmessen erhält man jedoch eine Bildschirmbreite von 7,1 cm, was 2,75 Inches entspricht. So wurde das Bild genau in der Mitte zerschnitten, dann aber auf dem Samsung Gerät um 0,5 Inches gestreckt.

Will man immer die gesamte verfügbare Bildschirmfläche nutzen, lässt sich dieses Problem nicht rein programmiertechnisch lösen, bis alle Geräte richtige Daten angeben.

5 Ausblick

Die Stitch-to-Tile Interaktion ist ein Ansatz, um das Spontaneous Device Sharing Problem zu lösen. Die Konzeption in Kapitel 3 und der Prototyp aus Kapitel 4 lassen erkennen, dass diese Geste eine vielversprechende ist. Dadurch, dass die eigentliche Geste und das Tiling in einem Aktion-Reaktion Verhältnis stehen, ist von Anfang an eine Modularität gegeben. Das bedeutet, dass man von Anwendungsfall zu Anwendungsfall die Tiling Art auswechseln kann und Stitch-to-Tile so relativ einfach sowohl in neue, als auch in existierende Applikationen verbauen kann.

Möchte man als App Entwickler eine beliebige Art von Tiling verwenden, muss man nur den Floating Action Button zuschalten und Stitching wird ermöglicht. Dies verhindert Kollisionen mit anderen Funktionen der Applikation.

Jedoch gibt es noch einige Probleme, von denen zwei besonders auffällig sind. Zum einen ist eine Möglichkeit die Geräte miteinander zu verbinden, ohne dies in einer Dialogbox bestätigen zu müssen, ein Muss, sodass das Stitching wirklich den Eindruck vermitteln kann, man hätte statt zwei Geräten nur eines mit zwei Bildschirmen vor sich.

Zum anderen besteht das Problem mit den falschen Bildschirmdimensionen und Pixeldichten, die von manchen Geräten geliefert werden. Eine mögliche, jedoch aufwendige, Lösung wäre eine Datenbank von Geräten zu erstellen, die falsche Informationen zurückgeben. Würde ein solches Gerät für Stitch-to-Tile verwendet werden, könnte man die echten Daten aus dieser Datenbank auslesen und so immer ein korrektes Visuelles Tiling durchführen.

Tabellenverzeichnis

Tabelle 1: Gegenüberstellung der Gesten.....	27
--	----

Abbildungsverzeichnis

Abbildung 1: Bump Geste: Aus [HIN03]	4
Abbildung 2: Bump Erkennung: Vgl. [HIN03]	4
Abbildung 3: Pen-Stitching: Aus [HIN04]	5
Abbildung 4: Dynamic Display Tiling: Aus [LI12]	7
Abbildung 5: iPodWall: Aus [WLS]	8
Abbildung 6: Entfernungen zwischen Bildschirmen und Personen	10
Abbildung 7: Panorama Foto auf Tablet	11
Abbildung 8: Panorama Foto von Tablet auf Smartphone erweitert	11
Abbildung 9: Display Tiling von Tablet auf Smartphone	12
Abbildung 10: Display Tiling von Smartphone auf Smartphone	13
Abbildung 11: Display Tiling und Umgang mit Bildschirmrändern.....	14
Abbildung 12: FAB zu Beginn der Interaktion	20
Abbildung 13: Ablauf der Interaktion	21
Abbildung 14: Stitch-Erkennung.....	22
Abbildung 15: Stitchrichtung bei mehr als 2 Geräten	24
Abbildung 16: Aktivitätsdiagramm Stitch-to-Tile.....	31
Abbildung 17: Verbindungsfenster.....	32
Abbildung 18: Sequenzdiagramm zur Herstellung der Verbindung	34
Abbildung 19: Zustandsmaschinen zur Kommunikation zwischen den Geräten.....	37
Abbildung 20: Sequenzdiagramm Serverseitig	39
Abbildung 21: Sequenzdiagramm Clientseitig.....	40

Literaturverzeichnis

[GOO12] Google Inc.: The New Multi-screen World: Understanding Cross-platform Consumer Behaviour. [Online] <https://www.thinkwithgoogle.com/research-studies/the-new-multi-screen-world-study.html> (Zuletzt besucht: 22.04.2015)

[GRU01] GRUDIN, Jonathan: Partitioning Digital Worlds: Focal and Peripheral Awareness in Multi Monitor Use. Proc. CHI2001, March 31-April 4, 2001, Seattle, WA, USA.

[HIN03] HINKLEY, Ken: Distributed and Local Sensing Techniques for Face-to-Face Collaboration. Proc. ICMI'03, November 5-7, 2003, Vancouver, British Columbia, Canada.

[HIN04] HINKLEY, Ken; RAMOS, Gonzalo; GUIMBRETIERE, Francois; BAUDISCH, Patrick; SMITH, Marc: Stitching: Pen Gestures that Span Multiple Displays. Proc. AVI 2004, May 25-28, 2004, Gallipoli (LE), Italy.

[HUT04] HUTCHINGS, Dugald R.; SMITH, Greg; MEYERS, Brian; CHERWINSKI, Mary; ROBERTSON, George: Display Space Usage and Window Management Operation Comparisons between Single Monitor and Multiple Monitor Users. Proc. AVI 2004, May 25-28, 2004, Gallipoli (LE), Italy.

[JOK15] JOKELA, Tero; CHONG, Ming; LUCERO, André; GELLERSEN, Hans: Connecting devices for collaborative interactions. In: Interactions, Ausgabe XXII.4 JULY - AUGUST 2015, S.39-43, ACM

[LEE11] LEE-DELISLE, Seb: PixelPhones. [Online] <http://seb.ly/2011/09/pixelphones-a-huge-display-made-with-smart-phones/> (Zuletzt besucht: 24.04.2015)

[LI12] LI, Ming; KOBELT, Leif: Dynamic Tiling Display: Building an Interactive Display. Proc. MUM '12 December 04 - 06 2012, Ulm, Germany.

[SBG12] Sleeping Beast Games: Spaceteam. [Online] <http://www.sleepingbeast-games.com/spaceteam/> (Zuletzt besucht: 18.09.2015)

[TRU08] TRUEMPER, Jacob M.; SHENG, Hong; HILGERS, Michael G.; HALL, Richard H.; KALLINY, Morris; TANDON, Basanta: Usability in Multiple Monitor Displays. In: ACM SIGMIS Database 39, 74–89 (2008)

[WIK15a] Wikipedia: Second Screen [Online] https://en.wikipedia.org/wiki/Second_screen (Zuletzt besucht: 18.09.2015)

[WIK15b] Wikipedia: Proxemics [Online] <https://en.wikipedia.org/wiki/Proxemics> (Zuletzt besucht: 20.09.2015)

[WLS] Welikesmall, Inc.: iPodWall [Online] <http://www.welikesmall.com/the-ipod-wall> (Zuletzt besucht: 27.04.2015)