

Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart  
Universitätsstraße 38  
D-70569 Stuttgart

Bachelorarbeit

# **Visuelle Unterstützung bei der Planung der Bildentwicklung in Lateinformationschoreografien**

Fabian Schmierer

<b>Studiengang:</b>	Informatik
<b>Prüfer/in:</b>	Dr. Steffen Koch
<b>Betreuer/in:</b>	Samuel Beck, M.Sc., Nina Dörr, M.Sc.
<b>Beginn am:</b>	5. Oktober 2022
<b>Beendet am:</b>	5. April 2023



## Kurzfassung

Das Gebiet der Visualisierung von Teamsportdaten wächst in den letzten Jahren stetig. Grund dafür sind die steigende Popularität und Kommerzialisierung von Sportarten wie Fußball. Für Randsportarten existieren indes jedoch kaum digitale Lösungen. Eine dieser Sportarten ist der Lateinformationstanz. In dieser Arbeit soll deswegen ein digitales Werkzeug entwickelt werden, mit dem Choreografien für Lateinformationen erstellt, analysiert und mit den Tänzern kommuniziert werden können. Der Fokus liegt hierbei insbesondere auf der Erstellung von Bildern einer Choreografie und der Bildentwicklung. Dieser Prozess wird bisher nur von einfachen Programmen unterstützt, mit welchen man die Positionen der Tänzer in einem Koordinatensystem einzeichnet und dies dann in einer 2D- oder 3D-Ansicht betrachtet. In Zusammenarbeit mit drei Formationstrainern wurden deshalb Anforderungen erhoben, die ein Programm zur Bildentwicklung erfüllen sollte. Anschließend wurden die Anforderungen in einem webbasierten Prototypen implementiert. Der Prototyp wurde daraufhin in einer qualitativen Studie mit vier Domänenexperten evaluiert. Hierbei stellte sich heraus, dass Choreografien mit dem Prototyp ohne großen Aufwand editiert und analysiert werden können. Als positive Merkmale wurden insbesondere die gewählten visuellen Kodierungen und die Möglichkeiten zur Analyse der Choreografie von den Teilnehmern hervorgehoben. Die Spezifikation von Bildübergängen, das visuelle Erkennen von Kollisionen und die Erstellung von Posen im 3D-Modus wurden ebenfalls gelobt. Jedoch brauchen diese Funktionen Überarbeitungen im Bereich des Interaktionsdesigns. Eine Zeitleiste, bestehend aus Takten mit jeweils acht Schlägen, in der die zeitliche Verordnungsung der Bilder eingetragen ist, wurde als zu vereinfacht bezeichnet. Basierend auf dem Feedback wurden abschließend neun Designrichtlinien herausgearbeitet, welche für digitale Werkzeuge zur Bildentwicklung angewandt werden können.



## Hinweis

In dieser Arbeit wird bei der Nennung von Personen (z.B. Nutzer, Teilnehmer, Formationstrainer, etc.) immer das generische Maskulinum verwendet und auf eine Mehrfachnennung von Geschlechtern, zum Beispiel mit einem Doppelpunkt, zur besseren Lesbarkeit verzichtet. Dies soll jedoch ausdrücklich niemanden ausschließen. Das heißt immer dann, wenn das generische Maskulinum für Personen verwendet wird, sind ausdrücklich alle Geschlechter miteinbezogen. Da der Tanzsport jedoch klassischerweise auf Geschlechterrollen basiert, insbesondere indem Paare aus einer Dame und einem Herren jeweils zusammen tanzen, werden im Folgenden immer die Begriffe „Dame“ und „Herr“ bzw. „Frau“ und „Mann“ verwendet, wenn nur ein bestimmtes Geschlecht gemeint ist. Zudem wurden die Geschlechter der Teilnehmer der Evaluation nochmals explizit genannt.

Zahlen bis zwölf wurden zur besseren Lesbarkeit immer dann nicht ausgeschrieben, wenn sie negativ waren, einen Nachkommaanteil hatten, von Studienteilnehmern so angegeben waren oder es sich bei ihnen um IDs handelte (z.B. für Bilder oder Tänzer).



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>17</b>
<b>2</b>	<b>Grundlagen</b>	<b>21</b>
2.1	Lateinformationstanz . . . . .	21
2.2	Technische und Algorithmische Grundlagen . . . . .	25
<b>3</b>	<b>Verwandte Arbeiten</b>	<b>35</b>
3.1	Teamsportvisualisierung . . . . .	35
3.2	Visualisierung für (Formations-)Tanz . . . . .	36
3.3	Positionierung dieser Arbeit . . . . .	39
<b>4</b>	<b>Methodik</b>	<b>41</b>
4.1	Designstudie: Vorgehensmodell . . . . .	41
4.2	Durchgeführte Designstudie . . . . .	44
<b>5</b>	<b>Implementierung</b>	<b>49</b>
5.1	Verwendete Technologien . . . . .	49
5.2	Prototyp . . . . .	51
5.3	Ansicht auf mobilen Geräten . . . . .	69
<b>6</b>	<b>Evaluation</b>	<b>73</b>
6.1	Zielsetzung . . . . .	73
6.2	Durchführung . . . . .	74
6.3	Resultate . . . . .	76
6.4	Diskussion . . . . .	81
6.5	Einschränkungen der Validität . . . . .	85
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>87</b>
	<b>Literaturverzeichnis</b>	<b>89</b>





# Abbildungsverzeichnis

1.1	Spielerprofil von Serge Gnabry auf Sofascore [Sof23] . . . . .	18
2.1	Beispielhaftes Bild aus der Choreografie Crazy Fire . . . . .	23
2.2	Beispiel für eine konvexe und eine nicht-konvexe Hülle . . . . .	26
2.3	Beispiel für das Ergebnis vom Graham Scan . . . . .	26
2.4	Beispiel zur Errechnung des Kurvenverhaltens zwischen zwei Linien . . . . .	28
2.5	Bedarf an öffentlichen Verkehrsmitteln in Guangzhou . . . . .	29
2.6	Schaubild eines Gauss-Kernels . . . . .	30
2.7	<i>Splat</i> in Form eines Gauss-Kernels . . . . .	31
2.8	Beispielhafte Diagonale, die vom Prototyp erkannt wird . . . . .	32
3.1	2D-Ansicht der Choreografie in ChoreoGraphics . . . . .	37
3.2	Visualisierung von Bildern in DanceStudio . . . . .	38
4.1	Vorgehensmodell nach Sedlmair et al. [SMM12] . . . . .	42
5.1	Übersicht aller Interfacebestandteile . . . . .	52
5.2	Modaler Dialog zur Erstellung einer neuen Choreografie . . . . .	53
5.3	Beispiel für Paare, die entweder getrennt oder zusammen tanzen . . . . .	54
5.4	Beispiel für Drag-and-Drop . . . . .	55
5.5	Beispiel für die Selektion mit dem Brush . . . . .	55
5.6	Beispielhafte Nutzung des Skalierwerkzeugs . . . . .	56
5.7	Beispielhafte Nutzung des Kontextmenüs . . . . .	56
5.8	Beispiel für die Darstellung von Nachbarn ausgewählter Tänzer . . . . .	57
5.9	Modaler Dialog zur Erstellung eines neuen Bildes . . . . .	58
5.10	Beispiel für die Rotation von Reihen von Tänzern . . . . .	58
5.11	Balkendiagramm zur Darstellung von Laufdistanzen zwischen Bilder . . . . .	59
5.12	Ausschnitt aus der Zeitleiste . . . . .	60
5.13	Beispielbild im Orientierungsmodus . . . . .	61
5.14	Modaler Dialog zur Änderung der Blickrichtung und der Ausrichtung des Körpers	62
5.15	Beispielhafter Bildübergang . . . . .	63
5.16	Prozess zur Spezifikation des Bildübergangs . . . . .	64
5.17	Prozess zum Markieren einer prägnanten, geometrischen Form . . . . .	65
5.18	Diskrete und kontinuierliche Heatmap . . . . .	66
5.19	3D-Ansicht auf ein Bild . . . . .	67
5.20	Poseneditor . . . . .	68
5.21	Mobile Versionen des Prototyps . . . . .	70
5.22	Landschaftsansicht auf einem iPad . . . . .	71
6.1	Modell für Design und Validierung von Visualisierungen . . . . .	74

6.2 Bewertungen gegebener Aussagen nach der Nutzung des Prototyps . . . . . 82

# Tabellenverzeichnis

4.1	Anforderungen an den Prototyp . . . . .	46
6.1	Vorerfahrung der Studienteilnehmer . . . . .	75



## Verzeichnis der Listings

2.1	CSS Media Query . . . . .	34
5.1	React Komponente für eine Liste . . . . .	50
5.2	Responsives Gitter . . . . .	51



# Verzeichnis der Algorithmen

2.1	Graham Scan . . . . .	27
2.2	Erkennung von Diagonalen . . . . .	32





# 1 Einleitung

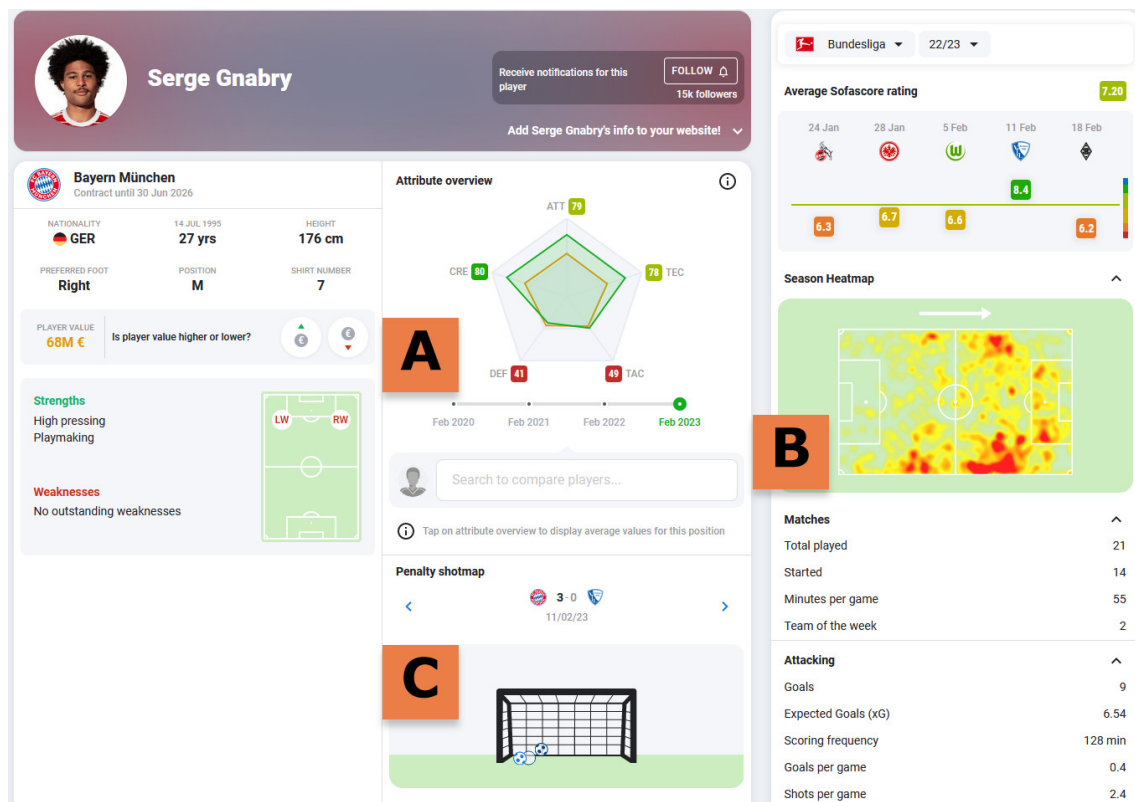
Das Gebiet der Teamsportvisualisierung wächst in den letzten Jahren stetig. Ein großer Treiber dafür sind die immer größere werdende mediale Verbreitung und Professionalisierung in Teamsportarten wie Fußball oder Basketball. Beim Betrachten eines Spiels der UEFA Champions League werden dem Zuschauer die Aufstellungen der Mannschaften in einem Overlay angezeigt und die aktuellen Leistungsdaten der Topstars werden in einer 3D-Ansicht auf das Spielfeld projiziert. Neben der Präsentation reiner Spieldaten für die Zuschauer wachsen auch die Möglichkeiten zur strategischen Analyse, die Visualisierungen verwenden. Ein Beispiel dafür ist die Webseite Sofascore<sup>1</sup>, die sportbezogene Daten aggregiert und visualisiert. Als Veranschaulichung betrachten wir das Sofascore-Spielerprofil von Serge Gnabry, welches in Abbildung 1.1 zu sehen ist. In einem Radardiagramm (Komponente A) sind für verschiedene positionsbezogene Attribute wie Technik, Verteidigung, Angriff, etc. Leistungswerte als grüne Linien eingetragen. In einer Heatmap (Komponente B) kann man die Positionen des Spielers auf dem Spielfeld ablesen und in einem Piktogramm (Komponente C), wohin er Elfmeter schießt. Dies lässt sich auch für andere Sportarten wie Basketball, Tennis, Hockey usw. durchführen.

In der Forschung werden ebenfalls Visualisierungen Bereich des Teamsports erstellt und evaluiert. Sportarten mit einem geringerem öffentlichen Interesse sind jedoch bisher kaum Gegenstand der Sportvisualisierung, so auch der Lateinformationstanz. Zwar ist Tanzsport an sich einer breiten Öffentlichkeit bekannt, jedoch hauptsächlich durch den Paartanz. So existieren alleine in Deutschland laut dem Allgemeinen Deutschen Tanzlehrerverband e.V. [ADT23] circa 850 Tanzschulen, welche die Kurse des Welttanzprogramms lehren, das aus einer Reihe ausgewählter Paartänze besteht. Dem Lateinformationstanz hingegen wurde in der Visualisierungsforschung bisher noch kaum Beachtung geschenkt. Es handelt sich dabei um eine Unterart des Formationstanzes, welche hauptsächlich in Deutschland und England in der Mitte des 20. Jahrhunderts entstanden ist. Beim Lateinformationstanz tanzt eine Formation bestehend aus sechs bis acht Tänzern eine vorgefertigte Choreografie. Jede Choreografie besteht aus fünf lateinamerikanischen Tänzen und deren Aufführung wird auf Wettbewerben von einer Reihe von Wertungsrichtern beurteilt. Besondere Relevanz haben hierbei die sogenannten *Bilder*. Eine Choreografie besteht immer aus einer Menge dieser Bilder, welche die Positionen aller Tänzer zu einem bestimmten Zeitpunkt während der Musik angeben. Auf die Einhaltung der Bilder und deren Erkennbarkeit wird ein besonderes Augenmerk gelegt. Die Bilder und die getanzen Übergänge zwischen ihnen bezeichnet man als *Bildentwicklung*. Da die Bildentwicklung mit den zu tanzenden Figuren abgestimmt werden muss, ist sie ein komplexer und oft iterativer Prozess, bei dem sich Choreografien über die Zeit hin noch verändern können. Neben der Planung der Bilder und der Bildentwicklung stellen auch die Kommunikation der Bildentwicklung im Training und das Austrainieren der Choreografie an sich weitere Schwierigkeiten da. Es ist somit ein qualitativ hochwertiger Prozess notwendig, der sich vom Planen der Bilder, über die Kommunikation an die Tänzer, bis hin zum Training der Choreografie

---

<sup>1</sup> URL: <https://www.sofascore.com/>

# 1 Einleitung



**Abbildung 1.1:** Spielerprofil von Serge Gnabry auf Sofascore [Sof23]. In Komponente A lassen sich Bewertungen für verschiedene Attribute wie Technik oder Defensive ablesen und mit anderen Spielern vergleichen. In Komponente B befindet sich eine Heatmap, welche zeigt, wo sich der Spieler während der Saison auf dem Spielfeld aufgehalten hat. In Komponente C kann abgelesen werden, an welche Stellen der Spieler Elfmeter schießt und welche davon zu einem Tor führten.

und deren iterativer Verfeinerung streckt. Bisher läuft dieser Prozess so ab, dass die Bilder auf Papier geplant werden, indem die Positionen der Tänzer (z.B. durch Kreise) in ein Koordinatensystem eingetragen werden. Es wird nun jedes Bild in ein eigenes Koordinatensystem gezeichnet und in Papierform oder als PDF-Datei an die Tänzer weitergegeben. Damit sind nachträgliche Änderungen an der Choreografie nur mit erheblichem Aufwand möglich. Auch fehlen Möglichkeiten zur Analyse der gebauten Choreografie. Es existieren zwar einige digitale Werkzeuge wie ChoreoMaker<sup>2</sup> oder Stagekeep<sup>3</sup>, jedoch bilden auch diese nur den analogen Papierprozess ab. Somit ist nur die digitale Planung der Bilder möglich. Es fehlen Möglichkeiten Aspekte wie die Laufwege- und distanzen der Tänzer, die Ausnutzung der Tanzfläche, etc. zu spezifizieren und zu analysieren.

<sup>2</sup> URL: <https://choreomaker.com/>

<sup>3</sup> URL: <https://beta.stagekeep.com/>

---

Gesucht sind somit Visualisierungen, mit denen man die Bilder und die Bildentwicklung planen, analysieren und mit den Tänzern kommunizieren kann. Im Rahmen dieser Arbeit wurde deshalb in Zusammenarbeit mit Formationstrainern ein webbasierter Forschungsprototyp entwickelt, der die genannten Probleme zu lösen versucht.

Die weitere Arbeit ist dabei wie folgt gegliedert: In Kapitel 2 werden zuerst die Grundlagen eingeführt, die für das Verständnis der Arbeit nötig sind. Dazu gehört zum einen eine ausführliche Beschreibung der Zieldomäne, als auch Erklärungen über Techniken und Algorithmen, die bei der Implementierung des Prototyps zum Einsatz kamen. Danach wird in Kapitel 3 ein Überblick über die bisher existierende Forschung in den Bereichen Team sportvisualisierung und Visualisierung für Formationstanz gegeben. Es folgt dann in Kapitel 4 die Erklärung der Methodik, die während der gesamten Arbeit verfolgt wurde. Anschließend erfolgt eine Vorstellung des entstandenen Prototyps und aller dahinter liegenden Konzepte in Kapitel 5. Die Ergebnisse einer Evaluation des Prototyps werden darauf folgend in Kapitel 6 vorgestellt. Als letztes werden dann die Kernpunkte der Arbeit nochmals in Kapitel 7 zusammengefasst und ein Ausblick auf zukünftige Forschungsmöglichkeiten gegeben.



## 2 Grundlagen

In diesem Kapitel werden einige Grundlagen eingeführt, die im Verlauf der Arbeit relevant sind. Dabei wird in Abschnitt 2.1 die Anwendungsdomäne erklärt, hier die Sportart Lateinformationstanz. In Abschnitt 2.2 werden dann Algorithmen und Techniken erklärt, die im Rahmen der Implementierung des Forschungsprototyps zum Einsatz kamen.

### 2.1 Lateinformationstanz

Unter Formationstanz versteht man das koordinierte Tanzen von mehr als zwei Personen, wodurch er sich vom Solotanz und vom Paartanz unterscheidet. Der Formationstanz ist dabei in vielen Kulturen mit einer langen Historie verbunden. Eine deutsche Ausprägung zum Beispiel waren folkloristische Kreistänze. Der Formationssport hingegen ist nicht so alt. Am Anfang des 20. Jahrhunderts wurde der Tanzsport noch vom Paartanz, hauptsächlich durch die Popularität des Wiener Walzers, dominiert. In der Mitte des 20. Jahrhunderts entwickelte sich dann vorrangig in Deutschland und England der Formationssport wie er auch noch heute betrieben wird. Am bekanntesten sind dabei der Standardformationstanz und der Lateinformationstanz, welche beide Teil des ersten deutschen Formationstanzsportturnieres im Jahr 1962 waren. Die Informationen zur Historie des Formationstanzes wurden dem Programmheft zur Deutschen Meisterschaft der Formationen 1991 [Han23] entnommen, in welchem weitere Details nachgeschlagen werden können. Im Folgenden soll nun auf die Charakteristika des Lateinformationstanzes im Detail eingegangen werden.

#### 2.1.1 Rahmeninformationen: Lateinformationssport

Die Regeln für Formationswettbewerbe im Bereich Latein legt in Deutschland der Deutsche Tanzsportverband e.V. in der Turnier- und Sportordnung [Deu23a] fest. Die beschriebene Satzung stand zum Zeitpunkt der Arbeit fest und kann sich ändern. Einige Details, welche insbesondere die nachher erwähnten Bilder betreffen, stammen aus einem Interview mit Domänenexperten (siehe Abschnitt 4.2.1).

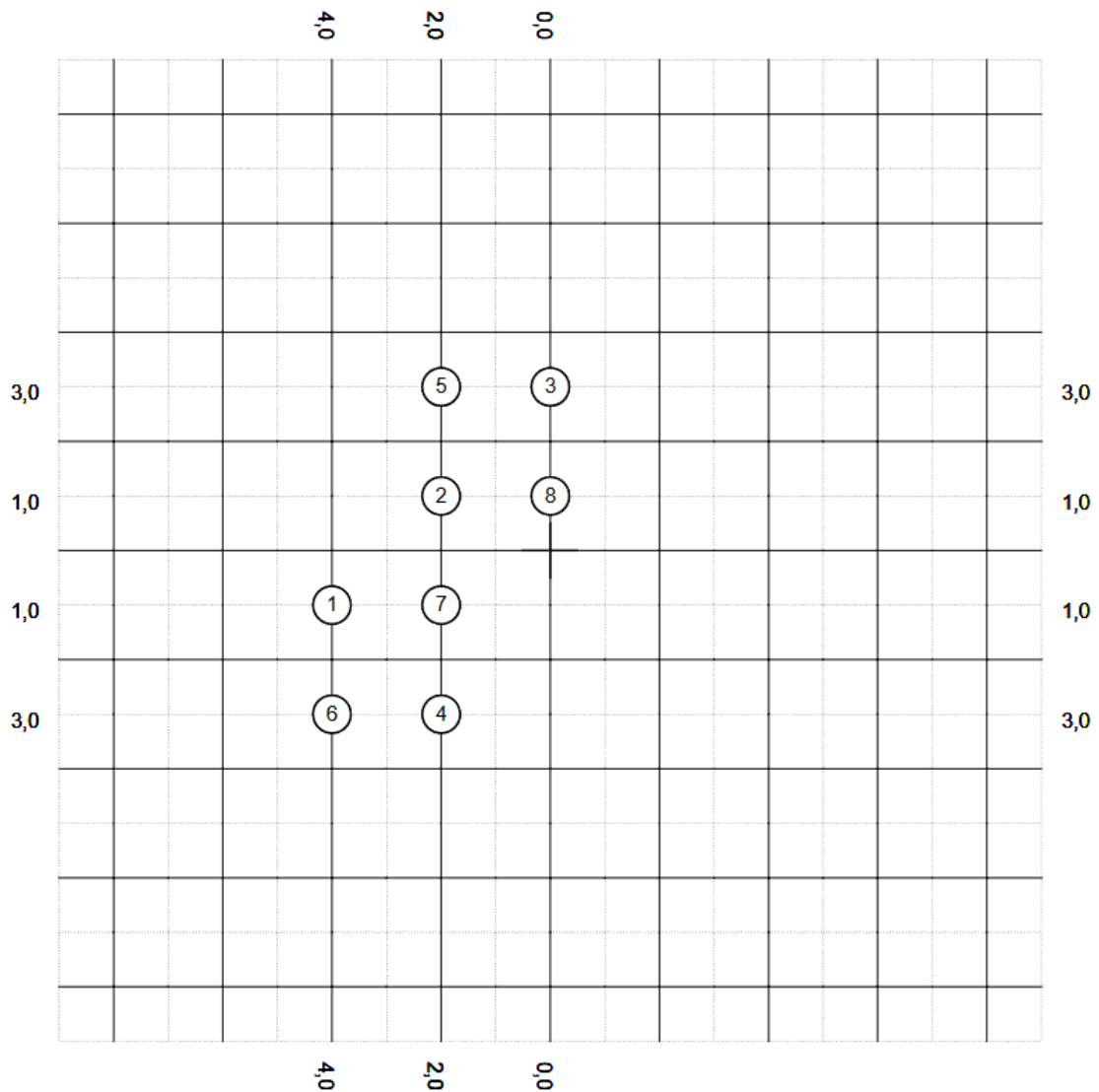
Eine Lateinformation besteht aus sechs bis acht Paaren, die eine bestimmte vorgegebene Choreografie tanzen. Die Choreografie besteht aus den fünf lateinamerikanischen Tänzen Samba, Cha-Cha-Cha, Rumba, Paso Doble und Jive. Jeder Tanz muss dabei in der Choreografie mindestens einmal enthalten sein. Eine bestimmte Ordnung ist nicht vorgegeben. Ein Formationsdurchgang und die zugehörige Musik bestehen aus einem Einmarsch, einem Hauptteil und einem Ausmarsch. Die Dauer des kompletten Durchgangs darf eine Zeit von sechs Minuten nicht überschreiten. Gewertet wird nur der Hauptteil, der zwischen drei und viereinhalb Minuten lang sein muss. Die Teile werden jeweils durch ein akustisches Signal getrennt.

Getanzt wird auf einer Tanzfläche, deren Merkmale von der Liga des jeweiligen Wettbewerbs festgelegt werden. Zum Beispiel muss die Tanzfläche in der 1. Bundesliga mindestens  $240\text{ m}^2$  groß sein und die Seiten dürfen eine Länge von 14 Metern nicht unterschreiten. In allen anderen Ligen (2. Bundesliga Nord/West/Süd, Regionalligen, Landesligen) muss die Tanzfläche mindestens  $180\text{ m}^2$  groß sein und die Seiten dürfen eine Länge von zwölf Metern nicht unterschreiten.

Die Musik, zu welcher getanzt wird, kann von jeder Formation frei gewählt werden, solange sie zur Charakteristik des Tanzes passt. Dazu zählt vor allem der Takt des jeweiligen Tanzes. Cha-Cha-Cha, Jive und Rumba werden in einem 4/4-Takt und Samba sowie Paso Doble in einem 2/4-Takt getanzt. Des Weiteren gibt es bei gleichem Takt Unterschiede in den Betonungen. Zwar werden Jive und Cha-Cha-Cha beide in einen 4/4-Takt getanzt, jedoch liegt die Betonungen beim Jive auf dem zweiten und dem vierten Schlag, während es beim Cha-Cha-Cha nur eine Betonung auf dem ersten Schlag im Takt gibt. Das Tempo der Musik kann ebenfalls passend zum aktuellen Tanz geändert werden. So ist der Jive beispielsweise ein vergleichsweise schneller Tanz während die Rumba ein langsames Tempo voraussetzt. Die Erstellung der Musik wird in den meisten Fällen von Vereinen der Bundesliga in Auftrag gegeben. Zur Refinanzierung wird die Musik von den Vereinen der Bundesliga oft an Vereine aus niedrigeren Ligen weiter lizenziert, so auch die Musik zur Choreografie, welche in dieser Arbeit als Testdatensatz verwendet wurde. Im Gegensatz zu Standardformationen gibt es für die Tänzer der Lateinformationen keine festen Bekleidungsregeln. Meist tragen die Herren schwarze Kleidung und die Damen Kleider. Die einzige Einschränkung ist die, dass die Kleidung innerhalb einer Formation einheitlich sein muss.

Neben den Bewegungsabläufen der Tänzer, den sogenannten Figuren, bestehen die Choreografien im Lateinformationssport aus einer Reihe von Bildern. Ein Bild bestimmt dabei die Position jedes Tänzers auf der Tanzfläche zu einem bestimmten Zeitpunkt während der Musik, meist während eines bestimmten Schlags im Takt. Die Anordnung der Tänzer innerhalb eines Bildes ergibt oft geometrische Formen, zum Beispiel Rechtecke, Rauten, Diagonalen und horizontale bzw. vertikale Reihen. Ein Beispielbild kann man in Abbildung 2.1 sehen. Jedoch können auch komplexere Muster auftreten. Darunter zählen zum Beispiel die Darstellung von Tieren wie einer Schwalbe sowie Pfeile mit beliebig vielen Spitzen. Der Herr und die Dame innerhalb eines Paares müssen sich in einem Bild nicht an der gleichen Position auf der Tanzfläche befinden und können unabhängig voneinander tanzen. Oft gilt dies für alle Paare gleichzeitig, das heißt entweder tanzen alle Paare getrennt voneinander oder zusammen. Neue Trends lockern diese bisherigen Konventionen auf. So kann es möglich sein, dass bei vier Paaren die Dame und der Herr zusammen tanzen, während bei den anderen vier Paaren beide getrennt voneinander tanzen. Die korrekte Einhaltung der Positionen der Bilder von allen Tänzern ist ein integraler Bestandteil des Formationssports, wodurch er sich vom Paartanz abgrenzt.

Die gesamte Choreografie besteht nun aus den Bildern und den Übergängen zwischen diesen, welche durch Figuren ertanzt werden sollen. Der komplette Prozess wird als Bildentwicklung bezeichnet. Auch für die Verwendung von Figuren sind einige Einschränkungen im Regelwerk des Deutschen Tanzsportverbandes e.V. enthalten. Auf diese wird jedoch nicht weiter eingegangen, da sie nicht für das Verständnis der weiteren Passagen benötigt werden. Im Folgenden sollen nun noch detailliertere Anforderungen an Choreografien genannt werden, die über das reine Regelwerk hinausgehen.



**Abbildung 2.1:** Ein Bild aus der Choreografie Crazy Fire, einmal getanzt durch eine Formationsgruppe (oben) und einmal schematisch im Koordinatensystem eines Bilderplans eingezeichnet (unten). Jeder Kreis im Bilderplan steht für ein Paar. Der Bilderplan zeigt die Ansicht der Tänzer. Die Aufnahme der Formationsgruppe zeigt die Ansicht eines Trainers. Das heißt das Bild wird jeweils von zwei unterschiedlichen Positionen aus betrachtet. Beide Materialien wurden von Samuel Beck bereitgestellt.

### 2.1.2 Bewertungskriterien für Choreografien

Wie bereits im letzten Abschnitt beschrieben, wird das Aufführen einer Choreografie durch eine Formationsgruppe von Wertungsrichtern beurteilt. Diese orientieren sich an den Wertungsrichtlinien im DTV für Formationswettbewerbe Standard und Latein [Deu23b], aus welchem auch die folgenden Informationen stammen. Das Ziel einer Formationsdarbietung wird in den Wertungsrichtlinien folgendermaßen definiert: „Durch tanzende Paare zur Musik einer Turnierart mit einer Auswahl an tanz-spezifischer Bewegungsabläufe synchron auf der Tanzfläche abwechslungsreiche, für Augenblicke stationäre oder bewegte, nachvollziehbare Bilder zu gestalten“ [Deu23b]. Grundsätzlich geschieht eine Bewertung durch die Vergabe von einem bis zehn Punkten in den untereinander gleichwertigen Wertungsgebieten: *Musik, Tänzerische Leistung, Ausführung der Choreografie* sowie *Durchgängigkeit und Charakteristik*. Ausdrücklich nicht bewertet werden der Ein- und Ausmarsch, das Erscheinungsbild der Tänzer, die Qualität der Musikwiedergabe und die Aussage der Choreografie.

Im Wertungsbereich *Musik* wird darauf geachtet, dass alle Tänzer den Takt und den Grundrhythmus einhalten und dass die Figuren und einzelnen Bewegungen zeitgleich ausgeführt werden.

Im Wertungsbereich *Tänzerische Leistung* wird die durchschnittliche tänzerische Leistung aller Paare der Formation und deren Konsistenz bewertet. Es wird auch darauf Wert gelegt, dass Übergänge zwischen Bildern nicht gelaufen, sondern ertanzt werden.

Der Wertungsbereich *Ausführung der Choreografie* ist vor allem für unsere Arbeit von besonderer Relevanz. Hier wird insbesondere auf die Bilder und das Stellen dieser Wert gelegt. Kriterien für eine erfolgreiche Choreografie und einen korrekt ausgeführten Formationsdurchgang sind unter anderem Folgende:

- Die Linienführung und die Entstehung von Linien ist präzise.
- Linien und Reihen sind gerade, haben also keine leichte Krümmung.
- Symmetrische Bilder sind korrekt um eine gedachte Mitte aufgebaut und asymmetrische Bilder sind klar als solche erkennbar.
- Die Abstände zwischen den Paaren bzw. den Tänzern sind gleichmäßig.
- Die Tanzfläche wird gleichmäßig ausgenutzt.

Zudem werden einige Schwierigkeiten in Choreografien herausgehoben, die in der Bewertung berücksichtigt werden sollen. Eine relevante Auswahl dieser umfasst:

- Sich bewegende Bilder sind schwieriger zu tanzen als stationäre Bilder.
- Eine Choreografie mit vielen einzigartigen Bildern ist schwerer als eine Choreografie mit wenigen, gleichförmigen Bildern.
- Je mehr Paare in einer Linie bzw. Reihe tanzen, desto schwieriger ist die Ausführung.
- Diagonalen sind in der Ausführung schwerer als Längsreihen, welche wiederum schwerer sind als Querreihen.



Im letzten Wertungsgebiet, *Durchgängigkeit und Qualität*, wird die Umsetzung der Musik und Bilder in Figuren betrachtet. Des Weiteren wird bewertet, wie geschlossen die Formation bei Wechseln von Musik, Tempo oder Bildern agiert.

Eine detailliertere Betrachtung des Formationssport wäre für den Zweck der Arbeit nicht zuträglich. Der Sport und dessen Regeln unterliegen einem ständigen Prozess der Anpassung. Des Weiteren wurde hier hauptsächlich auf den Lateinformationssport in Deutschland eingegangen. Dieser existiert zwar auch in anderen Ländern, vor allem in Mittel- und Osteuropa, jedoch sind die Änderungen nicht grundlegend. Der Formationssport kann kompetitiv zwischen Teilnehmern einzelner Nationen auf Europa- und Weltmeisterschaften ausgetragen werden.

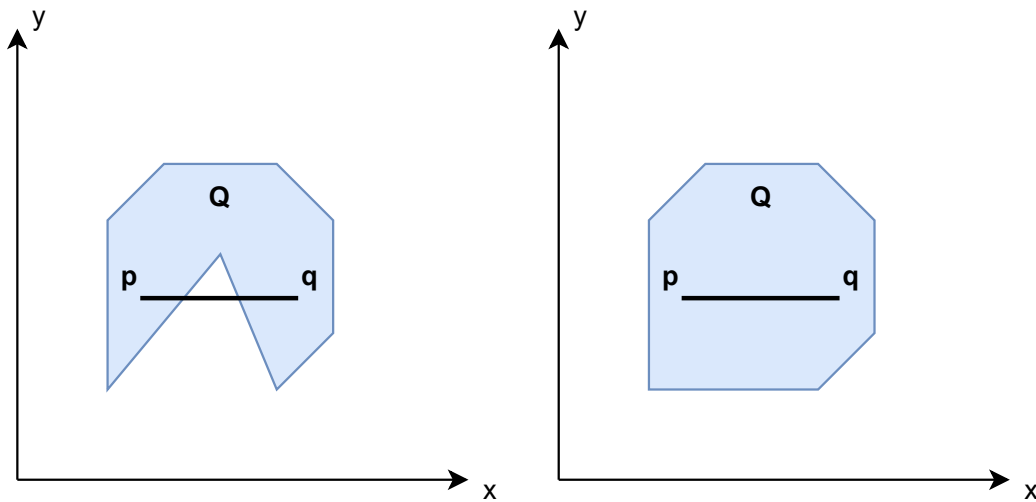
## 2.2 Technische und Algorithmische Grundlagen

Wir widmen uns nun den Grundlagen, welche für das Verständnis der Implementierung vonnöten sind. Auf der algorithmischen Seite wird mit dem Graham Scan in Unterabschnitt 2.2.1 ein Algorithmus zur Bestimmung der konvexen Hülle einer Menge von Punkten vorgestellt, was im Prototyp zur automatischen Berechnung prägnanter Formen in Bildern verwendet wird. Mit der Kernel Density Estimation wird in Unterabschnitt 2.2.2 ein Verfahren zur Erstellung von Heatmaps eingeführt, welche im Prototyp zur Visualisierung der Ausnutzung der Tanzfläche genutzt werden. In Unterabschnitt 2.2.3 wird ein Algorithmus zur Bestimmung von Geraden mit beliebigem Steigungswinkel vorgestellt, der nicht auf Grundlagen bisheriger Forschungsliteratur aufbaut. Auf der technischen Seite werden in Unterabschnitt 2.2.4 die Grundlagen responsiver Entwicklung für Webseiten erläutert.

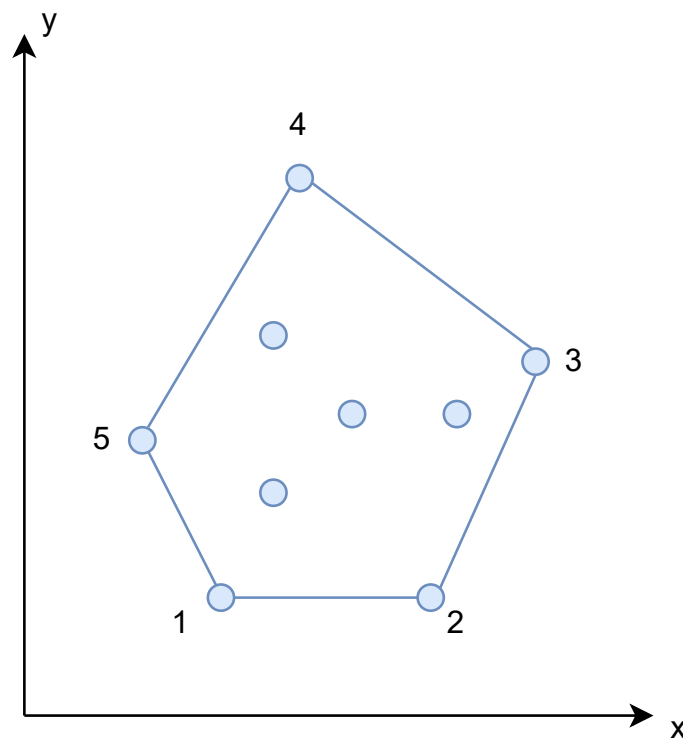
### 2.2.1 Graham Scan

Zur Ermittlung einer konvexen Hülle für eine gegebene Punktmenge existieren in der algorithmischen Geometrie viele Lösungsansätze. Einer der ersten Algorithmen ist der Graham Scan [Gra72], welcher 1972 von Ronald Graham vorgestellt wurde.

Das grundlegende Problem lautet hierbei wie folgt: Gegeben sei eine Menge  $P$  an Punkten im euklidischen Raum. In unserem Fall reicht hier die Einschränkung auf den zweidimensionalen Raum  $\mathbb{R}^2$ . Eine Teilmenge  $Q \subseteq \mathbb{R}^2$  ist konvex, wenn für alle Liniensegmente  $p, q \in Q$  gilt: Die Linie  $\overline{pq}$  ist vollständig in  $Q$  enthalten, sie schneidet also keine Kante. Beispiele hierfür sind in Abbildung 2.2 gegeben, wobei die linke Form keine konvexe Hülle ist, da die Linie  $\overline{pq}$  mindestens eine Kante schneidet. Die konvexe Hülle  $CH(Q)$  der Punktmenge  $Q$  ist die kleinste konvexe Hülle, die  $Q$  enthält. Im euklidischen Raum  $\mathbb{R}^2$  reduziert sich dies auf ein Polygon, also ein Vieleck, welches aus allen Liniensegmenten besteht, welche die konvexe Hülle begrenzen. Der Graham Scan liefert nun die konvexe Menge einer gegebenen Menge an Ausgangspunkten. Die Punkte sind in der Ergebnismenge gegen den Uhrzeigersinn angeordnet und starten bei dem Punkt mit dem niedrigsten y-Wert. Verbindet man immer zwei nacheinander in die Ergebnismenge eingefügte Punkte mit Liniensegmenten, so bekommt man das konvexe Polygon, welches alle eingegebenen Punkte beinhaltet. Ein Beispiel für das Ergebnis von Grahams Scan und die Einfügereihenfolge in die Ergebnismenge sind in Abbildung 2.3 zu sehen.



**Abbildung 2.2:** Links ist ein Beispiel für eine nicht-konvexe Hülle zu sehen, rechts ein Beispiel für eine konvexe Hülle. Die Hülle links ist nicht konvex, da  $\overline{pq}$  mindestens eine Kante des Polygons schneidet.



**Abbildung 2.3:** Beispiel für das Ergebnis vom Graham Scan. Alle Punkte sind als blaue Kreise in das Koordinatensystem eingetragen. Die konvexe Hülle ist über die Verbindung der Eckpunkte des resultierenden Polygons gekennzeichnet. Die Nummern zeigen die Reihenfolge, in welcher der Graham Scan die Punkte der Ergebnismenge hinzufügt.

**Algorithmus 2.1** Graham Scan

---

```

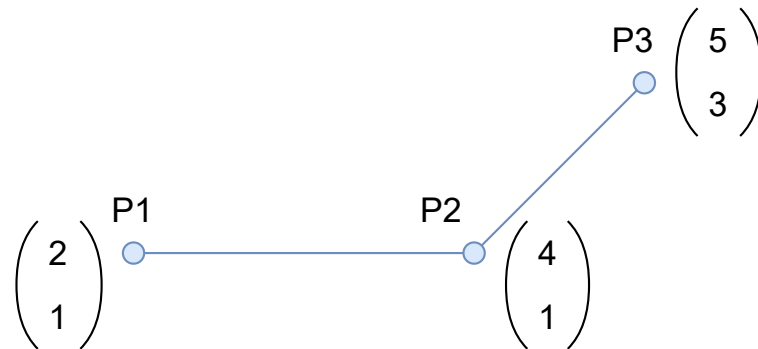
procedure GRAHAM-SCAN( $Q$ )
   $p_0 \leftarrow p \in Q$  mit minimaler  $y$ -Koordinate, der am weitesten links steht
   $\langle p_1, p_2, \dots, p_m \rangle \leftarrow$  restliche Punkte in  $Q$ , sortiert nach Polarwinkel um  $p_0$  gegen den
  Uhrzeigersinn
   $S \leftarrow$  leerer Stapel
  PUSH( $p_1, S$ )
  PUSH( $p_2, S$ )
  PUSH( $p_3, S$ )
   $\langle p_4, p_5, \dots, p_m \rangle \leftarrow \langle p_1, p_2, \dots, p_m \rangle$  ohne  $p_1, p_2, p_3$ 
  for  $i = 3$  to  $m$  do
    while Winkel zwischen NEXTTO TOP( $S$ ), TOP( $S$ ) und  $p_i$  macht keine Linkskurve do
      POP( $S$ )
    end while
    PUSH( $p_i, S$ )
  end for
  return  $S$ 
end procedure

```

---

Im Folgenden wird eine Variante des Algorithmus von Cormen et al. [CLRS22] aus dem Buch „Introduction to Algorithms“ zur weiteren Erklärung verwendet. Der Pseudocode dazu findet sich in Algorithmus 2.1. Die Operationen Push und Pop legen ein Element (hier einen Punkt) auf den Stapel oder nehmen diesen wieder herunter. Die Operation Top gibt das oberste Element des Stapels zurück ohne es zu entfernen. Die Operation NextToTop macht das gleiche für das vorletzte Element des Stapels. Um zu beurteilen, ob zwei verbundene Liniensegmente, welche durch die Punkte  $P_1, P_2, P_3$  definiert sind, eine Rechts- oder Linkskurve bilden, kann die z-Koordinate der Kreuzprodukte der Vektoren  $\overrightarrow{P_1P_2}$  und  $\overrightarrow{P_1P_3}$  miteinander verglichen werden. Dieses Kreuzprodukt lässt sich mit folgendem Ausdruck berechnen:  $(x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1)$ . Ist der Wert dieses Ausdrucks positiv, dann handelt es sich um eine Linkskurve. Ist der Wert negativ, dann ergeben die drei Liniensegmente eine Rechtskurve. Nimmt der Ausdruck den Wert 0 an, so haben wir den Sonderfall, dass die drei Punkte kollinear zueinander sind. In diesem Fall kann selbst entschieden werden, ob diese Punkte Teil der konvexen Hülle sein sollen. Wenn ja, dann ist die Menge der Punkte, die das Polygon aufspannen, nicht mehr minimal. Ein Beispiel für die Errechnung des Kurvenverhaltens ist in Abbildung 2.4 zu sehen. Dort ist der Wert des Ausdrucks  $(4 - 2)(3 - 1) - (1 - 1)(5 - 2) = 4$ . Da  $4 > 0$  ist, handelt es sich um eine Linkskurve.

Der Algorithmus hat eine Laufzeit von  $O(n \log n)$ , wobei  $n$  die Anzahl der Punkte in der Menge  $Q$  ist. Die limitierende Komponente ist hierbei die Sortierung der Punkte nach Polarwinkeln. Die Laufzeit kann jedoch im Rahmen der Arbeit hier vernachlässigt werden, da  $n = 16$  das Maximum ist.



**Abbildung 2.4:** Beispiel zur Errechnung des Kurvenverhaltens zwischen Linien, die durch die Punkte  $P_1$ ,  $P_2$  und  $P_3$  aufgespannt wird. Es handelt sich um eine Linkskurve.

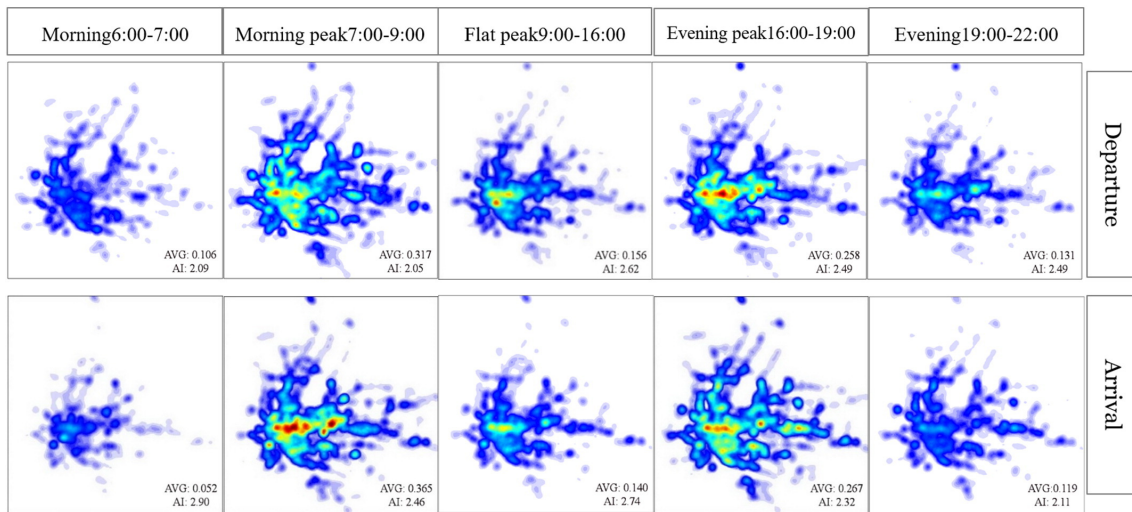
## 2.2.2 Kernel Density Estimation und Heatmaps

Das Ziel dieses Abschnitts ist es, Techniken darzustellen, mit denen man die räumliche Verteilung von Daten über eine quantitative Skala visualisieren kann. Ein typischer Weg ist es, die räumliche Dichte auf einer Farbskala und die Position im Raum auf eine Position in der Visualisierung abzubilden. In diesem Fall spricht man dann von sogenannten Heatmaps. Ein Beispiel ist in Abbildung 2.5 zu sehen. Die Visualisierung stammt aus einer Arbeit von Yu und He [YH17], in welcher der Bedarf an Bussen an bestimmten Orten in Guangzhou visualisiert wird. So ist in dieser direkt ersichtlich, dass der Bedarf zum Arbeitsbeginn zwischen 7:00-9:00 Uhr und während dem Feierabend zwischen 16:00-19:00 Uhr in einigen Bezirken besonders hoch ist und dort Buslinien mit höherer Frequenz befahren werden sollten.

Grundsätzlich kann man Heatmapdarstellungen in diskrete Gitter-Heatmaps und kontinuierliche, räumliche Heatmaps einteilen, wobei viele alternative Ansätze existieren. Ein Beispiel für eine kontinuierliche, räumliche Heatmap ist in Abbildung 2.5 zu sehen. Bei diskreten Gitter-Heatmaps wird der Raum in ein Gitter unterteilt und jeder Datenpunkt einer Gitterzelle zugeordnet, was sich *binning* nennt. Anschließend wird jede Gitterzelle einzeln visualisiert. Das Gitter muss hierfür nicht zwangsweise uniform sein.

Die weiteren Konzepte werden nun erst im eindimensionalen Fall erklärt. Eine Erweiterung auf zwei Dimensionen folgt danach. Gegeben seien nun Daten, die über eine quantitative Skala verteilt sind. Teilt man die Werte in gleich große *Bins* ein, so bekommt man ein Histogramm. Dies ist das eindimensionale Pendant zur Heatmap mit uniformem Gitter. Bei diesem Ansatz ist eine geeignete Wahl für die Anzahl der *Bins* essenziell. Wählt man die Anzahl der *Bins* zu groß, kann es zu vielen Ausreißern kommen. Bei einer zu geringen Anzahl an *Bins*, kann hingegen zu viel an Information verloren gehen.

Ein Weg dieses Problem zu umgehen ist die Schätzung der Probability Distribution Function (PDF) per Kernel Density Estimation. Dieses Verfahren wurde unter anderem von Parzen [Par62] entwickelt. Der Grundgedanke dieser Methode ist die Faltung aller Datenpunkte mit einem Kernel. Ein Kernel ist eine reellwertige, nicht-negative Funktion  $K$ , mit  $\int_{-\infty}^{\infty} K(x) dx = 1$ . Das heißt der Flächeninhalt, der vom Kernel und der x-Achse eingeschlossen wird, beträgt genau 1. Ein bekannter



**Abbildung 2.5:** Visualisierung für den Bedarf an öffentlichen Verkehrsmitteln in Guangzhou mittels mehrerer Heatmaps von Yu und He [YH17]. Eine Heatmap zeigt den Bedarf in einem bestimmten Zeitintervall. Es wird dabei zwischen Ankunfts- und Abfahrtsort unterschieden. Der Bedarf wird auf eine Farbskala abgebildet, in der blau für einen geringen Bedarf und rot für einen hohen Bedarf steht.

Kernel ist der Gauss-Kernel (siehe Gleichung 2.1). Das Schaubild des Kernels ist in Abbildung 2.6 zu sehen.

$$(2.1) \quad K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

Die Faltung zweier Funktionen ist mathematisch folgendermaßen definiert:

$$(2.2) \quad (f * g)(x) = \int_{-\infty}^{\infty} f(t)g(x-t) dt$$

Die Reihenfolge der Funktionen ist irrelevant, da die Faltung kommutativ ist. Da der gegebene Datensatz, jedoch keine Funktion ist, gibt es für die Faltung auch eine diskrete Version:

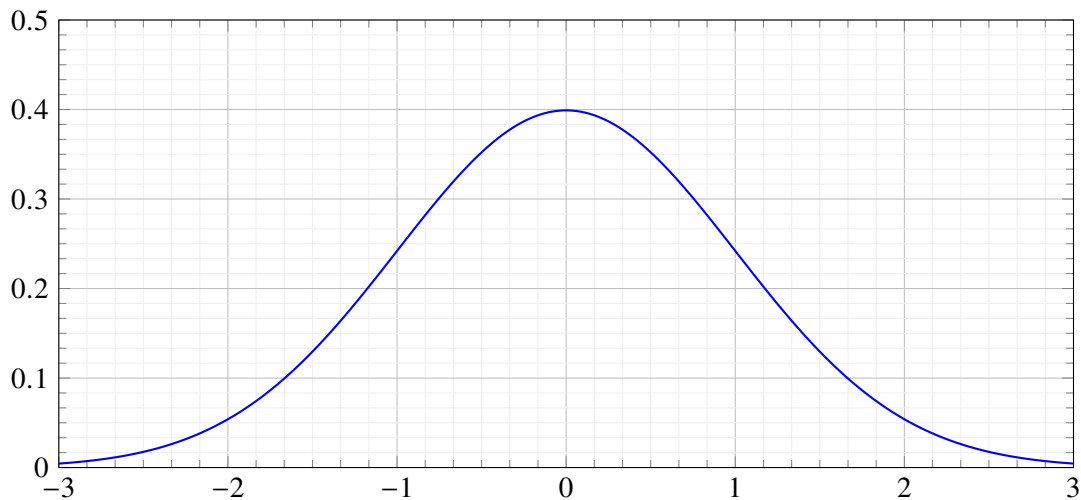
$$(2.3) \quad (f * g)(x) = \sum_{t=1}^n f(t)g(x-t)$$

Der Parameter  $n$  gibt dann die Genauigkeit der Abtastung an. Nun sind die beiden Funktionen  $f$  und  $g$  einmal der zu visualisierende Datensatz  $\{x_1, x_2, \dots, x_n\}$  und einmal der Kernel  $K$ . Damit lässt sich für jeden Punkt der quantitativen Skala ein geschätzter Dichtewert  $d_h$  bei gegebener Bandbreite  $h$  mit folgender Formel berechnen:

$$(2.4) \quad d_h(x) = \frac{1}{n} \sum_{t=1}^n K_h(x - x_i)$$

Die Bandbreite  $h$  gibt an, wie spitz der Kernel an der Stelle 0 zuläuft. Die Formel für einen Gauss-Kernel mit Bandbreite  $h$  lautet wie folgt:

$$(2.5) \quad K_h(x) = \frac{1}{\sqrt{2\pi h^2}} e^{-\frac{x^2}{2h^2}}$$



**Abbildung 2.6:** Graph der Funktion 2.1. Es handelt sich hierbei um einen Gauss-Kernel.

Im Histogramm würde der Kernel der Breite der einzelnen *Bins* entsprechen. Somit sollte auch hier eine Bandbreite  $h$  gewählt werden, die weder zu viel an Information verwirft, noch zu hohe Frequenzen zulässt. Eine Erweiterung in den zweidimensionalen Raum sieht dann im diskreten Fall wie folgt aus:

$$(2.6) \quad (f * g)(x, y) = \sum_{t=1}^n \sum_{q=1}^m f(t, q)g(x - t, y - q)$$

In der Bildverarbeitung wird dieses Verfahren als Tiefpassfilter verwendet. Da  $\lim_{x \rightarrow \infty} K(x) = \lim_{x \rightarrow -\infty} K(x) = 0$  gilt, kann man den Kernel beschneiden. Bei einem Gauss-Kernel mit  $h = 1$  sind beispielsweise alle Werte kleiner als  $-3$  und größer als  $3$  so klein, dass sie vernachlässigt werden können. Es reicht also, sich eine kleine Maske zu definieren, in welcher die abgetasteten Werte des zweidimensionalen Kernels stehen. Die Maske kann dann mit dem Originalbild gefaltet werden. Da die Faltung für jedes Bildpixel berechnet werden muss, gibt es im Bereich der Bildverarbeitung Verfahren zur Beschleunigung (z.B. die Fourier-Transformation).

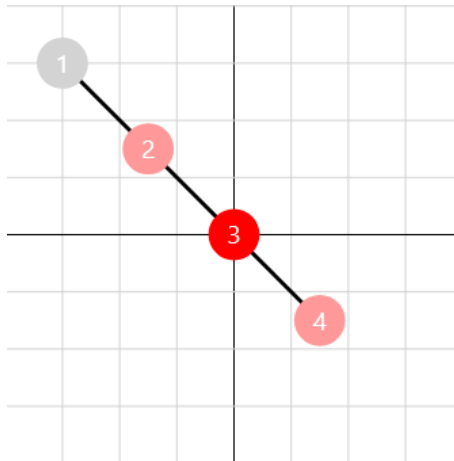
Zur Erstellung von Heatmaps hat sich ein anderes Verfahren durchgesetzt, das sogenannte *Kernel-Splatting*, welches von Westover [Wes91] entwickelt wurde und eigentlich für das Volumenrendering gedacht war. Die Grundidee entstammt dem inhärenten Unterschied zwischen den Rendering-Verfahren Raytracing und Rasterisierung. Während man beim Raytracing die Berechnung pro Pixel durchführt, wird die Berechnung bei der Rasterisierung pro Objekt durchgeführt. Somit kann man bei der Rasterisierung eine interaktive Bildgebung auch bei hohen Auflösungen garantieren. Dieselbe Idee kann auch bei der Generierung der Heatmap genutzt werden und wurde unter anderem auch von Thom et al. [TBKE14] zur Heatmapdarstellung für große Datenmengen erfolgreich angewandt. Anstatt also die Berechnung für jedes Pixel der Heatmap auszuführen, wird ein zweidimensionaler Kernel abgetastet und die Resultate in einer Textur, auch *splat* genannt, gespeichert. Für jeden Datenpunkt kann die Textur dann auf die Heatmap aufaddiert werden (additives Blending). Die Textur muss dabei so verschoben werden, dass das Zentrum des Kernels mit dem höchsten Funktionswert an der Stelle des Datenpunktes liegt. Ein Beispiel für eine Textur, in der ein zweidimensionaler Gauss-Kernel enthalten ist, ist in Abbildung 2.7 zu sehen. In einem letzten Schritt müssen die errechneten Werte aus beiden Methoden noch auf eine Farbskala abgebildet und auf dem Bildschirm angezeigt werden.



**Abbildung 2.7:** *Splat* in Form eines Gauss-Kernels. Übernommen aus Thom et al. [TBKE14].

### 2.2.3 Algorithmus zum Erkennen von Diagonalen

Nun wird ein Algorithmus zur Lösung des folgenden Problems erläutert: Gegeben sei eine Menge  $P$  von Punkten in  $\mathbb{R}^2$ . Von diesen Punkten wird nun ein Punkt, genannt  $p_{sel} \in P$ , ausgewählt. Es soll eine von links nach rechts fallende bzw. steigende Gerade mit ansonsten beliebiger Steigung gefunden werden, auf der  $p_{sel}$  und mindestens drei andere Punkte aus  $P$  liegen. Ein Beispiel dafür ist in Abbildung 2.8 zu sehen. Es wurde dabei eine Gerade gefunden, welche ausgehend vom ausgewählten Punkt 3 vom linken oberen Quadranten zum rechten unteren Quadranten verläuft. Dieser Fall wird auch weiter in diesem Abschnitt besprochen. Der Fall für das Finden einer Diagonale, welche im linken unteren Quadranten beginnt und im rechten oberen Quadranten endet, sieht ähnlich aus. Grundgedanke des Verfahrens ist es, alle Punkte des linken, oberen Quadranten mit allen Punkten des rechten, unteren Quadranten zu kombinieren. Dies spannt eine Testgerade auf. Dann errechnet man für den ausgewählten Punkt, ob dieser sich auf der gerade aufgespannten Gerade befindet. Wenn ja, wird noch nachgezählt, wie viele Punkte sich insgesamt auf dieser Geraden befinden. Ist diese Zahl mindestens vier, dann bricht die Berechnung ab, da eine Gerade gefunden wurde. Damit kann zwar der Sonderfall nicht abgedeckt werden, bei dem mehrere Diagonalen durch den ausgewählten Punkt  $p_{sel}$  gehen, jedoch kann dieser Fall bei der geringen Mengen an Punkten in einem Bild vernachlässigt werden. Das Verfahren ist als Pseudocode auch noch in Algorithmus 2.2 zu sehen. Der Algorithmus hat eine Laufzeit von  $O(n^3)$ , wobei  $n$  die Anzahl der Punkte ist.



**Abbildung 2.8:** Darstellung einer Diagonale, die im linken, oberen Quadranten beginnt und im rechten, unteren Quadranten endet. Punkt 3 ist hier der ausgewählte Punkt und deswegen dunkelrot markiert. Die direkten Nachbarn von Punkt 3 sind hellrot markiert.

---

#### Algorithmus 2.2 Erkennung von Diagonalen

---

```

procedure DIAGONAL( $P, p_{sel}$ )
   $P_l \leftarrow p \in P$  mit im linken, oberen Quadrant ausgehend von  $p$ 
   $P_r \leftarrow p \in P$  mit im rechten, unteren Quadrant ausgehend von  $p$ 
  for all  $p_l \in P_l$  do
    for all  $p_r \in P_r$  do
       $Diagonal \leftarrow \{p_r, p_l\}$ 
      if COLLINEAR( $p_l, p_{sel}, p_r$ ) then
         $Diagonal \leftarrow Diagonal \cup \{p_{sel}\}$ 
        for all  $p_k \in (P_l \cup P_r)$  do
          if COLLINEAR( $p_l, p_k, p_r$ ) then
             $Diagonal \leftarrow Diagonal \cup \{p_k\}$ 
            if  $|Diagonal| > 3$  then
              return  $Diagonal$ 
            end if
          end if
        end for
      end if
    end for
  end for
end procedure

```

---



### 2.2.4 Webentwicklung für diverse Endgeräte

Der Bereich der Webentwicklung wurde laut Nebeling und Norrie [NN13] in der letzten Dekade fortlaufend weiterentwickelt. Dies liegt hauptsächlich an der Diversifizierung der Endgeräte, mit denen Webanwendungen in Browsern genutzt werden. Webseiten werden nicht nur in Browsern auf klassischen Desktopsystemen verwendet, sondern auf einer ganzen Riege moderner, meist mobiler Systeme. Dazu zählen kleine Geräte wie Smartphones oder mittelgroße Geräte wie Tablets oder Convertibles. Die Geräte unterscheiden sich vor allem in der Größe, der standardmäßigen Ausrichtung der Bildschirme und der verwendeten Interaktionstechnik. Im Bereich der Bildschirmgröße reicht die Spanne der Möglichkeiten von hochauflösenden Breitbildmonitoren bis zu kleinen, meist hochkant verwendeten Bildschirmen auf Smartphones. Weitere Extrema wie Darstellungen auf Smartwatches sind zwar möglich, bilden aber bisher noch die Ausnahme. Auf der Seite der Interaktionsmöglichkeiten sind die Maus- und Tastatureingabe auf Desktopsystemen und Notebooks der touch- oder gestenbasierten Eingabe auf mobilen Geräten gegenüberzustellen. Moderne Webseiten sollten ihre Elemente also nicht nur an den verfügbaren Platz anpassen, sondern sie sollten auch einheitlich mit jeder Interaktionstechnik bedienbar sein.

Um dies zu erreichen, existierten zwei Designstrategien, nämlich *Graceful Degradation* und *Progressive Enhancement*. Bei beiden handelt es sich um geschichtete Ansätze, bei denen Schichten (Funktionalität, Interaktionselemente, etc.) je nach Rechenstärke, Bildschirmgröße und Interaktionstechnik entfernt oder hinzugefügt werden. Bei der Anwendung von *Graceful Degradation* startet man bei der Gestaltung der Benutzeroberfläche bei der Plattform mit den wenigsten Einschränkungen, also meist einem Desktoprechner. Falls ein Gerät eine bestimmte Funktionalität nicht unterstützt, dann wird sie auf diesem Gerät entfernt. Zum Beispiel sind aufwändige 3D-Grafikberechnungen auf mobilen Geräten mit weniger Rechenleistung schwerer umsetzbar als auf Computern mit einer dedizierten Grafikkarte. Auch ist ein präzises Klicken mit der Maus auf eine kleine Schaltfläche schwer auf ein fingerbasiertes Touchdisplay abzubilden. *Progressive Enhancement* im Gegenteil startet bei dem Gerät mit den meisten Einschränkungen und fügt stückweise Ebenen hinzu. Ein darauf folgender Trend nennt sich Responsive Webentwicklung und wurde für diese Arbeit verwendet. Deswegen wird die zugrundeliegende Technik im nächsten Unterabschnitt näher erläutert.

#### Responsive Webentwicklung

Bei der Erstellung responsiver Systeme müssen nach Gardner [Gar11] drei Aspekte beachtet werden. Der erste Aspekt ist die Verwendung eines fluiden Layouts. Bei Webseiten wird dabei der Zeichenbereich in zwölf gleich breite Spalten eingeteilt, in welchen die einzelnen Elemente dann angeordnet werden. Verändert sich die Größe des Browserfensters, dann werden alle Spalten und die darin liegenden Komponenten gleichmäßig mitskaliert.

Ein zweiter Aspekt sind eingebettete Medien. Diese können nicht beliebig skaliert werden. So verpixeln zu große Bilder und der Inhalt zu kleiner Bilder ist nicht mehr sichtbar. In diesem Fall muss man die Größe von eingebetteten Medien beschränken. Dies kann durch CSS-Attribute wie `min-width` und `max-width` spezifiziert werden.

Mit fluiden Layouts und mitskalierenden Komponenten, kann es trotzdem zu unschönen Effekten bei manchen Bildschirmgrößen kommen. Ist ein Bildschirm zum Beispiel sehr schmal, dann werden auch alle Buttons schmaler, sodass die Beschriftung der Buttons unterbrochen wird. Um dies zu

---

### Listing 2.1 CSS Media Query

---

```
@media only screen and (max-height: 575px) {  
  .adaptive-element {  
    display: none !important;  
  }  
}
```

---

verhindern, sollte man für verschiedene Arten von Geräten (Smartphones, Tablets, Desktop-PCs, etc.) verschiedene Layouts definieren. CSS bietet dafür seit Version 2.1 *media queries* an, mit denen man CSS-Attribute abhängig von der Größe des anzeigenden Bildschirms ändern kann. Ein Beispiel dafür ist in Listing 2.1 zu sehen. Dort wird ein HTML-Element nur angezeigt, wenn die innere Fenstergröße (ohne Browserleiste) eine Höhe von 575 Pixeln übersteigt. Hierfür muss dem HTML Element die Klasse `adaptive-element` zugeordnet sein. Die innere Größe des Fensters ist im weiteren Verlauf der Arbeit immer dann gemeint, wenn von der Bildschirmgröße gesprochen wird. Es existieren bereits einige Bibliotheken die vorgefertigte Lösungen zur Gestaltung responsiver Webseiten bereitstellen. Eine dieser Bibliotheken wird in Unterabschnitt 5.1.2 kurz beschrieben.

Wie bereits erwähnt, muss bei der Verwendung von mobilen Geräten nicht nur die Anpassung der Bildschirmgröße sondern auch die Art der Interaktion bedacht werden. Moderne Browser konvertieren hierbei bereits HTML-Events wie `onclick`, `onmousedown` oder `onmouseup`, die für Mauseingaben konzipiert wurden, in Pendanten für Geräte mit einer Toucheingabe. Statt einem Mausklick wird beispielsweise ein Finger auf dem Touchscreen erwartet.

## 3 Verwandte Arbeiten

Sportvisualisierung ist ein Bereich, der seit dem ersten Seminar über dieses Thema auf der IEEE VIS Konferenz 2013 stark gewachsen ist, wie Perin et al. [PVS+18] zeigen. Sie teilen in ihrer Arbeit die für Visualisierungen relevanten Daten in drei Kategorien ein, nämlich in Spielberichtsdaten, aufgezeichnete Daten und Meta-Daten. Spielberichtsdaten stellen zum Beispiel Events wie Tore in einem Fußballspiel dar. Bei aufgezeichneten Daten handelt es sich um Daten mit räumlich-zeitlicher Dimension, welche meist mit dem aufgezeichneten Bewegungsapparat der Athleten verbunden sind. Meta-Daten hingegen gehen meist über den Kontext des Spiels hinaus und können hochheterogen sein. Ein Beispiel für Meta-Daten im Fußball wären die Körpermaße der Spieler oder die Auslastung der Fußballstadien.

Eine Taxonomie speziell für Visualisierungen von Teamsportdaten wurde von Page und Moere [PM06] vorgeschlagen. Dort wird eine Einteilung in vier Kategorien vorgenommen. Diese sind die zu visualisierenden Daten, die Art der gewählten Visualisierung, die Zielgruppe sowie das Einsatzgebiet. Auffällig ist, dass keine der beiden Taxonomien den Formationstanz bzw. Tanzsport im Allgemeinen berücksichtigt.

Es wird im Folgenden nun einen kurzer Überblick über bereits existierende Arbeiten gegeben. Dazu werden in Abschnitt 3.1 Teamsportvisualisierungen im Allgemeinen mit Fokus auf räumlich-zeitlichen Daten betrachtet. Danach folgt in Abschnitt 3.2 eine Übersicht von existierenden Arbeiten im Bereich der Visualisierung für Tanzsport und Formationstanz. Zum Abschluss versuchen wir dann in Abschnitt 3.3 unsere Arbeit in die bestehenden Taxonomien einzuordnen.

### 3.1 Teamsportvisualisierung

Der Forschungsbereich der Teamsportvisualisierungen beschäftigt sich momentan vor allem mit der direkten Analyse von Spielen in populären Sportarten. Stein et al. [SJL+18] entwickelten ein System zur Analyse von Fußballspielen über TV-Übertragungen, bei dem die Visualisierung direkt in das übertragene Bild eingebettet wird und die Analyse somit nicht anhand abstrakterer 2D-Visualisierungen stattfinden muss. Dazu werden die Spieler mit Bilderkennungsverfahren in der Übertragung erkannt und dann je nach Wunsch verschiedene Metriken und daraus resultierende Visualisierungen berechnet. Dazu gehören unter anderem Spielertrajektorien, dominante Regionen, in denen ein Team das Spiel kontrolliert oder mögliche Passoptionen ballführender Spieler. In einer nachfolgenden Arbeit betteten Stein et al. [SBH+18] dies dann in ein konzeptuelles Workflow-Modell ein. Dies ermöglicht eine Form der Interaktion, bei welcher ein Nutzer „Was wäre wenn“-Szenarien erstellen kann, indem er beispielsweise Spieler auf dem Feld hin- und herbewegt. In beiden Fällen wiesen qualitative Evaluationen mit Fußballexperten darauf hin, dass solche Werkzeuge die Nachbetrachtung eines Fußballspiels sinnvoll ergänzen können. Auch Formationen, also die Aufstellung der Mannschaft, können trotz häufiger Positionswechsel anhand dieser Daten ermittelt werden, wie Bialkowski et al. [BLC+16] zeigten.

SnapShot von Pileggi et al. [PSBS12] visualisiert das Schussverhalten von Eishockeyspielern auf einem 2D-Feld aus der Vogelperspektive. Die Schusshäufigkeit bei einer bestimmten Entfernung zum Tor wird als konzentrischer Kreis eingezeichnet, wobei die Häufigkeit die Intensität der Kreisfarbe beeinflusst. BKViz von Losada et al. [LTB16] ist ein Programm zur Analyse von Basketballspielen und ermöglicht neben einer Analyse von Spielberichtsdaten und Meta-Daten auch das Nachverfolgen einzelner Spielzüge. Zudem lässt sich das Schussverhalten verschiedener Teams anhand einiger Visualisierungen wie Heatmap, radialer Heatmap und Scatterplot (auf dem Spielfeld) aus einer abstrakten Top-Down-Sicht erforschen. Sowohl SnapShot als auch BKViz wurden in Kooperation mit Domänenexperten evaluiert.

Owens und Jankun-Kelly [OJ13] versuchen ebenfalls eine Analyse von Spielzügen zu ermöglichen, jedoch im Bereich des American Footballs. Statt die Spielzüge direkt auf einem 2D-Feld aufzuzeigen, wird jeder Spielzug durch einen Bogen repräsentiert, der auf einer Linie, einer Repräsentation des Spielfelds, enden muss. Damit lässt sich der Start- und Endpunkt eines Spielzugs bestimmen. Die Farbe kodiert noch die Art des Spielzugs (Pass, Run, Penalty, etc.).

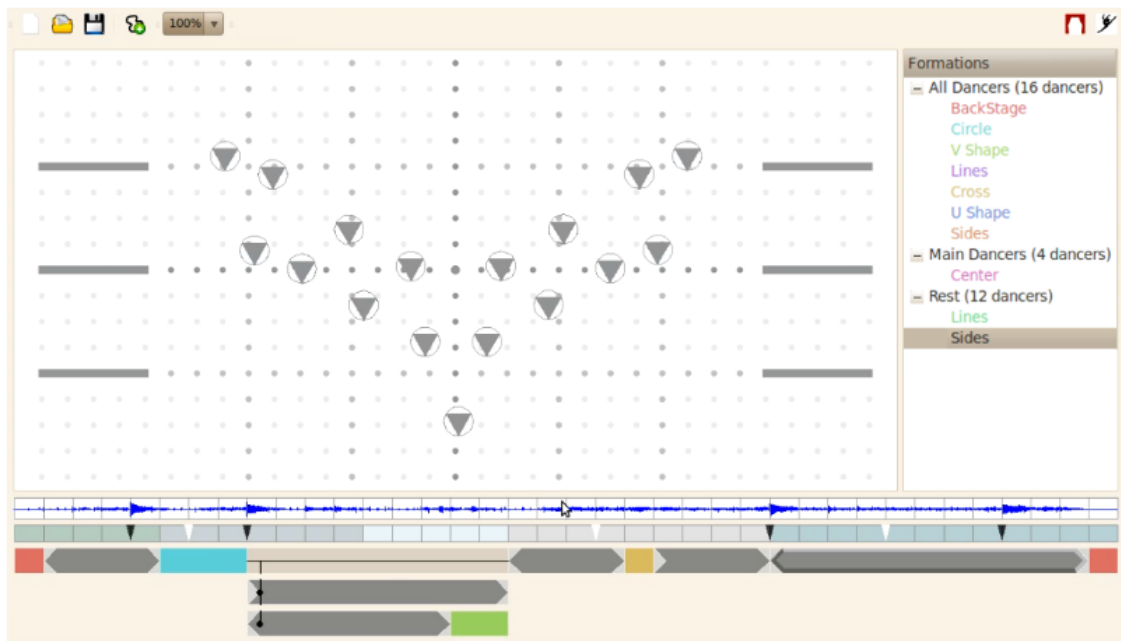
Es lässt sich anhand der vorgestellten Auswahl von Beispielen sehen, dass Teamsportdaten mit räumlich-zeitlicher Dimension eine Vielzahl von Visualisierungsmöglichkeiten bieten, sei es direkt im aufgenommenen Videomaterial, auf einer zweidimensionalen Repräsentation des Spielfelds oder in aggregierter und visuell kodierter Form. Auch muss angemerkt werden, dass es aufgrund des kommerziellen Erfolgs der genannten Sportarten eine Vielzahl an Werkzeugen am Rand des wissenschaftlichen Diskurses gibt. Ein, in Deutschland prominentes, Beispiel ist Viz Libero von Vizrt [Viz22], welches Möglichkeiten zur 3D-Analyse von Spielen zusammen mit Funktionalitäten zur Annotation (z.B. durch Pfeile, Hervorheben von Spielern, etc.) bietet.

## 3.2 Visualisierung für (Formations-)Tanz

Obwohl Visualisierungen im Bereich des Tanzsports weniger verbreitet sind, existieren hier bereits einige Arbeiten. ActionPlot von Carlson et al. [CSS11] fokussiert sich auf die Bewegungsanalyse von zeitgenössischem Tanz und richtet sich hauptsächlich an Experten in diesem Gebiet. Dabei werden verschiedene Aspekte des Tanzes wie Tempo, Bewegungsaufwand, Blickrichtung oder die Körperhaltung der Tänzer über verschiedene Teilvisualisierungen und Informationskanäle (Glyphen, Punkte, Farbkodierung) vermittelt. Es erfolgt jedoch keine Darstellung von Formationen oder koordinierten Tanzbemühungen.

DanceMoves von Arpatzoglou et al. [AKD+21] ist ein Werkzeug zur Analyse von Bewegungsabläufen von Tänzern in aufgezeichneten Videos. In mehreren koordinierten Ansichten kann dazu das Verhalten einzelner Körperteile über die Zeit beobachtet werden. Des Weiteren kann man die Ähnlichkeit der Bewegungen in zwei Videos über eine Pixelmatrix miteinander vergleichen. Auch Anfragen nach bestimmten Bewegungsabläufen an das System sind möglich. Jedoch ist das System auf die Bewegungsanalyse von einzelnen Tänzer und nicht für die Analyse ganzer Formationen ausgelegt.

Griggio et al. [GRL15] entwickelten das Programm Canvas Dance, mit welchem koordiniertes Tanzen mehrerer Personen in informellen Kontexten wie Nachtclubs unterstützt werden sollte. Den Tänzern wurden dafür nicht direkt Tanzbewegungen oder Formationen gezeigt, sondern die Bewegungen der einzelnen Tänzer wurden mit Sensoren aktueller Smartphones aufgenommen und

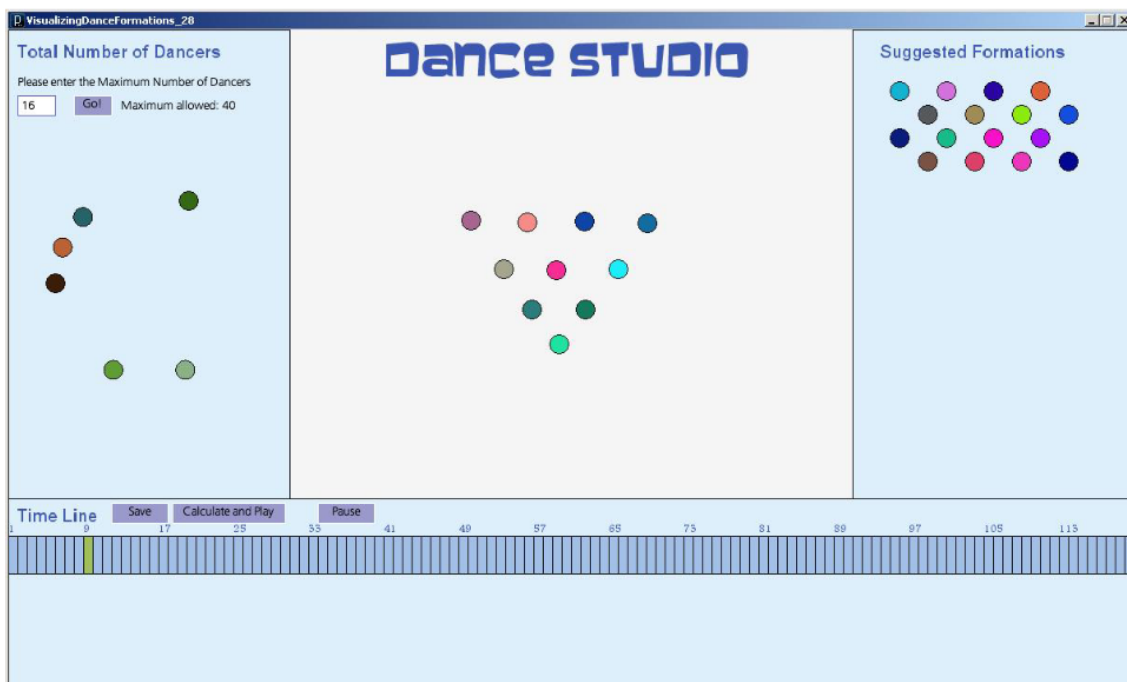


**Abbildung 3.1:** 2D-Ansicht der Choreografie in ChoreoGraphics von Schulz et al. [SMV13]. Die Tänzer sind durch Kreise auf der Tanzfläche eingezeichnet. Die Dreiecke in den Tänzersymbolen zeigen ihre Ausrichtung. Im Menü rechts sind vordefinierte Bilder zu finden. Die Zeitleiste unten ordnet das Bild zeitlich zur Musik ein.

dann auf einem Display abgebildet. Jedem Tänzer war dabei ein Kreis zugeordnet und er konnte durch eigene Bewegungen die Position und das Aussehen des Kreises verändern oder visuelle Effekte (z.B. Partikel) auslösen. Es zeigte sich jedoch in einer anschließenden Nutzerstudie, dass dies nur eingeschränkt zu einer Verbesserung der Koordination der Tänzer führte. Hier ist der Fokus der Arbeit anders als in unserer, da zwar koordinierte Tanzbewegungen visualisiert werden, diese jedoch nicht formell definiert sind und sich nicht an einer vorgefertigten Choreografie orientieren.

Ein Programm, das sich vor allem auf den Formationstanz konzentriert, wurde von Yoshida und Yonezawa [YY] vorgestellt. Hierbei fokussierten sich die Autoren auf die Darstellung der kollektiven Bewegungen und dem daraus resultierenden Eindruck einer Formation. Dazu wurde die Darstellung einfacher Avatare mit einer Bibliothek voraufgenommener Tanzbewegungen kombiniert und dies dann in einem dreidimensionalen Raum dargestellt. Das Erstellen der Bewegungsvorlagen erfolgte mit einem Kinect Sensor. Der Begriff des Bildes in einer Choreografie fällt in ihrer Arbeit nicht, jedoch existieren sogenannte „reference shapes“, mit denen sich die Choreografie bauen lässt. Ihre Arbeit unterscheidet sich von unserer Arbeit insofern, dass das Erstellen von Choreografien zwar im Vordergrund steht, der Prozess jedoch vollständig in einer 3D-Ansicht stattfindet. Eine 2D-Ansicht wird in ihrer Arbeit nicht erwähnt.

ChoreoGraphics von Schulz et al. [SMV13] erweitert diese Ideen und versucht alle Aspekte für die Planung einer Choreografie zu berücksichtigen. So existiert neben den per Motion-Capture erfassten Bewegungen in 3D auch eine 2D-Sicht der Tänzer, welche deren Position und Rotation beinhaltet. Übergänge zwischen einzelnen Bildern der Formation können vom Programm per bipartitem Matching berechnet werden und mit initialen Bedingungen können Trajektorien für die



**Abbildung 3.2:** DanceStudio von Muhammad [Muh09]. In der Programmmitte sind die Tänzer über verschiedenfarbige Kreise auf der Tanzfläche eingezeichnet. Rechts wird ein vordefiniertes Bild vorgeschlagen. Links kann die Anzahl der Tänzer verändert werden. Unter der Visualisierung ist das Bild auf einer diskreten Zeitleiste, bestehend aus Schlägen, eingetragen.

Tänzer erstellt werden. Währenddessen läuft eine Berechnung, welche die Übergänge auf mögliche Kollisionen testet. Des Weiteren existieren prozedurale Methoden wie Attraktoren und Distraktoren, mit welchem das Programm automatisch neue Bilder generieren kann. Darüber hinaus kann noch eine Audiospur importiert werden, um die Choreografie auf die Musik abzustimmen. Viele der Ideen sind ähnlich zu unserer Arbeit. Choreografien bestehen ebenfalls aus einer Menge von Bildern, welche die Positionen und Blickrichtungen der Tänzer zu einem bestimmten Zeitpunkt zeigen. Auch können Blaupausen für Bilder definiert und wiederverwendet werden. Ein Beispielbild in ChoreoGraphics befindet sich in Abbildung 3.1. Kombiniert wird dies mit einer 3D-Ansicht, in der aus einzelnen Tanzschritten ganze Bewegungssequenzen zusammengesetzt werden können. Im Gegensatz zu den Ansätzen in unserer Arbeit, ist die Spezifikation der Bilder und vor allem der Bildentwicklung deklarativ. So werden Bilder über die Attribute Form, Dichte und Muster definiert, was dann wiederum die Aufteilung der Tänzer im Raum beeinflusst. Es ist ebenfalls möglich, Teile der Bilder zu spiegeln. Bildentwicklungen, in ChoreoGraphics Evolutionen genannt, werden durch die Eingabe von Start- und Randbedingungen vom System berechnet und nicht direkt vom Choreografen definiert. Damit soll die Exploration vieler unterschiedlicher Choreografien erlaubt werden. Dies unterscheidet sich zu unserer Arbeit, da bei uns sowohl die Bilder als auch die Bildentwicklung imperativ spezifiziert werden.

Die klassischsten Möglichkeiten zur Planung und Darstellung von Formationen bietet DanceStudio von Muhammad [Muh09], das in Abbildung 3.2 zu sehen ist. Die Tänzer können in einer 2D-Ansicht aus der Vogelperspektive angeordnet werden. Dies kann für einige Stellen der Choreografie gemacht

werden. Alle Zwischenbilder werden dann per linearer Interpolation anhand der Keyframes ermittelt. Komplexere Übergänge und eine Kollisionserkennung sind nicht implementiert. Des Weiteren schlägt das Programm für jede Anzahl an Tänzern ein vordefiniertes, statisches Bild vor. Es ist dabei jedoch nicht für Standard- und Lateinformationstanz ausgelegt. Tänzer können nicht in Paaren angeordnet werden und eine Choreografie kann bis zu 40 Tänzer beinhalten. Allerdings ist das Projekt zum jetzigen Zeitpunkt nur in einer veralteten Version verfügbar. Bei unerlaubten Aktionen werden keine Fehlermeldungen geworfen, das Drag-and-Drop von Tänzern nicht diskret über einem eingeblendeten Gitter und Choreografie können nicht gespeichert werden. Einige der Ideen in dem vorgestellten Prototypen sind zu Konzepten in unserer Arbeit trotzdem ähnlich: So existiert in beiden eine diskrete Zeitleiste bestehend aus Schlägen und die standardmäßige Interpolation zwischen Bildern ist linear.

### 3.3 Positionierung dieser Arbeit

Diese Arbeit präsentiert ein Werkzeug, mit welchem Bilder in Lateinformationschoreografien erstellt, analysiert und mit den Tänzern kommuniziert werden können. Es wird hauptsächlich auf der Idee einer zweidimensionalen Darstellung aufgebaut, welche durch Analysemöglichkeiten erweitert wird. Unsere Arbeit grenzt sich insofern von bisherigen Arbeiten ab, dass der Fokus auf der Planung von Bildern für Choreografien von Lateinformationen liegt. Des Weiteren fehlen den meisten Programmen Möglichkeiten, die erstellten Formationen zu analysieren. Auch wurde bisher kein Werkzeug als responsive Webanwendung entwickelt, sondern zumeist als Desktopanwendung. Eine Webanwendung ist insofern von Vorteil, dass keine aufwändige Installation eines Programms nötig ist und man die Webseite von einem beliebigem Gerät aus abrufen kann. Dazu zählen insbesondere auch Smartphones. Somit können Tänzer die Visualisierung auch während des Trainings benutzen.

Im Kontext der Sportvisualisierung erweitert unsere Arbeit bisherige Arbeiten mit Methodiken, die noch kaum angewandt wurden. In die Taxonomie von Perin et al. [PVS+18] lässt sich unser Ansatz nur schwer einordnen, da hier keine bestehenden Daten visualisiert, sondern von den Nutzern generiert werden, um die Sportler zu unterstützen. Page und Moere [PM06] formulieren zudem den Bedarf nach Visualisierungen, die nicht an Medien oder Zuschauer, sondern direkt an die Sportler selbst gerichtet sind. Es existieren nur wenige Beispiele für solche Ansätze, bisher vor allem im Zusammenhang mit Wearable Computing. Ein Beispiel ist ein System namens SensorHogu von Chi [Chi05], bei welchem piezoelektrische Sensoren in der Schutzkleidung von Kampfsportlern eingebaut wurden. Damit sollte es Richtern auf Wettkämpfen vereinfacht werden, die Kämpfe zu bewerten. Unsere Anwendung hingegen soll die Sportler direkt während des Trainings unterstützen.

Auffällig ist jedoch, dass sich die genannten Taxonomien, wie auch die meisten Arbeiten im Bereich der Sportvisualisierung, immer an populären Teamsportarten wie Fußball orientieren. Hier ist das Antreten zweier gegnerischer Mannschaften von zentraler Bedeutung. Zwar gibt es auch im Bereich des Lateinformationstanzes eine Wettbewerbsstruktur, jedoch treten hier nie Teams klassisch gegeneinander an, sondern die Leistungen der einzelnen Formationsgruppen werden immer separiert voneinander bewertet.





## 4 Methodik

Als nächstes soll die Methodik erläutert werden, nach welcher im Rahmen dieser Arbeit vorgegangen wurde. Wir folgen dabei dem nutzerzentrierten Ansatz von Sedlmair et al. [SMM12], der im weiteren Verlauf auch als Designstudie bezeichnet wird. Eine Designstudie ist ein Projekt, in welchem ein Szenario aus der echten Welt untersucht wird. Dazu wird ein Visualisierungswerkzeug zusammen mit Experten entworfen, entwickelt und evaluiert. Dieser Ansatz empfiehlt sich vor allem bei problemgetriebener Forschung. In unserem Fall ist die Problemdomäne der Lateinformationstanz.

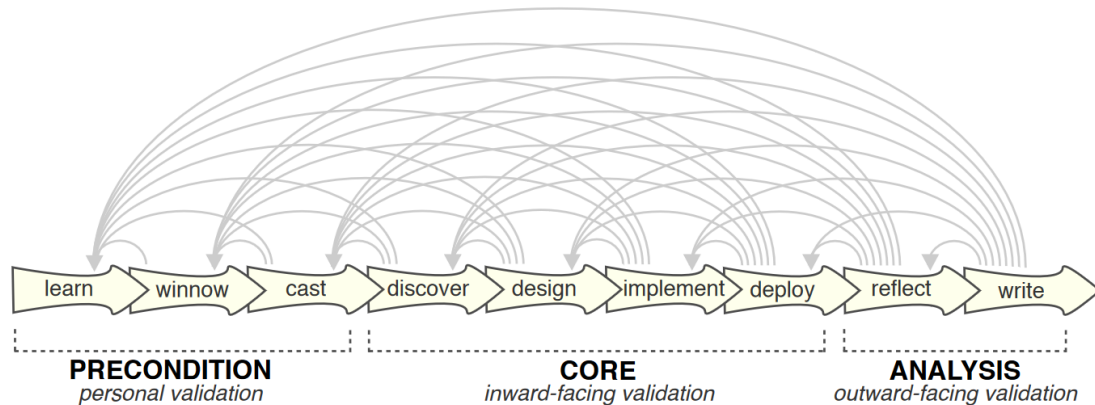
### 4.1 Designstudie: Vorgehensmodell

Sedlmair et al. [SMM12] nennen in ihrer Arbeit drei mögliche Beiträge als Resultat einer Designstudie: Eine Problemcharakterisierung, eine validierte Visualisierungslösung oder eine Reflektion über den Designprozess. Ein Bericht über die Designstudie muss nicht Beiträge in allen drei Bereichen liefern. Im Rahmen dieser Arbeit konzentrieren wir uns auf die ersten beiden Bereiche. Eine allgemeine Einführung in die Domäne befindet sich in Abschnitt 2.1. Die Anforderungen an die Visualisierung, die in Zusammenarbeit mit Experten erhoben wurden, befinden sich in Unterabschnitt 4.2.1. Des Weiteren wurde ein konkretes Visualisierungswerkzeug entwickelt, welches in Abschnitt 5.2 im Detail vorgestellt wird.

Sedlmair et al. [SMM12] schlagen in ihrer Arbeit ein konkretes Vorgehensmodell (vgl. Abbildung 4.1) vor, welches aus neuen Phasen besteht. Die Anordnung der Phasen ist sequentiell, da Artefakte vorheriger Phasen benötigt werden, jedoch können sich Phasen auch überlappen oder iterativ durchlaufen werden. Es werden im Folgenden für jede Phase die durchzuführenden Aktivitäten erklärt. Es wird zudem kurz beschrieben, wie die Phase von uns umgesetzt wurde.

In der *Learn* Phase geht es darum, sich ein grundlegendes Wissen über die existierende Visualisierungsforschung anzueignen. Dazu zählen unter anderem Designrichtlinien, Interaktionstechniken und das Wissen über Evaluationsmethoden. Da solches Wissen über die Zeit und mit steigender Anzahl an Projekten wächst, kann hier nur auf ein bereits abgeschlossenes Studienprojekt des Autors im Bereich der Visualisierung für Anfragesprachen für semantische Wissensgraphen [SSS22] verwiesen werden.

In der *Winnow* Phase geht es als Nächstes darum, Experten zu finden, mit welchen das Projekt durchgeführt wird. Sedlmair et al. [SMM12] schlagen vor, viele mögliche Partner in ersten Treffen kennen zu lernen und sich dann auf die Zusammenarbeit mit einer ausgewählten Menge an Personen zu konzentrieren, die adäquat erscheint. Ein häufiges Problem ist es, dass das Interesse der Experten am Projekt während der Durchführung sinkt oder das Bedürfnis nach der entwickelten Lösung nicht existiert. Die Auswahl der Experten wurde bereits vor Start des Projekts getroffen und konnte nicht geändert werden. Bei den Experten handelt es sich um Tänzer und Trainer einer Lateinformation mit jeweils mindestens zwölf Jahren Erfahrung. Einer der Experten war zudem einige Jahre Teil einer



**Abbildung 4.1:** Vorgehensmodell nach Sedlmair et al. [SMM12]. Die neun Phasen müssen hierbei nicht wasserfallartig abgearbeitet werden, was durch die grauen Pfeile verdeutlicht wird, welche alle Phasen miteinander verbinden.

Formation, die in der 1. Bundesliga und bei Deutschen Meisterschaften antrat. Die Experten trafen sich in diesem Zeitraum dreimal mit dem Autor dieser Arbeit, um über das Projekt zu diskutieren. Des Weiteren lag ein Beispieldatensatz zu Beginn des Projekts bereits als PDF und als JSON-Datei vor.

In der *Cast* Phase werden allen Projektteilnehmern verschiedene Rollen zugeordnet. Am Projekt beteiligt waren, neben dem Autor der Arbeit, zwei Betreuern und drei Domänenexperten. Sedlmair et al. [SMM12] haben hierfür bereits Rollen identifiziert. Die Domänenexperten sind dabei sogenannte Frontline-Analysten, welche das Visualisierungstool am Ende gewinnbringend nutzen sollen. Gatekeeper sind Personen mit der Befugnis das Projekt abubrechen. Dazu zählen in diesem Fall die Betreuer der Bachelorarbeit. Als Konnektoren werden Projektteilnehmer bezeichnet, welche Kontakt zwischen den Forschern und den Frontline-Analysten aufbauen. Dies war in unserem Projekt der Hauptbetreuer der Arbeit. Es existieren auch noch weitere Rollen, die jedoch auf keinen der Teilnehmer zutreffen.

In der *Discover* Phase geht es darum, sich Wissen über die Anwendungsdomäne anzueignen, Arbeitsabläufe und dabei auftretende Probleme zu verstehen und Anforderungen an das Visualisierungswerkzeug zu erheben, welches die ermittelten Probleme lösen soll. Ein erster Schritt dafür war das Lesen von Domänenliteratur. Für dieses Projekt handelt es sich hauptsächlich um die Informationen, die vom Deutschen Tanzsportverband e.V. veröffentlicht wurden. Als zweites wurde die sogenannte *fly-on-the-wall* Methode angewandt, bei der Forscher den normalen Ablauf in der Domäne beobachten, ohne die Prozesse zu unterbrechen. Dies geschah durch den Besuch eines Trainings der Tanzsportgruppe, welche die konsultierten Experten trainieren. Als Letztes folgte noch ein einstündiges Interview mit den Experten, welches dafür bestimmt war, konkrete Anforderungen zu erheben. Die Details dazu befinden sich in Unterabschnitt 4.2.1.

In einem nächsten Schritt, der *Design* Phase, werden, basierend auf den erhobenen Anforderungen, Ideen für Prototypen und deren Design entwickelt. Hier sollten sowohl die Datenabstraktionen, visuelle Kodierungen als auch mögliche Interaktionsmechanismen betrachtet werden. Auch dieser Prozess ist iterativ und sollte zu Beginn die Prüfung vieler möglicher Entwürfe enthalten. Für dieses Projekt wurde zuerst eine Papier-Mock-Up mit niedrigem Detailgrad (*low-fidelity*) erstellt. Dieser

wurde mit den Betreuern der Arbeit diskutiert und verfeinert. Es folgte dadurch eine Verfeinerung hin zu einem digitalen Mock-Up mit höherem Detailgrad (*high-fidelity*), der mit dem Online-Werkzeug Figma<sup>1</sup> erstellt wurde. Auch dieser Prototyp wurde wieder mit den Betreuern der Arbeit diskutiert. Zwar sollten auch solche frühen Artefakte bereits mit den Domänenexperten diskutiert werden, jedoch war dies aus zeitlichen Gründen nicht möglich.

Die eigentliche Programmierung des Visualisierungswerkzeugs folgt in der *Implement* Phase. Hierbei wurde nach den Grundsätzen des *rapid prototyping* vorgegangen. Es wurden Programmbestandteile entwickelt, in wöchentlichen Treffen mit den Betreuern diskutiert und je nachdem akzeptiert, überarbeitet und verfeinert oder verworfen. Ein Beispiel ist die visuelle Kodierung der Körper- und Blickrichtung der Tänzer, für die verschiedene Lösungsansätze implementiert und miteinander verglichen wurden. Der Forschungsprototyp wurde zudem regelmäßig unbeteiligten Testnutzern, meist Informatikstudenten, gegeben, um Fehler im Programm zu finden. In einer expliziten Testsitzung vor dem zweiten Interview wurden so beispielsweise 34 Fehler entdeckt und später behoben. Des Weiteren sollten auch die Domänenexperten in den Prozess der Implementierung miteinbezogen werden und während der Entwicklung noch Feedback geben. Dies wurde in Form eines zweiten Interviews umgesetzt, welches circa vier Monate nach Projektstart durchgeführt wurde. Der Prototyp wurde zwei der Experten zeitgleich vom Autor dieser Arbeit vorgeführt. Dabei konnten die Experten jederzeit Fragen stellen und Feedback äußern. Aufgrund der zeitlichen Positionierung des Interviews konnten nicht alle Kritikpunkte und Vorschläge implementiert werden. Das Interview wird in Unterabschnitt 4.2.2 näher beschreiben.

In der letzten Kernphase der Entwicklung, der *Deploy* Phase, wird die fertige Software nun ausgerollt und evaluiert, um herauszufinden, inwiefern die Visualisierung den Domänenexperten hilft die Probleme aus der *Discover* Phase anzugehen. Die Verteilung des Programms wurde mit Google Firebase realisiert, einem Dienst der in Unterabschnitt 5.1.4 kurz vorgestellt wird. Das Programm lässt sich über folgende URL abrufen: <https://botafigure.web.app/>. Sedlmair et al. [SMM12] schlagen für die Evaluation die Durchführung einer Feldstudie vor. Dies würde jedoch über den Rahmen dieser Arbeit hinausgehen, weswegen wir uns in dieser Arbeit auf eine *Think Aloud* Methode beschränkt haben (vgl. Unterabschnitt 4.2.3). Die Ergebnisse der Evaluation befinden sich in Kapitel 6.

Abschließend folgen noch die *Reflect* und die *Write* Phase. Während es sich bei zweiterer um das Schreiben der Arbeit handelt, besteht die *Reflect* Phase aus der Betrachtung des angewandten Prozesses. Das heißt Richtlinien sollten bestätigt, überarbeitet oder komplett verworfen werden. Auffällig in diesem Projekt war der relativ überschaubare Zeitrahmen von sechs Monaten. Für viele der Phasen schlagen Sedlmair et al. [SMM12] mehrere Monate an Zeit vor. Dies ist jedoch im Rahmen einer Abschlussarbeit nicht möglich. Des Weiteren sollte man beachten, dass die Terminfindung mit Experten oft schwierig ist. So kam es dazu, dass die eigentliche Programmierung, abgesehen von einigen Technikdemos und einem Prototypen für das Interview, erst zwei Monate nach Projektbeginn starten konnte. Auch das zweite Interview fand recht spät während der Entwicklung statt, weswegen einige Funktionen nicht mit vollem Umfang implementiert werden konnten. Es sollte deshalb auf eine frühe Terminfindung geachtet werden, die mindestens einen bis eineinhalb Monate vor dem eigentlichen Termin starten sollte.

---

<sup>1</sup> URL: <https://www.figma.com/de/>

### 4.2 Durchgeführte Designstudie

Im Folgenden werden die drei Interviews beschrieben, welche für unsere Designstudie durchgeführt wurden. Da es sich beim dritten Interview um die Evaluation handelt, wird in Unterabschnitt 4.2.3 nur die Theorie der eingesetzten Methodik geschildert. Details zur Durchführung der Evaluation und die Ergebnisse werden dann in Kapitel 6 ausführlich dargelegt.

#### 4.2.1 Erstes Interview: Anforderungsanalyse

Das erste Interview mit drei Domänenexperten fand circa eineinhalb Monate nach Projektbeginn statt. Ziel des Interviews war es ein tieferes Verständnis über die Anwendungsdomäne zu erlangen. Dies beinhaltet das Verständnis über den Prozess, wie Choreografien mitsamt ihrer Bilder entwickelt und trainiert werden. Insbesondere sollten neben positiven Aspekten des bereits existierenden Arbeitsablaufs Probleme aufgedeckt und Anforderungen an eine Visualisierungslösung erhoben werden.

Die Befragung der Experten fand in Form eines semistrukturierten Interviews statt, dessen Abschrift sich im Ergänzungsmaterial zu dieser Arbeit [Sch23] befindet. Es wurde eine Reihe von Fragen vorbereitet, welche sich an einem Leitfaden für semistrukturierte Interviews von Wood [Woo97] orientierten und in loser Reihenfolge abgearbeitet wurden. Nachdem die Experten nach ihrer Erfahrung im Bereich des Lateinformationssports befragt wurden, erklärten sie ihren bisherigen Prozess, um Choreografien zu entwickeln bzw. zu digitalisieren. Zuerst wurden die Bilder dafür von einem der Experten grob in ein Koordinatensystem eingezeichnet. Danach wurden die Bilder digitalisiert. Dafür verwenden die Experten bisher ein rudimentär entwickeltes Programm, welches keine Analysemöglichkeiten bietet und bei dem insbesondere der Funktionsumfang stark kritisiert wurde. Gelobt wurde hingegen eine gute Bedienbarkeit per Mehrfachauswahl von Tänzern und Drag-and-Drop zum Ändern der Position. Danach wurden Funktionen diskutiert, die das zu entwickelnde Programm haben sollte. Die Experten konnten ihre Vorschläge hierbei anhand einer frühen Technikdemo präsentieren und erklären.

Daraus wurden im Anschluss eine Reihe von Anforderungen definiert, die während der Entwicklung umgesetzt werden sollten. Die Anforderung wurden dabei entweder explizit von den Experten formuliert oder aus den impliziten Wünschen von uns abgeleitet. Alle Anforderungen sind in Tabelle 4.1 aufgelistet. Einige stammen dabei aus dem zweiten Interview, sind aber hier aus Vollständigkeitsgründen bereits aufgelistet. Die Anforderungen wurden nach der Erfassung klassifiziert und in sieben Kategorien eingeteilt. Alle Anforderungen, die erst nach dem zweiten Interview dazu kamen, sind in Unterabschnitt 4.2.2 explizit genannt. In der Tabelle wurde auch nicht berücksichtigt, dass das Programm auf mobilen Geräten korrekt angezeigt werden sollte, um eine Nutzung während des Trainings zu ermöglichen.

<b>A</b>	<b>Allgemeines</b>
A1	Neue Choreografie erstellen
A2	Choreografie speichern
A2	Choreografie laden
<b>B</b>	<b>Erstellung und Visualisierung von Bildern</b>
B1	Positionen der Tänzer anzeigen
B2	Neues Bild erstellen
B3	<i>Bild als Vorlage speichern</i>
B4	Bestehendes Bild duplizieren
B5	Vorheriges Bild auswählen
B6	Nachfolgendes Bild auswählen
B7	Beliebiges Bild auswählen
B8	Bild rotieren
B9	Tänzer verschieben
B10	Mehrere Tänzer verschieben
B11	Tänzerabstand skalieren
B12	<i>Geraden mit beliebigem Steigungswinkel rotieren</i>
B13	Paar trennen/vereinen
B14	<i>Alle Paare trennen/vereinen</i>
B15	Nachbarn zur Orientierung anzeigen
B16	Zeitliche Verordnung der Bilder anzeigen
B17	Bild löschen
B18	Notiz zum Bild hinzufügen
B19	Farbe der Tänzer ändern
B20	Gitterauflösung ändern
<b>C</b>	<b>Ausrichtung von Kopf und Körper</b>
C1	Ausrichtung von Kopf und Körper anzeigen
C2	Ausrichtung von Kopf und Körper ändern
C3	Ausrichtung von Kopf und Körper für mehrere Tänzer ändern
<b>D</b>	<b>Transitionen</b>
D1	Vorheriges Bild anzeigen
D2	Übergang von vorherigem Bild zu aktuellem Übergang anzeigen
D3	Übergang selbst als stückweise Gerade definieren
D4	Übergang zwischen Bildern animieren
<b>E</b>	<b>Prägnante Formen in Bildern</b>
E1	Prägnante Form einfügen
E2	Prägnante Form anzeigen
E3	Prägnante Form löschen
<b>F</b>	<b>Choreografieanalyse</b>
F1	Länge der Laufwege zwischen zwei Bildern anzeigen
F2	Gesamtlänge der Länge der Laufwege anzeigen
F3	Kollisionen bei Bildübergängen detektieren
F4	Ausnutzung der Tanzfläche anzeigen

<b>G</b>	<b>3D-Ansicht und Posen</b>
G1	Positionen der Tänzer in einer 3D-Ansicht zeigen
G2	Kopf- und Körperausrichtung in der 3D-Ansicht zeigen
G3	Pose für Tänzer definieren
G4	Pose für mehrere Tänzer definieren
G5	Definierte Pose speichern

**Tabelle 4.1:** Anforderungen an den Prototyp, welche aus Interviews mit Formationstrainern stammen. Die Anforderungen sind zur besseren Übersicht in Kategorien eingeteilt und werden bei der Vorstellung des Prototyps in Kapitel 5 über ihre ID in der ersten Spalte referenziert. Anforderungen aus dem zweiten Interview sind durch kursive Schrift hervorgehoben.

#### 4.2.2 Zweites Interview: Review

Das zweite Experteninterview wurde circa vier Monate nach Projektbeginn mit zwei der drei Experten des ersten Interviews durchgeführt. Ziel des Interviews war es, den bis dahin existierenden Prototypen zu präsentieren und Feedback für die restliche Entwicklung einzuholen. Das Interview war wie zuvor semistrukturiert aufgebaut und dauerte circa eine Stunde. Der Interviewleitende erklärte den Prototyp und führte durch alle implementierten Funktionen, während die Experten dies jederzeit unterbrechen konnten, um Fragen zu stellen oder Kritik beziehungsweise Lob am Prototyp zu äußern. Nach Abschluss eines Themengebiets wurden die Experten zusätzlich nochmals aufgefordert, ihre Gedanken zu explizit auszuformulieren. Generell äußerten sich die Experten dem Prototypen gegenüber positiv und freuten sich schon auf die Arbeit mit diesem. Kritik äußerten sie an der Sinnhaftigkeit des Einzeichnens prägnanter Formen in die Bilder, gaben dazu jedoch keine weiteren Verbesserungsvorschläge. Die Experten brachten jedoch weitere Vorschläge an, die in Tabelle 4.1 als Anforderung **B3**, **B12** und **B14** zu finden sind. Zudem gab es noch Kritik bei einigen kleineren Themen. Zum Beispiel sollte es möglich sein, den 3D-Modus auch auf mobilen Geräten anzeigen zu lassen und für die Transitionen sollten alternative Farbschemata angeboten werden. Da das Interview schon recht spät während der Implementierungsphase stattfand, konnten nicht mehr alle Vorschläge umgesetzt werden. Stattdessen wurden die Vorschläge nach fallender Relevanz sortiert und die niedriger Priorisierten wurden für eine spätere Implementierung vermerkt. Dazu zählt zum Beispiel eine Funktion, mit der bei der Erstellung von Posen beide Arme gleichzeitig rotiert werden können. Eine Abschrift des Interviews befindet sich ebenfalls wieder im Ergänzungsmaterial [Sch23].

#### 4.2.3 Drittes Interview: Evaluation

Zur Evaluation des entstandenen Prototyps wurde eine Laborstudie mit der sogenannten *Think Aloud* Methode angewandt. Es handelt sich hierbei um eine etablierte Methodik der qualitativen Forschung und wurde zum Beispiel von Stein et al. [SJL+18] für Visualisierungen genutzt. Die Anwendung des *Think Aloud* Protokolls ist dann nützlich, wenn die Visualisierung von einer kleinen Menge von Experten evaluiert wird, deren Zahl nicht für eine aussagekräftige quantitative Studie reicht. Wir orientieren uns im Folgenden bei der Beschreibung an der Arbeit von Boren und Ramey [BR00], welche die Durchführung von Studien mit der *Think Aloud* Methode unter Berücksichtigung von

Kenntnissen aus der Kommunikationstheorie anreichern. Die Grundidee ist, den Teilnehmern der Studie eine Reihe von Aufgaben zu geben, welche sie mit dem Forschungsprototyp lösen sollen. Währenddessen sollen die Teilnehmer nun ihre Gedanken laut verbalisieren, so als wären sie alleine im Raum. Falls die Teilnehmer 15-60 Sekunden stumm bleiben, fordert sie der Studiendurchführende auf, das Verbalisieren fortzusetzen. Ansonsten nimmt der Studiendurchführende eine passive Rolle ein. Der Anteil der Interaktion zwischen Studienteilnehmer und Durchführendem ist in der Literatur ein viel diskutiertes Thema. Boren und Ramey [BR00] empfehlen, die Interaktion auf ein mögliches Minimum zu reduzieren, da ein häufiger Wechsel zwischen Monolog und Dialog für die meisten Teilnehmer schwierig ist. Interaktion ist nur dann vorgesehen, wenn Teilnehmer Bestätigungssignale erwarten oder ein Eingriff im Prozess notwendig ist. Beispiele dafür sind Programmfehler oder der Fall, dass ein Teilnehmer die Aufgabe nicht beenden konnte oder wichtige Aspekte bei der Lösung der Aufgabe vernachlässigt hat. In letzterem Fall wird empfohlen, den Teilnehmer zu bestärken und zum Beispiel anzumerken, dass er einen wichtigen Beitrag geleistet hat, ein Problem zu identifizieren. Dem Teilnehmer sollte zu jedem Zeitpunkt klar sein, dass nicht er sondern der Forschungsprototyp das zu untersuchende Objekt ist. Des Weiteren wird empfohlen, das Verbalisieren der Gedanken in einer simplen Voraufgabe zu üben. Großen Einfluss auf die Qualität der Ergebnisse haben zudem die Aufgaben, welche die Teilnehmer lösen sollen. Charters [Cha03] beschreibt Eigenschaften für passende Aufgaben in einer *Think Aloud* Studie. Die Aufgaben sollten von mittlerer Schwierigkeit sein und mit sprachlichen Mitteln gelöst werden können, das heißt automatische Abläufe der Teilnehmer sollten vermieden werden. Aufgaben sollten zudem sinnvoll heruntergebrochen werden, um das Arbeitsgedächtnis der Teilnehmer nicht zu überladen. Hierbei kann es helfen, den Teilnehmer die Fragen zusätzlich in schriftlicher Form zu reichen. Wie genau wir die Methodik in unserer Evaluation genutzt haben, wird in Abschnitt 6.2 genauer erläutert.





## 5 Implementierung

Im Folgenden wird der Forschungsprototyp vorgestellt, der im Rahmen dieser Arbeit entstanden ist. In Abschnitt 5.1 wird auf eine Auswahl der Bibliotheken und Sprachen eingegangen, die zur Erstellung des Prototyps verwendet wurden. Danach wird der Prototyp in Abschnitt 5.2 detailliert anhand der Desktop-Anwendung vorgestellt. Auf die Unterschiede mit den mobilen Geräten wird in Abschnitt 5.3 eingegangen.

### 5.1 Verwendete Technologien

Der Prototyp wurde webbasiert als *single-page application* in TypeScript<sup>1</sup> implementiert. Einige Komponenten wurden von einem privaten Prototyp übernommen und erweitert. Im Folgenden werden nun die relevantesten verwendeten Bibliotheken vorgestellt.

#### 5.1.1 React

React [Rea23] ist eine open-source Bibliothek für die Erstellung von Nutzerinterfaces (UI) in JavaScript oder TypeScript. Alle Elemente des Interfaces werden in React in Komponenten gekapselt. Diese besitzen jeweils einen eigenen Zustand, von welchem die Darstellung abhängt. Änderungen des Interfaces finden nicht über die direkte Manipulation von Textfeldern oder Buttons statt, sondern über die Änderung des Zustands der Komponente. React kümmert sich automatisch um die Änderungen am Document Object Model der Webseite. Hierbei werden nur die Komponenten angepasst, welche vom geänderten Zustand abhängen. Interaktive UIs werden in React mit einer HTML-ähnlichen, deklarativen Syntax namens JSX (für JavaScript) oder TSX (für TypeScript) verfasst. React-Komponenten sind wie in HTML in einer Baumstruktur angeordnet, das heißt eine React-Komponente kann beliebig viele andere React-Komponenten beinhalten. Zudem ist es möglich, den eigenen Zustand mit den Kindkomponenten zu teilen. Listing 5.1 zeigt eine einfache React-Komponente, welche eine Liste von Texten generiert. Es handelt sich um eine funktionale Komponente, da die Funktion nur einen Parameter (props) hat. Alternativ können auch Klassen aus der ECMAScript 6 Spezifikation von JavaScript verwendet werden. Die UI wird dann ähnlich zu HTML aufgebaut. Weitere verwendete Bibliotheken aus dem React-Ökosystem sind *react-spring*<sup>2</sup> für die Erstellung von Animationen und *Immer.js*<sup>3</sup> für Zustandsupdates an der Choreografie.

---

<sup>1</sup> URL: <https://www.typescriptlang.org/>

<sup>2</sup> URL: <https://www.react-spring.dev/>

<sup>3</sup> URL: <https://immerjs.github.io/immer/>

### Listing 5.1 React Komponente für eine Liste

---

```
function SimpleList(props: SimpleListProps) {
  return (
    <ul>
      {props.listItems.map(item => (
        <li key={item.id}>item</li>
      ))}
    </ul>
  );
}
export default SimpleList;
```

---

#### 5.1.2 (React) Bootstrap

Bootstrap [Boo23] ist eine open-source Bibliothek zur Entwicklung von Weboberflächen, welche viele Komponenten wie Buttons oder modale Dialoge beinhaltet. Zudem besitzt Bootstrap ein Gitter-System, das die Entwicklung responsiver Webseiten im Gegensatz zur Nutzung von klassischem CSS und *media queries* vereinfacht. In React Bootstrap wurden alle vorgefertigten Bootstrap-Elemente in React-Komponenten umgebaut. Für dieses Projekt stammen alle UI-Elemente (bis auf die eigentliche Visualisierung und die Zeitleiste) aus React Bootstrap. In Listing 5.2 ist beispielhaft der Code für ein responsives Gitter gegeben. In einer `<Container>` Komponente wird dazu eine Menge an `<Row>` Komponenten definiert.

#### 5.1.3 Three.js

Bei Three.js [Thr23] handelt es sich um eine open-source JavaScript-Bibliothek zur Erstellung von 3D-Computergrafikinhalten. Dabei wird auf WebGL zurückgegriffen, um alle Berechnungen durch die Grafikkarte zu beschleunigen. Three.js hilft zum Beispiel bei der Beleuchtung der Szenen und beim Laden von 3D-Modellen. Die Bibliothek wurde in diesem Projekt für die 3D-Darstellung der Bilder und das Erstellen nutzer-definierter Posen eingesetzt. Alle 3D-Modelle stammen von der Webseite Mixamo [Mix23] und dürfen in privaten und kommerziellen Projekten eingesetzt werden.

#### 5.1.4 Firebase

Firebase [Fir23] ist ein von Google entwickeltes Werkzeug, welches eine Reihe an Dienstleistungen anbietet, mit welcher das Publizieren von Webseiten vereinfacht wird. Darunter zählen unter anderem das Hosting der Webseite, Datenbankanbindung, Nutzerauthentifizierung, automatische Benachrichtigungen und noch viele mehr. Firebase wurde für dieses Projekt eingesetzt, um die Visualisierung ohne Installationsschwierigkeiten im Rahmen der *Deploy* Phase an die Experten weiterzugeben.

**Listing 5.2** Responsives Gitter

```

function ResponsiveGrid() {
  return (
    <Container>
      <Row>
        <Col sm={2}>Toolbar</Col>
        <Col sm={10}>Inhalt</Col>
      </Row>
    </Container>
  );
}
export default Responsive Grid;

```

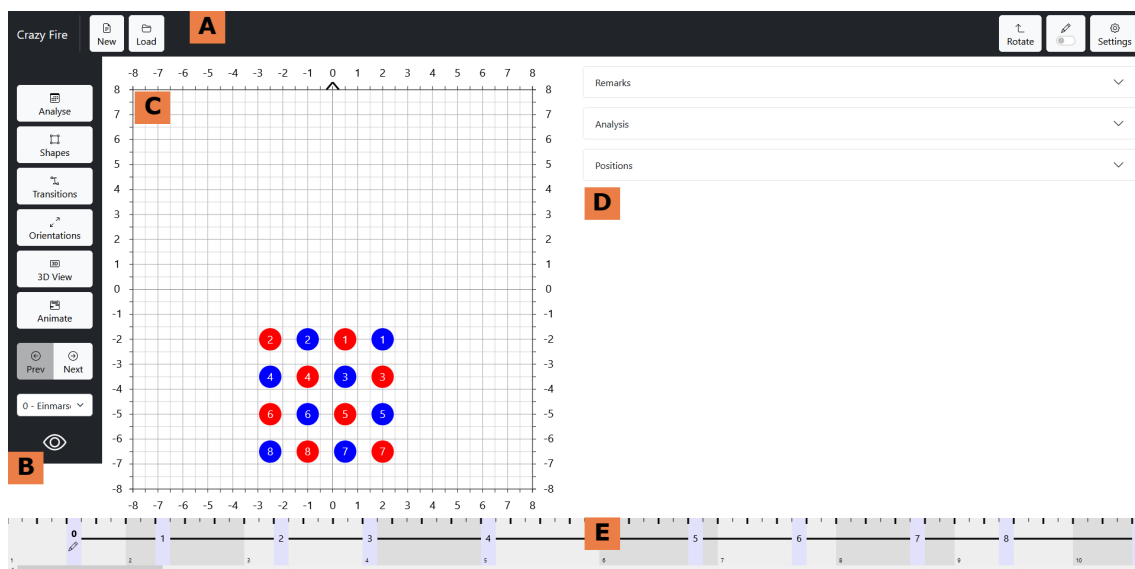
## 5.2 Prototyp

Grundsätzlich fiel bei der Anforderungsanalyse auf, dass es zwei Hauptinteressensgruppen gibt, nämlich die Trainer bzw. Choreografen einer Formationsgruppe und die Tänzer. Trainer bauen Choreografien, das heißt sie müssen die Bilder und die Bildentwicklung erstellen und potenzielle Änderungen vornehmen. Die Tänzer der Formation sind passive Akteure, das heißt sie betrachten die Bilder und die Bildentwicklung, die vom Trainer vorgegeben werden. Um eine Version für beide Interessensgruppen zu bauen, lässt sich das Programm in zwei Modi nutzen, dem Editiermodus und dem Ansichtsmodus. Während im Editiermodus Änderungen an der Choreografie möglich sind (zum Beispiel das Verschieben eines Tänzers in einem Bild), ist dies im Ansichtsmodus nicht möglich. Im Verlauf dieses Kapitels werden noch weitere Unterschiede aufgezeigt. Des Weiteren folgt nun eine detaillierte Vorstellung aller Funktionalitäten. Außerdem wird mithilfe der IDs aus Tabelle 4.1 aufgezeigt, welche Anforderung die beschriebene Funktion erfüllt.

In Abbildung 5.1 ist der Prototyp direkt nach dem Programmstart zu sehen. Standardmäßig startet das Programm im Ansichtsmodus. Bereich A beinhaltet eine Navigationsleiste, bei deren Design sich an bekannten Softwaresystemen wie Word oder Excel orientiert wurde, da Nutzer damit meist vertraut sind. Links in der Navigationsleiste wird der Name der Choreografie angezeigt. Im gezeigten Beispiel handelt es sich um die Choreografie „Crazy Fire“, die als Testdatensatz bei der Entwicklung diente. Daneben befindet sich ein Button, der einen modalen Dialog öffnet, mit dem eine neue Choreografie erstellt werden kann (A1, siehe Abbildung 5.2). Nachdem man einen Namen und optional eine Beschreibung für die neue Choreografie eingegeben hat, muss man auswählen, wie viele Paare die Bilder der neuen Choreografie enthalten. Ein Bild muss mindestens drei und darf höchstens acht Paare enthalten. Als letztes muss noch aus einer Liste vordefinierter Vorschläge das erste Bild der neuen Choreografie ausgewählt und ein Name für dieses Bild vergeben werden. Für den Standardfall von acht Tänzern werden folgende Bilder angeboten: Ein Rechteck, zwei horizontale Reihen, zwei vertikale Reihen, ein Pfeil mit einer Spitze und eine Doppeldiagonale. Im Beispiel wird eine Choreografie mit dem Namen „Demochoreografie“ und acht Paaren pro Bild erstellt. Das erste Bild in dieser Choreografie heißt „Demobild“ und beinhaltet eine Doppeldiagonale.

Neben dem Button zur Erstellung einer neuen Choreografie befindet sich der Button zum Laden einer bereits bestehenden Choreografie, welche als JSON-Dateien abgespeichert wird (A2). Momentan werden die Choreografien lokal auf dem eigenen System gespeichert, weswegen eine entsprechende Choreografie-datei im Dateisystem auszuwählen ist. Im Editiermodus gibt es neben dem Button zum

## 5 Implementierung



**Abbildung 5.1:** Übersicht aller Interfacebestandteile. Bereich A enthält eine Navigationsleiste zum Speichern und Laden von Choreografien. Bereich B enthält eine Werkzeugleiste, in welcher verschiedene Visualisierungsmodi aktiviert werden können. Bereich C enthält die Visualisierung der Tanzfläche und der Positionen der Tänzer im Bild. Bereich D enthält mehrere Akkordeons, in welchen unter anderem die Positionen der Tänzer in einer Tabelle abgelesen werden und Notizen zum Bild vermerkt werden können. Bereich E enthält eine Zeitleiste, welche in Takte und Schläge aufgeteilt ist und in der die Bilder anhand ihrer ID eingetragen sind.

Laden einer Choreografie noch einen weiteren Button zum Speichern der Choreografie. Drückt man diesen, dann lädt der Browser die bearbeitete Version der geladenen Choreografie als JSON-Datei in den Download-Bereich des Betriebssystems (A3). Auf der rechten Seite der Navigationsleiste befinden sich ebenfalls drei Buttons. Mit dem „Rotate“-Button kann die Ansicht des aktuell ausgewählten Bildes um jeweils 90° gegen den Uhrzeigersinn rotiert werden (B8), um das Bild aus unterschiedlichen Richtungen betrachten zu können. Während die Tänzer mit Blick nach vorne stehen (z.B. zur Tribüne der Wertungsrichter), sitzt der Trainer im Training oft in Blickrichtung der Tänzer. Er braucht somit eine gespiegelte Ansicht. Aufgrund des Feedbacks der Domänenexperten wurden vier verschiedene Ansichten eingebaut, um auf potenzielle Blickrichtungen seitens der Trainer reagieren zu können. Die Sinnhaftigkeit dessen wurde von den Domänenexperten bestätigt. Die Funktion zum Rotieren der Bilder ist nur im Ansichtsmodus verfügbar, um zu vermeiden, dass die Choreografie fälschlicherweise mit einer falschen Bildrotation gebaut wird. Über den Button mit dem Stiftsymbol kann zwischen dem Ansichtsmodus und dem Editiermodus gewechselt werden. Bei jedem Wechsel zwischen den Modi wird man in einem modalen Dialog nochmal auf den neuen Modus hingewiesen. Dieser Dialog verschwindet automatisch wieder. Der Button ganz rechts ist für das Öffnen der Einstellungen gedacht. Auch hierfür öffnet sich wieder ein modaler Dialog. Dort können zum Beispiel der Titel und die Beschreibung der Choreografie verändert werden.

Der Bereich B in Abbildung 5.1 beinhaltet eine vertikale Werkzeugleiste. Ein Problem bei der Visualisierung von Bildern ist die Menge an Information, die den Tänzern vermittelt werden muss. Dazu gehören unter anderem die Position von bis zu 16 Tänzern, die Ausrichtung von Kopf und

Create new choreo
×

---

**Specify your new choreo**

Name

Description

Dies ist eine Choreografie für Demonstrationszwecke

Number of couples

8

**Specify your first pattern**

Name

First pattern

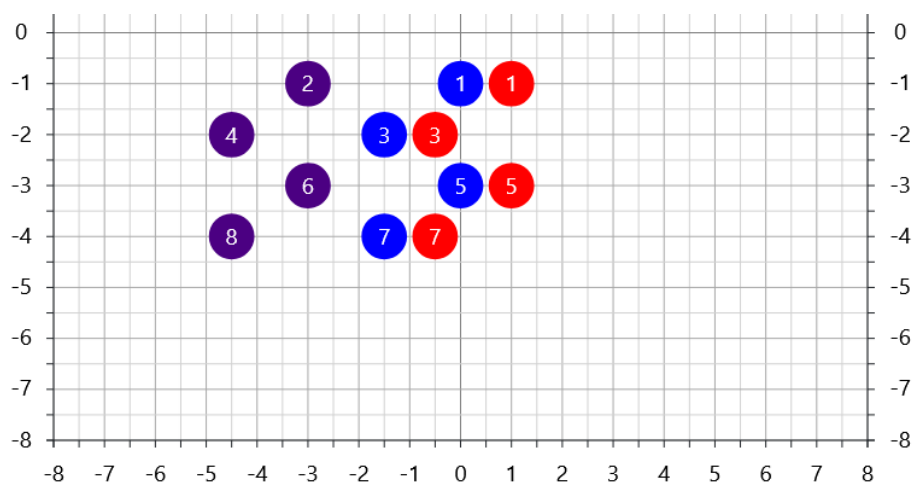
Double Diagonal
▼

Close
Create

**Abbildung 5.2:** Modaler Dialog zur Erstellung einer neuen Choreografie. Es müssen dazu mindestens eine Name für die Choreografie, die Anzahl der Tänzer und ein erstes Bild spezifiziert werden.

Körper oder die eingenommen Pose. Um visuellen *Clutter* zu vermeiden, existieren für das Programm verschiedene Modi, um Informationen nur bei Bedarf anzuzeigen. Wie bereits in Unterabschnitt 4.2.1 erwähnt, wurden alle Anforderungen klassifiziert und in eine von sieben Kategorien eingeteilt. Diese bilden die Grundlage für die verschiedenen Modi, die in der vertikalen Symbolleiste über die Buttons angewählt werden können. Bei den Modi handelt es sich um „Analyse“, „Prägnante Formen“, „Transitionen“, „Orientierungen“ und „3D-Ansicht“. Die genannten Modi können über die ersten fünf Buttons von oben ausgewählt werden, welche wie Schalter gebaut sind. Wird beispielsweise „Transitions“ ausgewählt, ist der Modus so lange aktiv (sichtbar durch die dunkle Hinterlegung des Buttons), bis dieser oder ein anderer Modus ausgewählt wird. Auf die einzelnen Modi, sowie den Button mit der Beschriftung „Animate“, der im Editiermodus durch einen Button mit der Beschriftung „New“ ersetzt wird, wird später noch im Detail eingegangen. Zusätzlich existieren in der Werkzeugleiste noch die Buttons mit den Beschriftungen „Prev“ und „Next“, mit denen jeweils das vorherige Bild (**B5**) oder das nachfolgende Bild (**B6**) der Choreografie, falls sie denn existieren, ausgewählt werden können. In der Drop-Down-Liste darunter kann ein beliebiges Bild aus der Choreografie anhand seines Namens ausgewählt werden (**B4**).

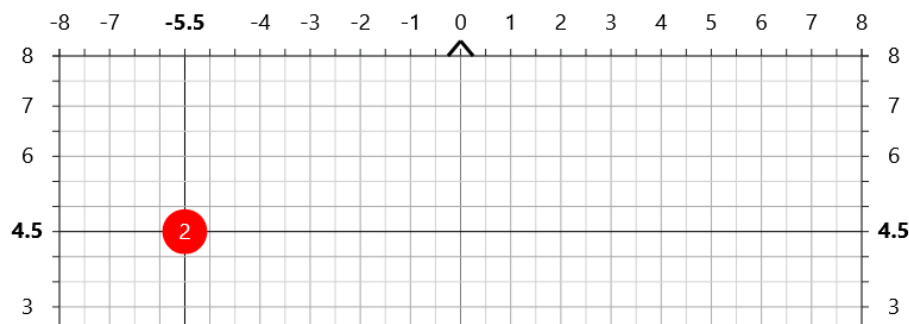
Der Bereich C aus Abbildung 5.1 beinhaltet die eigentliche Visualisierung der Bilder. Die Positionen werden in ein zweidimensionales, kartesisches Koordinatensystem eingezeichnet, analog zum Papierprozess. Die Skalen der beiden Achsen reichen von -8 bis 8. Die Tänzer sind in Form von



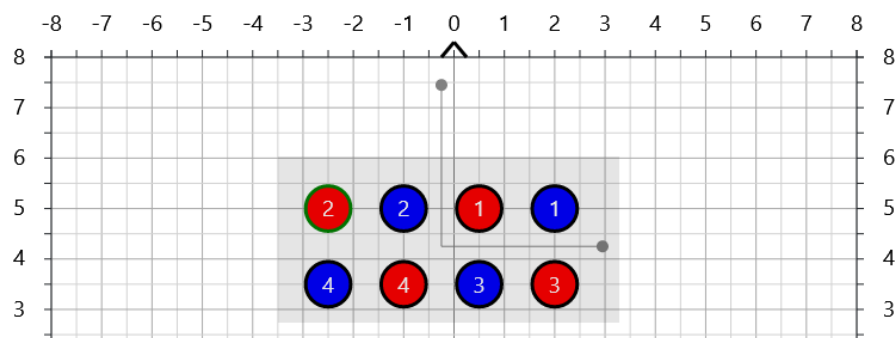
**Abbildung 5.3:** Beispiel für Paare, die entweder getrennt (Herren sind blau dargestellt, Damen in rot) oder zusammen (lila) tanzen.

Kreisen in die Visualisierung eingetragen (**B1**). Die Nummer innerhalb des Kreises entspricht indes der Nummer des Paares. Die Anzahl der Paare und somit auch der Tänzer in der Visualisierung lässt sich, wie bereits erwähnt, bei der Choreografieerstellung einstellen. Positionen können in beide Richtungen nur in Schritten von einem halben Meter eingenommen werden. Eine genauere Spezifizierung der Bilder ist nicht notwendig, da dies laut den befragten Experten nicht umsetzbar ist. Die Kreise der Herren sind blau gefärbt, während die Kreise der Damen rot gefärbt sind. Die Auswahl der vordefinierten Farbe soll eine gute Unterscheidbarkeit gewähren und zielt nicht auf eine geschlechtsspezifische Konnotation ab. Des Weiteren kann die Farbe der Herren und Damen in den Einstellungen frei gewählt werden (**B19**). Wie bereits in Unterabschnitt 2.1.1 erklärt, können die Damen und Herren der Paare sowohl getrennt als auch zusammen tanzen. Paare, die zusammen tanzen, sind standardmäßig lila eingefärbt. Jedoch kann auch diese Standardfarbe in den Einstellungen jederzeit geändert werden. Ein Beispiel dafür ist in Abbildung 5.3 zu sehen. Dort tanzen die Herren und die Damen in den Paaren mit ungerader Nummerierung getrennt und in den Paaren mit gerader Nummerierung zusammen. In das Koordinatensystem ist zudem zur besseren Lesbarkeit der Positionen der Tänzer ein Gitter aus hellgrauen Linien eingefügt. Die Auflösung des Gitters kann in den Einstellungen von einem halben Meter in der Domäne (also auf der Tanzfläche) zu einem Meter verändert werden (**B20**). Der schwarze Pfeil in der Visualisierung an der Position (0, 8) stellt die Richtung dar, welche die Tänzer als von ihnen aus gesehen „vorne“ bezeichnen würden. Damit wird sichergestellt, dass die Nutzer zu jedem Zeitpunkt wissen, welche Ansicht auf die Tanzfläche momentan ausgewählt ist.

Das Interaktionsdesign übernimmt bekannte Metaphern, um die Usability zu erhöhen. Die Tänzer können hierbei als Pendant zu Desktop-Icons angesehen werden. Im Editiermodus kann die Position eines Tänzers auf der Tanzfläche per Drag-and-Drop verändert werden (**B9**). Um die Positionsfindung während des Drag-and-Drops zu unterstützen, wird die Achsenbeschriftung für die aktuelle Position an den Achsen links, rechts, oben und unten fett hervorgehoben und benachbarte Beschriftungen bei Bedarf temporär ausgeblendet. Zudem werden die Gitterlinien, die sich an der Position des Tänzers kreuzen, schwarz eingefärbt, um es dem Nutzer zusätzlich zu vereinfachen die Position des ausgewählten Tänzers mit den Positionen der anderen Tänzer zu vergleichen. Somit kann man insbesondere prüfen, ob Tänzer zusammen auf einer horizontalen oder vertikalen Gerade



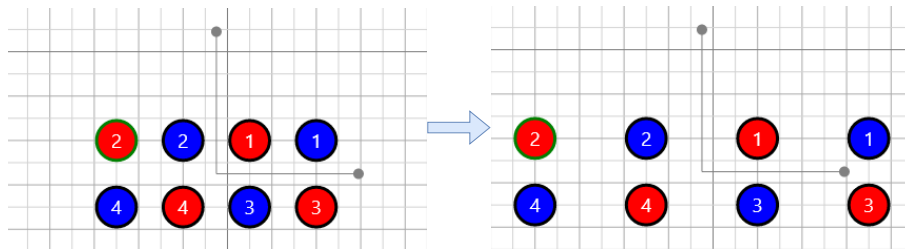
**Abbildung 5.4:** Dame 2 wird per Drag-and-Drop an die neue Position  $(-5.5, 4.5)$  verschoben.



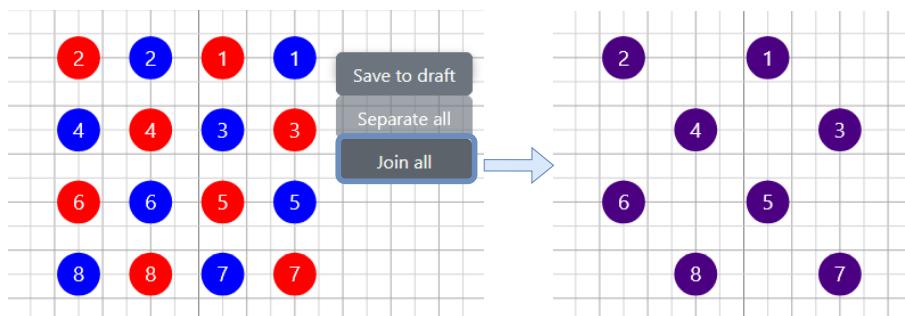
**Abbildung 5.5:** Tänzer der Paare 1 bis 4 wurden mit dem Brush selektiert.

stehen. Aufgrund der Häufigkeit von Geraden soll diese Funktion die Bilderstellung vereinfachen. Verschiebt man ein Paar, welches im Moment zusammen tanzt, dann werden die Positionen von Dame und Herr beide angepasst. Während des Drag-and-Drops wird die Position des verschobenen Tänzers jeweils immer auf die nächstliegende Gitterposition im 0.5-Meter-Abstand gerundet. Möchte man lieber das klassische Drag-and-Drop der Desktopsysteme haben, bei dem während des Verschiebevorgangs noch keine feste Gitterposition eingenommen und diese erst nach dem Loslassen der Maustaste berechnet wird, so kann dies in den Einstellungen umgestellt werden. Jedoch werden dann die Hilfslinien und hervorgehobenen Achsenbeschriftungen nicht mehr für die aktuell richtige Stelle angezeigt, sondern erst für die nächste Position im Gitter. Deswegen ist diese Einstellung standardmäßig nicht aktiviert.

Wie auf dem Desktop auch, können mehrere Tänzer mit einem rechteckigen Brush ausgewählt werden. In Abbildung 5.5 sind zum Beispiel alle Tänzer aus den Paaren eins bis vier durch den Brush ausgewählt. Visuell wird dies durch die schwarze Umrandung der Tänzer markiert. Ein Spezialfall stellt die Auswahl des ersten Tänzers da. Dieser wird grün umrandet, wie in unserem Beispiel die Dame 2. Auf die Besonderheit des grün umrandeten Tänzers wird später noch eingegangen. Möchte man weitere Tänzer selektieren ohne die Auswahl zuvor aufzuheben, dann lässt sich dies, analog zum Desktop machen, indem man einen Tänzer mit der Maus auswählt und dabei die Steuerungstaste (Strg/Ctrl) gedrückt hält. Wählt man mit dieser Methode einen bereits selektierten Tänzer aus, wird er aus der Menge selektierter Tänzer entfernt. Wird nun einer der selektierten Tänzer per Drag-and-Drop verschoben, werden auch alle anderen selektierten Tänzer relativ dazu verschoben (**B10**). Die Hilfslinien und die Hervorhebung der Achsenbeschriftung gibt es jedoch weiterhin nur für den Tänzer, der mit der Maus auch tatsächlich verschoben wird.



**Abbildung 5.6:** Links: Die Menge selektierter Tänzer vor dem Skalieren. Rechts: Ergebnis nach horizontalem Skalieren, indem die Maus um eine Gitterbreite nach rechts verschoben wurde.

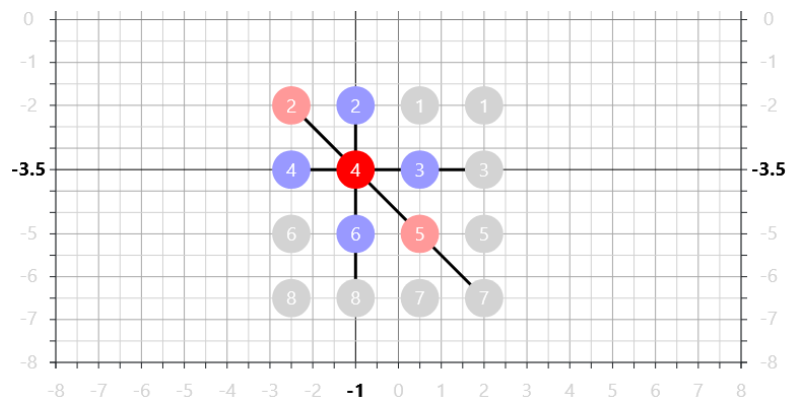


**Abbildung 5.7:** Links: Initialer Zustand der Choreografie. Per Rechtsklick auf den Hintergrund wurde das Kontextmenü geöffnet. „Join All“ wird ausgewählt. Rechts: Ergebnis der Operation.

Hat man mindestens zwei Tänzer selektiert, dann wird in der Mitte zwischen allen selektierten Tänzern eine Skalierungskomponente eingeblendet, die ähnlich wie in Spieleengines oder 3D-Modellierungsprogrammen funktioniert (**B11**). In Abbildung 5.6 ist dies dargestellt. Dort sind links in der Abbildung die Tänzer der Paare 1 bis 4 wieder ausgewählt. Nun wird das Ende der rechten Linie der Skalierungskomponente mit der Maus ausgewählt und um die Breite des Abstands zwischen zwei Gitterzellen nach rechts verschoben. Daraufhin werden alle Tänzer von der gedachten Mitte um den jeweiligen Abstand zu ihr weggeschoben. Die vertikale Skalierung funktioniert analog mit dem Ende der vertikalen Linie der Skalierungskomponente. Da während der Berechnung die Positionen der Tänzer auf die Gitterpunkte gerundet werden, kommt es bei weiteren Verschiebungen bisher noch zu Ungenauigkeiten. Diesbezüglich könnte in Zukunft untersucht werden, ob mithilfe von formbasierten Ansätzen weniger Ungenauigkeiten auftreten.

Mit einem Rechtsklick lässt sich ein Kontextmenü öffnen. Ist die Maus hierbei auf einem Tänzer, dann erscheint ein Button, mit dem man den Tänzer mit seinem Partner verbinden oder trennen kann. Dies ist davon abhängig, ob der Tänzer unter dem Mauscursor momentan von seinem Partner getrennt ist oder nicht (**B13**). Öffnet man das Kontextmenü mit einem Rechtsklick auf den Hintergrund, dann werden drei Buttons eingeblendet. Mit den Buttons mit der Beschriftung „Join All“ und „Separate All“ kann man alle Paare auf einmal trennen oder miteinander verbinden (**B14**). Der Ablauf, um alle Paare miteinander zu verbinden, ist in Abbildung 5.7 dargestellt. Mit dem Button mit der Beschriftung „Save as draft“ kann das Bild als Blaupause gespeichert und bei der Erstellung neuer Bilder später wiederverwendet werden (**B3**).





**Abbildung 5.8:** Alle relevanten Nachbarn zur Orientierung von Dame 4 werden mit einer weniger gesättigten Farbe angezeigt. Dazu wird die Dame 4 mit dem Mauscursor ausgewählt. Alle Tänzer, die keine relevanten Nachbarn sind, werden hellgrau gezeichnet, um visuell in den Hintergrund zu rücken.

Alle bisher genannten Funktionen, mit der Änderungen an der Choreografie vorgenommen werden können, sind im Ansichtsmodus nicht vorhanden. Da der Ansichtsmodus vor allem für die Tänzer gedacht ist, wurde eine Funktion zur Vereinfachung der Orientierung innerhalb eines Bildes eingefügt. Während dem Tanzen achten Tänzer nicht nur auf ihre exakte Position, sondern auch auf das Verhältnis zur restlichen Formation. Es würde den ästhetischen Eindruck mindern, wenn ein Tänzer an der richtigen Stelle steht, während der Rest der Formation um 20 Zentimeter von der Idealposition verschoben dasteht. Deswegen muss sich der Tänzer sauber im Bild einordnen, auch wenn dies unter Umständen nicht exakt gestellt wird. In den meisten Fällen orientieren Tänzer sich an den nächststehenden Tänzern, also an allen direkt neben oder vor ihnen. In manchen Fällen bietet sich auch noch der Tänzer dahinter zur Orientierung an. Des Weiteren erwähnten die Domänenexperten bei den Interviews immer wieder, dass vor allem Diagonalen Probleme machen, da sie bei der Ausführung oft in Kurven resultieren. Bewegt man nun im Ansichtsmodus den Mauscursor über einen Tänzer, dann werden er und seine Nachbarn farblich hervorgehoben (**B15**). Dies soll anhand Abbildung 5.8 verdeutlicht werden. Dort wurde die Dame 4 mit der Maus markiert und wird daher als einzige mit voller Farbsättigung dargestellt. Alle relevanten Nachbar (dazu gehören die direkten Nachbarn links, rechts, vor und hinter dem Tänzer sowie auf den erkannten Diagonalen) werden farblich leicht entsättigt dargestellt. Die restlichen Tänzer werden hellgrau eingefärbt, um visuell in den Hintergrund zu treten. Die Verbindungen zu den Nachbarn, sowie die komplette Diagonale werden über eine schwarze Linien hervorgehoben. Zur Berechnung relevanter Diagonalen wird der Algorithmus aus Unterabschnitt 2.2.3 verwendet. Damit eine Diagonale als solche erkannt wird, müssen sich mindestens vier Tänzer auf ihr befinden, da in den Bildern sonst zu viele falsch-positive Diagonalen erkannt würden. Alternativ könnte man die Choreografen relevante Diagonalen selbst einzeichnen lassen, jedoch ist dies bei Choreografien mit vielen Bildern mit einem erheblichen Mehraufwand verbunden. Daher haben wir uns in dieser Arbeit für eine automatisierte Lösung entschieden.

Um der Choreografie ein neues Bild hinzuzufügen, muss man im Editiermodus auf den Button mit der Beschriftung „New“ in der Werkzeugleiste drücken (**B2**). Es wird daraufhin ein modaler Dialog geöffnet, der in Abbildung 5.9 zu sehen ist. Zuerst muss der Name des neuen Bildes und optional eine Beschreibung eingegeben werden, genauso wie bei einer neuen Choreografie. Danach folgt noch

Create new pattern
×

---

Name

First remarks

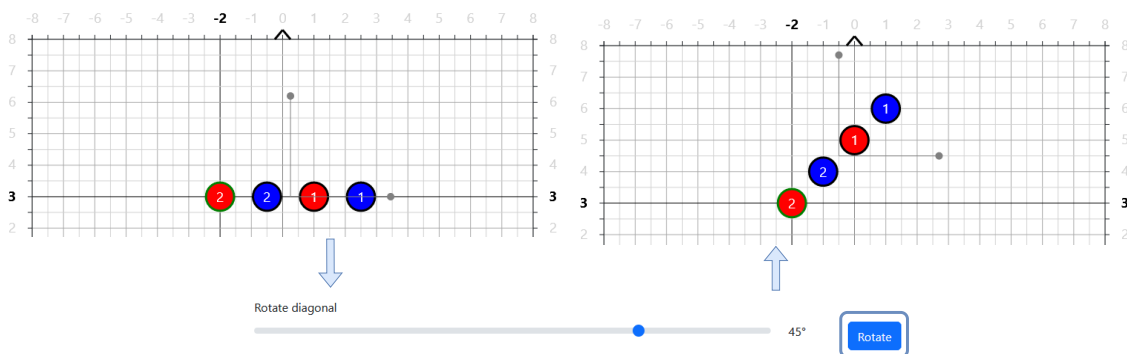
Es handelt sich um ein zur Choreographie neu hinzugefügtes Bild. Es wird dafür eine bereits vorher definierte Blaupause verwendet

Duplicate existing pattern  
 Use standard pattern  
 Use draft pattern

Demoblaupause
▼

Create

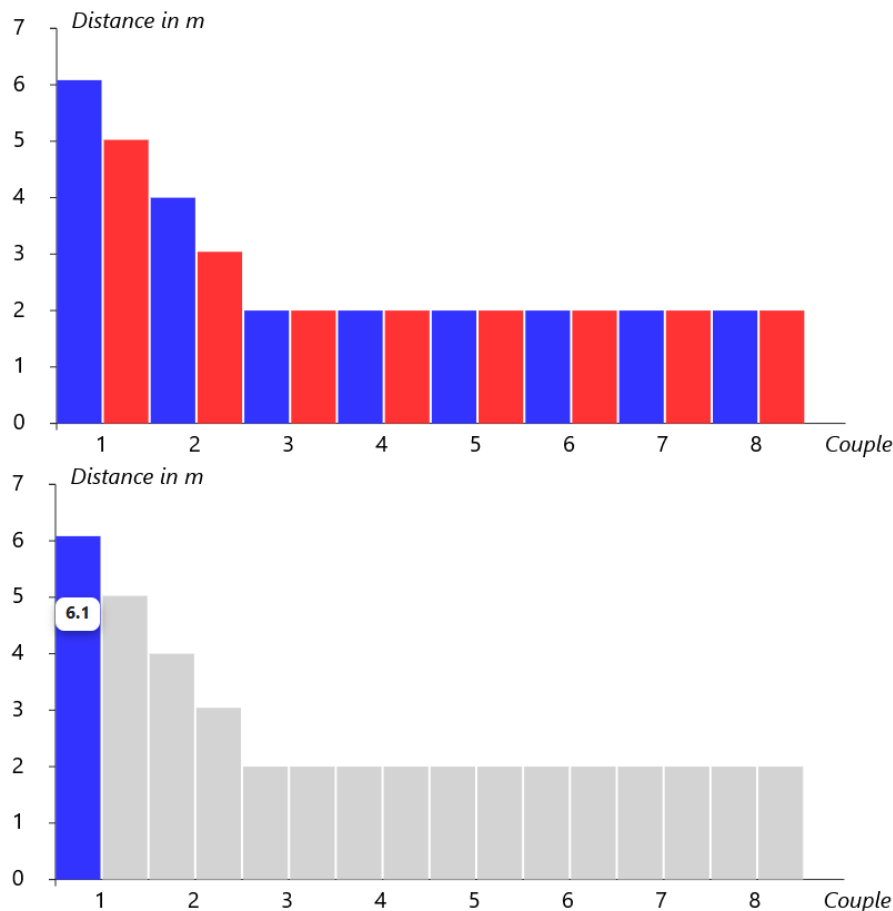
**Abbildung 5.9:** Modaler Dialog zur Erstellung eines neuen Bildes für die Choreografie. Dazu müssen mindestens ein neuer Name für das Bild und die Anfangspositionen der Tänzer ausgewählt werden. Bei letzterem kann man entweder ein bisher bestehendes Bild duplizieren oder ein Standardbild bzw. eine selbst definierte Blaupause verwenden.



**Abbildung 5.10:** Beispiel für die Rotation von Reihen von Tänzern. Hier wird die ausgewählte Reihe um 45° gegen den Uhrzeigersinn um Dame 2 herum rotiert.

die Entscheidung, ob man ein bereits existierendes Bild aus der aktuell geöffneten Choreografie, ein Standardbild, oder eine eigene Blaupause duplizieren will. Im Beispiel wird die bereits vom Nutzer eingefügte Blaupause mit dem Namen „Demoblaupause“ verwendet. Die Standardbilder sind dieselben, die man beim Erstellen der ersten Bildes einer neuen Choreografie auswählen kann.

Eine Funktion, die von den Experten während des zweiten Interviews gefordert wurde, ist die Erstellung von diagonalen Reihen mit vorgegebener Steigung (**B12**). Dazu muss man mit dem Brush oder mit Steuerung+Mausklick eine Menge von Tänzern auswählen, die alle kollinear zueinander stehen. Dies ist eine Bedingung für die Rotation einer Reihe von Tänzern. Als zweite Voraussetzung



**Abbildung 5.11:** Oben: Balkendiagramm zur Darstellung der Laufdistanz von Tänzern zwischen dem aktuell dargestellten und vorherigen Bild. Unten: Balkendiagramm, wenn Herr 1 in der Visualisierung mit der Maus ausgewählt wurde.

muss der als erstes ausgewählte, grün markierte Tänzer derjenige mit dem niedrigsten Wert auf der x-Achse sein, das heißt er muss am weitesten links stehen. Dann kann über einen Schieberegler, welcher in Bereich D aus Abbildung 5.1 erscheint, die neue Steigung der Reihe an Tänzern ausgehend vom grün markierten Tänzer ausgewählt werden. Für die Steigung können Werte zwischen  $-90^\circ$  und  $90^\circ$  in  $5^\circ$ -Schritten ausgewählt werden. Positive Werte sorgen hierbei für eine Rotation gegen den Uhrzeigersinn und negative Werte für eine Rotation im Uhrzeigersinn. Das Programm versucht, die Länge der ursprünglichen Reihe beizubehalten. Ein beispielhafter Ablauf ist in Abbildung 5.10 zu sehen. Das Berechnen einer Geraden mit einer beliebigen Steigung ist nicht immer perfekt möglich, da die Tänzer weiterhin nur auf den Gitterpositionen stehen dürfen. Da die Funktion erst im zweiten Interview angefordert wurde, konnte sie bisher nur in einfacher Form implementiert werden. In zukünftigen Iterationen könnte eine Erweiterung hinzugefügt werden, um Strukturen um einen beliebig ausgewählten Tänzer zu rotieren.

In Bereich D aus Abbildung 5.1 befinden sich drei Akkordeons mit Zusatzinformationen zur Choreografie. Im ersten Akkordeon „Remarks“ befinden sich die Notizen zum aktuell angezeigten Bild (**B18**). Mit einem Mausklick auf den Inhalt des Akkordeons öffnet sich ein modaler Dialog, in welchem die Notiz in einem Textfeld editiert werden kann. Das Akkordeon darunter ist zur



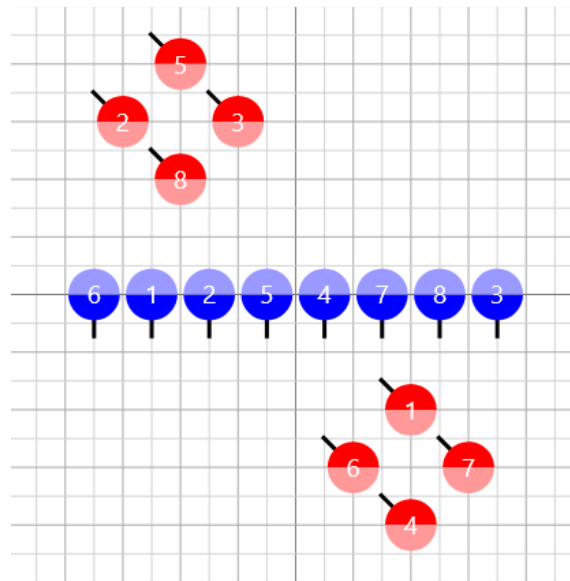
**Abbildung 5.12:** Ausschnitt der ersten vier Takte in der Zeitleiste. Beim Hovern über ein Bild wird eine Miniaturvorschau angezeigt.

Analyse der Weglänge der Übergänge zwischen dem aktuellen Bild und dem vorherigen Bild eingebaut. In einem Balkendiagramm ist für jeden Tänzer die Laufdistanz in Metern in Form eines Balkens eingetragen (**F1**). Die Farbe des Balkens entspricht der Farbe des Tänzers in der Visualisierung. Bewegt man den Mauszeiger über einen der Tänzer in der Visualisierung, dann wird im Balkendiagramm nur der Balken des markierten Tänzers farblich dargestellt. Zudem wird in einem Pop-Up-Fenster die Laufdistanz in Metern, gerundet auf die erste Nachkommastelle, angezeigt. Eine Ausnahme bildet das erste Bild einer Choreografie. Da hier kein vorangegangenes Bild existiert, wird das Balkendiagramm nicht angezeigt. Das Balkendiagramm ist in beiden Formen in Abbildung 5.11 zu sehen. Im dritten Akkordeon werden alle Positionen der Tänzer nochmals in Form einer Tabelle präsentiert.

Der Bereich E aus Abbildung 5.1 beinhaltet die Zeitleiste, in der die temporale Verordnung der Bilder visualisiert ist (**B16**). Die Zeitleiste ist durch zwei alternierende Graufarben in Takte unterteilt. Jeder Takt beinhaltet acht Schläge, welche die schwarzen Striche am oberen Rand der Zeitleiste darstellen. Die Nummer des Taktes steht links unten als Text. Die Bilder sind, repräsentiert durch ihre jeweilige ID, an einem bestimmten Taktschlag eingetragen, der zur besseren Sichtbarkeit hellblau unterlegt ist. Bewegt man den Mauszeiger über einen Schlag, so wird dieser dunkelgrau hervorgehoben. Ist für diesen Schlag ein Bild verfügbar, zeigt ein Pop-Up-Fenster dessen Vorschau. Dies wird in Abbildung 5.12 nochmals veranschaulicht. Klickt man mit der Maus auf diesen Schlag, wechselt das aktive Bild und somit auch die Visualisierung und die Daten der Akkordeons in den Bereichen C und D. Das momentan visualisierte Bild ist in der Zeitleiste durch ein Stiftsymbol hervorgehoben. Wählt man ein Bild mit einer der anderen Möglichkeiten an, zum Beispiel dem Drop-Down-Menü in der Werkzeugleiste, dann wird die Zeitleiste verschoben, sodass das Bild in dieser an zentraler Position zu sehen ist. Drückt man auf einen bereits ausgewählten Schlag, der ein Bild enthält, öffnet sich ein modaler Dialog. In diesem kann die temporale Verordnung, also Takt und Schlag, zu dem das Bild gestellt sein muss, über Schieberegler angepasst werden. Die Verschiebung wird dabei durch das vorherige und das nachfolgende Bild begrenzt. Das heißt Bild 0 kann nicht so verschoben werden, dass es zeitlich erst nach Bild 1 liegt. Zudem existiert in diesem Dialog noch ein Button, mit welchem das gewählte Bild gelöscht werden kann. Bevor das Bild endgültig gelöscht wird, wird mittels einer Warnung noch nach einer Bestätigung gefragt. Im Folgenden werden nun die fünf Visualisierungsmodi vorgestellt, mit welchen die Art der dargestellten Informationen in der Visualisierung angepasst werden können. Alle Modi lassen sich, wie bereits erwähnt, über die Werkzeugleiste am linken Bildschirmrand toggeln und schließen sich gegenseitig aus.

### 5.2.1 Modus: Orientierungen (Anforderungsbereich C)

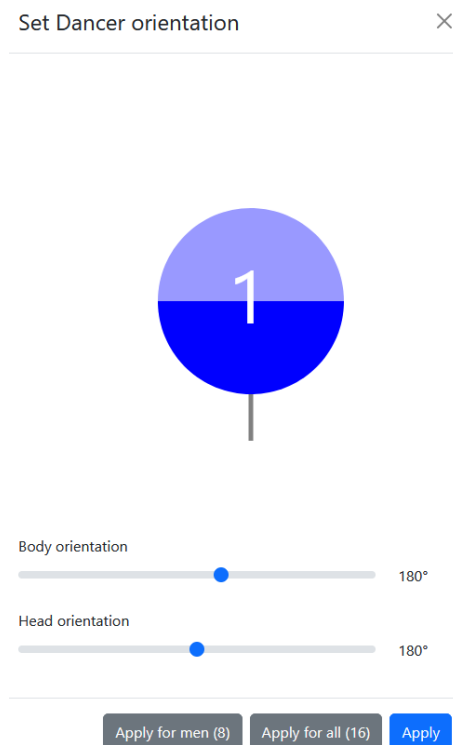
Der Modus „Orientierungen“ ist für die Visualisierung der Blickrichtung und der Ausrichtung der Körper der Tänzer gedacht (**C1**). Die Ausrichtung der Körper ist hier über die Färbung der Tänzer kodiert. Während ohne die Auswahl des Orientierungsmodus alle Tänzer durch einfarbige



**Abbildung 5.13:** Beispielbild im Orientierungsmodus. Der dunklere Halbkreis einer Glyphe kodiert die Ausrichtung des Körpers der jeweiligen Tänzers. Die Blickrichtung wird über die schwarze Linie angezeigt. Die Nummer in der Glyphe beschreibt weiterhin die Zugehörigkeit des Tänzers zu einem Paar.

Kreise dargestellt wurden, bestehen Sie in diesem Modus aus zwei Halbkreisen mit gleicher Farbe und unterschiedlicher Sättigung. Der gesättigtere Halbkreis gibt dann die Ausrichtung des Körpers an. Die schwarze Linie, die von den Rändern des dunkleren Halbkreises ausgeht, beschreibt die Blickrichtung der Tänzer. Dies ist beispielhaft in Abbildung 5.13 zu sehen. Dort sind alle Herren um  $180^\circ$  nach hinten gedreht und blicken geradeaus. Die Damen haben ihren Körper nach vorne ausgerichtet und schauen in einem Winkel von  $45^\circ$  nach links. Für die visuelle Kodierung der Ausrichtungen wurden auch einige andere Versionen getestet. Ein vorheriger Vorschlag war es, die Ausrichtung des Körpers nur über einen Halbkreis ohne eine Änderung der Farbe zu visualisieren. Dieser wurde aufgrund von Missverständnissen seitens der Tänzerposition verworfen.

Um die Ausrichtung der Körper bzw. die Blickrichtung für einen oder mehrere Tänzer zu ändern, muss man im Editiermodus auf einen der Tänzer mit der Maus klicken. Es öffnet sich dann ein modaler Dialog, in welchem über zwei Schieberegler die beiden Ausrichtungen in  $45^\circ$ -Schritten angepasst werden können. Der Dialog, wenn man Tänzer 1 in Abbildung 5.13 anklickt, ist in Abbildung 5.14 zu sehen. Die Blickrichtung, und somit auch die Ausrichtung des Kopfes, wird durch die Ausrichtung des Körpers begrenzt. Man kann die Blickrichtung deswegen, ausgehend von der Ausrichtung des Körpers, um maximal  $90^\circ$  nach links oder rechts bewegen. Danach muss man noch auswählen, für wen die eben eingestellten Ausrichtungen gelten sollen. Zur Auswahl stehen der mit der Maus ausgewählte Tänzer über den Button mit der Beschriftung „Apply“ (C2), alle Tänzer die durch den Brush ausgewählt wurden oder alle Herren/Damen im Brush (C3). Im Beispiel wäre mit „Apply“ nur der Herr 1 betroffen. Mit „Apply for all“ sind im Beispiel alle 16 Tänzer betroffen. Dies ist der Ausnahmefall, wenn kein Tänzer durch den Brush ausgewählt wurde. Alternativ kann man mit „Apply for men/women“ nur die Herren bzw. Damen im Brush auswählen. Auch hier sind dann wieder alle Herren bzw. Damen betroffen, da niemand durch den Brush ausgewählt wurde.

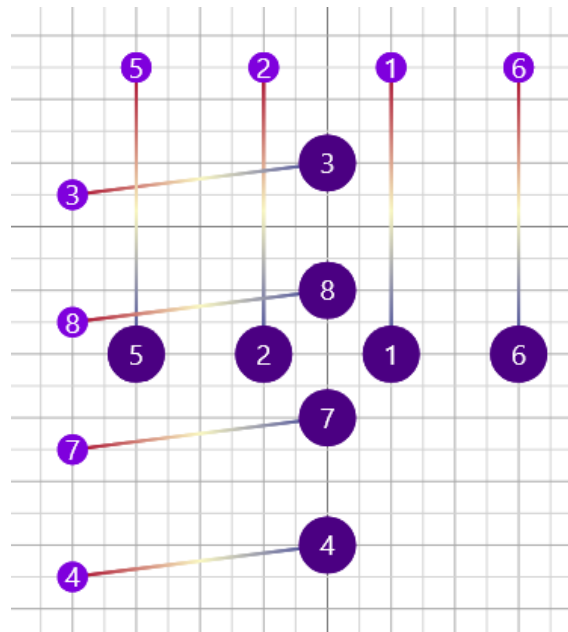


**Abbildung 5.14:** Modaler Dialog zur Änderung der Blickrichtung und der Ausrichtung des Körpers über Schieberegler. Die Ausrichtung des Kopfes ist an die Ausrichtung des Körpers gekoppelt mit einem Sichtfeld von 180°.

Die Zahlen in Klammern hinter den Beschriftungen der Buttons geben zusätzlich Auskunft darüber, wie viele Tänzer durch die Änderungen betroffen sind (auch wenn ihre Ausrichtung praktisch nicht verändert wird).

### 5.2.2 Modus: Transitionen

Der Modus „Transitionen“ spezifiziert und visualisiert die Übergänge zwischen zwei Bildern. Der dargestellte Übergang ist dabei derjenige zwischen dem aktuell ausgewählten und dem vorangegangenen Bild. Die Visualisierung der Tänzer des ausgewählten Bilds bleibt hierfür gleich. Die Positionen der Tänzer des vorherigen Bildes werden nach dem gleichen Schema dargestellt, jedoch mit entsättigteren Farben und einem kleineren Kreisradius (**D1**). Der eigentliche Übergang wird dann über stückweise, zusammenhängende Geraden modelliert (**D2**). Hat man noch keine Änderungen an den Übergängen vorgenommen, handelt es sich nur um eine einzige Gerade, dargestellt durch eine Linie zwischen der aktuellen und vorherigen Position eines Tänzers bzw. Paares. Dieser Fall ist in Abbildung 5.15 dargestellt. Da die Visualisierung mit bis zu 16 einzeln dargestellten Tänzersymbolen in zwei Bildern und mindestens 16 Linien für Übergänge (potentiell durch Zwischenhalte noch mehr) viele graphische Elemente beinhaltet, kann man den Übergang eines einzelnen Tänzers oder Paares auswählen und die restlichen werden ausgeblendet. Klickt man entweder auf den Tänzer aus dem vorherigen oder aktuellen Bild oder auf den Übergang dazwischen, werden nur diese drei Elemente in der gesamten Visualisierung angezeigt. Dies lässt sich mit

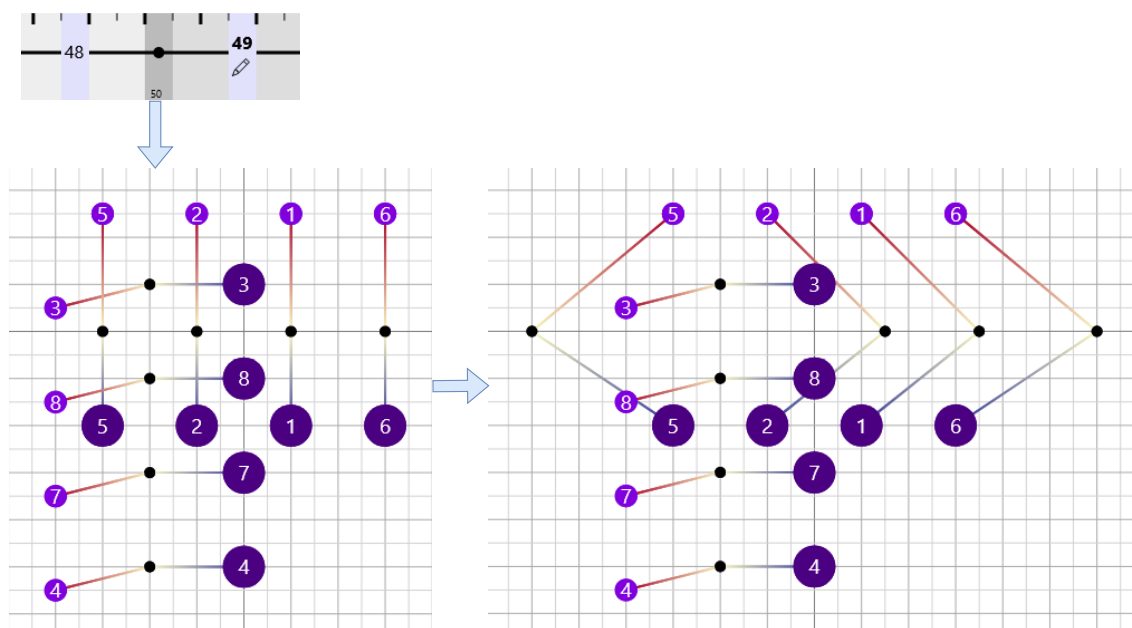


**Abbildung 5.15:** Ein beispielhafter Bildübergang. Die Positionen der Tänzer im ausgewählten Bild sind über die Kreise mit größerem Radius dargestellt. Die Kreise mit kleinerem Radius zeigen die Positionen der Tänzer im vorherigen Bild. In diesem Fall bewegen sich die Tänzer zusammen mit ihrem Partner. Der Übergang zwischen den Bildern ist für jeden Tänzer über eine Linie angezeigt. In diesem Fall bewegen sich die Tänzer auf einer Geraden zu ihrer aktuellen Position. Der Farbübergang auf der Linie symbolisiert den zeitlichen Verlauf während einer Transition. Kreuzen sich zwei Linien mit einem ähnlichem Farbton, dann ist dort die Gefahr hoch, dass zwei Tänzer bzw. Paare miteinander kollidieren.

einem Klick auf den Hintergrund wieder rückgängig machen. Möchte man den Übergang nun genauer spezifizieren, kann man in der Zeitleiste zwischen dem aktuellen und vorherigen Bild einen Zwischenstopp einfügen, indem man auf den Schlag klickt (**D3**). Daraufhin wird die Position des Zwischenstopps auf dem bisherigen Übergang per linearer Interpolation der bisherigen Positionen berechnet und bei Bedarf noch auf die nächstliegende Gitterposition gerundet. Der Zwischenstopp ist über einen schwarzen Punkt kodiert. Dieser Punkt lässt sich per Drag-and-Drop an eine beliebige neue Gitterposition schieben. Somit hat man zwei Teilübergänge: den Übergang von der Position des vorherigen Bildes zum Zwischenstopp und der Übergang vom Zwischenstopp zur Position im aktuellen Bild. Durch mehrere zusammenhängende Geraden als Übergang lassen sich somit auch kurvige Übergänge annähern. Der gerade beschriebene Prozess ist in Abbildung 5.16 nochmal bildlich zu sehen.

Der Farbgradient auf den Linien visualisiert den zeitlichen Verlauf während des Übergangs. Je rötlicher die Linie, desto früher befindet man sich im Übergang, je bläulicher desto später. Diese voreingestellte RdYIBu-Farbskala aus `d3-scale-chromatic`<sup>4</sup> lässt sich in den Einstellungen verändern, ist jedoch standardmäßig eingestellt, sodass Nutzer mit Deuteranopie die Übergänge

<sup>4</sup> URL: <https://github.com/d3/d3-scale-chromatic>

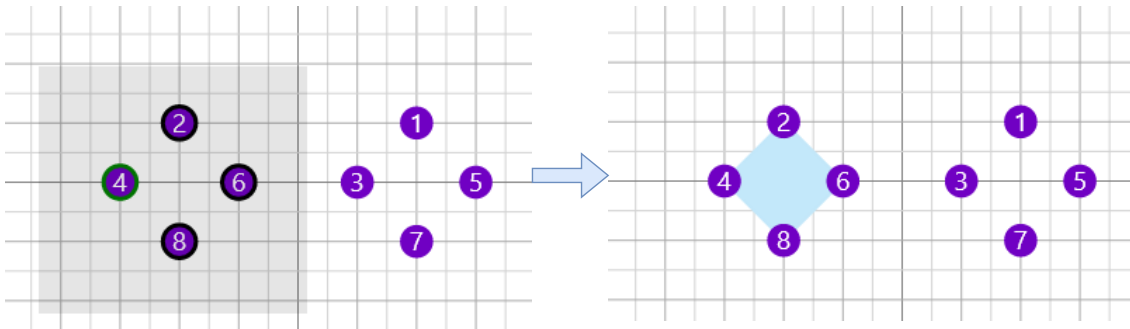


**Abbildung 5.16:** Prozess zur Spezifikation des Bildübergangs mit Zwischenstopps. Zuerst wird in der Zeitleiste der Schlag ausgewählt, bei dem die Tänzer den Zwischenstopp überqueren sollen. Die Position des Zwischenstopps wird per linearer Interpolation auf dem bisherigen Übergang berechnet und als schwarzer Punkt eingezeichnet. Dieser kann dann mit der Maus per Drag-and-Drop verschoben werden.

auch wahrnehmen können. Schneiden sich nun zwei Linien mit einem ähnlichen Farbton, deutet dies auf eine Kollision der Tänzer hin (**F2**). Im Modell wird hierbei davon ausgegangen, dass die Geschwindigkeit der Tänzer beim Übergang konstant ist. In Abbildung 5.15 kreuzen beispielsweise die Linien für die Übergänge der Paare 2 und 8. Somit können Trainer bzw. Choreografen schon beim Bau der Choreografie mögliche Kollisionen erkennen und den Bildübergang entweder umbauen oder auf diese Stellen während des Trainings besonders achten.

Die Übergänge zwischen Bildern können des Weiteren noch animiert dargestellt werden (**D4**). Befindet man sich im Ansichtsmodus und drückt auf den Button mit der Beschriftung „Animate“, dann öffnet sich ein modaler Dialog, über welchen sich die Animation starten lässt. Darin muss man spezifizieren, wie viele Bilder ausgehend vom aktuell gewählten Bild in der Animation enthalten sein sollen und wie viel Pause zwischen den einzelnen Übergängen gemacht werden sollen. Ist beispielsweise Bild 1 ausgewählt und man möchte eine Animation für die nächsten zwei Bilder starten, dann bewegen sich die visuellen Darstellungen der Tänzer erst entlang dem definierten Übergang zwischen den Bildern 1 und 2 und von da aus nach einer kurzen Pause (z.B. eine Sekunde) entlang dem Übergang zwischen den Bildern 2 und 3. Danach wird man in einem modalen Dialog gefragt, ob man die Animation nochmals abspielen möchte. Die Länge der Pause in Sekunden kann über einen Schieberegler, der zum Start der Animation in Bereich D erscheint, noch verändert werden. Die Geschwindigkeit, mit der die eigentliche Animation abläuft, lässt sich in den Einstellungen ebenfalls verändern.





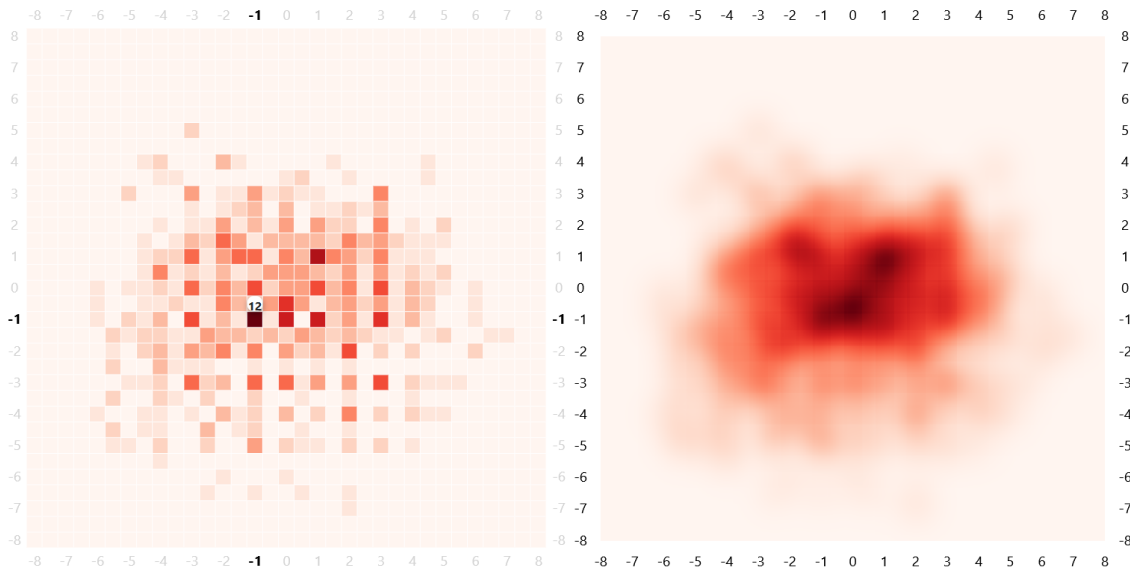
**Abbildung 5.17:** Prozess zum Markieren einer prägnanten, geometrischen Form. Dazu wird eine Menge von Tänzern mit dem Brush markiert. Die konvexe Hülle davon wird im Hintergrund berechnet und als blaues Polygon in die Visualisierung eingezeichnet.

### 5.2.3 Modus: (Prägnante) Formen

Bilder enthalten oft prägnante geometrische Formen die von einfachen Rechtecken bis zu Darstellungen von Tieren reichen können. Um Tänzern die Intention hinter einem Bild klar zu machen, lassen sich prägnante Formen in ein Bild einzeichnen (**E1**). Dazu wählt man in der Visualisierung mit dem Brush eine Menge von Tänzern aus. Im Hintergrund wird dann die konvexe Hülle, wie in Abbildung 2.2 erklärt, berechnet und als blaues Polygon in die Visualisierung eingezeichnet (**E2**). Dieser Prozess ist in Abbildung 5.17 beispielhaft zu sehen. Wird einer der markierten Tänzer nachträglich bewegt, wird die Form neu berechnet. Um eine eingezeichnete Form zu entfernen, kann man mit einem Rechtsklick auf die Form ein Kontextmenü öffnen, in welchem sich nur ein Button mit der Aufschrift „Delete“ befindet. Klickt man auf diesen Button, dann wird die gewählte Form wieder gelöscht (**E3**).

### 5.2.4 Modus: Analyse

Möchte man die vollständige Choreografie genauer analysieren, dann kann dies im Analysemodus durchgeführt werden. In Unterabschnitt 2.1.2 wurde bereits angesprochen, dass Choreografien die Tanzfläche so gut wie möglich nutzen sollten. Diese Funktion kann in unserer Anwendung mittels zwei verschiedener Heatmaps analysiert werden: Zum einen mit einer diskreten Gitterheatmap und zum anderen mit einer kontinuierlichen, räumlichen Heatmap (**F4**). Beide sind für die Choreografie „Crazy Fire“ in Abbildung 5.18 zu sehen. Die Heatmap ersetzt hierbei die Visualisierung der Tänzer und kann nur einzeln betrachtet werden. Standardmäßig ist dies die diskrete Heatmap, was in den Einstellungen jedoch geändert werden kann. Grundlage für beide Heatmaps ist die Frequenz, mit der Tänzer in den Bildern auf einer Position standen. Tanzen Paare zusammen, so zählen diese wie Tänzer nur einfach, da ein Paar nicht signifikant mehr Platz braucht als ein einzelner Tänzer. Zwischenhalte bei Bildübergängen zählen nicht zu den Daten, die für die Heatmap genutzt werden. Bei der diskreten Heatmap wird an jeder Gitterposition ein Rechteck gezeichnet. Die Farbe des Rechtecks kodiert dann, wie oft Tänzer während der Choreografie an dieser Position standen. Je dunkler der Rotton ist, desto häufiger standen die Tänzer an dieser Position. Bewegt man den Mauszeiger über die Kachel, wird die genaue Anzahl in einem Pop-Up-Fenster über der Kachel als Text angezeigt. Zudem werden die entsprechenden Achsenbeschriftungen fett markiert, damit die

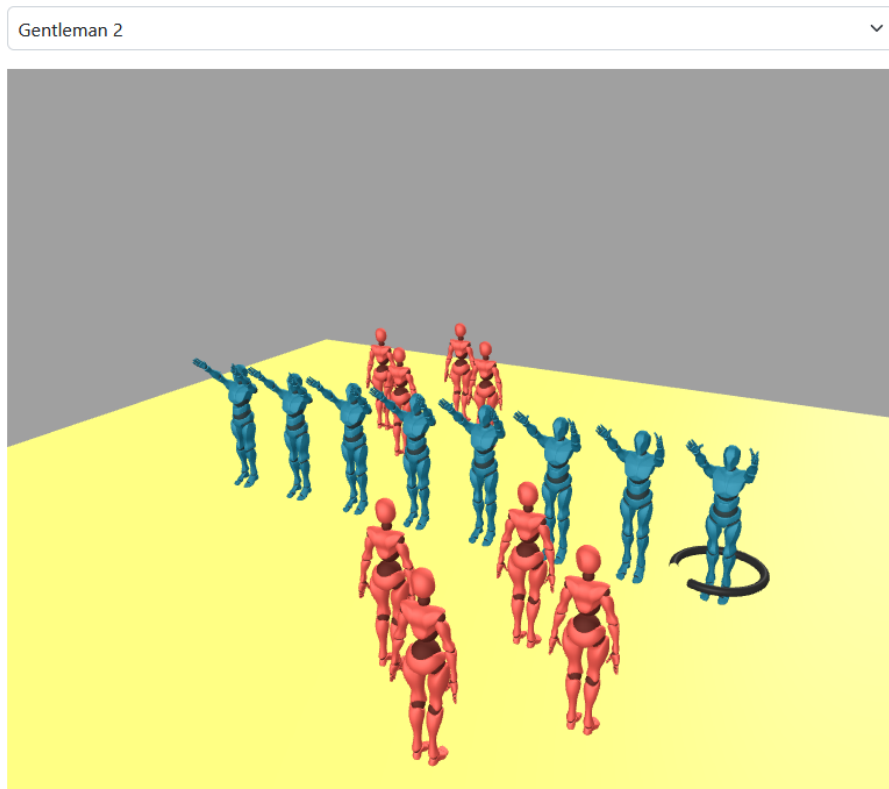


**Abbildung 5.18:** Links: Diskrete Gitterheatmap. Bewegt man den Mauszeiger über eine Zelle, wird in einem Pop-Up-Fenster darüber die Frequenz angezeigt, wie oft Tänzer in Bildern während der gesamten Choreografie auf dieser Position standen. Die Position der Zelle wird an der Achsenbeschriftung hervorgehoben. Rechts: Kontinuierliche, räumliche Heatmap.

Position der Kachel abgelesen werden kann. Zur besseren Vergleichbarkeit zwischen den Heatmaps mehrerer Choreografien lässt sich mit einem Schieberegler die maximale Frequenz verändern, welche die Intensität der Farben in der Heatmap steuert. So lassen sich kleinere Choreografien mit weniger Bildern immer noch mit großen Choreografien vergleichen. Für die kontinuierliche Heatmap wird *splating* verwendet, welches bereits in Unterabschnitt 2.2.2 vorgestellt wurde. Die kontinuierliche Darstellung ist hauptsächlich dafür gedacht, Hotspots zu erkennen. Parameter wie die Bandbreite des Kerns können nicht verändert werden. Im Heatmap-Modus wird nur das Akkordeon mit dem Namen „Analysis“ angezeigt, da die beiden anderen Akkordeons bildspezifische Informationen enthalten. Im angezeigten Akkordeon wird in einem Balkendiagramm, analog zu Abbildung 5.11, die akkumulierte Laufdistanz aller Tänzer über die komplette Choreografie hinweg visualisiert (F2).

### 5.2.5 Modus: 3D-Ansicht

Eine 3D-Ansicht der Bilder existiert laut den befragten Experten bereits in anderen Programmen zur Bilderstellung- und -entwicklung. Das größte Problem war hierbei meist, dass eine dreidimensionale Darstellung anhand primitiver geometrischer Objekte (z.B. Kegel) keinen Mehrwert gegenüber der zweidimensionalen Darstellung bietet. Deswegen wurde für diesen Prototyp die 3D-Ansicht dahingehend erweitert, dass die Körperposen, die in den Bildern eingenommen werden sollen, an realistischeren 3D-Modellen definiert werden können. Die zweidimensionale Visualisierung wird in diesem Modus durch eine dreidimensionale Ansicht ersetzt (vgl. Abbildung 5.19). Das gleiche Bild als 2D-Visualisierung kann in Abbildung 5.13 betrachtet werden. In der Szene ist jeder Tänzer durch ein eigenes 3D-Modell visualisiert. Die roten Modelle stehen für die Damen und die blauen Modelle

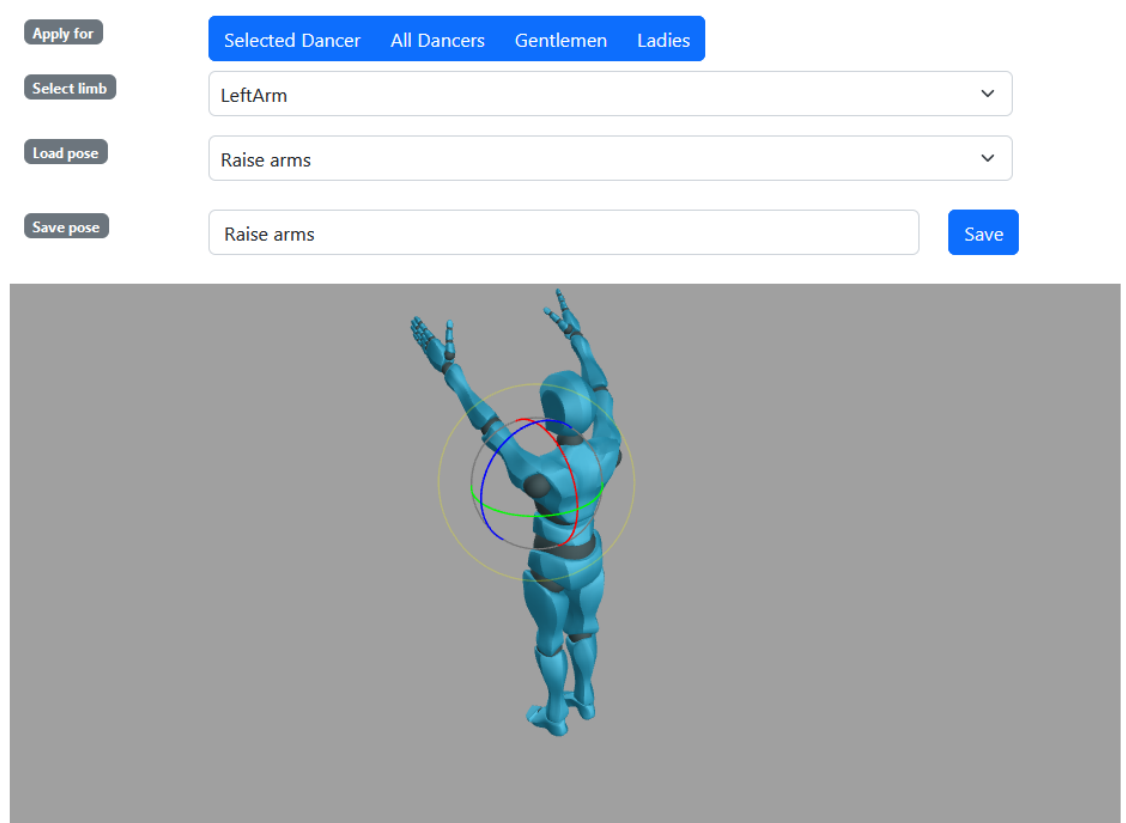


**Abbildung 5.19:** 3D-Ansicht auf das Bild aus Abbildung 5.13. Die roten Modelle stehen für die Damen im Bild und die blauen Modelle für die Herren. Der Herr ganz rechts wurde über die Drop-Down-Liste ausgewählt und ist mit einem 3D-Torus markiert. Die Herren haben zudem eine selbst definierte Pose bekommen, bei der sie beide Arme heben.

stehen für die Herren. Die Farben können bisher noch nicht wie in der 2D-Ansicht geändert werden. Die Informationen für die Position der Tänzer, die Blickrichtung und die Ausrichtung des Körpers werden aus den Daten der 2D-Visualisierung übernommen. Mit gehaltener rechter Maustaste kann man die Kamera durch die Szene bewegen, mit der linken Maustaste die Ansicht drehen und mit dem Mausekranz zoomen. Die Kamera dreht sich des Weiteren automatisch mit, wenn die 2D-Ansicht rotiert wird. Damit man die Tänzer bei jeder Kameraeinstellung sehen kann, ist Backface-Culling aktiviert, sofern die Kamera sich unter der Tanzflächen-Plane befindet. Beim Backface-Culling werden Rückseiten von Polygonen in 3D-Modellen nicht dargestellt, um Rechenzeit zu sparen. Über eine Drop-Down-Liste kann ein Tänzer ausgewählt werden, welcher dann in der 3D-Sicht von einem sich drehenden schwarzen Torus umrundet wird. Diese Auswahl ist bisher noch umständlich und könnte in weiteren Entwicklungszyklen durch eine direkte raycasting-gestützte Alternative ersetzt werden.

Um die 3D-Modelle der Tänzer Posieren zu lassen, existiert im Editiermodus ein Poseneditor, welcher anstatt der Akkordeons angezeigt wird. Dieser ist in Abbildung 5.20 zu sehen. In der unteren Hälfte des Poseneditors wird hierfür eine weitere 3D-Ansicht geladen. In dieser wird das Modell des Tänzers dargestellt, der in der 3D-Bildansicht über die Drop-Down-Liste ausgewählt wurde. Für das Modell kann dann eine neue Pose erstellt werden. Dazu wählt man in der Drop-

## 5 Implementierung



**Abbildung 5.20:** Ausschnitt aus dem Poseneditor. In der Drop-Down-Liste, die mit „Select limb“ beschriftet ist, kann ein Gelenk ausgewählt werden, dessen Rotation in der 3D-Ansicht am unteren Rand des Editors rotiert werden kann. In der Drop-Down-Liste darunter können gespeicherte Posen geladen werden. Darunter können Posen mit einem Namen gespeichert werden. Über die vier blauen Buttons am oberen Rand des Editors kann die Pose auf den ausgewählten Tänzer, alle Herren, alle Damen oder alle Tänzern angewendet werden.

Down-Liste, welche mit „Select limb“ beschrieben ist, ein Körperteil aus, dessen Rotation verändert werden soll. Im gezeigten Beispiel ist der linke Arm ausgewählt. An der Stelle des ausgewählten Körperteils wird dann ein Interaktionselement eingeblendet, welches auch in Spieleengines und 3D-Modellierungsprogrammen zur Änderung von Rotationen an 3D-Modellen eingesetzt wird. Die Berechnung der Knotenpositionen im Gitternetz des Modells übernimmt hierbei das *SkinnedMesh*, eine Klasse von *Three.js* für 3D-Modelle mit virtuellen Knochen im Inneren. Es werden also keine eigenen *Inverse Kinematics* Berechnungen durchgeführt. Nun kann über die vier Buttons am oberen Rand des Editor ausgewählt werden, ob nur der selektierte Tänzer, alle Tänzer, alle Herren oder alle Damen die neue Pose zugewiesen bekommen sollen. Zudem können Posen mit einem Namen gespeichert und so auch in weiteren Bildern wiederverwendet werden. Die Interaktion mit der 3D-Szene des Poseneditors geschieht analog zu der 3D-Ansicht des Bildes.

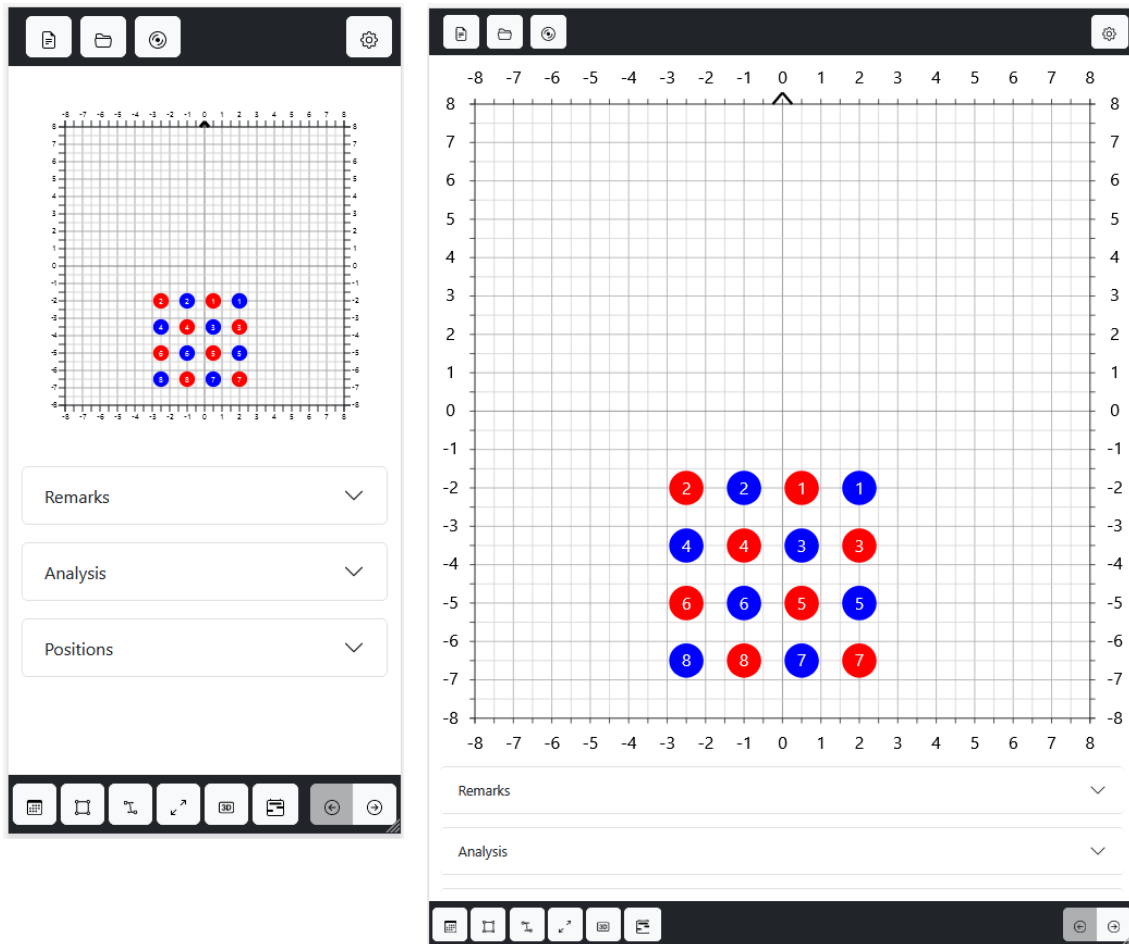
## 5.3 Ansicht auf mobilen Geräten

Wie bereits in Abschnitt 5.2 angesprochen, ist die Anwendung für die zwei Hauptzielgruppen Tänzer und Trainer bzw. Choreografen erstellt worden. Da Tänzer im Training meist Geräte wie Smartphones verwenden, wurde die Webanwendung responsiv entwickelt und diesem Anforderungsprofil angepasst. Deswegen fällt auf allen mobilen Geräten (mit Ausnahme der Landschaftsansicht von Tablets) die Möglichkeit in den Editiermodus zu wechseln weg. Somit können Tänzer nicht unabsichtlich die Choreografie verändern. Laptops und Convertibles zählen in dieser Erklärung nicht zu klassischen mobilen Geräten, auch wenn diese portabel sind. Mobile Geräte sind hier mit einer Bildschirmbreite von weniger als 992 Pixeln definiert. Hierfür muss beachtet werden, dass hochauflösende Bildschirme für diese Betrachtung herunterskaliert werden. Das heißt es kommt primär auf das Seitenverhältnis des Geräts und nicht auf die Pixelanzahl an. Hierfür ein kleines Beispiel zum Verständnis. Das Apple iPad hat eine Auflösung von 1620x2160 Pixeln, hat jedoch ein Bildschirmdiagonale von 10,2 Zoll. Somit ist das iPad im Vergleich zu Full-HD Bildschirmen relativ klein, hat aber eine hohe Pixeldichte. Der Emulator skaliert dies nun mit dem Faktor 0.5 (also einer *device pixel ratio* von 2) auf eine Auflösung von 810x1080 Pixel runter. Beispiele der mobilen Versionen in den Porträtsichten auf einem emulierten Samsung Galaxy S10 und einem Apple iPad befinden sich in Abbildung 5.21. In diesen Versionen wird der Bereich D (die Akkordeons) aus der Desktopversion nicht mehr rechts neben der Visualisierung gerendert, sondern darunter, um der Visualisierung der Bilder keinen Platz in der Breite wegzunehmen. Des Weiteren ist die Werkzeugleiste am unteren Bildschirmrand platziert. Dies wird bei vielen mobilen Applikationen so umgesetzt, da diese meist mit dem Daumen der haltenden Hand bedient werden. Da die meisten Smartphones bzw. Tablets jedoch deutlich höher sind als eine Daumenlänge, ist es schwierig an Interaktionselemente im oberen Bildschirmbereich zu gelangen. Deswegen wurden die Buttons aller häufig benutzten Aktionen am unteren Bildschirmrand platziert. Eine weitere Anpassung findet für schmalere Bildschirme mit einer Breite von mindestens 992 Pixeln statt. So werden die Akkordeons hier stückweise kleiner, um der Visualisierung der Bilder keinen Platz wegzunehmen. Dies ist beispielhaft in der Landschaftsansicht eines Apple iPad in Abbildung 5.22 zu sehen. Auch die Werkzeugleiste wird bei sinkender Bildschirmbreite verschmälert, um nicht in die Visualisierung der Bilder hineinzuragen. In einem ersten Schritt wird hierzu die Beschriftung der Buttons ausgeblendet. Damit die Funktion eines Buttons immer noch ersichtlich ist, wird in einem Tooltip die Beschriftung zusätzlich noch eingeblendet. In einem zweiten Schritt werden die Buttons dann noch verschmälert und die beiden nebeneinander angeordneten Buttons, um das vorherige und nachfolgende Bild auszuwählen, werden übereinander platziert. Des Weiteren fehlt aus Platzgründen die Drop-Down-Liste zum Auswählen eines beliebigen Bildes. Diese Version ist in Abbildung 5.22 zu sehen.

Eine letzte Änderung gibt es bezüglich der Zeitleiste in Bereich E. Diese lässt sich auf schmalen Geräten nicht sinnvoll bedienen, da zur übersichtlichen Nutzung einiges an Platz benötigt wird. Deswegen fällt sie ab einer Bildschirmbreite von weniger als 992 Pixeln weg. Auch sollte die Leiste in keiner Version wichtige Interaktionselemente für den Nutzer verdecken. Deswegen wird sie auf allen Systemen nur bei Bildschirmhöhen von mindestens 575 Pixeln eingeblendet.

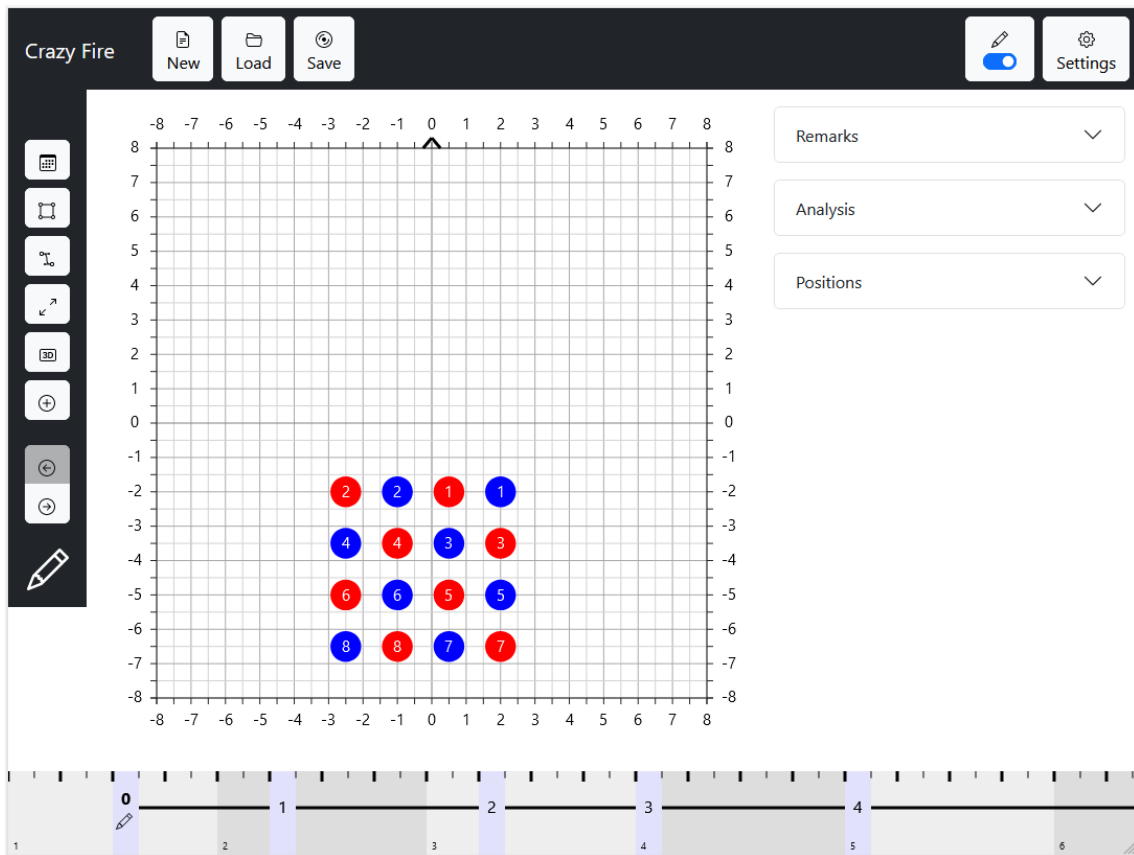
Eine Optimierung für Smartphones in der Landschaftsansicht existiert noch nicht. Hier ist hauptsächlich die geringe Höhe des Bildschirms ein Problem, bei welcher es schwer möglich ist, das ganze Bild zu zeigen, ohne es zu klein zu skalieren. Die Interaktion auf Tablets muss in weiteren Iterationen der Entwicklung ebenfalls noch untersucht und verbessert werden. So ist momentan

## 5 Implementierung



**Abbildung 5.21:** Mobile Versionen der Webapplikation bei einer Portrait-Ansicht. Links: Version auf einem emulierten Samsung Galaxy S10 mit Android 11 und einer Auflösung von 360x760 Pixeln. Rechts: Version auf einem emulierten iPad mit iPadOS 14.7.1. und einer Auflösung von 810x1080 Pixeln.

das Drag-and-Drop auf Touchoberflächen, das nur im Editiermodus auf Tablets möglich ist, nicht ausgereift. Zudem gibt es noch Probleme mit dem Safari Browser auf iPadOS, in welchem die Achsenbeschriftungen falsch angezeigt werden. Diese Probleme ließen sich ohne ein Testgerät und nur mit Emulatoren anderer Browser noch nicht nachbilden. Auch dies sollte in weiteren Entwicklungszyklen behoben werden.



**Abbildung 5.22:** Mobile Version der Webapplikation der Landschaftsansicht auf einem emulierten iPad mit iPadOS 14.7.1. und einer Auflösung von 1080x810 Pixeln.





# 6 Evaluation

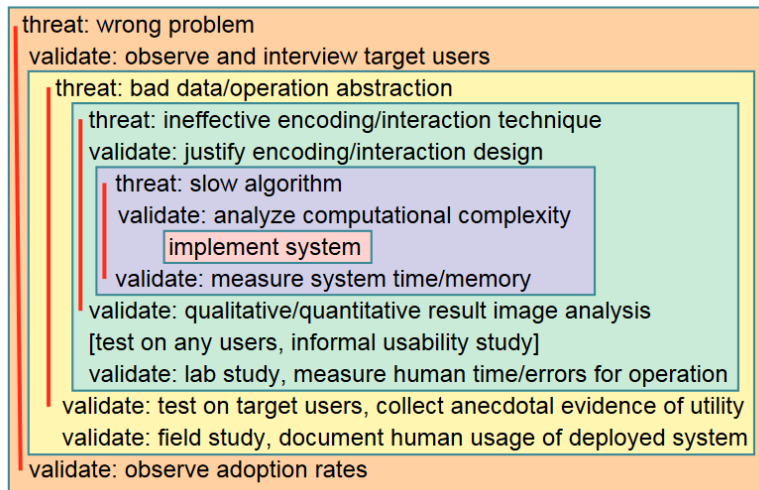
Im diesem Kapitel wird erläutert, wie der Prototyp evaluiert wurde. Dazu wird in Abschnitt 6.1 als Erstes die Zielsetzung dieser Evaluation genannt. Die Beschreibung der konkreten Umsetzung folgt dann in Abschnitt 6.2. Danach werden die Ergebnisse der Studie in Abschnitt 6.3 aufgezählt. Die Ergebnisse werden dann in Abschnitt 6.4 interpretiert und diskutiert. Am Ende folgt in Abschnitt 6.5 noch eine Betrachtung aller Störvariablen, die die Ergebnisse der Studie potenziell beeinträchtigt haben könnten.

## 6.1 Zielsetzung

Das Ziel der Evaluation war zu testen, ob es Formationstrainern möglich ist, mit wenig Aufwand Choreografien zu erstellen, sowie bestehende Choreografie zu editieren und zu analysieren. Das erste Ziel kann hierbei nur teilweise untersucht werden, da das Erstellen neuer Choreografien ein zeitintensiver Prozess ist, welcher oft auch durch das Feedback der Tänzer im Training beeinflusst wird. Da die Kommunikation von Choreografien mit den Tänzern darauf aufbaut, dass Tanzlehrer bereits Choreografien mit dem Prototyp gebaut haben, kann auch dieser Aspekt aufgrund des zeitlichen Rahmens der Arbeit nicht untersucht werden.

Die Art der Evaluation von Visualisierungen und das resultierende Feedback wird unter anderem im *Nested model for Visualization Design and Validation* von Munzner [Mun09] beschrieben. Eine Illustration davon lässt sich in Abbildung 6.1 finden. Das Modell besteht aus mehreren ineinander geschachtelten Levels. Diese beschäftigen sich von außen nach innen mit der Charakterisierung der Problemdomäne, der Abstraktion der Daten bzw. Operationen auf den Daten, dem Design visueller Abbildungen der Daten und Interaktionsmöglichkeiten, sowie der Entwicklung von effizienten Algorithmen. Die Resultate eines Levels dienen als Eingaben für die nächste Schachtelungsebene. Nachdem alle geschachtelten Levels abgeschlossen sind, sollte für jedes Level noch eine Form der Validierung durchgeführt werden. Da Fehler in äußeren Levels nach innen weiter propagiert werden und erst danach durch die Validierung entdeckt werden würden, sollte man vor dem Eintritt in ein geschachteltes Level ebenfalls erst eine Validierung durchführen. Damit lassen sich, je nach Level, Probleme und Feedback in vier Kategorien einteilen:

1. Das falsche Problem wird gelöst: Die Visualisierung löst kein Problem des Nutzers, das heißt es wurden fehlerhafte, unvollständige oder inkonsistente Anforderungen erhoben.
2. Eine unpassende Abstraktion der Daten wurde gewählt, das heißt die Abbildung der Daten der Zieldomäne auf Datenstrukturen der Informatik ist problematisch.
3. Inadäquate visuelle Kodierungen und Interaktionstechniken wurden verwendet, das heißt die Art wie Daten visuell aufbereitet werden und wie man mit ihnen interagiert, ist problembehaftet umgesetzt.



**Abbildung 6.1:** *Nested model for Visualization Design and Validation* von Munzner [Mun09]. Das Modell besteht aus ineinander geschachtelten Levels. Um eine Aktivität auf einem Level zu beginnen, sollte das umgebene Level abgeschlossen sein, da die darin generierten Artefakte Eingaben für das aktuelle Level sind.

4. Es wurde der falsche Algorithmus angewendet, das heißt die Laufzeit der Implementierung ist zu hoch.

Da während unserer Studie die *Think Aloud* Methodik verwendet wurde, kann die Art des Feedbacks nicht eingegrenzt werden. Teilnehmer können und sollen alle ihre Gedanken bei der Nutzung des Prototyps laut formulieren. Daher kann es passieren, dass Teilnehmern eine Funktion fehlt. Dies kann auf eine unvollständige Beschreibung der Domäne hindeuten. Andererseits kann ein Teilnehmer sich zum Beispiel über ein schwer verständliches Nutzerinterface beklagen, was auf ein problematisches Interaktionsdesign hindeutet.

## 6.2 Durchführung

Um die Nutzung des Prototyps zu beobachten, wurde eine Laborstudie mit vier Domänenexperten durchgeführt. Alle Teilnehmer der Studie waren Trainer von Lateinformationen. Einer der Teilnehmer hatte bereits an den ersten zwei Interviews teilgenommen. Die restlichen Teilnehmer wurden über private Kontakte eines Betreuers dieser Arbeit rekrutiert. An der Evaluation haben drei Männer und eine Frau teilgenommen. Die Teilnehmer waren zwischen 22-55 Jahre alt. Die Vorerfahrungen aller Studienteilnehmer sind in Tabelle 6.1 zusammengefasst.

Als erstes wurden die Teilnehmer über den Zweck der Studie und ihre Rechte aufgeklärt. Dies durften die Teilnehmer durch das Unterschreiben einer Einwilligungserklärung bestätigen. Die Experten bekamen für die Teilnahme an der Studie keine Entschädigung. Danach durften sie einen Fragebogen ausfüllen, in welchem ihr Alter, ihr Geschlecht, ihre Erfahrung als Trainer und Tänzer von Lateinformationen und ihre Erfahrung in der (digitalen) Entwicklung von Choreografien abgefragt wurden. Bevor der eigentliche Durchgang begann, absolvierten die Teilnehmer eine Trainingsphase. In dieser bekamen sie ein sechsminütiges Video präsentiert, in welchem alle für die

	Teilnehmer 1	Teilnehmer 2	Teilnehmer 3	Teilnehmer 4
Erfahrung als Tänzer	17 Jahre: Landesliga, Oberliga	8 Jahre: Landesliga, Oberliga, Regionalliga, 2. Bundesliga	8.5 Jahre: Landesliga bis 2. Bundesliga	35 Jahre: Alle Ligen, Weltmeisterschaft
Erfahrung als Trainer	4 Jahre: Landesliga	3 Jahre: Regionalliga, 2. Bundesliga	1.5 Jahre: Landesliga	21 Jahre: Alle Ligen
Erfahrung in der Erstellung von ChoreografieN	4 Jahre: keine Neuentwicklung, sondern Erarbeitung bestehender Choreografien	Anpassung existierender Bilder aus dem A-Team, Erstellung zusätzlicher Bilder	2 komplette Choreografien erstellt (für 6 und 8 Paare)	Mehrfach komplette Choreografien erstellt
Erfahrung mit digitalen Werkzeugen	4 Jahre: Digitalisierung bestehender Choreografien	Anpassung bereits existierender Bilder, Neuentwicklung von Bildern	Erstellung der Choreografien mit ChoreoMaker	Choreografien wurden mit privatem Programm erstellt

**Tabelle 6.1:** Vorerfahrung der Studienteilnehmer

Studie relevanten Funktionen des Prototyps erklärt wurden. Danach durften die Teilnehmer den Prototyp 15 Minuten lang frei austesten und dem Studienleiter bei Verständnisproblemen Fragen stellen. Als letzten Schritt in der Trainingsphase wurde den Teilnehmern der *Think Aloud* Prozess erklärt und sie durften ihn an folgender Frage trainieren: „Bitte berechnen Sie die Lösung der Aufgabe  $15 \cdot 25$  und sprechen Sie alle Ihre Gedanken dabei laut aus“. Fragen dieser Art eignen sich laut Boren und Ramey [BR00] besonders gut, da Zahlen beim Lösen eines mathematischen Problems in derselben Form verbalisiert werden, wie sie auch im Kurzzeitgedächtnis liegen.

Darauf folgte der Hauptteil der Studie. In diesem bekamen die Teilnehmer auf Papier eine Reihe von Aufgaben, welche Sie in sequentieller Reihenfolge abarbeiten sollten. Zur Bearbeitung der Fragen wurde eine neue Choreografie für die Evaluation erstellt. Die Teilnehmer wurden angewiesen, jede Frage bei Beginn laut vorzulesen. Danach sollten Sie die Frage bearbeiten und alle ihre Gedanken währenddessen verbalisieren. Taten Sie dies in einem Zeitraum von 15-60 Sekunden nicht mehr, dann wurden sie vom Studienleiter nochmals mit der Aufforderung „Reden Sie ruhig weiter“ daran erinnert, den *Think Aloud* Prozess fortzusetzen. Der Studienleiter griff nur ein, wenn die Teilnehmer auf Programmfehler stießen, um den Prozess der selbstständigen Verbalisierung durch bilaterale Kommunikation nicht ständig zu unterbrechen. In einem Fehlerfall wurden die Teilnehmer nochmals darauf hingewiesen, dass sie keinen Fehler gemacht haben, sondern einen Beitrag zu Verbesserung des Programms geleistet haben. Waren die Teilnehmer der Meinung mit einer Aufgabe fertig zu sein, dann sollten sie dies kurz signalisieren. Für die Studie wurde das Programm auf einem Desktop-PC mit Maus und Tastatur genutzt. Der Auflösung des Bildschirms betrug mindestens 1920x1080 Pixel. Nachdem die Teilnehmer die Aufgaben abgeschlossen hatten,

bekamen Sie zehn Aussagen über verschiedene Aspekte des Prototyps und sollten ihre Zustimmung bzw. Ablehnung gegenüber der Aussage auf einer Likert-Skala von -2 bis 2 ausdrücken, wobei 2 für „Ich stimme voll und ganz zu“ und -2 für „Ich stimme überhaupt nicht zu“ stehen. In einem letzten Schritt wurden die Teilnehmer dann noch aufgefordert, Aspekte am Prototyp hervorzuheben, die sie als besonders gelungen erachten. Danach sollten sie Aspekte nennen, die sie für problematisch und verbesserungswürdig halten. Ein Durchgang dauerte, je nach Studienteilnehmer, zwischen 60 und 90 Minuten. Die Bearbeitung der Aufgaben mit der *Think Aloud* Methode, sowie die anschließende Diskussion wurden als Tonspur aufgezeichnet. Zusätzlich wurde der Bildschirm aufgezeichnet, auf welchem die Teilnehmer die Aufgaben durchführten. Die Abschriften der Audioaufnahmen sind im Ergänzungsmaterial zu dieser Arbeit [Sch23] verfügbar.

### 6.3 Resultate

Im Folgenden werden die Ergebnisse der Studie vorgestellt. Diese beruhen sowohl auf der Beobachtung der Teilnehmer beim Lösen der Aufgaben, als auch auf dem Feedback der Teilnehmer in den Diskussionsrunden. Zur übersichtlicheren Gliederung wird immer eine Aufgabe aus dem Aufgabenkatalog genannt und dann die Resultate über den Themenbereich präsentiert, welchen diese Aufgabe abdecken sollte.

**Laden Sie die Choreografie mit dem Namen „Evaluationschoreografie“. Sie befindet sich im Ordner „Downloads“.**

Die Aufgabe konnte von allen Teilnehmer ohne Probleme abgeschlossen werden. Alle Teilnehmer waren sich zuerst nicht sicher, ob die Choreografie auch wirklich erfolgreich geladen wurde. Sie konnten dies dann aber am Namen in der Navigationsleiste und an der Anzahl der Bilder in der Zeitleiste ablesen. Ein Teilnehmer merkte an, dass es sinnvoll wäre, die Choreografie online zu speichern und so mit den Tänzern zu teilen.

**Welche Ansicht haben Sie momentan auf die Choreografie. Erklären Sie in eigenen Worten den Inhalt der Visualisierung (d.h. die Tanzfläche und deren Inhalt).**

Ziel dieser Aufgabe war es zu prüfen, ob die Teilnehmer die grundlegenden, visuellen Kodierungen verstanden haben. Dazu gehören insbesondere die Erkennung von Damen und Herren anhand von Farben, die Positionen der Tänzer im Koordinatensystem und die Zuordnung zu den Paaren mit den Nummern in den Kreisen der Tänzer. Auch diese Aufgabe konnte von allen Teilnehmern erfolgreich bearbeitet werden. Zwei der Teilnehmer hatten jedoch zunächst Probleme herauszufinden, welche Farbe den Kreisen der Damen bzw. der Herren zugeordnet sind. Ein Teilnehmer löste dies, indem er die Farbe im Einstellungsmenü einsah und der andere Teilnehmer bewegte einen der Tänzer und beobachtete die Veränderung der Positionswerte in der Tabelle. Der erste Teilnehmer schlug daher vor, im Ansichtsmodus eine kleine Legende für die Farben der Tänzer anzubringen.

**An welcher Position steht die Dame 8 im Bild 0 und mit welchem Körperteil steht sie auf der Position?**

Die Aufgabe konnte von allen Teilnehmern gelöst werden. Zwei der Teilnehmer gelangten an die Information über die hervorgehobene Beschriftung der Achsen, die anderen zwei Teilnehmer lasen die Position in der Tabelle im Akkordeon rechts neben der Visualisierung ab. Die Information, mit welchem Körperteil die Dame auf der Position steht, konnte von zwei Teilnehmern direkt aus dem Akkordeon mit der Aufschrift „Remarks“ abgelesen werden. Die anderen beiden Teilnehmer fanden

diese Information nicht auf den ersten Blick und suchten alle Interfaceelemente nacheinander ab. Ein Teilnehmer versuchte hierfür den Orientierungsmodus zu verwenden. Er argumentierte, dass der Körper des Tänzers nach vorne rotiert ist und der Tänzer deswegen mit der Körpermitte auf dem Punkt steht. Der andere Teilnehmer versuchte erst, die Information im 3D-Modus zu finden. Schlussendlich konnten alle Teilnehmer die vorgesehene Information finden.

**An welcher Bildschirmposition würde sich ein Trainer befinden, wenn er vor den Tänzern steht?**

Alle Teilnehmer konnten diese Aufgabe erfolgreich bearbeiten und erklärten korrekt, dass das schwarze Dreieck am Rand der Visualisierung den Blick der Tänzer in Richtung der Tribüne mit den Wertungsrichtern zeigt. Nur einer der Teilnehmer drehte die Visualisierung danach noch. Dieser merkte an, dass er es bevorzugen würde, wenn das rotierte Bild auch im Editiermodus verfügbar wäre.

**An welcher Position befinden sich während der Choreografie Tänzer am häufigsten?**

Die Aufgabe konnte von allen Teilnehmern gelöst werden. Alle wechselten ohne Probleme in den Editiermodus und identifizierten die dunkelste Kachel in der diskreten Heatmap als Position mit höchster Ausnutzung. Zwei der Teilnehmer veränderten die höchste Frequenz als Parameter für die Farbskala in der Heatmap ohne daraus jedoch weitere Erkenntnisse zu gewinnen.

**Welche Teile der Tanzfläche werden besonders oft genutzt und welche nicht?**

Die Aufgabe konnte von allen Teilnehmern gelöst werden. Alle Teilnehmer identifizierten, dass die Randbereiche weniger stark ausgelastet sind und dass sich die Tänzer hauptsächlich im mittleren Teil der Tanzfläche aufhalten. Dafür wurde nur die diskrete Heatmap verwendet. Ein Teilnehmer merkte an, dass er zwischen dieser Aufgabe und der Aufgabe zuvor keinen Unterschied sehe. Alle Teilnehmer hoben die Möglichkeit zur Analyse der Ausnutzung der Tanzfläche in der Diskussionsrunde nochmal hervor und betonten die Wichtigkeit der Funktion, da es sich um ein Wertungskriterium handelt, dessen Analyse in anderen Programmen nicht möglich sei.

**Finden Sie die Dame oder den Herren mit der niedrigsten Laufdistanz während der gesamten Choreografie.**

Die Aufgabe konnte von drei der vier Teilnehmer gelöst werden. Jeder der Teilnehmer versuchte zur Lösung der Aufgabe das Balkendiagramm im Analysemodus zu verwenden. Der Teilnehmer, der die Aufgabe nicht lösen konnte, missinterpretierte die Balken als Distanz für Paare und nicht für einzelne Tänzer. Die anderen Teilnehmer hoben die Funktion zum Ablesen von Laufdistanzen zwischen Bildern positiv in der Diskussionsrunde hervor. Ein Teilnehmer teilte mit, dass die Länge von Laufwegen öfters ein Streitthema zwischen den Tänzern und dem Formationstrainer sei.

**Erstellen Sie ein neues Bild für die Choreografie, in dem ein Rechteck enthalten sein soll. Ordnen Sie das Bild, so gut wie es geht, mittig an.**

Jeder Teilnehmer erstellte erfolgreich ein neues Bild und verwendete dafür das Standardbild mit dem Namen „Rectangle“. Zwei Teilnehmer lobten explizit die Standardbilder und meinten, dass sie dies aus anderen Programmen bisher nicht kannten. Ein weiterer Teilnehmer fragte in der Diskussionsrunde nochmal gesondert nach einer Funktion, mit der eigene Bilder als Blaupausen gespeichert werden können und war erfreut, dass diese Funktion bereits existiert. Er war der Meinung, dass dies das Erstellen von Choreografien vereinfacht. Der Übergang vom Erstellen des Bildes zum Bearbeiten des Bildes funktionierte hingegen bei zwei der Teilnehmer nicht problemlos. Ein Teilnehmer befand sich noch im Analysemodus und war sich nicht sicher, ob das neue Bild erfolgreich erstellt wurde. Der andere Teilnehmer befand sich im Transitionsmodus und versuchte in

diesem ohne Erfolg, die Positionen der Tänzer per Drag-and-Drop zu ändern. Um dem Teilnehmer die Bearbeitung nachfolgender Aufgaben zu ermöglichen, griff der Studienleiter hier korrigierend ein. Beim zweiten Teil der Aufgabe, also dem mittigen Anordnen des Rechtecks, kam es sehr darauf an, wie die Teilnehmer die Aufgabe interpretierten. So trennten zwei der Teilnehmer beispielsweise die Frauen und die Männer voneinander und versuchten, zwei separate Rechtecke mit ihnen zu bilden. Hierbei fiel auf, dass diese Teilnehmer erfolgreich die Kombination aus Steuerungstaste und Linksklick nutzten, um mehrere Tänzer zu selektieren. Dies klappte nicht bei allen Teilnehmern trotz potenzieller vorangegangener erfolgreicher Anwendung. Entweder versuchten die Teilnehmer dabei, einen Tänzer nur über einen Linksklick auszuwählen oder sie kombinierten den Linksklick mit dem Drücken der Umschalttaste. Ein Teilnehmer schlug vor, dass alle Damen bzw. alle Herren bei einem Doppelklick automatisch ausgewählt werden. Beim Drag-and-Drop einzelner Tänzer hatten die Teilnehmer keine Probleme. Auch die Mehrfachauswahl mit der Maus konnten alle Teilnehmer ohne Probleme anwenden. Ein Teilnehmer wünschte sich eine Funktion, mit der die ausgewählten Tänzer über die Pfeiltasten der Tastatur verschoben werden können. Zwei Teilnehmer merkten explizit an, dass ein Raster mit Abständen von einem halben Meter nicht ausreicht, um alle Anwendungsfälle beim Bau einer Choreografie abzudecken. Die Anforderungen variierten hierbei je nach Trainer. Während der Trainer aus der Landesliga noch Drag-and-Drop mit den Auflösungen von 1 Meter, 0,5 Meter und 0,25 Meter forderte, wollte der Trainer aus der 1. Bundesliga zusätzlich Auflösungen von 0,33 Meter und 0,1 Meter. Dies steht im Widerspruch zu den Anforderungen, die wir in den ersten zwei Interviews mit Tanzlehrern aus der Landesliga erhoben haben. Zwei der Teilnehmer versuchten für eine mittige Anordnung des Rechtecks die Abstände der Tänzer im vorgegebenen Bild zu verändern. Jedoch benutzte nur einer dafür auch das Skalierwerkzeug. Dies konnte er nicht erfolgreich einsetzen. Um diesen Fehler rückgängig zu machen, löschte er das Bild und fügte das Rechteck wieder neu hinzu. Keiner der Teilnehmer versuchte, die Tänzer zu Paaren zusammenzufügen.

**Verändern sie die Choreografie nun so, dass das neu erstellte Bild zum ersten Schlag von Takt 36 steht.**

Die Aufgabe konnte von allen Teilnehmern gelöst werden. Einer der Teilnehmer versuchte, das Bild per Drag-and-Drop auf der Zeitleiste zu verschieben.

**Fügen Sie ein weiteres Bild mit einem Rechteck hinzu, bei welchem die hinterste Reihe an Tänzern eine um 45° fallende Diagonale von links nach rechts stellt.**

Die Aufgabe konnte von drei der vier Teilnehmer gelöst werden. Diese Teilnehmer nutzten zur Rotation der Tänzerreihe das vorgesehene Menü rechts von der Visualisierung. Ein Teilnehmer rotierte die Tänzerreihe erst um 45° nach oben. Zwei der drei Teilnehmer hatten zuerst nicht den Tänzer am weitesten links als Rotationszentrum ausgewählt und bekamen eine Fehlermeldung. Mit den Informationen aus der Fehlermeldung konnten sie die Aufgabe dann erfolgreich abschließen. Der Teilnehmer, der die Fehlermeldung nicht bekam, merkte an, dass es besser wäre, wenn eine Reihe von Tänzern um einen beliebig ausgewählten Tänzer rotiert werden kann, da Bilder oft Kippunkte enthalten. Der vierte Teilnehmer erstellte statt einem Bild mit einem Rechteck ein Bild, das eine Doppeldiagonale enthielt und argumentierte, dass die Tänzer nun in einer Diagonale stünden. Jedoch hatte diese Diagonale keinen Neigungswinkel von 45°.

**In welche Richtung schauen die Damen in Bild 8?**

Die Aufgabe konnte von allen Teilnehmern gelöst werden. Zwei Teilnehmer merkten an, dass ihnen die visuellen Kodierungen der Blickrichtung und Ausrichtung der Körper der Tänzer nach dem Einführungsvideo noch nicht vollständig klar waren. Nach kurzer Zeit konnten jedoch alle

Teilnehmer die Daten visuell ablesen. Ein Teilnehmer kritisierte zudem noch, dass sich Elemente überdecken, wenn Tänzer zu nahe beieinander stehen. Er schlug deswegen vor, die Größe der graphischen Elemente (z.B. den Kreisradius) für den Nutzer variabel zu gestalten.

**Sind die Damen in diesem Bild (bezieht sich auf die Aufgabe zuvor) den Herren mit dem Körper zugewandt oder nicht?**

Die Beobachtungen sind identisch zur Aufgabe zuvor. Alle Teilnehmer konnten die Aufgabe lösen.

**Betrachten Sie Bild 2. Richten Sie die Damen nun so aus, dass sie in Richtung der Herren stehen und schauen.**

Alle Teilnehmer lösten die Aufgabe erfolgreich. Einer der Teilnehmer fand den Dialog zur Änderung von Blickrichtung und Ausrichtung des Körpers nicht und erwartete, dass man die Tänzer direkt in der Visualisierung mit der Maus drehen kann.

**Welche Pose haben die Damen in Bild 3 eingenommen?**

Die Aufgabe konnte von allen Teilnehmern gelöst werden. Alle Teilnehmer wechselten dazu in den 3D-Modus und beschrieben die Posen der Damen korrekt. Die Nutzung der Kamera in der 3D-Szene stellte für keinen der Teilnehmer ein Problem dar. Drei der Teilnehmer versuchten, die Damen in der 3D-Szene auszuwählen. Sie probierten dabei entweder die 3D-Modelle mit der Maus anzuklicken oder eine Auswahl aus der 2D-Visualisierung in 3D-Ansicht zu transportieren. Außerdem versuchten sie, die Damen über den Button mit der Aufschrift „Ladies“ im Poseneditor auszuwählen. Ein Teilnehmer setzte dadurch die Posen der Damen zurück und beschrieb deswegen die Standardpose anstatt die, für die Choreografie erstellte, Pose.

**Sorgen Sie nun dafür, dass im gleichen Bild alle Herren den linken Arm heben.**

Jeder der Teilnehmer löste die Aufgabe auf die vorgesehene Weise. Die Funktion zum Ändern von Posen wurde von drei der vier Teilnehmer in der Diskussionsrunde gelobt. Jedoch gab es immer noch den Nachsatz, dass nicht alle Trainer diesen Aufwand (v.a. in der Praxis) investieren würden. Zwei der Teilnehmer speicherten die Pose ab. Einer davon erwähnte die Funktion nochmals lobend in der Diskussionsrunde und beschrieb ein Anwendungsszenario, bei dem zunächst eine Bibliothek von möglichen Posen erstellt wird und daraufhin die eigentlichen Bilder erstellt werden. Dann könnten die vorgefertigten Posen auf die Tänzer übertragen werden.

**Finden Sie eine potenzielle Kollision von zwei Tänzern in der Choreografie und erklären Sie, wieso es sich hier um eine Kollision handelt.**

Die Aufgabe konnte von allen Teilnehmern gelöst werden. Zwei der Teilnehmer erkannten Kollisionen mithilfe der Farbübergänge auf den Transitionen. Diese Teilnehmer lobten die Funktion in der Diskussion später nochmals gesondert, da sich Kollisionen so besonders leicht erkennen lassen. Ein weiterer Teilnehmer nutzte ebenfalls die Transitionen, aber ignorierte die Farbübergänge. Er zweifelte an, ob die linearen Transitionen reale Bildübergänge exakt genug modellieren können, um sinnvolle Aussagen über Kollisionen treffen zu können. Der vierte Teilnehmer in dieser Aufzählung nutzte die Funktion zum Animieren von Bildübergängen, um Kollisionen zu erkennen. Der Teilnehmer wurde in der Diskussionsrunde gefragt, ob er die Bedeutung des Farbverlaufs auf den Transitionen verstanden hätte, was er verneinte. Allgemein fiel auf, dass die Teilnehmer Schwierigkeiten hatten, die Visualisierung zu verstehen, wenn die visuellen Elemente sich zu sehr überlappten. Ein Teilnehmer teilte in der Diskussionsrunde mit, dass in seiner Formation Kollisionen gar kein Problem seien, da bei ihrer Choreografieerstellung nicht zuerst die Bilder, sondern die Schritte und Figuren existieren, die getanzt werden. Diese werden dann so angepasst, dass in einem weiteren Schritt die Bilder entstehen.

### **Beim Übergang von Bild 19 auf 20: Ändern sie den Laufweg von Paar 3, sodass dieses nicht mehr mit Paar 7 kollidiert.**

Die Aufgabe wurde von zwei der vier Teilnehmer auf die vorgegebene Art gelöst. Dabei sollten Zwischenstopps in den Laufweg von Paar 3 eingefügt und Positionen dieser Zwischenstopps dann per Drag-and-Drop verändert werden. Zwei Teilnehmer veränderten nicht die Zwischenstopps, sondern tauschten Paare im Bild und dann auch in den vorherigen Bildern, um dort Kollisionen zu vermeiden. Dies kann aufwändig sein, da das Tauschen von zwei Paaren oder Tänzern Auswirkungen auf alle Bildübergänge in der Choreografie hat. Ein Teilnehmer schlug deswegen eine Funktion vor, mit der zwei Tänzer oder Paare in jedem Bild der Choreografie die Plätze tauschen. Die Teilnehmer, welche den Laufweg von Paar 3 änderten, kritisierten die Nutzbarkeit beim Einfügen von Zwischenstopps. Beide Teilnehmer versuchten zuerst, einen Zwischenstopp direkt auf dem Übergang von Paar 3 einzufügen. Beide hielten das Einfügen von Zwischenstopps über die Zeitleiste für unintuitiver als ein direktes Einfügen in die Visualisierung. Auch merkten sie an, dass es möglich sein sollte, nur von einem Paar den Übergang zu ändern. Zwei der Teilnehmer lobten explizit die Möglichkeit eine abgeschwächte Form eines Bildes einzufügen, welches nur Punkte für den Laufweg der Tänzer definiert. Zwei der Teilnehmer waren sich nicht sicher, ob im Transitionsmodus das vorherige oder das nächste Bild zusammen mit dem aktuellen Bild angezeigt wird.

### **Animieren Sie nun diesen Übergang**

Die Aufgabe konnte von allen Teilnehmern gelöst werden. Einer der Teilnehmer benötigte Zeit für die Suche nach dem Button zum Animieren, da dieser nur im Ansichtsmodus verfügbar ist. Er versuchte dabei immer wieder, die Animation direkt über die Zeitleiste zu starten. Zwei Teilnehmer waren sich nicht sicher, ob die Animation vom Startbild oder vom Zielbild aus beginnt und fanden dies schlussendlich durch Ausprobieren heraus. Die anderen beiden Teilnehmer hingegen bemängelten, dass bei Übergängen, in denen sich Paare trennen oder vereint werden, der Punkt, an dem dies geschieht, nicht selbst spezifiziert werden kann. Ein weiterer Teilnehmer schlug vor, die Animationen auch im 3D-Modus abzuspielen, da dies laut ihm in anderen Programmen zur Bilderstellung möglich sei.

### **Speichern Sie die Choreografie ab.**

Bei dieser Aufgabe fiel auf, dass keiner der Teilnehmer den Button zum Speichern auf Anhieb fand. Alle Teilnehmer befanden sich wegen der vorherigen Aufgabe noch im Ansichtsmodus und nicht im Editiermodus. In der Diskussionsrunde kritisierten alle Teilnehmer, dass Funktionen wie das Speichern der Choreografie nur in bestimmten Modi verfügbar sind.

### **Allgemeines Feedback**

Grundsätzlich lobten alle Teilnehmer die einfache Nutzung des Prototyps. Insbesondere die Werkzeugleiste links neben der Visualisierung wurde von zwei Teilnehmern positiv hervorgehoben. Zur Navigation durch die Choreografie, also das Wechseln zwischen den Bildern, nutzte nur ein Teilnehmer vorwiegend die Werkzeugleiste. Die anderen drei Teilnehmer benutzten für diesen Zweck hauptsächlich die Zeitleiste. Zwar wurde sie aufgrund des Fehlens in anderen Programmen positiv wahrgenommen, jedoch kritisierten drei Teilnehmer die Vereinfachung in Takte mit jeweils acht Schlägen. Laut ihnen kann es vorkommen, dass die Musik einen Wechsel des Tanzes nach weniger als acht Schlägen vorsieht und somit auch ein neuer Takt begonnen werden muss. Außerdem existiert für Teams ab der 2. Bundesliga die Möglichkeit sogenannte *Highlights* in ihre Choreografie einzubauen. Ein Highlight kann zum Beispiel eine Lankenau-Drehung sein, bei welcher es sich um



eine Pirouette der Damen auf einem Bein handelt. Diese Highlights können länger als acht Takte gehen und werden kontinuierlich bis zum Ende weiter gezählt. Zwei Teilnehmer kritisierten des Weiteren, dass Bilder nicht in der Mitte der Choreografie eingefügt werden können.

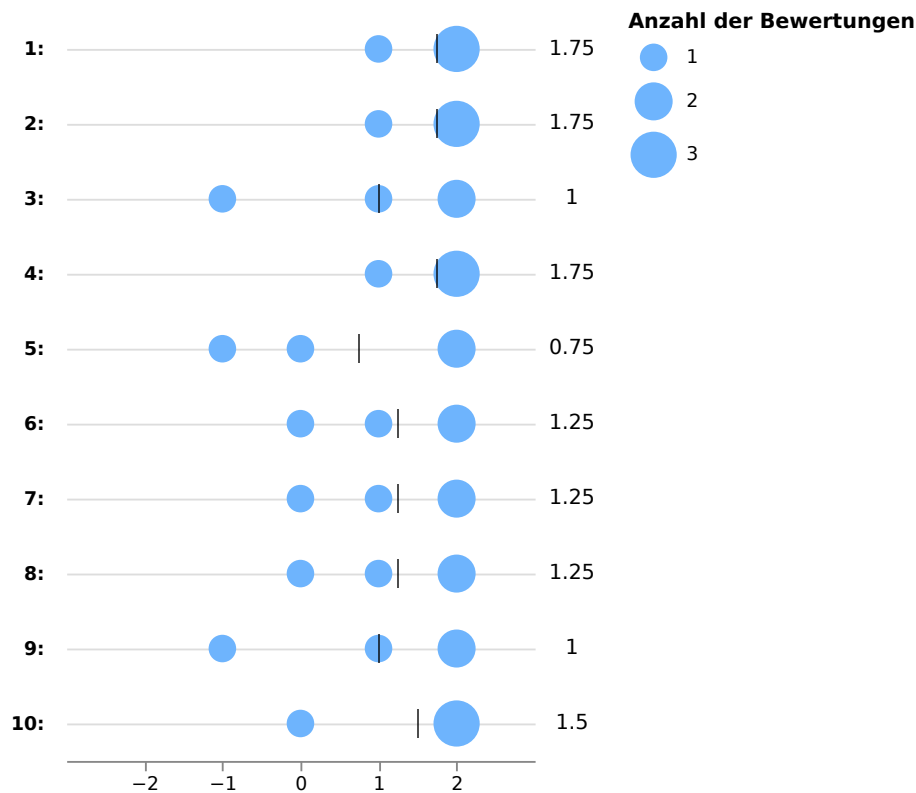
Die quantitativen Ergebnisse, welche über die Bewertung von Aussagen am Ende zustande kamen, sind in Abbildung 6.2 zu sehen. Die Aussagen waren die Folgenden:

1. Die Darstellung von Bildern aus Choreografien ist einfach und intuitiv.
2. Die Positionen einzelner Tänzer können einfach abgelesen werden.
3. Die Darstellung von Blickrichtung und Ausrichtung des Körpers sind einfach und intuitiv.
4. Die Bewertung der Ausnutzung der Tanzfläche ist einfach und intuitiv.
5. Das visuelle Erkennen von Kollisionen hilft bei der Erstellung der Choreografie.
6. Das Erstellen neuer Bilder ist einfach und kann ohne hohen Aufwand durchgeführt werden.
7. Die Bedienung des Prototyps ist im Zusammenhang mit der Zeitleiste einfach und intuitiv.
8. Das Erstellen von Posen in 3D ist einfach und intuitiv.
9. Eine 3D-Ansicht hat einen Mehrwert und sollte parallel zur 2D-Ansicht existieren.
10. Das Einfügen von Zwischenstopps und die Animation der Choreografie sind einfach und intuitiv.

## 6.4 Diskussion

Im Folgenden werden die Ergebnisse analysiert und diskutiert. Außerdem sollen mögliche Lösungsansätze genannt werden. Grundsätzlich war es den Experten möglich, die Choreografie mit vertretbarem Aufwand zu editieren und zu analysieren. Es fiel insbesondere auf, dass die Usability einen maßgeblichen Einfluss darauf hatte, wie die Experten eine Funktion bewerteten. Ihre Bereitschaft, die Funktion in ihren eigenen Arbeitsablauf zu integrieren, hing hauptsächlich von der Nutzbarkeit der Funktion ab. Es fiel insbesondere auf, dass die Aufteilung in einen Ansichts- und einen Editiermodus vorwiegend Schwierigkeiten mit sich bringt. So suchten die Teilnehmer in der Studie immer wieder den Button zum Animieren der Choreografie oder waren verwundert, wieso das Drag-and-Drop auf einmal nicht mehr funktioniert. Die Aufteilung in zwei Modi sollte somit nur für unterschiedliche Geräte erfolgen, das heißt die Nutzung des Ansichtsmodus sollte nur auf mobilen Endgeräten möglich sein. Eine Alternative wären eigene Accounts mit Berechtigungsstufen für Trainer und Tänzer. Beim Laden der Choreografie sollte zudem ein Dialog erscheinen, um Nutzern visuell klar zu machen, dass der Ladevorgang erfolgreich war.

Die Zerteilung des Programms in verschiedene Modi, um visuelle Überladung zu vermeiden, wurde von den Teilnehmern gelobt und kann als sinnvoll angesehen werden. Jedoch sollten gerade beim Visualisieren der Transitionen weitere Möglichkeiten angeboten werden, die Anzahl der graphischen Elemente zu reduzieren. Bisher ist es nur möglich einen Übergang auszuwählen und die anderen auszublenden. Dies könnte man mit dem Brush kombinieren, um eine Reihe von Übergängen auszuwählen.



**Abbildung 6.2:** Bewertungen gegebener Aussagen auf einer Likert-Skala nach der Nutzung des Prototyps. 2 steht für „Ich stimme voll und ganz zu“ und -2 steht für „Ich stimme überhaupt nicht zu“ steht. Der schwarze Strich steht für das arithmetische Mittel der Bewertungen aller Teilnehmer. Dieser Wert wurde rechts daneben auch nochmals in Textform dazu geschrieben.

Die Wahl der visuellen Kodierungen kann ebenfalls als sinnvoll bewertet werden. Die Positionen und Ausrichtungen der Tänzer, die Laufwege oder die Rotation ganzer Bilder lassen sich leicht visuell ablesen. Dies kann auch anhand der Zustimmungswerte für die Aussagen 1-3 in Abbildung 6.2 abgelesen werden. Es stellte sich jedoch heraus, dass für die visuellen Abbildungen wie die Farbe der Symbole für Damen und Herren mehr Hilfestellungen benötigt werden. Hierfür könnten Legenden neben der Visualisierung oder Tutorialeinblendungen bei der erstmaligen Nutzung des Systems helfen. Auffällig war, dass alle visuellen Kodierungen, die bereits im Papierprozess vorhanden waren und im Prototyp angewandt wurden, von den Teilnehmer der Studie sofort verstanden wurden. Dies deutet darauf hin, dass eine allgemeingültige, abstrakte, graphische Syntax im Lateininformationssport helfen würde, um das Verständnis der visuellen Notationen zu vereinfachen. Des Weiteren hat sich herausgestellt, dass es wichtig ist, die Größe der visuellen Elemente an das aktuelle Bild anpassen. Elemente, die sich überlappen, haben vor allem bei der Betrachtung von Bildübergängen oft dazu geführt, dass die Information nicht erkennbar war. Dieses Problem war auch bei der Darstellung von Blickrichtung und Ausrichtung des Körpers von Tänzern präsent.

Die Analyse der Ausnutzung der Tanzfläche mit einer Heatmap und die Analyse der Laufristanzen mit einem Balkendiagramm wurden ebenfalls von den Teilnehmern gelobt und konnten von ihnen korrekt angewandt werden. Die Freude über solche Funktionen kann zum Teil auch daher kommen,

dass solche Funktionen in noch keinem bisher verfügbaren Programm integriert waren und daher so viel Zustimmung erhielt. Auf die Nachfrage, welche weiteren Analysemöglichkeiten das Programm bieten sollte, fiel den Teilnehmern keine Antwort ein. Dies bestätigt auch der Zustimmungswert von 1.75 für Aussage 4 in Abbildung 6.2. Die fehlende Anwendung der kontinuierlichen Heatmap könnte auch daran liegen, dass man im Einstellungsmenü zu ihr wechseln muss. Möglicherweise sollte diese Option prominenter neben der Visualisierung im Analysemodus erscheinen.

Die Darstellung von Transitionen zur Ermittlung von Kollisionen wurde ebenfalls positiv aufgenommen, jedoch ist hier der Zustimmungswert von 0.75 für Aussage 5 in Abbildung 6.2 etwas niedriger als bei den anderen Aussagen. Dies könnte daran liegen, dass das Problem in manchen Formationen gar nicht existiert oder der Farbübergang auf den Transitionen ebenfalls nicht offensichtlich genug erklärt wurde. Hier könnte man auch mit einer Legende neben der Visualisierung arbeiten. Des Weiteren sollte das Anpassen der Transitionen überarbeitet werden, sodass Nutzer Zwischenstopps ohne einen Umweg über die Zeitleiste direkt in der Visualisierung einfügen können. Zwar liegt der Zustimmungswert für Frage 10 in Abbildung 6.2 bei 1.5, jedoch hatten Nutzer oft Probleme dabei, Zwischenstopps einzufügen. Ebenfalls sollte Nutzern verdeutlicht werden, welche Bilder und welcher Übergang im Transitionsmodus momentan angezeigt werden. Dies könnte man zum Beispiel über das visuelle Highlighting der Bilder und des Übergangs in der Zeitleiste bewerkstelligen.

Das Einfügen neuer Bilder konnte von den Teilnehmern problemlos durchgeführt werden und hat einen Zustimmungswert von 1.25 für Frage 6 in Abbildung 6.2. Gerade die Funktionen zum Speichern und Wiederverwenden von Bildern und dem Duplizieren existierender Bilder stießen auf große Zustimmung. Sobald das neue Bild erstellt wurde, sollte das Programm automatisch in den normalen Modus wechseln, um das Editieren des Bildes zu erlauben und dem Nutzer zu verdeutlichen, dass der Erstellvorgang erfolgreich war. Beim Editieren der Bildern fiel auf, dass alle Interaktionsmechanismen, die direkt in der Visualisierung stattfanden, den Nutzern sehr einfach fielen. Dazu zählen insbesondere das Drag-and-Drop und die Mehrfachauswahl von Elementen. Andere Funktionen, deren Interaktion nicht in die Visualisierung integriert waren, benötigten deutlich länger, um von den Teilnehmern richtig angewandt zu werden.

Die Bedienung der Zeitleiste stellte für die Teilnehmer kein Problem dar (Zustimmungswert von 1.25 für Frage 7 in Abbildung 6.2). Die Zeitleiste war ursprünglich dafür gedacht, Bilder zeitlich in der Choreografie zu verorten, erwies sich aber während der Studie als ein Mittel zur effizienten Navigation durch die Choreografie. Aufgrund der Kritik der Teilnehmer sollte man das Design der Zeitleiste überdenken und die Vereinfachung mit acht Schlägen pro Takt abschaffen. Ein sinnvoller Lösungsansatz, der ebenfalls mit einem der Teilnehmer diskutiert wurde, ist, dass man der Zeitleiste keine Takte, sondern ganze Tänze über den Button mit dem Plus-Symbol hinzufügt. Dort könnte man dann den Tanz bzw. das Highlight integrieren und danach spezifizieren wie lange der nächste Abschnitt geht (in Anzahl der Takte) und wie viele Schläge ein Takt haben soll. Außerdem benötigt die Zeitleiste eine direktere Interaktion, zum Beispiel das Drag-and-Drop von Bildern, und mehr Möglichkeiten, um eine iterative Bildentwicklung zuzulassen. Dazu gehört das Einfügen von Bildern mitten in der Choreografie und nicht erst am Ende. Auch sollten Zwischenstopps wieder gelöscht werden können, ohne dass das gesamte nachfolgende Bild gelöscht wird.

Der 3D-Modus wurde vor allem in der Diskussion besonders positiv hervorgehoben und hat einen Zustimmungswert von 1.25 für Aussage 8 in Abbildung 6.2. Laut den Experten sollte dieser parallel zu einer 2D-Ansicht existieren (Zustimmungswert 1 für Aussage 9 in Abbildung 6.2), da dieser weitere zusätzliche Informationen kodiert. Die manuelle Definition von Posen kannten die Teilnehmern nicht aus anderen Programmen und stieß deswegen potenziell auch auf größere

Zustimmung. Jedoch existiert hier Verbesserungspotenzial hinsichtlich der Usability. Die Interaktion sollte so optimiert werden, dass sie direkt in der 3D-Ansicht stattfindet. Tänzer sollten mit der Maus direkt über das Anklicken der Modelle selektiert werden können. Außerdem sollte es möglich sein, die einzelnen Gelenke im Poseneditor direkt im Modell anwählbar zu machen. Weiterhin sollte der Dialog zur Erstellung von Posen überarbeitet werden, sodass Nutzer das Selektieren von Tänzern und das Anwenden von Posen auf die Tänzer nicht mehr miteinander verwechseln.

Die Anforderungen an den Prototyp hingen stark von der Vorerfahrung der Teilnehmer ab. So kritisierten nur Teilnehmer, die bereits in der 2. Bundesliga oder höher getanzt hatten, dass die Gitterauflösung für das Drag-and-Drop zu niedrig ist und dass eine Vereinfachung mit acht Schlägen pro Takt in der Zeitleiste nicht angebracht ist. Die unterschiedlichen Anforderungen dieser Zielgruppen lassen sich hierbei nur schwer vereinen. Die Vorerfahrung der Teilnehmer mit technischen Lösungen zur Choreografieerstellung hat ebenfalls Einfluss darauf, wie sehr gewisse Funktionen angenommen werden. Die Teilnehmer, die eher dazu bereit waren, wirklich Zeit in das Programm zu investieren (z.B. indem sie Posen im 3D-Modus erstellten), haben den Prototyp tendenziell besser bewertet. Um solche Probleme zu umgehen, könnte man einen Anfängermodus einbauen, in dem Funktionen wie die 3D-Ansicht deaktiviert sind, um diese nicht zu überfordern.

Grundsätzlich gab es innerhalb des Feedbacks aller Teilnehmer trotzdem viele Gemeinsamkeiten, die beim Design der visuellen Kodierungen und den Interaktionsmechanismen beachtet werden sollten. Wir haben im Folgenden versucht, dieses Feedback in neun Design-Regeln zu verallgemeinern. Überschneidungen mit anderen Leitfäden für Usability, zum Beispiel den *Eight Golden Rules of Interface Design* von Shneiderman [Shn98], sind durchaus gewollt. Zudem betonen sie nur nochmal, dass ein Programm, welches hauptsächlich von Tanzlehrern ohne technischen Hintergrund benutzt wird, eine konsistente, einfache und verständliche Benutzeroberfläche haben sollte:

1. **Imitiere den Papierprozess:** Tänzer und Trainer verstehen alle visuellen Kodierungen einfacher, die bereits im Papierprozess benutzt werden.
2. **Erkläre eigene visuelle Kodierungen:** Man kann nicht davon ausgehen, dass Nutzer eigens definierte visuelle Kodierungen nach einmaliger Erklärung bereits verstehen. Verwende Legenden, Pop-Ups oder Tutorials, um visuelle Kodierungen zu erklären.
3. **Nutze Metaphern zur Interaktion:** Interaktive Elemente sollten auf existierenden Metaphern, zum Beispiel dem Desktop, existieren. Die Zielgruppe dieser Programme ist oft nicht technikaffin, weswegen komplizierte Interaktionen nur spärlich oder gar nicht eingesetzt werden sollten.
4. **Vermeide visuelle Überladung:** Es sollten nicht zu viele graphische Elemente in der Visualisierung angezeigt werden. Falls sich dies nicht umgehen lässt, sollten interaktive Werkzeuge wie adaptives Highlighting von Elementen oder die Änderung der Größe graphischer Elemente angeboten werden.
5. **Aktionen brauchen visuelle Reaktionen:** Jede Aktion sollte dem Nutzer visuelles Feedback geben.
6. **Direkte Manipulation:** Die Interaktion sollte so weit wie möglich direkt in der Visualisierung erfolgen und nicht über separate Dialoge.
7. **Einmal definieren, überall verwenden:** Daten wie Bilder oder Posen, sollten nur einmal aufwändig vom Nutzer manuell generiert und danach wiederverwendet werden.

8. **Kontextabhängiges Editieren:** Jede Funktion, die Änderungen an einem Bild vornimmt, sollte eine Option anbieten, sodass die Änderung für alle Tänzer, für alle Damen, für alle Herren oder nur für einen Tänzer angewandt wird.
9. **Sequenziell zuerst, iterativ bei Bedarf:** Choreografien werden in der Regel sequenziell vom ersten zum letzten Bild hin gebaut und nur bei Bedarf während des Trainings geändert. Ein Programm sollte somit hauptsächlich den sequentiellen Prozess unterstützen, zum Beispiel indem Daten aus dem letzten Bild übernommen werden. Jedoch sollten zwischen Bildern ebenfalls neue Bilder eingefügt werden können.

## 6.5 Einschränkungen der Validität

Bei der Wahl des Studiendesigns können bestimmte Limitierungen die Ergebnisse der vorherigen Kapitel beeinflusst haben. Diese werden im Folgenden kurz genannt und diskutiert. Als erstes ist die Anzahl der Teilnehmer zu nennen, hier vier Personen. Diese Zahl ist vor allem im Vergleich zu quantitativen Studien nicht repräsentativ. Jedoch argumentieren Tory und Moller [TM05], dass bereits wenige Experten ausreichen, um die Nutzbarkeit einer Visualisierungslösung zu evaluieren. Man sollte jedoch beachten, dass mit Experten nicht nur Domänenexperten, sondern auch Experten in den Bereichen Visualisierung und Mensch-Computer-Interaktion gemeint sind. Des Weiteren ist eine Beschränkung auf wenige Domänenexperten im Bereich der Sportvisualisierung nicht unüblich. So evaluierten Stein et al. [SJL+18] in ihrer Arbeit eine Visualisierungslösung im Bereich der Fußballanalyse ebenfalls mit vier Domänenexperten. Auch kann die Stichprobe an Teilnehmern in unserem Fall als repräsentativ angesehen werden, da Formationstrainer aus allen Ligen, von der Landesliga bis in die 1. Bundesliga und zu Weltmeisterschaften an der Studie teilgenommen haben. Des Weiteren haben Frauen und Männer teilgenommen, die sich alle in unterschiedlichen Altersbereichen befinden und unterschiedliche Levels an Vorerfahrung mitbrachten. Es kann auch erwähnt werden, dass einer der Teilnehmer bereits zuvor in den ersten beiden Interviews als Stakeholder am Projekt beteiligt war und bei der Anforderungsanalyse und bei der Verbesserung des Systems half. Es kann deswegen nicht ausgeschlossen werden, dass der Teilnehmer aus diesem Grund positiveres Feedback gab. Es ist zudem davon auszugehen, dass seine Leistung während der Bearbeitung der Aufgaben deutlich höher ausfiel, da er bereits mehr Vorerfahrung mitbrachte.

Ein weiterer Punkt sind die variierenden Umgebungen, in denen die Teilnehmer an der Studie teilgenommen hatten. Um so viele Teilnehmer wie möglich zu rekrutieren, wurde ihnen angeboten, entweder virtuell oder vor Ort an der Universität an der Studie teilzunehmen. Teilnehmer, die virtuell teilnahmen, hatten öfters Probleme sich an das *Think Aloud* Protokoll zu halten. Ob dies tatsächlich an der Online-Teilnahme lag oder an der Persönlichkeit dieser Teilnehmer, bleibt unklar. Darüber hinaus können Faktoren wie die Ablenkung durch die Umgebung virtuell deutlich schlechter als im Labor kontrolliert werden.

Darüber hinaus ist ein weiterer Punkt die Sprache, in der das Interview stattgefunden hat. Die Fragen waren alle auf Deutsch gestellt und wurden von den Teilnehmern auch auf Deutsch beantwortet. Alle Beschriftungen im Prototyp hingegen waren auf Englisch. Teilnehmer fragten deswegen sowohl während der Trainingsphase, als auch während der Bearbeitung der Aufgaben immer wieder nach Übersetzungen. Dies könnte die Leistung einzelner Teilnehmer bei der Bearbeitung der Aufgaben negativ beeinflusst haben.

Weiterhin fiel auf, dass die Teilnehmer bei der Bearbeitung einige Funktionen ignorierten, insbesondere das Skalierwerkzeug und die kontinuierliche Heatmap. Des Weiteren äußerte ein Teilnehmer bereits während der Trainingsphase, also bei der freien Exploration des Prototyps, Lob und Kritik, die in der Diskussionsrunde nochmal aufgegriffen wurde. Daher sollte bei einer weiteren Evaluation überlegt werden, ob eine freie Exploration sinnvoll scheint, sodass der Teilnehmer alle Funktionen nutzen und praxisnahes Feedback geben kann.

Als letzter Punkt sollte noch der Fokus dieser Studie genannt werden. Da die Teilnehmer eine vorgegebene Liste an Aufgaben abarbeiteten, lassen sich keine Aussagen über die tatsächliche Nutzung außerhalb eines Laborumfelds treffen. Insbesondere wurde nicht untersucht, ob Formationstrainer im Prototyp komplette Choreografien mit vertretbarem Aufwand erstellen können. Außerdem wurde die Nutzung des Prototyps auf mobilen Geräten während des Trainings nicht untersucht.

## 7 Zusammenfassung und Ausblick

In dieser Arbeit wurde in Zusammenarbeit mit Formationstrainern ein Prototyp entwickelt, mit welchem Choreografien für Lateinformationen erstellt, analysiert und mit Tänzern kommuniziert werden können. Dafür wurden in einem ersten Interview Anforderungen erhoben, die ein digitales Werkzeug für diese Zwecke haben sollte. Besonderes Augenmerk wurde hierbei auf die Positionen der Tänzer, deren Blickrichtung und Körperrotation, die Darstellung von Bildübergängen, die Spezifikation von Posen in einer 3D-Ansicht und die zeitliche Verordnungsart von Bildern gelegt. Der Prototyp wurde in einem zweiten Interview von den Formationstrainern bewertet und anschließend verbessert. In einer qualitativen Evaluation wurde der Prototyp von vier Domänenexperten unter Anwendung der *Think Aloud* Methode genutzt und bewertet. Dabei stellte sich heraus, dass die verwendeten visuellen Kodierungen geeignet sind, Daten wie die Positionen der Tänzer und deren Blick- bzw. Körperausrichtung zu vermitteln. Die Analyse von Laufdistanzen zwischen Bildern und der Ausnutzung der Tanzfläche stellten sich als intuitiv und nützlich heraus. Die 3D-Ansicht stieß größtenteils auf positives Feedback. Jedoch zeigte sich, dass das Erstellen von Posen mit hohem Aufwand verbunden ist. Die Erstellung von Zwischenstopps und Bildübergängen und die damit verbundene, visuelle Erkennung von Kollisionen wurden von den Domänenexperten zwar positiv aufgenommen, jedoch sollten hier die visuelle Kodierung erklärt und die Interaktion mit dem Prototyp überarbeitet werden. Basierend auf dem gesammelten Feedback wurden neun Designrichtlinien formuliert, die zukünftige Programme zur Choreografieerstellung beachten sollten.

Für zukünftige Arbeiten sollte der Prototyp anhand des gesammelten Feedbacks überarbeitet werden. Zudem muss die Interaktion auf mobilen Geräten optimiert werden. Des Weiteren schlagen Sedlmair et al. [SMM12] in ihrem Workflow-Modell vor, die Visualisierung in einer Feldstudie zu untersuchen, zum Beispiel mit einer ethnographischen Studie. In dieser könnte man Daten darüber sammeln, ob und wie Tänzer die Anwendung im Training einsetzen. Dies konnte aufgrund des zeitlichen Rahmens der Arbeit nicht gemacht werden. Zudem empfehlen Tory und Moller [TM05] bei einer Evaluation mit Experten nicht nur Domänenexperten, sondern auch Experten aus den Bereichen Visualisierung und Mensch-Computer-Interaktion heranzuziehen. Dies könnte bei der weiteren Evaluation des Prototyps helfen. Weitere Aspekte, die berücksichtigt werden könnten, wären beispielsweise die Einbindung der verwendeten Musik in das Programm. Für die Vereinfachung der Posenerstellung in 3D könnte man untersuchen, wie man Techniken aus dem Bereich der Bilderkennung und künstlichen Intelligenz einsetzen könnte, um Posen oder Bewegungsabläufe mit einer Webcam aufzunehmen und direkt auf ein 3D-Modell zu übertragen. Zudem wurde die Choreografie bisher nur als präskriptives Modell betrachtet, welchem die Tänzer in einem Formationsdurchgang folgen. In einem nächsten Schritt wäre es möglich, den Formationsdurchgang aufzunehmen, per Bilderkennung die Positionen der Tänzer zu bestimmen, und als deskriptives Modell wieder in das Programm einzubinden. Die beiden Modelle könnte man dann nebeneinander

darstellen, um zu bewerten, wie exakt die getanzte Choreografie und die vorgegebenen Positionen der Bilder eingehalten wurden. Hierbei wäre es auch denkbar die erstellten Daten in erweiterter oder virtueller Realität darzustellen.



# Literaturverzeichnis

- [ADT23] Allgemeiner Deutscher Tanzlehrerverband e.V. *Webseite ADTV*. 2023. URL: <https://adtv.de/> (zitiert auf S. 17).
- [AKD+21] V. Arpatzoglou, A. A. Kardara, A. Diehl, B. Flueckiger, S. Helmer, R. Pajarola. „Dancemoves: A visual analytics tool for dance movement analysis“. In: (2021) (zitiert auf S. 36).
- [BLC+16] A. Bialkowski, P. Lucey, P. Carr, I. Matthews, S. Sridharan, C. Fookes. „Discovering Team Structures in Soccer from Spatiotemporal Data“. In: *IEEE Transactions on Knowledge and Data Engineering* 28.10 (2016), S. 2596–2605. DOI: [10.1109/TKDE.2016.2581158](https://doi.org/10.1109/TKDE.2016.2581158) (zitiert auf S. 35).
- [Boo23] Open Source. *Bootstrap*. 2023. URL: <https://getbootstrap.com/> (zitiert auf S. 50).
- [BR00] T. Boren, J. Ramey. „Thinking aloud: Reconciling theory and practice“. In: *IEEE transactions on professional communication* 43.3 (2000), S. 261–278 (zitiert auf S. 46, 47, 75).
- [Cha03] E. Charters. „The use of think-aloud methods in qualitative research an introduction to think-aloud methods“. In: *Brock Education Journal* 12.2 (2003) (zitiert auf S. 47).
- [Chi05] E. Chi. „Introducing wearable force sensors in martial arts“. In: *IEEE Pervasive Computing* 4.3 (2005), S. 47–53. DOI: [10.1109/MPRV.2005.67](https://doi.org/10.1109/MPRV.2005.67) (zitiert auf S. 39).
- [CLRS22] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. *Introduction to algorithms*. MIT press, 2022 (zitiert auf S. 27).
- [CSS11] K. Carlson, T. Schiphorst, C. Shaw. „ActionPlot: A Visualization Tool for Contemporary Dance Analysis“. In: *Proceedings of the International Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*. CAe ’11. Vancouver, British Columbia, Canada: Association for Computing Machinery, 2011, S. 113–120. ISBN: 9781450309080. DOI: [10.1145/2030441.2030466](https://doi.org/10.1145/2030441.2030466). URL: <https://doi.org/10.1145/2030441.2030466> (zitiert auf S. 36).
- [Deu23a] Deutscher Tanzsportverband e.V. *Turnier- und Sportordnung des Deutschen Tanzsportverbandes e.V.* 2023. URL: <https://www.tanzsport.de/de/service/downloads-und-formulare/verband/satzungen-und-ordnungen> (zitiert auf S. 21).
- [Deu23b] Deutscher Tanzsportverband e.V. *Wertungsrichtlinien im DTV für Formationswettbewerbe Standard und Latein*. 2023. URL: <https://www.tanzsport.de/files/tanzsport/downloads/sportwelt/formationen/wrichtl-f.pdf> (zitiert auf S. 24).
- [Fir23] Google LLC. *Firebase*. 2023. URL: <https://firebase.google.com/> (zitiert auf S. 50).
- [Gar11] B. S. Gardner. „Responsive web design: Enriching the user experience“. In: *Sigma Journal: Inside the Digital Ecosystem* 11.1 (2011), S. 13–19 (zitiert auf S. 33).

- [Gra72] R. L. Graham. „An efficient algorithm for determining the convex hull of a finite planar set“. In: *Info. Proc. Lett.* 1 (1972), S. 132–133 (zitiert auf S. 25).
- [GRL15] C. F. Griggio, M. Romero, G. Leiva. „Towards an Interactive Dance Visualization for Inspiring Coordination Between Dancers“. In: *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA '15. Seoul, Republic of Korea: Association for Computing Machinery, 2015, S. 1513–1518. ISBN: 9781450331463. DOI: [10.1145/2702613.2732925](https://doi.org/10.1145/2702613.2732925). URL: <https://doi.org/10.1145/2702613.2732925> (zitiert auf S. 36).
- [Han23] Hans-Joachim Schäfer. *Die Geschichte des Formationstanzens*. 2023. URL: <https://www.tanzsport.de/de/sportwelt/formationen-standard-latein/faszination> (zitiert auf S. 21).
- [LTB16] A. G. Losada, R. Therón, A. Benito. „BKViz: A Basketball Visual Analysis Tool“. In: *IEEE Computer Graphics and Applications* 36.6 (2016), S. 58–68. DOI: [10.1109/MCG.2016.124](https://doi.org/10.1109/MCG.2016.124) (zitiert auf S. 36).
- [Mix23] Adobe Inc. *Mixamo*. 2023. URL: <https://www.mixamo.com/#/> (zitiert auf S. 50).
- [Muh09] M. N. Muhammad. „Visualizing Dance Formations: The Choreographer’s Tool“. In: (2009) (zitiert auf S. 38).
- [Mun09] T. Munzner. „A Nested Model for Visualization Design and Validation“. In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (2009), S. 921–928. DOI: [10.1109/TVCG.2009.111](https://doi.org/10.1109/TVCG.2009.111) (zitiert auf S. 73, 74).
- [NN13] M. Nebeling, M. C. Norrie. „Responsive design and development: methods, technologies and current issues“. In: *Web Engineering: 13th International Conference, ICWE 2013, Aalborg, Denmark, July 8-12, 2013. Proceedings 13*. Springer. 2013, S. 510–513 (zitiert auf S. 33).
- [OJ13] S. G. Owens, T. Jankun-Kelly. „Visualizations for exploration of american football season and play data“. In: *The 1st workshop on sports data visualization. IEEE*. Bd. 1. 2013, S. 15 (zitiert auf S. 36).
- [Par62] E. Parzen. „On estimation of a probability density function and mode“. In: *The annals of mathematical statistics* 33.3 (1962), S. 1065–1076 (zitiert auf S. 28).
- [PM06] M. Page, A. Moere. „Towards Classifying Visualization in Team Sports“. In: *International Conference on Computer Graphics, Imaging and Visualisation (CGIV'06)*. 2006, S. 24–29. DOI: [10.1109/CGIV.2006.85](https://doi.org/10.1109/CGIV.2006.85) (zitiert auf S. 35, 39).
- [PSBS12] H. Pileggi, C. D. Stolper, J. M. Boyle, J. T. Stasko. „SnapShot: Visualization to Propel Ice Hockey Analytics“. In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012), S. 2819–2828. DOI: [10.1109/TVCG.2012.263](https://doi.org/10.1109/TVCG.2012.263) (zitiert auf S. 36).
- [PVS+18] C. Perin, R. Vuillemot, C. D. Stolper, J. T. Stasko, J. Wood, S. Carpendale. „State of the art of sports data visualization“. In: *Computer Graphics Forum*. Bd. 37. 3. Wiley Online Library. 2018, S. 663–686 (zitiert auf S. 35, 39).
- [Rea23] Open Source. *React*. 2023. URL: <https://reactjs.org/> (zitiert auf S. 49).

- [SBH+18] M. Stein, T. Breitzkreutz, J. Haussler, D. Seebacher, C. Niederberger, T. Schreck, M. Grossniklaus, D. Keim, H. Janetzko. „Revealing the Invisible: Visual Analytics and Explanatory Storytelling for Advanced Team Sport Analysis“. In: *2018 International Symposium on Big Data Visual and Immersive Analytics (BDVA)*. 2018, S. 1–9. DOI: [10.1109/BDVA.2018.8534022](https://doi.org/10.1109/BDVA.2018.8534022) (zitiert auf S. 35).
- [Sch23] F. Schmierer. *Visuelle Unterstützung bei der Planung der Bildentwicklung in Lateinformatiionschoreografien - Ergänzendes Material*. Apr. 2023. DOI: [10.5281/zenodo.7791174](https://doi.org/10.5281/zenodo.7791174). URL: <https://doi.org/10.5281/zenodo.7791174> (zitiert auf S. 44, 46, 76).
- [Shn98] B. Shneiderman. „Designing the user interface: strategies for effective human-computer interaction“. In: *Designing the user interface: strategies for effective human-computer interaction* (1998) (zitiert auf S. 84).
- [SJL+18] M. Stein, H. Janetzko, A. Lamprecht, T. Breitzkreutz, P. Zimmermann, B. Goldlücke, T. Schreck, G. Andrienko, M. Grossniklaus, D. A. Keim. „Bring It to the Pitch: Combining Video and Movement Data to Enhance Team Sport Analysis“. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2018), S. 13–22. DOI: [10.1109/TVCG.2017.2745181](https://doi.org/10.1109/TVCG.2017.2745181) (zitiert auf S. 35, 46, 85).
- [SMM12] M. Sedlmair, M. Meyer, T. Munzner. „Design Study Methodology: Reflections from the Trenches and the Stacks“. In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012), S. 2431–2440. DOI: [10.1109/TVCG.2012.213](https://doi.org/10.1109/TVCG.2012.213) (zitiert auf S. 41–43, 87).
- [SMV13] A. Schulz, W. Matusik, L. Velho. „ChoreoGraphics: An Authoring Tool for Dance Shows“. In: *Journal of Graphics Tools* 17.4 (2013), S. 159–176. DOI: [10.1080/2165347X.2014.909341](https://doi.org/10.1080/2165347X.2014.909341). eprint: <https://doi.org/10.1080/2165347X.2014.909341>. URL: <https://doi.org/10.1080/2165347X.2014.909341> (zitiert auf S. 37).
- [Sof23] Sofascore. *Sofascore Spielerprofil Serge Gnabry*. 2023. URL: <https://www.sofascore.com/player/serge-gnabry/187433> (zitiert auf S. 18).
- [SSS22] F. Schmierer, M. Stengel, R. Smith. „Vergleich visueller Abfragesprachen für semantische Wissensgraphen im Bereich des kulturellen Erbes“. Bachelor Forschungsprojekt, Universität Stuttgart. Okt. 2022 (zitiert auf S. 41).
- [TBKE14] D. Thom, H. Bosch, R. Krueger, T. Ertl. „Using large scale aggregated knowledge for social media location discovery“. In: *2014 47th Hawaii international conference on system sciences*. IEEE. 2014, S. 1464–1473 (zitiert auf S. 30, 31).
- [Thr23] Open Source. *Three.js*. 2023. URL: <https://threejs.org/> (zitiert auf S. 50).
- [TM05] M. Tory, T. Moller. „Evaluating visualizations: do expert reviews work?“ In: *IEEE Computer Graphics and Applications* 25.5 (2005), S. 8–11. DOI: [10.1109/MCG.2005.102](https://doi.org/10.1109/MCG.2005.102) (zitiert auf S. 85, 87).
- [Viz22] Vizrt. *Vizrt*. 2022. URL: <https://www.vizrt.com/> (zitiert auf S. 36).
- [Wes91] L. A. Westover. *Splatting: a parallel, feed-forward volume rendering algorithm*. The University of North Carolina at Chapel Hill, 1991 (zitiert auf S. 30).
- [Woo97] L. E. Wood. „Semi-structured interviewing for user-centered design“. In: *interactions* 4.2 (1997), S. 48–61 (zitiert auf S. 44).

- [YH17] C. Yu, Z.-C. He. „Analysing the spatial-temporal characteristics of bus travel demand using the heat map“. In: *Journal of Transport Geography* 58 (2017), S. 247–255 (zitiert auf S. 28, 29).
- [YY] Y. Yoshida, T. Yonezawa. „Choreographic design visualization of enormous dancers for authoring and browsing dance motion and formation“. In: () (zitiert auf S. 37).

Alle URLs wurden zuletzt am 31. 03. 2023 geprüft.