

The present work was submitted to the faculty of Computer Science and Mathematics of the Goethe University Frankfurt

Dancing Trainer: A System For Humans To Learn Dancing Using Multimodalities

Master Thesis

Presented by

Romano, Gianluca

5337053

First examiner: Prof. Dr. Hendrik Drachsler

Second examiner: Prof. Dr. Alexander Mehler

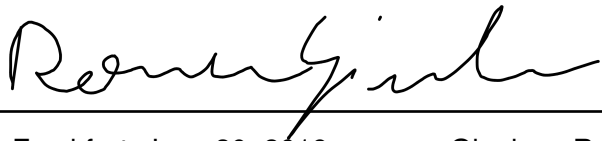
Supervisor: Dr. Jan Schneider

Frankfurt, June 30, 2019

Erklärung gemäß Master-Ordnung Informatik 2015 §35

Abs. 16

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel verfasst habe. Ebenso bestätige ich, dass diese Arbeit nicht, auch nicht auszugsweise, für eine andere Prüfung oder Studienleistung verwendet wurde.

A handwritten signature in black ink, appearing to read 'Romano Gianluca', written over a horizontal line.

Frankfurt, June 30, 2019

Gianluca Romano

Acknowledgments

“To master a new technology, you have to play with it.”

—Jordan Peterson

My deepest gratitude goes to my family. I would especially like to thank my father and mother for always supporting and motivating me in the past. Without them this thesis would have never been possible.

I would also like to thank Prof. Dr. Hendrik Drachsler and Prof. Dr. Alexander Mehler for supporting this thesis. I would like to express my gratitude to Dr. Jan Schneider for being a great supervisor, without his help this journey would have been extremely hard.

And finally, a big shout-out to the Salsa teacher that helped me out with recording Salsa steps to build the recognition and to the people that assisted me with the user tests.

Abstract

Dancing is the ability to feel the music and express it in rhythmic movements with the body. But learning how to dance can be challenging because it requires proper coordination and understanding of rhythm and beat. Dancing courses, online courses or learning with free content are ways to learn dancing. However, solutions with human-computer interaction are rare or missing. The Dancing Trainer (DT) is proposed as a generic solution to fill this gap. For the beginning, only Salsa is implemented, but more dancing styles can be added. The DT uses the Kinect to interact multimodally with the user. Moreover, this work shows that dancing steps can be defined as gestures with the Kinect v2 to build a dancing corpus. An experiment with 25 participants is conducted to determine the user experience, strengths and weaknesses of the DT. The outcome shows that the users liked the system and that basic dancing steps were learned.

Zusammenfassung

Tanzen ist die Fähigkeit, die Musik zu fühlen und das Gefühl in rhythmische Bewegungen mit dem Körper auszudrücken. Allerdings kann das Tanzenlernen eine Herausforderung darstellen, weil gute Koordination und Rhythmusgefühl gefragt sind. Tanzkurse mit einem Lehrer, Online Kurse oder kostenlos verfügbare Videos sind Wege um Tanzen zu lernen. Trotz technologischer Möglichkeiten, gibt es wenige oder gar keine Lösungen mit Human-Computer Interaction (HCI). Daher wird ein generisches System namens Dancing Trainer (DT) präsentiert. Für den Anfang ist Salsa als einzige Tanzkomponente implementiert, aber dem System können weitere Tanzstile als Komponente hinzugefügt werden. Der DT verwendet die Kinect v2, um multimodal mit dem Benutzer zu interagieren. Außerdem wird gezeigt, dass Tanzschritte als Gesten mit der Kinect v2 definiert werden können, um einen Tanzkorpus aufzubauen. Es wurde ein Experiment mit 25 Teilnehmern durchgeführt, um die User Experience, Stärken und Schwächen des DTs herauszufinden. Das Ergebnis zeigt, dass die Teilnehmer das System mögen und, dass Salsagrundschritte gelernt werden konnten.

Contents

1	Introduction	1
2	Structure	5
3	Related Work	7
3.1	When Technology meets Learning	7
3.2	Applications of the Microsoft Kinect Camera for Windows v2	8
4	Mathematical Definitions	11
4.1	Vectors and Operations	11
4.2	Cartesian Coordinate Systems	12
4.2.1	2 Dimensional	12
4.2.2	3 Dimensional	12
4.3	Confusion Matrix	13
5	Design Concept of the Dancing Trainer	15
5.1	Embedding the Dancing Trainer into a Learning Analytics Framework	15
5.1.1	Stakeholders	15
5.1.2	Objectives	17
5.1.3	Data	17
5.1.4	Instruments	18
5.1.5	External Constraints	18
5.1.6	Internal Limitations	19

5.2	Simplifications and Restrictions	19
5.2.1	Beat Tracking	19
5.2.2	Dance Style: Salsa	20
5.3	Defining the Requirements	22
5.3.1	Functional Requirements	22
5.3.2	Non functional Requirements	23
5.4	User Interfaces and Navigation	26
5.4.1	MainWindow	26
5.4.2	SalsaWindow	28
5.4.3	TutorialWindow	31
5.5	Architecture of the Dancing Trainer	33
6	Implementation	35
6.1	Standard for the Beat Annotated Music Library	37
6.1.1	Data Organization	37
6.1.2	Content	38
6.2	Multimodal Feedback Design	39
6.2.1	Online Feedback	40
6.2.2	Offline Feedback	43
6.3	Construction of Multimodalities in the Dancing Trainer	46
6.3.1	Working with the Kinect	46
6.3.2	Step Recognition with the Kinect	51
6.3.3	Classification of Dance Movements to the Beat	56
6.3.4	Is your Body Moving when Dancing?	58
6.3.5	Putting Emotions into Dancing	59
6.3.6	Having a Good Posture in Salsa	60
7	Evaluation	61
7.1	Recognition of Feedback	61
7.2	Experiment	68
8	Result	79

9 Discussion	83
10 Outlook	87
11 Conclusion	91
Abbreviations	93
List of Tables	95
List of Figures	97
Listings	99
Bibliography	101

1 Introduction

Dancing is the ability to feel the music and express it in rhythmic movements with the body. It is well known and helps establishing fitness and healthiness [Han95]. Today there exist many different dance styles, e.g Hip-Hop, Break dance, Waltz and Salsa. However, dancing is not an easy task for everyone and can be trained. Thus, you can learn dancing.

There are different possibilities learning how to dance. Probably, the most classic way is visiting a dancing course. The teacher and course members help you training and are your main source you receive feedback from. Plus, you can socialize with other people. In the case of a pair dance like Salsa you can directly practice with a partner. Nowadays, online resources are a valid option, too. You can either pay for online courses or learn with free video tutorials. Another way is the usage of video games to practice dancing. One example of such a game is the *Just Dance* Series. Every method has its advantages and disadvantages. Learning alone with online resources can be good because one can practice in his own tempo, chose the time to practice to his liking and in the case of free tutorials there are no extra costs. On the other side you benefit from your teacher or the course members when visiting a dancing course, although you have to pay for it and the training sessions are fixed. Using video games introduce another method to learn dancing. That is learning with machines or technology. However, playing video games is more about having fun rather than learning. Using machines and technology to learn how to dance, currently is uncommon if it is about dancing. Thus, there are rare or missing solutions for it. A smartphone is used in [Don+17] to assist dancers with an audible Salsa beat. Similarly [San+18] evaluated acceleration as a metric for the Salsa beat. Other solutions [Ale+11; Sah+13; Kya+15; MST17; Pro+17; KKK17; Aic+18] target specific dance styles, but there is no generic trainer you can choose dance styles you want to learn from. Note that only

[KKK17; Pro+17; MST17] use the Microsoft Kinect Camera for Windows v2 (Kinect v2) sensor and the others [Ale+11; Sah+13; Kya+15; Aic+18] the older Kinect. [Kit+14] used multiple older Kinects and synchronized them to tackle the problem of occlusion in pair dance.

This work presents another human-machine interaction based approach. The system developed is called Dancing Trainer (DT) and it is used to learn different dance styles. The usage of *Just Dance* for learning purposes is not good because the user has to focus too much on mimicking choreographies while following an avatar and instructions. You rather train reading the icons and memorizing the choreography for one specific song other than recognizing the beat and moving to it. At the end of the choreography you obtain a good score with the least effort possible. The same accounts for *SingStar*, another video game of another domain. The domain is singing but you do not learn singing by playing *SingStar*. These are indicators that video games focus aspects that are not suitable for learning. However, the DT is developed with the purpose to be a generic trainer for different dance styles. The intent of the DT is to support users to learn dancing and this work investigates the user experience of the DT. Therefore, the system has to classify dancing steps. Other solution like [KKK17; Pro+17; MST17] implemented their own and different classifiers. This work uses Visual Gesture Builder (VGB) to create gestures that classify dancing steps. Thus, the question if dance movements can be defined as gestures for the Kinect v2 is tackled.

The DT uses a predefined music library that provides annotations of the beat for each song. The music library is called Beat Annotated Music Library (BAML). It offers different songs users can choose from and dance to. The beat annotation is used for the recognition if the user is dancing to the beat or not. The gestures that are built to recognize the steps are used for this purpose, too. The DT focuses on the implementation of basic dancing steps as gestures that are building blocks of more complex choreographies. While dancing, the user also receives feedback from the DT that comes from the multimodal communication between the Kinect v2 and the user, e.g. gestural or visual [CK00]. In this work Beat Tracking is excluded and Salsa is implemented as the first dance style of the DT. Beat Tracking is a challenging task especially for Salsa because different instruments play at different rhythms that overlap.

Salsa is chosen as the first dance style to be implemented because there are too many different dance styles. Covering all of them in this work is not feasible. The DT is designed

such that dance styles can be added at a later stage of the development.

To summarize, the main contributions of this work are:

- A generic system to learn basic dancing steps of different dance styles that is called the Dancing Trainer (DT).
- The definition of dancing steps as gestures for the Kinect v2 with VGB as a tool.
- The Beat Annotated Music Library (BAML) that provides the Beats Per Minute (BPM) as a ground truth for beat recognition tasks.

Finally, a pilot study in form of a user test is conducted to gain knowledge about the user experience, strengths and weaknesses of the DT. Furthermore, it helps finding new development steps like changing, discarding or adding features or functionalities. The conducted experiment contains 25 users where the DT was used to learn about Salsa and train basic forth and back Salsa steps. The evaluation yields that the users had fun using the DT and that basic Salsa steps could be learned, meaning that basic dancing steps can be defined as gestures with VGB.

2 Structure

This work is structured in 11 chapters. The first two chapters introduce this work describing the importance of dancing, the state-of-the art, the main contributions, briefly the results and the structure of this work. Chapter 3 talks about related work and states applications of the Kinect v2 and how technology can influence learning and education. The next chapter defines mathematical conventions used in this work. Chapter 5 is about the design concept of the DT with a generic framework for Learning Analytics (LA) applications. Beat Tracking and Salsa are explained as restrictions and simplifications made in this work. Then, the requirements of the system are elaborated, the User Interfaces (UIs) with their navigation are shown and the architecture is proposed. Knowing the design, chapter 6 focuses on the implementation of the BAML and the design of the feedback that users receive when dancing Salsa. Thereby, multimodality for the DT is introduced. Another topic of this chapter is the implementation of the different feedback with the Kinect v2. Chapter 7 evaluates the recognition systems built to recognize feedback and the user tests conducted for the DT. Chapter 8 summarizes the whole work by giving the results. On the contrary, chapter 9 discusses the results by mentioning current problems like strengths and weaknesses of the DT. Chapter 10 and 11 end this work by giving suggestions to possible future works that are inspired by the problems, and by giving a conclusion with some statements from the user tests.

3 Related Work

3.1 When Technology meets Learning

The field of education is not untouched by technology. Learning can be seen as the gain of knowledge such that it becomes usable for the learner. Knowledge can be obtained differently, e.g. by accessing it. Technology offers the possibility to communicate the content faster and easier which makes the access easier, too. In fact, it is seen as an enhancement of didactic approaches to communicate, visualize or transmit content by [Har12]. Another way of learning can be through interaction between e.g. students and teachers. The interaction can be in class or due to technology e.g. in Massive Open Online Courses (MOOCs) or online seminars/courses. In normal classes, the student receives an evaluation from the teacher from time to time that is based upon objective criteria and subjective thoughts. The evaluation process repeats but it depends on e.g. being an elementary, high school or university student. In this case technology can be used to enhance the communication of the evaluation.

But, it can be used to e.g. do the evaluation or be a medium to interact with. By this, it is not necessarily meant the evaluation of tests but of data that teachers grasp through interaction with e.g. social skills and machines has to do differently. Data has to be collected to infer personalized behavior like the time spent on exercises or the clicking behavior on the main page of an online course. Based on the behavior the student receives feedback. Giving feedback as a teacher to a student has a big impact on his performance [HT07]. The process of collecting and processing data, and reporting the outcome to the user for learning purposes is called LA. More precisely, the term was defined on the first International Conference of Learning Analytics and Knowledge in 2011: *"Learning Analytics is the measurement, collection, analysis*

and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs".¹ LA becomes multimodal when sensors capture multiple modalities humans communicate with such as face expressions or gestures to infer feedback from that according to [HT07] has a big impact on learning. With the advancement of technology, Intelligent Tutoring Systems (ITSs) are introduced of being able to substitute real human tutors to some extent [ABR85]. The Kinect v2 and its functions are part of such an advancement.

E.g. [Kou+17] use the Kinect in learning games to see if there is an impact on cognitive, motoric and academic skills for children with special educational need. It turned out that there is a positive one. The work of [Zha+18] presents the Kinect v2 as a cost efficient device for educational virtual reality development to e.g. serve the purpose of body motion and voice recognition. The system presented by [Sch+16] tracks motion and voice with the Kinect v2 to give the user feedback on his presentation performance. These are some examples how technology, that is not only used for communication but as interaction medium, can be involved in learning.

3.2 Applications of the Microsoft Kinect Camera for Windows v2

The Kinect v2 is a device with many functionalities. It has a depth and infrared sensor. So, the Kinect is applicable where these types of sensors are needed. E.g. [Fan+15] used the depth sensor of the Kinect v2 for mobile robot navigation. They tested the depth sensor for indoor and outdoor purposes with the influence of sunlight. According to their results the Kinect v2 provided robust depth images that are suitable to detect thin objects even when exposed to sunlight.

[Jia+17] proposed a post-rectification approach of depth images retrieved by the Kinect v2 for a better 3D reconstruction of indoor scenes. This can be used for robot navigation, too. The Kinect v2 can capture humans and represent them as skeletons that provide information about joints in a 3D Cartesian Coordinate System (CCS). The work of [YON17] used the

¹ <https://tekri.athabasca.ca/analytics/call-papers>

rotation of the joints which are fed in their deep learning approach to predict dance motions and transfer them to a robot. Other approaches that regard using a Kinect for dancing, can be seen in [Ale+11; Sah+13; Kya+15; MST17; Pro+17; KKK17; Aic+18]. All developed their own approach of a dancing tutoring system. Some classified dance movements, some did not. [Ale+11] compared the velocity vectors of a predefined choreography with the recording using the first order Derivative of Gaussian. A similar approach is chosen by [Aic+18] who aligned the performances and compared their similarity. A Support Vector Machine was used as a more sophisticated approach by [Sah+13]. [Kya+15] extracted features from a skeleton captured by a Kinect and used those in combination with a gesture database to derive a set of ballet movements. A projection of a teacher was used to show the movements to the user. [MST17] used a similar idea and used the Kinect v2 to produce a virtual model of the user. The user could see his model in comparison to a reference model that performed dance movements stored in a database. In contrast, [KKK17] built a Neural Network (NN) with a reduced input space to classify Korean pop dance moves and showed that it surpassed e.g. Support Vector Machines.

Overall, the Kinect is used for diverse recognition tasks like gait, posture and hand gesture recognition. This was used for medical purposes in [SDT17] for the improvement of gait and posture to prevent falls and in [SS15] to detect falls with the use of depth images and postural information. [Gal+14] stated that the Kinect can recognize bigger movements like standing or sitting for people with Parkinson. Smaller movements like hand clasping were not recognizable. Gait information retrieved was used to recognize more abstract things like self-esteem as proposed by [Sun+17].

The Kinect v2 is a cost efficient and versatile applicable device e.g. in education, robot navigation, dance and medicine.

4 Mathematical Definitions

4.1 Vectors and Operations

Let $P = (p_1, p_2, \dots, p_n), Q = (q_1, q_2, \dots, q_n) \in \mathbb{R}^n$ be points in an n-dimensional space. The vector between them is defined as:

$$\vec{PQ} = (q_1 - p_1, q_2 - p_2, \dots, q_n - p_n) \quad (4.1)$$

The scalar product of two vectors $\vec{u}, \vec{v} \in \mathbb{R}^n$ is defined as:

$$\vec{u} \cdot \vec{v} = u_1 \cdot v_1 + u_2 \cdot v_2 + \dots + u_n \cdot v_n \quad (4.2)$$

The length of a vector $\vec{u} \in \mathbb{R}^n$ is defined as:

$$\|\vec{u}\| = \sqrt{u_1^2 + u_2^2 + \dots + u_n^2} \quad (4.3)$$

The angle α of two vectors $\vec{v}, \vec{u} \in \mathbb{R}^n$ is defined as:

$$\alpha = \cos^{-1} \left(\frac{\vec{v} \cdot \vec{u}}{\|\vec{v}\| \cdot \|\vec{u}\|} \right) \quad (4.4)$$

4.2 Cartesian Coordinate Systems

4.2.1 2 Dimensional

Let $\vec{x}_1, \vec{x}_2 \in \mathbb{R}^2$ be the vectors that build the 2D CCS with the origin $\vec{0} = (0, 0)$, $\vec{x}_1 = (1, 0)$ growing to the right and $\vec{x}_2 = (0, 1)$ growing down (figure 4.1).

4.2.2 3 Dimensional

Let $\vec{x}, \vec{y}, \vec{z} \in \mathbb{R}^3$ be the vectors that build the 3D CCS with the origin $\vec{0} = (0, 0, 0)$, $\vec{x} = (1, 0, 0)$ growing to the right, $\vec{y} = (0, 1, 0)$ growing up and $\vec{z} = (0, 0, 1)$ growing forward (figure 4.1).

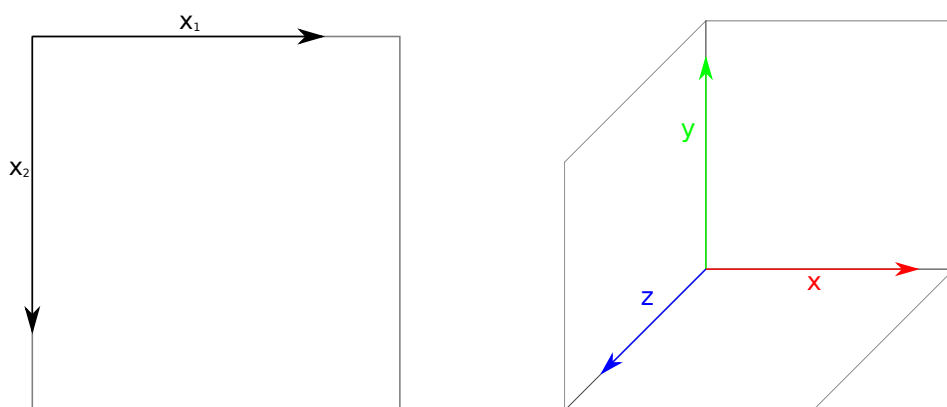


Figure 4.1: 2D (left) and 3D (right) CCSs.

Object Orientation

The orientation of an object in a 3D CCS is defined by quaternions. Quaternions can be written as a linear combination of real and imaginary parts. A quaternion q is defined as:

$$q = a + bi + cj + dk \quad (4.5)$$

where a, b, c, d are real numbers and i, j, k their imaginary part, and $i^2 = j^2 = k^2 = ijk = -1$.

4.3 Confusion Matrix

A confusion matrix reports the outcome of a classification problem and opposes the ground truth with the prediction. Table 4.1 shows how a binary confusion matrix looks like, where True Negatives (TNs) are negative instances that are predicted negative, True Positives (TPs) are positive instances that are predicted positive, False Negatives (FNs) are positive instances that are predicted negative and False Positives (FPs) are negative instances that are predicted positive.

		Prediction	
		Negative	Positive
Ground Truth	Negative	TN	FP
	Positive	FN	TP

Table 4.1: Binary Confusion Matrix.

You can calculate different measurements that rate the classification system. Ratings that are used in this work are Precision, Recall and F-Score:

$$Precision = \frac{TP}{TP + FP} \quad (4.6)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.7)$$

$$F - Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (4.8)$$

5 Design Concept of the Dancing Trainer

5.1 Embedding the Dancing Trainer into a Learning Analytics Framework

[GD12] proposed a generic framework as a guide to build LA applications or systems. Note that there are other frameworks to build LA systems such as proposed by e.g. [WHH16; Agu+16; GY19; Mit+18]. However, all of them target more specific cases and can be related back to the generic one. But the model presented by [Mit+18] is used specifically for the DT to design the underlying feedback in section 6.2 because the DT involves a multimodal interaction with users with the Kinect v2. Thus, the framework of [GD12] is applied to build the DT as a generic system to learn dancing and to position the DT as a LA application among other ways to learn dancing like (online) dancing courses. This section argues about the six dimensions of LA: stakeholders, objectives, data, instruments, internal limitations and external constraints.

5.1.1 Stakeholders

Making a proper distinction of stakeholders affects the implementation of the DT. Therefore, *data clients* and *data subjects* as introduced in [GD12] are defined for a Business-to-Business, Business-To-Person and Person-To-Person relationships. *Data clients* benefit from the produced data making them a data consumer. In contrary, the *data subjects* produce data, making them a data producer. Business-to-Person relationships are between businesses or non human instances, and persons. A Person-To-Person relationship is between two or more per-

sons. Business-To-Business relationships are between non human instances. You can argue that persons, like a teacher, belong to a business, thus acting in a Business-to-Person relationship. In fact, the borders are blurred. But, it does make sense to draw a line, because the interests of businesses can differ from the interests of teachers, e.g. being more profit than learner oriented.

Business-To-Business

Depending on the context, businesses are either *data subject*, *data client* or both, that produce, consume or exchange data in a more profit oriented way. Note, that data can be transformed or used in a way such that it is not anymore related to the DT. E.g. the provider of the DT can provide derived data to other businesses. The provider acts as a *data subject* and the other business as *data client*. If the other business continues processing, deriving and providing data to further businesses, it acts as *data subject*, too.

Business-To-Person

In a Business-To-Person relationship companies, institutions or other non human instances can provide the DT to persons. This automatically makes the business the *data client* and the human learner the *data subject*. The business can use or abuse the data produced by the *data subjects*, e.g. for development of the DT, selling out the data or to build other products.

Person-To-Person

In a Person-to-Person relationship one person is the *data client* and the other the *data subject* and *data client*. Take a teacher-student relationship. The teacher might be interested in the students improvements. He can use the data to recommend further training steps to take for the student. On the other hand the student consumes the data immediately after producing it, being its own teacher.

5.1.2 Objectives

The main objective of the DT is helping its users to learn how to dance. This can be broken down into *reflection* and *prediction* [GD12]. Let us use the relationships of the previous subsection to see which instances reflect on what and what they predict. In general, businesses have to reflect upon the collected data and predict their further movement. This depends on the businesses domain. In case of a Business-to-Person relationship where the DT is provided directly, the provider has to reflect on the current position of the DT. He has to predict the new version with its features. The usage of those can either benefit or abuse learners. In a Person-To-Person relationship the teacher reflects on the learners performances, on himself and also on the system. Then, the learner has to be able to self reflect on his dancing performance. Therefore, the DT has to provide information about past dancing sessions.

5.1.3 Data

The data used by the DT is retrieved directly from the user. Information about the joints of the learner are processed to detect motions or face expressions. This has to be saved to some extent to provide feedback after a dancing session. If the saved data is not related to the user, then he does not have to worry about personal abuse of it. However, the DT has the potential to do so. An account personalizes the use of the DT that leads to a collection of personal user data. Motion data can be abused by insurances to determine or indicate the clients physical conditions. In fact, motion data can predict the health conditions of users [SS15; SDT17]. Therefore, it is important to protect this type of personal data. But the provider might need motion data to develop a better version of the DT. Plus, he can profit from selling out this type of data. The user should be given the possibility to decide which data he wants to be sent to the provider. He has insight for what purpose data will be used. So, giving the user the decision of the usage of the data can be part of a solution, too. General data of the users progression can be argued to be open to some extent. Teachers might need some reference to support a better progress of the user. An agreeable trade-off can be to keep data rather open for a

Person-To-Person relationship and protected for a Business-To-Person relationship. Data can be anonymized to protect the user from abusive use. Businesses can benefit from the data for development but selling it does not fall back on the consumer.

5.1.4 Instruments

A computer is required to start the DT from. A separate screen, that is big enough to see and understand the feedback and instructions is beneficial but not mandatory. The Kinect v2 is another instrument of the DT. It is necessary to collect real time data about the users motion and classify dancing steps or other feedback. Machine learning is the theoretical background for classification problems. But the Kinect v2 does not have to be the only technology. Other devices to derive feedback from, can be added, too. Providing feedback to the user has an impact on learning [HT07]. [Sch+16] suggested to show simple feedback one at a time to not overwhelm the user. Plus, moving or doing sports like dancing improve healthiness and help relieving stress [Han95]. Furthermore, it affects your mood positively and makes you feel better [KK07; CL14].

5.1.5 External Constraints

One external constraint regulates the use and abuse of data by the governed laws, like e.g. the General Data Protection Regulation (GDPR) in the European Union. Another constraint is the necessity of the devices, technology and enough space to dance. The DT can either be seen as concurrence or enhancement to established learning approaches for dancing, like (online) dancing courses. This depends on the mindset towards digitalization. Digital driven solutions have to be accepted socially. Some might fear that their job will be taken away, but others see an opportunity to improve their dancing schools or businesses. In fact, the DT can be used in dancing schools for absolute beginners or to repeat basics for dancing experts.

5.1.6 Internal Limitations

The user has to be able to understand the feedback that is processed and displayed by the DT. Understanding what a system wants or tries to tell is not trivial. Misunderstanding feedback can influence the impact of it, thus the learning progress of the user. Besides the already highlighted social acceptance as external constraint, the user has to accept the DT, too. If you only train on your own you might dismiss the benefits to socialize with others in a dancing course. However, using the DT as an enhancement e.g. in dancing schools, users are still able to socialize that may facilitate their acceptance of the DT.

5.2 Simplifications and Restrictions

5.2.1 Beat Tracking

The beat of a song can be defined as the moments where you nod your head or tap your feet to. Not everyone is able to feel the beat equally. For machines, this can be even more challenging [GM11]. Music is a composition of notes, that is defined by the measure and *tactus*. The measure/bar denotes a slice of time dedicated to a defined number of beats. The time signature of e.g. 4/4 defines the number of beats that is given at the beginning of each piece. This number is called *tactus* and is the technically term for the beat. However, the beat can be perceived differently from the *tactus*. The *tatum* is defined as the lowest regular pulse a listener can perceive. The *tactus*/beat is a down sampled version of the *tatum*. For machines, computing those values can be hard because of background noises or the singers voices. The instruments used in a music piece has also to be dealt with because their measure, *tactus* and *tatum* can be different and overlapping. The current trend solving Beat Tracking goes towards NNs. A Recurrent Neural Network (RNN) in combination with a Dynamic Bayesian Network (DBN) was used in [Kre+16] to track the first beat of a bar, called downbeat. [BKW14] proposed a system for heterogenous music styles that uses multiple and music type specific RNNs. The RNN with the highest activation determines the beat. [GK17] adopt a Convolutional Neural Network (CNN) for Beat Tracking, too. The beats are inferred with dynamic programming from the Beat Activation Function (BAF) and transferred to a robot to dance to the beat. On the

other hand [Ohk+15] used visual data retrieved by a depth sensor, and audio signals to track the beat and let a robot dance to it. For the DT, Beat Tracking would be essential because of the freedom to choose the music you want to dance to. However, Beat Tracking is not part of this work due to its challenges. To overcome this, section 6.1 presents the BAML where the beat is annotated by the BPM of the song.

5.2.2 Dance Style: Salsa

Ideally, the DT shall support as many dance styles as possible. But there exist too many to be implemented in this work. So, the focus is only on the basics of Salsa.

Originally, the word *salsa* means *sauce*. The term was first used in 1963 from Iris Sanabria and the label Fania Records. After a spread of the term in the 1960s, in 1970 it was recognized as standard. So, the term Salsa as we know it today originated by the marketing of the music industry. Salsa is a latin music and dance style that has undergone various changes in history. It is influenced by the relations between Africa, South and North America. The influence is due to immigration, tourism and the enslavement at that time. Music styles that influenced Salsa are Jazz, caribbean Music, Rock and Pop. The mixture of different music styles fits the symbolic meaning of mixing and blending a sauce. So nowadays, there does not exist one specific dance style called Salsa, but many. [Hen08]

Like other latin music, Salsa is perceived with a fast and passionate rhythm. Characteristic instruments for Salsa are the *conga* drum and two wooden sticks called *clave*. Other instruments are e.g the piano, a cow bell or trumpets. The *conga* and *clave* give the beat to dance to. Counting the beat in Salsa is done in 2 bars. Thus, you count from 1 to 8 with a 4/4 time signature. The rhythm that the *conga* plays is called the *tumbao* and highlights the steps to make. The *tumbao* is played by to open tones and a stressed one. The *clave* can be played on 1-2.5-3,6-7 or 2-3,5-6.5-7. Note the intermediate counts of the *clave*. All the instruments play a different rhythmic pattern that overlaps. This makes hearing the beat in Salsa difficult, even for experts. Figure 5.1 and 5.2 show the steps to make at the beats 1 to 8 for forth and back, and side movement. The images are inspired by common vizualizations for basic Salsa steps in e.g. books. Let us explain the steps only for forth and back steps because it is analogously

for side steps. You start standing with both feet aligned. Then step 1 is stepping with the left foot forward, step 2 tapping with the right foot, step 3 returning to the starting position, step 4 pausing, step 5 stepping with the right foot backward, step 6 tapping with the left foot, step 7 returning to the starting position, and step 8 pausing. The pauses on beat 4 and 8 are pauses to break and change the direction of the movement. Depending on the style, you start stepping with the left foot forward on beat 1 or 2. Here, the first option is chosen. [Ren04] The first step depends also on your sex because Salsa is a pair dance. The male role steps with the left foot forward, while the female role steps with the right foot backward. However, the steps remain the same, such that the sex is not considered for learning alone with the DT.

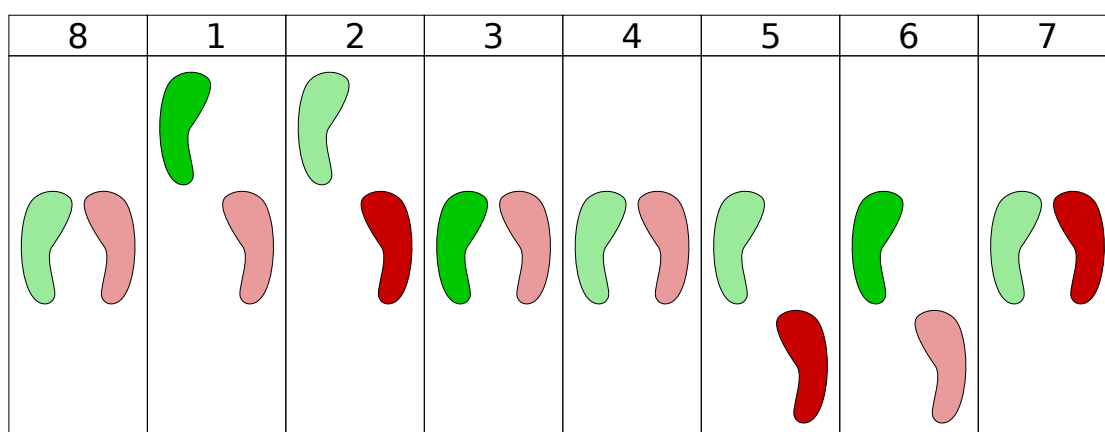


Figure 5.1: Salsa stepping order of the basic forth and back steps. The numbers indicate the beat/step with 8: neutral, 1: left forward, 2: right tap, 3: left back to neutral, 4: neutral, 5: right back, 6: left tap, 7: right back to neutral

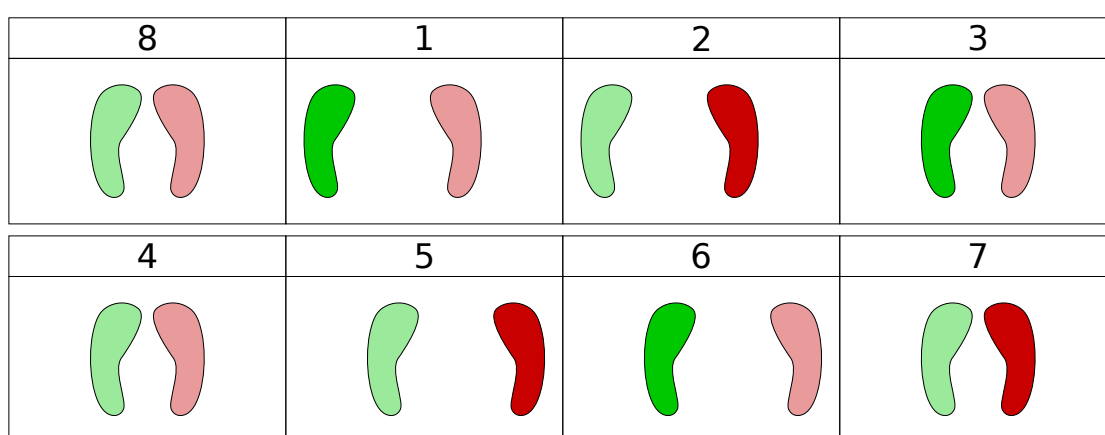


Figure 5.2: Salsa stepping order of the basic side steps. The numbers indicate the beat/step with 8: neutral, 1: left side, 2: right tap, 3: left back to neutral, 4: neutral, 5: right side, 6: left tap, 7: right back to neutral

5.3 Defining the Requirements

Designing a system or software is not an easy task. There are various strategies you can follow to design a system. In all cases a good design can separate experienced from inexperienced designers. Experienced designers e.g. spend more time understanding a problem and do a requirement analysis [SNV06]. Some design models like the V or the Spiral model are explained by [Zhu05]. Both and others do a requirement analysis. Plus, [Zhu05] says that a good design effects the final software on e.g. efficiency, correctness, reusability and more. Therefore, this section is dedicated to define the requirements of the DT. These are split into two different types, functional and non functional requirements.

5.3.1 Functional Requirements

Functional requirements describe what the system has to provide. Therefore, different use cases are defined to infer the functional requirements. In table 5.1 you can see use cases sorted by priority. Note that use cases 8 and 9 are also important for the DT overall, but there are other cases whose implementation is needed before. Trainers for different dance styles have to be implemented before users can choose them. The restrictions of implementing only Salsa in this work is explained in section 5.2. That is why they have a lower priority. For this work use cases with a priority greater than 1 are considered for implementation. These are the use cases from 1 to 6.

Having defined the use cases for the DT, inferring the functional requirements is possible. Table 5.2 briefly states what the DT has to do by listing the functional requirements in order to implement the use cases. The number before the dot refers to the use case. The number after the dot is an implementation measure of the according use case.

ID	The user wants to ...	Priority
1	... play, pause and stop dancing sessions.	3
2	... know if he is doing the basic Salsa steps correctly.	3
3	... know how he is performing while dancing.	3
4	... understand the system.	2
5	... review his performance.	2
6	... have an introduction to Salsa.	2
7	... have fun.	1
8	... see his learning progress.	1
9	... choose trainers for different dance styles	1

Table 5.1: Use cases with priority values from 1 (lowest) to 3 (highest). Use cases with a priority greater than 1 are considered.

5.3.2 Non functional Requirements

Non functional requirements or qualities either exceed the functional ones or describe how good the systems behaves. There are many non functional requirements and they can be categorized into run time (internal) and not (external) [Wie13]. This work aims for six qualities: correctness, maintainability, performance, reuseability, scalability and usability.

Correctness is the property of functions producing the wanted output. E.g. the specification of a button can be that, once pressed, the system starts playing a song. The function is correct if the system starts playing the song when the button is pressed. It is important to know if functions are correct because errors can be passed through functions resulting into cumulative false outputs. Another view of correctness in this case, is the ability to recognize Salsa steps and to predict if the user is dancing to the beat. This also holds for other tasks that include recognition, like the recognition of feedback. Let us measure the performance of recognition tasks with F-Scores and assume it to be good if it is at least 80%. You can see the evaluation in section 7.1.

Maintainability is the goal to maintain the system and to not loose track of its state. Correcting errors by repairing or replacing functions is one component. Ensuring future maintenance to be less difficult is another one. Maintenance can be defined as the mean time to repair. But this has to be measured first. So, let us focus on the measures of the DT that potentially reduce the mean time to repair. These are a good documentation of the DT, a creation of an error log and the storage of raw clips that are used for the gestures. This work is part of the documentation

ID	The DT ...
1.1	... offers music to be selected by the user.
1.2	... loads music files.
1.3	... starts playing the music.
1.4	... pauses the music.
1.5	... stops the music.
1.6	... memorizes the date of the session.
1.7	... memorizes the duration of the session.
1.8	... starts the capturing of the Kinect v2.
1.9	... pauses the capturing of the Kinect v2.
1.10	... starts counting the beat.
1.11	... pauses counting the beat.
1.12	... resets counting the beat.
2.1	... detects Salsa steps.
2.2	... shows the user the detection of Salsa steps.
3.1	... displays a live video recording of the user.
3.2	... recognize feedback at run time.
3.3	... displays feedback at run time.
3.4	... recognizes if the user is dancing to the beat.
3.5	... recognizes if the user engages his whole body when dancing.
3.6	... recognizes if the user is looking down.
4.1	... explains the user how to understand the system.
5.1	... saves recordings of the dancing session.
5.2	... shows recordings of the dancing session.
5.3	... saves a time line of the produced feedback.
5.4	... shows a time line of the produced feedback.
5.5	... saves a time line of recognized and suggested Salsa steps.
5.6	... shows a time line of recognized and suggested Salsa steps.
6.1	... explains the user basic Salsa steps.

Table 5.2: Functional requirements that are inferred from the use cases of table 5.1. The number before the dot of the ID refers to the use case and the number after it to an implementation measure.

but the code can be found online, too.¹ Knowing how the system is built reduces the time to search for the source of the error. The error log tells where the error has occurred, and helps understanding and solving it. It is important to store the raw data of the clips to be able to reuse them if tools for the Kinect v2 are updated and changed. Instead of re-recording the clips, *KSConvert.exe* converts raw clips into processed ones, that are needed to build the gestures, thus the classification of dancing steps.

¹ <https://github.com/CanIALugRoamOn/DancingTrainer>

Performance measures the computation of the system. Some measurements are the response time and throughput. Let us assume that both are good enough if the real time components (beat counting) of the system are not lagging significantly. Put differently, the system is to the beat. This means that e.g. updates on the UI while dancing are not happening with a delay. Otherwise it can lead to confusion of the user. Here, another way to understand performance is the learning performance of the user. But, as stated in section 5.3.1 the learning progress is not covered by this work.

Reusability is the property to reuse existing components of the system, e.g. for future versions or other systems. Being at the beginning of the development of the DT this is important. Reusing existing components speed up future development, ensures correctness for existing and verified parts and benefits extending the current system. One purpose building the DT is to be extended with different dance styles. A modular architecture of the DT ensures that parts of the system can be reused. Section 5.5 talks about the (modular) architecture of the DT.

Scalability is about the growth of the system and focuses on the question if the system can be used on a larger scale. The purpose of the DT is to provide a generic system to support learning different dance styles. Therefore, it has to be possible to add components to the DT that handle the different dance styles. This is achieved through the architecture of the DT explained in section 5.5. Thereby, elements of the DT that are/should be scalable depend on the defined requirements, e.g. how many accounts can be managed if the user could create one, or should the recognition be outsourced (cloud, server) and how much computation can it handle. The first attempt is to organize the DT in modules that makes it flexible to the addition of future requirements, functions and other scalable components. All computation is done on the executing machine.

Usability refers to the difficulty of learning the usage of a system and specifies how easy it is to use. In the case of a LA application, [GD12] mentioned that it is not self-evident that users understand how to interpret a LA application, here the DT. Therefore, UIs are kept simple and inspired on windows of the operating system *Windows*, the most operating system used.² However, the UI displayed while dancing is unique and focused on showing beat and steps as minimalist as possible, but as precise as necessary. A tutorial mode is provided that explains

²https://www.w3schools.com/browsers/browsers_os.asp

the unique UI and basic Salsa steps. The UIs are introduced in the next section 5.4.

5.4 User Interfaces and Navigation

Having specified functional and non functional requirements, there is an idea how the DT shall work. At this point, UIs for the DT has to be defined users can interact with. Therefore, this section is dedicated of defining the navigation between and interaction with them. The UIs that are presented are conceptual and they can be altered. However, the concept of those are based on the requirements in section 5.3.1 and 5.3.2. This yields three windows, the *MainWindow*, *SalsaWindow* and *TutorialWindow*. In order to describe UI elements the C# vocabulary is used.³

5.4.1 MainWindow

The *MainWindow* is the default window and appears once the DT is started. It works as an entry for the different dancing trainer components. This means that the trainer for the supported dance styles can be selected from the *MainWindow*. It shall support the basic functionalities every trainer has in common. These can be playing, pausing and stopping functions. So, the *MainWindow* can meet the functional requirements from 1.1 to 1.9 of table 5.2. The according dancing trainer component can be determined automatically by the selected music genre that is mapped to it. In this concept the functional requirement 3.1 of table 5.2 is included in the *MainWindow*. Note that it can be included in every different dancing trainer component separately, too. Figure 5.3 shows an implementation of the *MainWindow* and the following list refers to the numbered UI elements:

1. Minimize Window Button
2. Maximize Window Button
3. Close Window Button
4. Music Genre Label
5. Music Genre ComboBox

³<https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms?view=netframework-4.6.1>

Offers a selection of different genres. Press to select a genre, and automatically a dance style if the genre is mapped to it.

6. Load Button

Loads the respective dancing trainer component of the selected genre. It can only be pressed if a song is selected. The *SalsaWindow* is opened if users want to train Salsa.

7. Song Label

8. Song ComboBox

Offers a selection of songs that correspond to the previously selected genre. Press to select a song. It can only be pressed if a genre is selected.

9. Play Image

Starts playing the music, capturing with the Kinect v2 and counting the beat when pressed. It can only be pressed if a dancing trainer component was loaded before by pressing the *Load Button*.

10. Pause Image

Pauses the music, the Kinect v2 capturing and the beat counter when pressed. It can only be pressed if the *Play Image* was pressed previously.

11. Stop Image

Stops the music, the Kinect v2 capturing and resets the beat counter when pressed. It can only be pressed if the *Play Image* was pressed previously.

12. Kinect Image

Shows the current image that the Kinect v2 captures when plugged in.

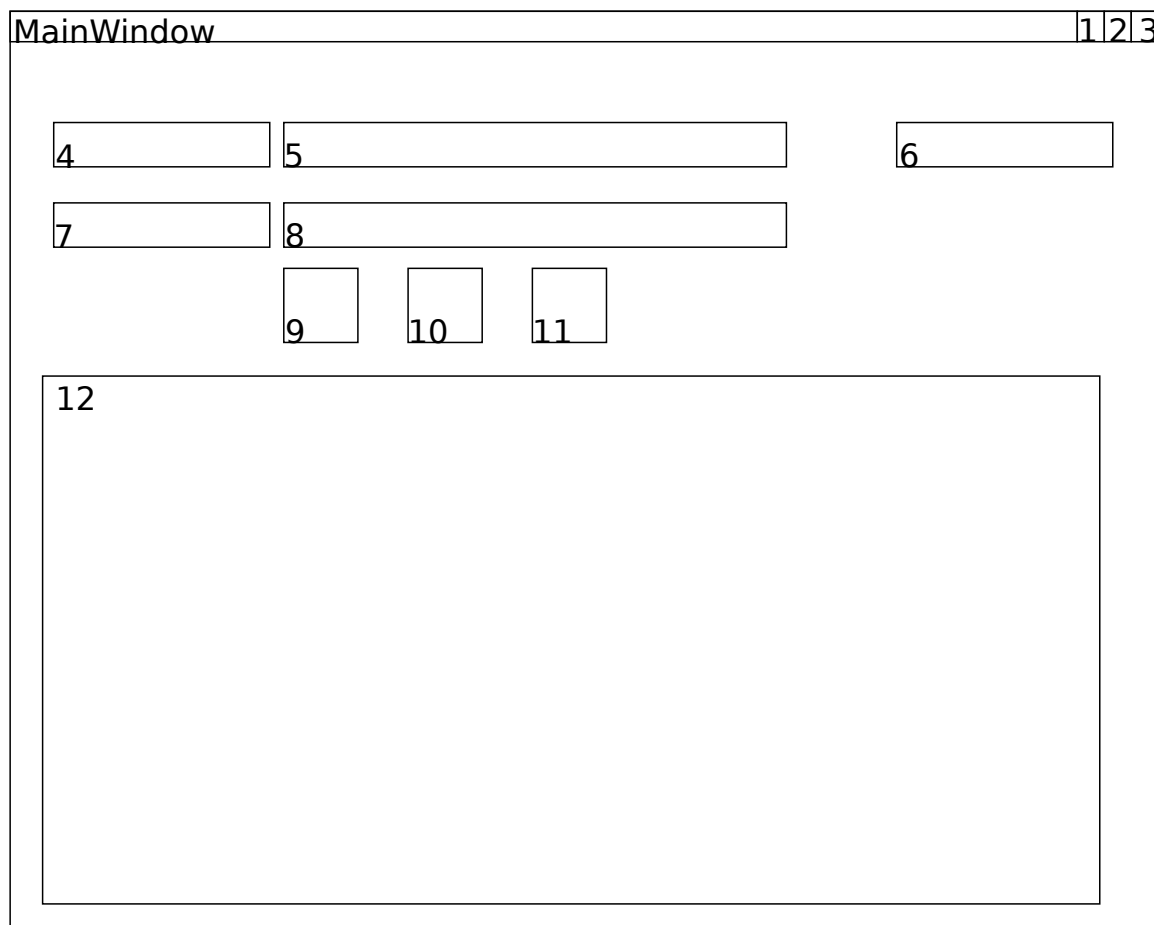


Figure 5.3: Concept of the *MainWindow* with the explanation of the UI elements in subsection 5.4.1.

5.4.2 SalsaWindow

The *SalsaWindow* represents one possible dancing trainer component. It can be opened from the *MainWindow* by selecting a genre and loading a song. The *SalsaWindow* provides the user with the possibility to train and learn Salsa. It displays a beat counter, feedback with small instructions and basic Salsa steps. Users can save recordings of their performance and feedback, and open them for review. Plus, users can change settings to adapt the difficulty to their skill level. Different levels of difficulty can be achieved by removing the display of supporting elements like the beat counter or the Salsa steps, or by enabling a beat audio support that highlights the beat of the song with e.g. a *click* sound. Moreover, users can change

the type of Salsa steps to practice. These are the forth and back, and side steps as shown in figures 5.1 and 5.2. If users are new to Salsa or not used to the interface, an introduction may be needed. Therefore, users can change to a *TutorialWindow* that helps how to interpret the interface, and do basic Salsa steps. Furthermore, opening the *TutorialWindow* changes the way the *SalsaWindow* behaves. In normal dancing sessions the *SalsaWindow* just shows the next step to make, unconditionally from the user dancing or not. However, changing to tutorial, the *SalsaWindow* shows the next step and waits for the user to perform it before showing the next one. Salsa steps work as input such that the detection of them is shown. With that, the *SalsaWindow* covers the functional requirements 1.10 to 1.12, 2.1, 2.2, 3.2 to 3.6, 5.1 to 5.6 of table 5.2. Figure 5.4 shows an implementation of the *SalsaWindow* and the following list refers to the numbered UI elements:

1. Minimize Window Button
2. Maximize Window Button
3. Close Window Button
4. File MenuItem
 - Includes an *Open MenuItem* and *Save MenuItem* to save and open performances for review.
5. View MenuItem
 - Changes the view (mode) of the *SalsaWindow* between normal and tutorial.
6. Settings MenuItem
 - Includes menu items for further settings like audio beat support, display of Salsa steps, the beat counter and the choice between practicing forth and back or side steps.
7. Beat Counter Label
 - Shows the current Salsa beat.
8. Left Forward Image
 - Shows the image of the left foot.
9. Right Forward Image
 - Shows the image of the right foot.
10. Left Side Image

Shows the image of the left foot.

11. Left Neutral Image

Shows the image of the left foot.

12. Feedback Icon Image

Displays the feedback icon.

13. Feedback Instruction Label

Shows the respective instruction of the feedback icon.

14. Right Neutral Image

Shows the image of the right foot.

15. Right Side image

Shows the image of the right foot.

16. Left Back Image

Shows the image of the left foot.

17. Right Back Image

Shows the image of the right foot.

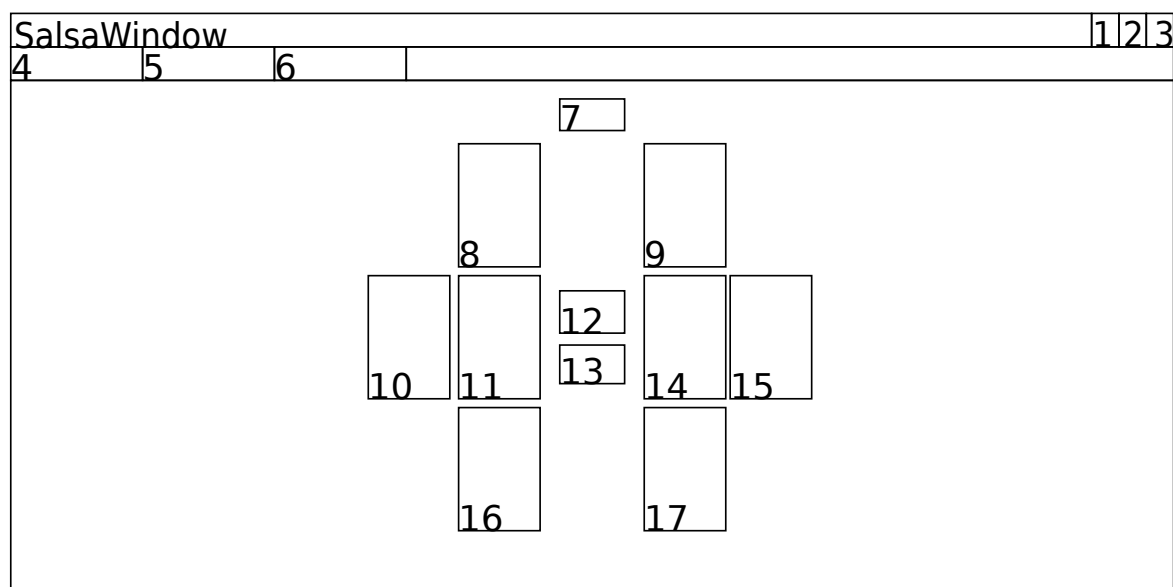


Figure 5.4: Concept of the *SalsaWindow* with the explanation of the UI elements in subsection 5.4.2.

5.4.3 TutorialWindow

The *TutorialWindow* has the purpose to help the user understanding the system and introducing basic Salsa steps to absolute beginners. You can get to the *TutorialWindow* by changing the view from the *SalsaWindow* introduced in the previous subsection 5.4.2. Remember that changing the view to tutorial changes how the *SalsaWindow* operates. The steps of the user work as input to the *SalsaWindow* that shows the next step and waits until it is performed. If so, the next step is shown and it is waited again for the user performing the next step. Thus, users can train Salsa steps and memorize them before getting into a normal dancing session. Feedback is not processed when attempting the tutorial mode, too. So, users can concentrate entirely on the steps and dance independently from the real beat of the song.

But before, the user has to walk through the *TutorialWindow*. It has a description supported by an image for visualization to explain users how to understand the UI, do basic Salsa steps and interpret the feedback. Users can switch between contents or close the window if they already have the knowledge. The *TutorialWindow* meets the functional requirements 4.1 and 6.1 of table 5.2. Figure 5.5 shows an implementation of the *TutorialWindow* and the following list refers to the numbered UI elements 5.5:

1. Minimize Window Button
2. Maximize Window Button
3. Close Window Button
4. Description Image
Visualizes content to let users know about basic Salsa steps and the DT.
5. Description TextBox
Describes content to let users know about basic Salsa steps and the DT.
6. Previous Button
Shows the previous tutorial content.
7. Next Button
Shows the next tutorial content.
8. Close Button

Closes the window.

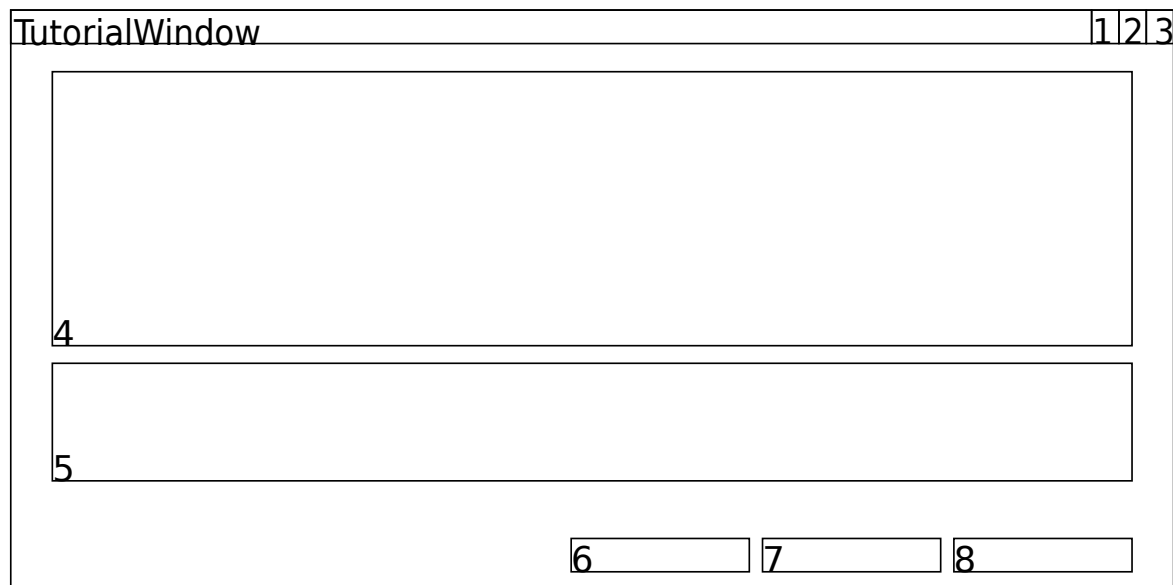


Figure 5.5: Concept of the *TutorialWindow* with the explanation of the UI elements in subsection 5.4.3.

5.5 Architecture of the Dancing Trainer

The architecture of the DT is designed regarding three key components: Beat Tracking, dancing step recognition and feedback. Beat Tracking is not tackled in this work but it has a high potential to be included in future works, and enhance the DT. That is why it is important to regard Beat Tracking for the design of the architecture right from the start. Further, the DT has to give developers the possibility to add dancing components. It directly aims to implement non functional requirements like scalability, reusability and modularity.

Figure 5.6 shows an extract of a class diagram of the current developed DT. You can see that the three different key components are targeted by the classes *SalsaBeatManager*, *GestureDetectorSalsa* and *Feedback*. The interfaces, *IBeatManager*, *IDancingtrainerComponent* and *IFeedback* are used to provide a standard such that other dancing components can be added. Every dance component needs an unique UI, feedback and beat manager because they change between dance styles. This makes the DT scalable, modular and allows code to be reusable. The sub architecture is governed by play, pause and stop functions. This means that some classes must have functions to play, pause and stop the dancing session or the recording with the Kinect v2. The interfaces can be used to ensure this sub architecture, too.

Here, the *SalsaBeatManager* is directly connected with the *MainWindow* (mainWin). The connection is not mandatory because it is also connected with the *SalsaWindow* (salWin). It does not hinder the execution because one dance component (Salsa) is implemented. However, with multiple dancing components it can be good or bad to have it connected like that. This depends if you want to have one beat manager for each dance component or not. This means to decide which class shall hold the object for the beat, the *MainWindow* or *SalsaWindow*. Here it is done by the *MainWindow*, but it is necessary to pass the beat to the *SalsaWindow* to classify if the user is to the beat or not.

The *SalsaWindow* holds each feedback separately. *Focus* refers to the *Look Straight* and *smile* to the *Smile* feedback. Feedback for upper body motion is implemented by *movebody*. The field *offbeat* implements the feedback if the user is to the beat or not, *Reset Dancing*. Note

6 Implementation

A concept of a system gives an image of how the system has to look like and how it should behave. However, the concept is still an idea. Therefore, this chapter is dedicated to the implementation of the concept that is presented in chapter 5. Further, it covers the definition of the BAML, and the design and implementation of feedback with the Kinect v2.

The UIs are similar to and inspired by the ones presented in section 5.4. A *MediaElement* that is attached to the *MainWindow* implements playing, pausing and stopping the music, beat and capturing with the Kinect v2. Note that it is possible to attach it to every dancing trainer component, like the *SalsaWindow*, separately. In both cases it is inconvenient for users to control playing, pausing and stopping with the mouse. Users have to move between the place they dance and where they control the DT from. For that reason, the DT additionally implements speech recognition for these controls to facilitate the overall usage. It enables with the loading of a dance component and reacts on the words *Play*, *Pause* and *Stop*.

The MenuBar of the *SalsaWindow* is divided in the following parts: *File*, *View* and *Settings*. In *File* users can save their performance and open it for review. The *View* allows the user to change between normal and tutorial view. The *Settings* cover the decision to practice basic forth and back or side Salsa steps and to display elements to change the difficulty of dancing. Users can enable beat audio support that highlights the beat with a *click* sound, or disable UI elements like the beat counter or the step suggestions for extra difficulty. The design of the UIs mainly uses common known *Windows* elements like Buttons, Labels or TextBoxes. But the dashboard seen while dancing is designed uniquely for the purpose of learning Salsa. It combines the images of basic Salsa steps like in figures 5.1 and 5.2, feedback and a beat counter exactly as suggested by the concept. Figure 6.1 shows an example how it looks like

for different scenarios. The beat counter is implemented by a periodic timer. Its interval is the Milliseconds Per Beat (MSPB) that is calculated from the BPM. The beat controls the step suggestion and the Kinect v2 recognizes the feedback.

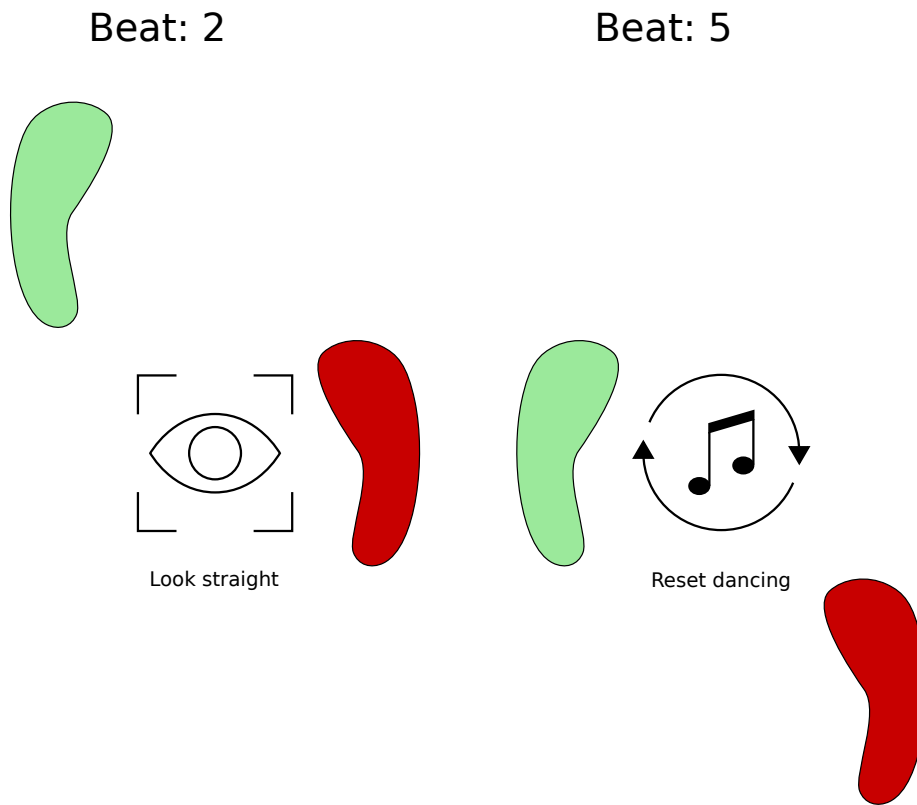


Figure 6.1: Examples of the dashboard during normal dancing sessions with *look Straight* feedback for step 2 (left) and *Reset Dancing* for step 5 (right).

Understanding the beat counter and the steps require prior knowledge of Salsa. Especially the feedback can be misinterpreted. Therefore, the DT implements the *TutorialWindow* that helps users gaining the knowledge they need to enter normal dancing sessions. In general, tutorials can be different, e.g. an expert explains the content directly to the user. But assigning a teacher to every student is not feasible. Instead, media like texts, images, videos or a combination can be used. In this work, a descriptive text supported by an image is used in the *TutorialWindow* to cover an introduction to basic Salsa steps, the dashboard and the feedback displayed while dancing.

Before using the DT you have to set it up correctly. The Kinect v2 is placed in front of a big screen such that the user faces the screen and the Kinect v2. The Kinect v2 faces the user,

too. The screen should be placed such that the user can look straight at it and does not have to bend his neck. Otherwise you automatically bend your neck looking at the screen if it is placed too high or too low. Thus, the user is forced to look straight and probably will not bend his neck. A normal PC screen or a bigger one, that you can connect to the PC, can be used. But it should be big enough such that the user can see everything in a distance from 2 meters from the Kinect v2. This is important to be recognized well by the Kinect v2. Being too close or far away does hinder the recognition. Plus, the user needs enough space to be able to dance properly and freely. A good place to set up the DT for private usage would be in the living room. Figure 6.2 shows the described setup of the DT.

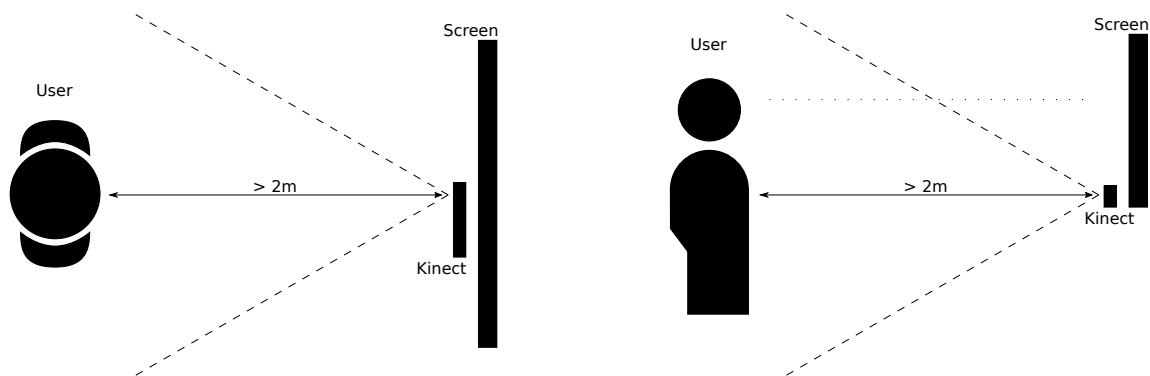


Figure 6.2: Setting up the DT. Left: top view. Right: profile view.

6.1 Standard for the Beat Annotated Music Library

6.1.1 Data Organization

The BAML is a directory with music as .wav files and a meta data file as .csv that contains meta data such as length and BPM of the music. Due to ambiguity of genres, the music files are not split in different directories according to their genre. A music piece can be assigned to different genres increasing the occurrence of that piece by the number of different assigned genres. Thus, organizing the music files in a single directory saves space.

6.1.2 Content

The content has been taken from www.audeeyah.com. The website provides free music to download and the BPM of the tracks that is taken as ground truth. Not all of the songs that are chosen has Salsa as genre. But they sound similar and are added to have more variety. The meta data for one song is saved in a .csv file. It contains the name of the song, the artist, its length in seconds, a collection of genre, the BPM, the path to the music file, the path to the BPM file and the path to the music file with audible beat. The BPM file contains the beat as audio that is denoted by a *click* sound. Table 6.1 shows the current collection of the BAML.

Song	Length [sec.]	Genre	BPM
Pan Comido	161	Latin,Salsa	110
Celebrating Life	142	Salsa	115
Adios	108	Salsa	110
Sunchine City	137	Salsa	100
Easygoing	127	Salsa	75
Move On	124	Salsa	155
Anhelo	117	Salsa	90
Friendships	142	Salsa	115
Golden Thoughts	137	Salsa	70
Exotic Envy	120	Salsa	160
Legendary Emotions	142	Salsa	115
Carnival	142	Brazilian Dance,Latin Pop	135
Club Sol	144	Brazilian Dance,Latin Pop	120
Copa Copa	137	Brazilian Dance,Latin Pop	115
La Fiesta	192	Brazilian Dance,Latin Pop	120
More Than Words Can Ever Say	128	Latin Pop	105
No More Tears	126	Latin Jazz,Latin Pop	130
Que Si Que No	102	Brazilian Dance,Latin Pop	150
Dance	142	Brazilian Dance,Latin Pop	126

Table 6.1: Content of the current *baml.csv*. The artist is www.audeeyah.com. The last three columns are filenames and not included in this table.

6.2 Multimodal Feedback Design

Dancing requires good coordination that comes from daily practice. Receiving proper feedback is an important property when learning [HT07]. It helps users to understand mistakes, how to correct them and what they are already doing well. Therefore, it is essential for the DT to transmit proper feedback to the user. [CK00] worked out five modalities how humans exchange knowledge. These are aural, gestural, linguistic, spatial and visual. This work makes use of them to communicate learning content or feedback to the user.

In fact, communication can be divided in en- [Kre03] and decoding [Pai90] messages. The usage of technology offers new perspectives for communication in learning and introduces Multimodal Learning Analytics (MMLA). The term MMLA was first used at the International Conference on Multimodal Interaction (ICMI) in 2012 [SWM12; Wor12] and later defined as: *"Multimodal Learning Analytics (MMLA) sits at the intersection of three ideas: multimodal teaching and learning, multimodal data, and computer-supported analysis. At its essence, MMLA utilizes and triangulates among non-traditional as well as traditional forms of data in order to characterize or model student learning in complex learning environments."* [Wor+16]. [Mit+18] classified multimodal data for learning that computers can use and defined a cycle for MMLA applications called Multimodal Learning Analytics Model (MLeAM). The model highlights the transition from the physical to the digital world by describing the processes how data is transformed. For a system like the DT this means that the user is captured with the Kinect v2, the resulting information processed and then fed back to the user. Here, the Kinect v2 functions as a translator of messages, en- and decoding them. The capturing refers to decoding and the feedback to encoding. Machine learning techniques are applied to derive feedback and work as a transition from gathered data to the provided feedback. However, this does not describe how to design the feedback. In fact, the final feedback model depends highly on the learning activity, thus it has to be designed for the purpose of the application [Mit+18]. This section is about designing a feedback model tailored to the DT with multimodalities for computers and the communication of feedback to the user.

In this work feedback is categorized in *Online* and *Offline Feedback*. *Online Feedback* is

displayed in real time while *Offline Feedback* can be reviewed after a dancing session was saved. Thereby, the feedback is inferred from data captured by the Kinect v2. The positive impact of feedback is indispensable [HT07] but too much feedback displayed in real time can be distractive to the user while dancing [Sch+15]. Thus, the feedback is displayed one at a time, shall not be much and very simple to understand [Sch+16]. A Salsa teacher was interviewed to know about common beginners mistakes and feedback was derived from it. Therefore, the DT support the following feedback: *Move Body*, *Look Straight*, *Reset Dancing* and *Smile*.

Online Feedback is simple but not detailed. It gives the user the opportunity to adapt his behavior instantly. *Offline Feedback* can be more sophisticated and covers this issue because users can spent more time reviewing it. The DT provides the user with two time lines. One time line summarizes all *Online Feedback* and highlights when it starts, is displayed and ends. The other time line shows a plot between the recognized steps of the user and the ones that the DT suggested.

6.2.1 Online Feedback

For this work the design of feedback only focuses on aural, gestural and visual modalities. The DT displays feedback to the user one at a time even if more feedback is recognized. That is because too much feedback can be distractive [Sch+15] and as a consequence the user puts too much attention to it such that his dancing performance suffers. Therefore, the recognized feedback has to be selected by the DT. The selection of feedback is inspired by the idea of teachers selecting their students when they raise their hand if they want to say something. The teacher functions as a scheduler and the students as the feedback. Students can raise their hand, say something when being selected, and lower their hand when they finish speaking or if they change their mind and do not want to speak anymore. Analogously, feedback has a recognition start, a display start and a recognition end. Note that the recognition of feedback can end without being displayed e.g. because another feedback is currently displayed. Thus, displayed feedback blocks others like two students that can not talk at the same time. Moreover, if feedback is being displayed it should not be displayed for some time such that other feedback gets a chance to be displayed and the user does not get overwhelmed with too

much consecutive feedback. In fact, the user needs some time to consume the feedback and adapt his behavior. The DT displays feedback for at least three seconds to give users enough time to consume it. Feedback is blocked for five seconds if the feedback is not recognized anymore to not display it over and over again. The display of it stops, too.

Aural

Following the beat in Salsa is not easy. There are different instruments playing at different rhythms. The DT supports the user with an acoustic signal that marks the beat if he enables audio beat support. The beat is highlighted with a *click* sound and plays together with the music. The music's volume is lower such that the *click* can be heard better. This is also a method to vary the difficulty.

Gestural

The DT interprets movements and steps as gestures, with the Kinect v2 representing the body with 25 joints. Therefore, it reacts on the gestures of the user. In a tutorial session the DT waits for the user to perform the next step that is displayed. Performing it, triggers the recognition and the next step is shown. But doing the steps correctly is not everything in dancing. Body language and emotions are forms of communication, too [CK00]. A bad posture while dancing can affect health. If the position of the head is bad, it can cause back issues in a long term [DWMZ96]. Beginners tend to look down to check their feet. But, it is important to look straight and have a good posture. Therefore, the DT recognizes if the neck is tilted too much and gives the *Look Straight* feedback. But, focusing too much on posture and as a consequence dancing rigidly, is not good neither. Engaging the whole body in Salsa is important to be considered as a fluent dancer. Therefore, upper body movement is also captured and if the user forgets to engage his whole body, the *Move Body* feedback reminds him. At last, smiling is a good way to have fun and transmit positive emotions to others while dancing [KK07; CL14; Luc14]. It is an important asset to let others feel good about you dancing. Therefore, the face expression is evaluated. For practice, it suffices if a little smile is recognized, because smiling excessively

can distract to focus on dancing. If no smile is recognized, then the user receives the *Smile* feedback.

Visual

The DT interprets body behavior as gestures and supports different visual features to assist the user while dancing Salsa. These are a beat counter, the step suggestion and the feedback. The beat counter shows the current beat to the user because it is not easy to hear it for Salsa music. Knowing the beat, means directly knowing the step, and the user can use the beat counter as a reminder of the step to make. Moreover, the DT suggests the steps directly to the user like in the guide of figure 5.1 or 5.2. You can see that every step is mapped to a beat and that the user can use both as an orientation. The beat counter and step suggestion are inspired from the idea that learners try to imitate and follow the Salsa steps of the teacher that is counting the beat. The different color for the left and right foot is intended to understand the images quicker because dancing can be fast. For the same reason, darker color indicates the foot that has to be moved. The simplicity of feedback is achieved by icons that are supported by small instructions that prevent users from misunderstanding it (figure 6.3). The instructions give a hint how to adapt the behavior. The feedback and its instructions are displayed between the feet of the step suggestion. The centered position allows users to see the feedback while looking on the steps. Beat counter, step suggestion and feedback are aligned as in the example of figure 6.1.

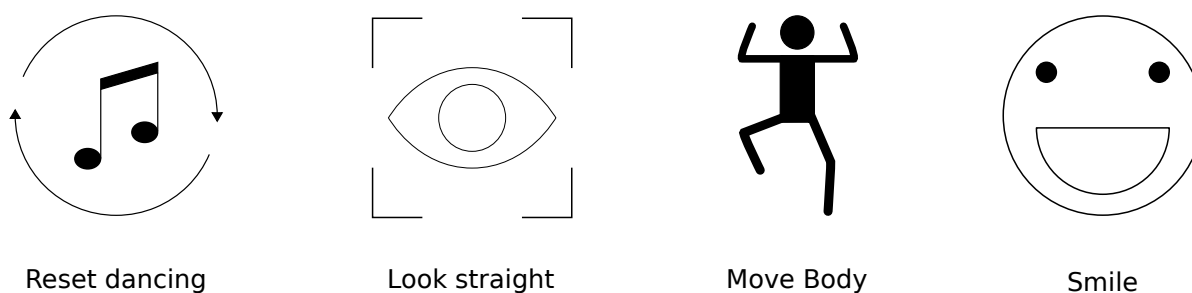


Figure 6.3: Feedback icons and instructions.

6.2.2 Offline Feedback

During sessions the user is concentrated on dancing and understanding the feedback he receives. Transmitting too much feedback can be distractive and the user does not benefit from it [Sch+15]. But a more detailed feedback can show the user more about his performance. After a dancing session the user can save and review it. The *Offline Feedback* is more sophisticated because the user has more time to spend on, understand it and infer new behaviors. For this work the DT provides two time lines as *Offline Feedback*. One shows the summarized *Online Feedback* received during the whole dancing sessions, and the other shows a plot of the recognized Salsa steps of the user and the ones the system suggests. The user can see if he is too late, early or in sync with the suggested steps. Figure 6.4 shows a possible output of the *Offline Feedback* whose data is saved in a JSON file (listing 6.1).

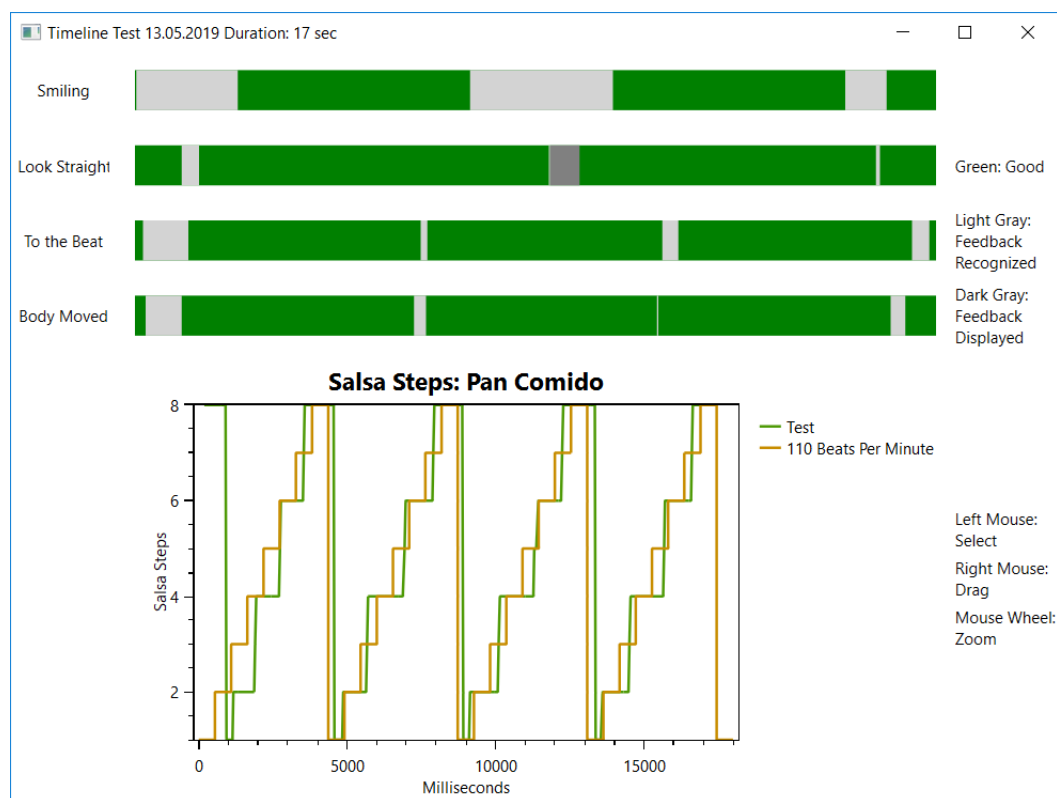


Figure 6.4: An example of the *Offline Feedback*. Top: Summary of *Online Feedback* that distinguishes between good (green), recognized (light gray) and displayed (dark gray). Down: Plot of ms and Salsa steps between the suggested (orange, 110 BPM) and the recognized steps (green, Test).

Note that values for the feedback are given in milliseconds and can be not a number (NaN), too. This is because not every feedback is displayed or terminated. This happens if the dancing session gets stopped. If so, then it is assumed that the end of the feedback is equal to the end of the dancing session. Besides, the actual display end is not considered. The time line shall be kept simple and understandable. Otherwise cases where the display end is greater than the recognition end or vice versa occur. This adds complexity to the visualization.

```
1 {"Date": "13.05.2019",
2  "TotalDuration": 17379.7635,
3  "Song": "Pan Comido",
4  "BPM": 110,
5  "Name": "Test",
6  "Feedback": [{"Instruction": "Reset Dancing",
7                "Feedback Start": 186.02020000000002,
8                "Display Start": "NaN",
9                "Feedback End": 1165.0811},
10 {"Instruction": "Move Body",
11    "Feedback Start": 227.89290000000003,
12    "Display Start": "NaN",
13    "Feedback End": 1001.0959},
14 {"Instruction": "Look straight",
15    "Feedback Start": 8966.7207,
16    "Display Start": 9021.4052000000011,
17    "Feedback End": 9640.4087}],
18 "PlotSalsaSteps": [{"ms": 186.02020000000002, "beat": 8},
19                     {"ms": 902.032, "beat": 8},
20                     {"ms": 933.3265, "beat": 1},
21                     {"ms": 1134.5677, "beat": 1},
22                     {"ms": 1165.0811, "beat": 2},
23                     {"ms": 1866.5728000000001, "beat": 2},
24                     {"ms": 1897.3221, "beat": 3},
25                     {"ms": 1935.2157000000002, "beat": 4},
26                     {"ms": 2699.3567000000003, "beat": 4},
27                     {"ms": 2732.6637, "beat": 5},
28                     {"ms": 2766.0117, "beat": 6},
29                     {"ms": 3507.1893000000005, "beat": 6},
30                     {"ms": 3535.172, "beat": 7}]}
```

Listing 6.1: JSON example for *Offline Feedback*.

6.3 Construction of Multimodalities in the Dancing Trainer

6.3.1 Working with the Kinect

Infrared / depth camera	Size Field of view	512 x 424 px 70.6 x 60 degrees
Color camera	Size Field of view	1920 x 1080 px 84.1 x 53.8 degrees
Capturing distance	Minimum Maximum	0.5 m 4.5 m
Framerate		30 Frames Per Second (FPS)
Microphone Array		4 pieces

Table 6.2: Technical information about the Kinect v2. [Fan+15][Jia+17]

Technical Requirements

In order to be able to use the Kinect v2 you have to meet some requirements. They can be found on the official websites but the information is not consistent. E.g. the german xbox website recommends Windows 7 while the english xbox website recommends Windows 8.¹² The download page for the Kinect SDK recommends Windows 8, too.³ The operating system and IDE used for this work is Windows 10 Education and Visual Studio 2017. Since the hardware recommendation does not differ much the Kinect v2 shall work with the following requirements:

- 32-bit (x86) or 64-bit (x64) processor
- 2 GB RAM or more
- dual core with 2,66 GHz
- USB 3.0

Another mentionable requirement is the space needed. Make sure that you have enough space such that the Kinect v2 can capture you well. The capturing distance for the Kinect v2 is from

¹ <https://support.xbox.com/de-DE/xbox-on-windows/accessories/kinect-for-windows-setup#1b54b150c26247ddb1cd8eed876f879>

² <https://support.xbox.com/en-US/xbox-on-windows/accessories/kinect-for-windows-v2-setup>

³ <https://www.microsoft.com/en-us/download/details.aspx?id=44561>

0.5 m to 4.5 m. Other technical data is given in table 6.2. You can see an image of the device in figure 6.5.

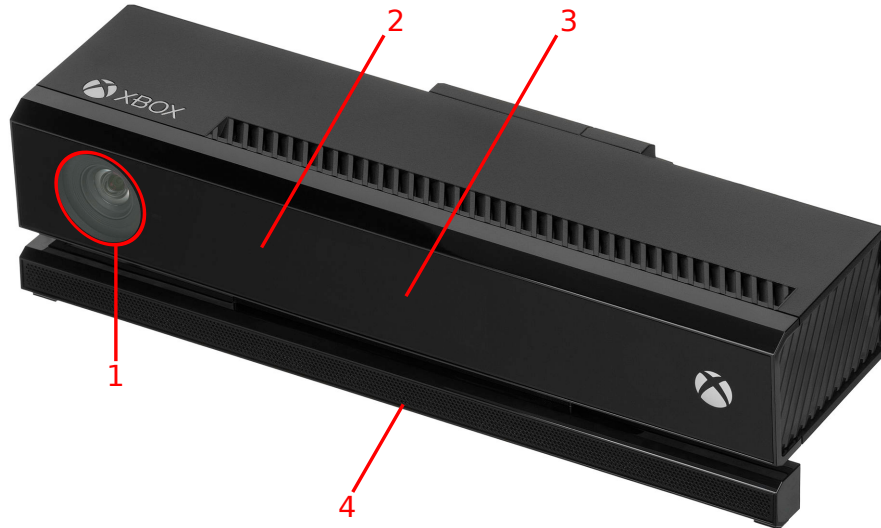


Figure 6.5: The Kinect v2 with 1: Color Camera, 2: Microphone Array, 3: Infrared Camera, 4: Infrared Projector

Skeleton Data

The Kinect v2 captures humans as bodies. The data collected and describing the body is called skeleton. The Kinect v2 represents the skeleton with 25 joints and refers to the skeleton of human bodies. The joints are defined in a 3D CCS of the camera. Every joint has a position and orientation, except the joints of the feet. They do not have an orientation. Figure 6.6 shows all the joints of the human body that are captured by the Kinect v2.

Operating in Different Spaces

It is important to know about the different spaces and the coordinate systems of the Kinect v2, too. The Kinect v2 has 4 different spaces: color-, depth-, infrared- and cameraspace. Most of them has different coordinate systems. This means that data has to be mapped to the right space during run time before processing it further. One example is displaying the skeleton of

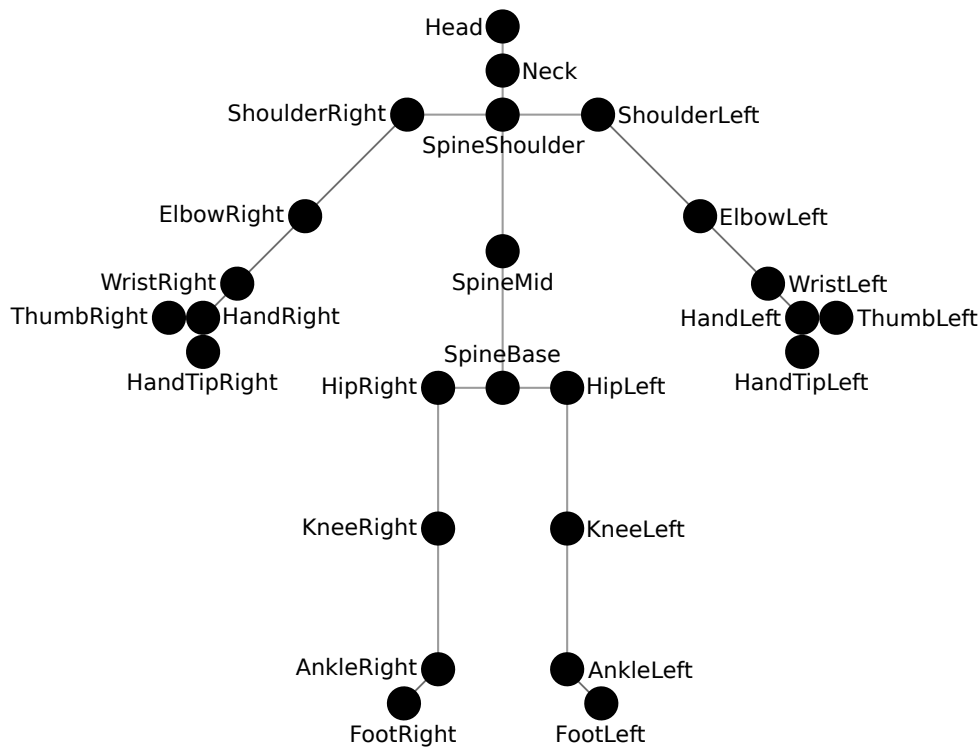


Figure 6.6: Image of the human body and the joints of the skeleton that are captured by the Kinect v2.

a person's body. You have to map the skeleton data that is given in the cameraspace to the colorspace to display the skeleton correctly.

The CCS for the colorspace is 2D and has its origin at the top left corner (subsection 4.2.1). The coordinates of the axes range from 0 to 1 representing percentages of the total width and height of the image seize. Looking at table 6.2 remember that the seize of color images is 1920 (width) x 1080 (height) pixels. E.g. let us take the point $p=(0.5,0.5)$. The point p corresponds to the middle point of the image with pixel values of (920,540).

Looking at table 6.2 you see that depth and infrared images have the same image seize. So, their spaces are treated the same way. The CCS is 2D and the origin is on the top left corner (subsection 4.2.1. The coordinates reach from 0 to 1 representing percentages of the total width and height of the respective image. Note that the seize is smaller then for images in the colorspace. Let us take the same example as in the previous subsection. With a seize of 512 x 424 pixels for infrared-/depthspace, p would have a the coordinates (256,212).

The cameraspace uses a 3D CCS as defined in subsection 4.2.2. The camera is its origin and the information represented in it are the joints of a skeleton. From the sensor's point of view the axes grow as follows: x left, y up and z straightforward. Joints are given as a 3D vector and their orientation is given by quaternions (equation 4.5). Note that tilting the camera affects the axes and that calculations may include the tilting angle of the camera.

Gesture Detection

Kinect Studio (KS) is a software that gets installed with the Kinect SDK. It offers the developer to engage with the different functionalities of the Kinect v2. You can view different sensor data like depth, color and infrared. You can record clips (videos), play them and monitor a 2D or 3D view. Recording clips is an important step when building gestures. The VGB is another tool that is installed with the installation of the Kinect SDK. It helps creating your own gestures that fit your application. In this case basic Salsa steps are the gestures to create and recognize. In order to do that, clips has to be recorded (KS) and tagged (VGB). The tagging of clips is done manually for each frame and identifies if the desired gesture happens or not. Tagged clips are used for training or testing of the built gesture recognition. Note that the tagging of the clips is crucial for the behavior of the learned model. E.g. if your gesture shall only be recognized when you use your right arm then frames shall only be tagged when this is the case. If the left arm is used then you can tag those frames as false. It is suggested to tag clips as consistent as possible to not include errors from tagging, and to record raw clips (15 Bit) and convert them into processed (11 Bit) to prevent re-recording from update changes.⁴ Clips can be converted from raw into processed with *KSConvert.exe*. The VGB distinguishes between 2 gesture types: discrete and continuous.

Discrete gestures are binary and always tagged as true or false. They describe when the gesture happens or not, e.g. stretching the right arm to the side. The output is a confidence value ranging from 0 (not confident) to 1 (confident) that the gesture is detected. All frames where the right arm is fully stretched to the side are tagged as true. The others are tagged as false. The data shall resemble the gesture as accurate as possible. In some cases you

⁴ <https://channel9.msdn.com/Blogs/k4wdev/Custom-Gestures-End-to-End-with-Kinect-and-Visual-Gesture-Builder>

want to include special clips with a specific behavior such that the recognition can be trained to recognize it, too. E.g. add clips where the gesture is performed slower if speed is not a factor in the gesture. Discrete gesture are implemented by *Decision Trees*. *Decision Trees* are part of supervised machine learning because the classification of the training data is known. It is a binary tree where each node contains a decision. The decision is binary and can only be true or false, and it is made upon a feature (e.g. x coordinate) of a data instance. The feature that splits the data set the best is selected by a certain criteria, like the gini impurity or information gain. This process continues until the data set can not be split anymore or if a maximal depth of the tree is defined. The leafs of the tree represent the final classification of the tree. The final classification does not have to be clear. But in the case the tree is not deep enough or the set can not be separated anymore, the class with the most supported instances in a leaf node can be chosen as final decision. *Decision Trees* are simple classifiers and can only be used if the data is linearly separable (Figure 6.7). Graphically, the data can be divided clearly by one line. [Gar17]

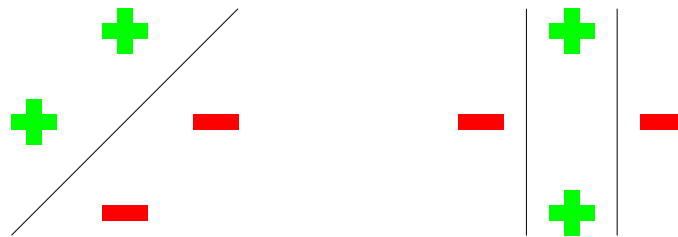


Figure 6.7: Graphical visualization of linear separability. Left: linearly separable. Right: Not linearly separable.

Continuous gestures are used whenever you want to show the progress of the gesture. The classification is denoted by a float value between 0 and 1. Intuitively 0 means that the gesture was not started and 1 that it is completed. But this depends on the use case and therefore can be different. Let us take the same gesture as in the example for the discrete one. As a continuous gesture 0 means that the right arm is not stretched at all and 1 that it is fully stretched. If the gesture would be a steering motion then 0.5 can represent no steering, while 0 is steering left and 1 steering right. Continuous gestures are implemented by *Random Forests*. *Random Forests* are part of supervised machine learning, too. If the data is not linear separable multiple *Decision Trees* are combined to create a *Random Forest*. Using multiple classifiers and combining them into one is known as *Ensemble Learning*. The core idea is that each

Decision Tree is trained on a different training set. The underlying training data can be split randomly such that the different *Decision Trees* are not correlated. The method used to split the data into multiple training sets is called *Bagging*. Moreover, a subset of features can be chosen randomly for each *Decision Tree* to prevent strong predictors to be selected over and over again. Otherwise, this would result into a high correlation between *Decision Trees*. The classification of a *Random Forest* can be made by a majority voting of each *Decision Tree*. That means that the class with the most votes among all *Decision Trees* is chosen as final classification. [Gar17]

Gestures can be very simple or complex. Depending on the use case either a discrete or continuous gestures can be used for a simple gesture. However, when dealing with more complex gestures then it may not be enough to use only one gesture type. You rather break the gesture down into more simple ones that are recognized with a discrete or continuous gesture. Thus, multiple gesture types are combined into one more complex gesture to e.g. recognize dancing steps.

6.3.2 Step Recognition with the Kinect

A step is a complex movement to be recognized that has different phases like lifting the foot, moving and placing it down. In this work, steps are recognized by using the joint information of the captured body. With that, there are several approaches to choose from. Two of them are coded heuristics or machine learning. First describes the crafting of rules to recognize the steps. Second describes an automated way to find those rules. Intuitively, the automated method should be preferred. However, the downside is the lack and difficulty of control. Coded heuristics can hard code elements a *Decision Tree* has to find hardly or can not find at all, like in the case of linear separability (Figure 6.7).

Coded heuristics can use the difference of the foot joints and the floor to see if the foot is raised or not. The floor can be retrieved directly as plane from the Kinect v2. But there are two problems with this approach. The recognition of dancing steps depend on the detection of the floor. The difference between the floor and the joints can vary depending on the shoes users wear. E.g. woman may dance with high heels. Both make the manually finding of thresholds a

guessing process.

That is why a *Decision Tree* was tested to see if and how these thresholds could be learned. The tree was trained with information about potentially important joints. These are the joints in the lower body below the hips. The following list shows the features for one data instance:

- Left and right foot position and their distance to the floor
- Left and right ankle position, orientation and their distance to the floor
- Left and right knee position, orientation and their distance to the floor

This results into a total of 36 features for the *Decision Tree*. Note that there are no orientation for the foot joint because the Kinect v2 does not capture it. The training and testing data is captured with KS. Different distances to the Kinect v2 are captured to make the decision independent of the distance to the Kinect v2. So, six classification labels were chosen as follows:

1. Standing far
2. Standing near
3. Left grounded, right raised, near
4. Left grounded, right raised, far
5. Left raised, right grounded, near
6. Left raised, right grounded, far

The trained *Decision Tree* had a depth of 14 and a rounded average Precision, Recall and F-Score of 1 (equations 4.6-4.8). The data set with $n=60968$ instances was split 1:1 for training and testing. However, the scores do not reflect the quality of the step recognition, because it was nor built or tested for it. This experimental approach was conducted in *Python* despite the programming language of the DT being *C#*. This approach was not followed because of the assumed sensitivity toward shoes. In fact, the feature with the highest importance was the z coordinate of the ankle. The data did not cover different shoes and was produced by the developer. However, a machine learning approach seems promising. Instead of crafting the data and allowing individual errors, VGB should be used to directly define dancing steps as gestures. Plus, VGB takes over the task deriving data such as velocity or acceleration that otherwise had to be calculated thoroughly.

Define Salsa Steps As Gestures

This work uses VGB for the Kinect v2 to build gestures that recognize Salsa steps. Custom *Decision Trees* can be built but the usage of a tool that suits the Kinect v2 is assumed to facilitate the process. Thus, features like relative position, velocity or acceleration are already supported and do not have to be calculated separately. Gestures can be discrete, continuous or a combination of both and are built in the following process: recording of clips, tagging of clips, training and testing of the detector. Note that the tagging depends on the gesture type, thus the type has to be known and the gesture created before. The training phase builds the final detector.

A Salsa teacher was invited to perform basic Salsa steps. The clips were recorded raw to not record them from scratch if the underlying software changes. The recordings contain the steps of table 6.3. Every clip was recorded with faster and slower motion and for training and testing. That makes 32 clips in total. The steps were recorded with a different stepping velocity such that the gesture is trained to not regard it. The stepping velocity depends on the speed of the song. The higher the BPM the faster you need to dance. Every step was performed six times for training and three times for testing to have a train-test ratio of 2:1. The clips are converted into processed with *KSConvert.exe* because VGB only uses processed clips.

Category	Starting Step	Salsa Steps
Basic Forth and Back	left forth	half (1-2-3-4)
	right back	half
	left forth	full (1-2-3-4-5-6-7-8)
	right back	full
	left back	full
	right forth	full
Basic Sidewards	left	full
	right	full

Table 6.3: Recorded clips with KS. Every clip has a fast and slow version for training and testing. The total number of clips is 32.

The gestures created to recognize basic Salsa steps are *ForthAndBackProgress_Left*, *FootTapping_Left* and *FootTapping_Right* for forth and back, and *SideStepProgress_Left*, *Side-*

FootTapping_Left and *SideFootTapping_Right* for side steps. Table 6.4 shows each gesture with the according training clips, gesture type and correspondence of Salsa steps.

Gesture Name	Training Clip	Gesture Type	Salsa Step
ForthAndBackProgress_Left	left forth full fast left forth full slow left forth full demo* right forth full fast right forth full slow	continuous	1, 3, 4, 5, 7, 8
FootTapping_Left	right back half fast right back half slow left forth full slow anti bobbing** left forth full demo*	discrete	2
FootTapping_Right	left forth half fast left forth half slow	discrete	6
SideStepProgress_Left	left side full demo* left side full fast	continuous	1, 3, 4, 5, 7, 8
SideFootTapping_Left	right side half slow	discrete	2
SideFootTapping_Right	left side half slow	discrete	6

Table 6.4: Gestures to recognize basic Salsa steps that use joints in the lower body and regard left and right. (*) recorded much slower and with stances to target the independence of speed (**) bobbing caused false classification

Note that not all clips were used because it turned out to work better with less. Further, some gestures have auxiliary clips denoted by asterisks. They were recorded afterwards to optimize the recognition. So called *demo* clips emphasize even more on the independence of speed by containing standing positions and an even slower execution of steps. An *anti bobbing* clip was added because bobbing caused a false classification and triggered the gesture falsely. Note that *FootTapping_Left* has more training clips than its counterpart, *FootTapping_Right*. This is due to the attempt of designing the gesture more robust.

The tagging of the discrete gestures is straightforward. For both, forth and back, and side steps, frames were tagged as true if step 2 or 6 happened to capture the tapping motion of the foot. For both continuous gestures, they have 0.5 as a neutral value when you stand with both feet aligned. Stepping with the left foot forward or to the side causes an increase up to 1. Analogously, a decrease down to 0 is caused when stepping with the right foot back or to the side. Figure 6.8 visualizes this behavior with the execution of step 5 for basic forth and back

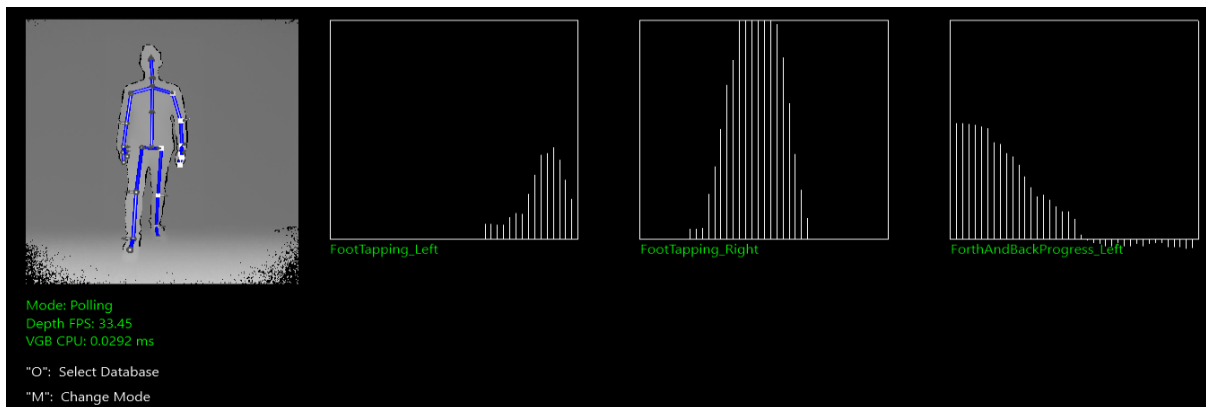


Figure 6.8: Interaction of the gestures for step 5 of the forth and back Salsa steps. From left to right: user, *FootTapping_Left*, *FootTapping_Right* and *ForthAndBackProgress_Left*.

steps.

The clips were tagged as consistent as possible because erroneous tagging affects the training, thus the final classification. However, it is not feasible to tag clips with 100% correctness or consistency. This is not only due to human tagging but abrupt motions that cause breaks between frames. This means that there is no fluent transition of motion between two frames. It depends on the speed of motion or the frame rate of the capturing device being too low. This makes tagging confusing, especially for continuous gestures where the progress has to be described while respecting the motion breaks. A reason why some gestures need extra clips can be due to differences in tagging. Finished with tagging, the gestures were built and tested in the live preview. Figure 6.8 shows this for step 5. You can see that both tapping gestures are triggered. But *FootTapping_Left* has a lower confidence value than *FootTapping_Right*. In fact, there is no tapping with the left foot when stepping back for step 5. But, you can see that the triggering happened at the end. This means a transition to step 6. Further, *FootTapping_Right* can be ignored when stepping back even if it is triggered. During testing with the live preview it was observed that the overall confidence of *FootTapping_Left* is lower than *FootTapping_Right*. However, *FootTapping_Right* was triggered more often, even if the step did not happen. Figure 6.8 captures both behaviors exactly. This could be caused by the attempt of designing *FootTapping_Left* more robust which results into a drop of the overall confidence.

These facts are used to come up with thresholds to recognize Salsa steps. The correct interpretation of the gestures prevents from false recognition. Thereby, the output of the gesture

detectors are combined to exclude each other, e.g. *FootTapping_Right* is only considered if a left step forward was recognized before. The empirically chosen thresholds to recognize Salsa steps are shown in table 6.5. Note that the thresholds are defined with a little bit of freedom.

Step	Forward and Backward Steps	Side Steps
1	$\text{ForthAndBackProgress_Left} \geq 0.95$	$\text{SideStepProgress_Left} \geq 0.95$
2	$\text{ForthAndBackProgress_Left} \geq 0.95$ and $\text{FootTapping_Right} \geq 0.6$	$\text{SideStepProgress_Left} \geq 0.95$ and $\text{SideFootTapping_Right} \geq 0.15$
3, 4, 7, 8	$\text{ForthAndBackProgress_Left} \geq 0.45$ and $\text{ForthAndBackProgress_Left} \leq 0.55$	$\text{SideStepProgress_Left} \geq 0.45$ and $\text{SideStepProgress_Left} \leq 0.55$
5	$\text{ForthAndBackProgress_Left} \leq 0.05$	$\text{SideStepProgress_Left} \leq 0.05$
6	$\text{ForthAndBackProgress_Left} \leq 0.05$ and $\text{FootTapping_Left} \geq 0.05$	$\text{SideStepProgress_Left} \leq 0.05$ and $\text{SideFootTapping_Left} \geq 0.7$

Table 6.5: Conditions to recognize basic Salsa steps.

Instead of recognizing step 1 when the progress value is exactly 1, it is recognized at least at 0.95. It can happen that the progress value of individual movements is not 1 in every case. But, selecting thresholds more freely can help including those cases, while remaining sure that the gesture happens. Another reason is that gestures are detected with a delay. The gesture happens first and then it is detected. So, defining thresholds like that can help to decrease this delay without miss detecting gestures.

6.3.3 Classification of Dance Movements to the Beat

Gestures are defined to recognize basic Salsa steps, but it remains unclear how to determine if the user is dancing to the beat or not. The way to classify if dancing movements are to the beat, is to determine the beat of the user and compare it with the original beat. Therefore, the time stamp of the recognition is compared to the beat that the DT suggests. For simplicity, it is assumed that steps has to happen consecutively. This means, that the recognition can not jump from e.g. step 2 to 5. The user is forced to do all steps from 1 to 8 correctly. If not, then the DT tries to find the most probable step in the next cycle if the step was not recognized again.

The original beat is given by the BPM and a stopwatch measures the time stamps of the

recognition. Having multiple timing elements (stopwatch, timer) can potentially cause measuring problems, although observations do not show that. To prevent greater time deviations the difference between the starting times of the timer for the beat and the stopwatch is subtracted, too.

Nevertheless, humans might not perform the step at the exact moment. Therefore, the DT shall still classify correct steps even if the user is a little bit too early or too late. A timing window that allows users to still be to the beat has to be chosen. Humans perceive elements as instantaneous if they happen with a delay of 100 ms [Ric67]. The delay is too short to be noticeable for humans such that it becomes meaningless. Thus, setting the window for correct steps to 100 ms sounds promising. But, real beginners rather follow the teacher or in this case the step suggestion (figure 6.1) and need time to react. Considering a delay of 100 ms sounds too optimistic. Instead, let us look at the average reaction time of humans to make it easier following the beat, especially for beginners. The average reaction time of humans lies between 220 ms and 384 ms [Kos12]. For this work, a window with a delay of 250 ms to be to the beat is chosen as an educational guess.

Another method could be to play the step suggestion faster. However, the DT is not a game where users have to react as fast as possible. Users are not prepared and focused on dancing. Besides, giving the step suggestion to the beat can enforce users to not focus too much on it. Probably, users will be too slow when reacting only, and start focusing on the music and feeling the beat. The step suggestion becomes an orientation.

Knowing the original beat, the beat that the user produces with his Salsa steps and the window of 250 ms, being to the beat can be classified. Let t_{orig} be the time in ms of the current original beat given by the BPM and t the time in ms of the beat that the user produces. There are 3 cases for correct steps if the recognized Salsa step is equal to the suggested one:

1. $t_{orig} = t$

The user is perfectly to the beat

2. $t_{orig} - 250 \leq t < t_{orig}$

The user is to the beat but a little bit too early

3. $t_{orig} < t \leq t_{orig} + 250$

The user is to the beat but a little bit too late

6.3.4 Is your Body Moving when Dancing?

The whole body is engaged while dancing and beginners unintentionally forget about it because they are more concerned about e.g. the coordination of their feet. However, it is at least as equally as important because otherwise dancing can be perceived as stiff. Plus, reminding beginners to move make them enjoy dancing [KK07; CL14]. Involving the whole body is especially true for Salsa because the steps are the only elements that define it. This makes Salsa a free dance style. In fact, there are different ways to dance Salsa (section 5.2). In this sense it is more than necessary to engage the upper body. Therefore, the DT has to recognize upper body movement during dancing sessions.

The Kinect v2 is used to capture the motion vectors of joints such that upper body motion can be classified. Motion vectors are calculated by calculating the difference between the joint positions of the current and previous frame (equation 4.1). So, they represent the motion of a joint from one frame to the next one. Generally, basic Salsa steps are about moving forth and back, or sideways. Let us consider only forth and back movement because sideways movement can be handled analogously.

This means, that each joint of the upper body has a forward or backward movement when dancing. If there is no engagement of the upper body the motion ideally represents the (opposite) direction of the z axis of the Kinect v2. The z Axis functions as a reference motion vector. So, the more the upper body moves the more it does not represent the ideal forward or backward movement, thus the z Axis. Here, upper body motion is denoted by the angle (equation 4.4) between the ideal and the real movement, that is represented by the motion vectors of the joints (equation 4.1). The joints that are considered for the classification of upper body motion are:

- Left and right hands
- Left and right wrists
- Left and right elbows
- Left and right shoulders

Other joints like *Head*, *SpineShoulder* or *SpineBase* are not considered because they are more rigid and not as free to move as elbows or hands. Before calculating the motion vectors of the joints, their 3D position is smoothed by rounding. This is important because values jitter due to capturing and cause motion to be recognized even if the user is standing still and not moving. Smoothing the values by rounding reduces the jittering. Values were rounded on the second decimal because it was observed to smooth enough and not too much.

Smaller angles are excluded from classification due to the natural motion that happens when shifting your weight while walking. That is why angles in the range of 35 and 145 degrees are considered as motions. This means, that the respective joint contains enough motion that is not considered walking or rigidly stepping forth and back. The final classification is made by a weighted vote between joints where enough movement was detected. The weighting is made up empirically with 20% distributed equally on the shoulders and 80% distributed equally on the other joints. The shoulders are weighted less because they are connected to the spine, thus they are more rigid. The result is a confidence value if upper body motion is happening or not. If the confidence is greater or equal than 0.77 the upper body is moving and engaged while dancing.

6.3.5 Putting Emotions into Dancing

Although the DT focuses on the goal to assist the user learning how to dance, having fun and communicating positive emotions can not be ignored. Putting emotions into dancing is important as stated by the interviewed Salsa teacher. Plus, it lets you feel good [KK07; CL14]. Smiling is a facial expression that expresses happiness and pleasure and helps getting positive emotions better across. In fact, having fun can benefit learning, too [Luc14]. The DT uses the face capturing of the Kinect v2 to recognize if the user is smiling or not. There are three possible results of the smile detection: smiling, not smiling and maybe smiling. The *Smile* feedback of the DT is triggered only if the detection is sure that the user is not smiling. This avoids that users constantly have to put a big smile on their face. Moreover, it can be perceived as unnatural and distract from dancing.

6.3.6 Having a Good Posture in Salsa

In general posture is part of nonverbal communication and can transmit characteristics like confidence or insecurity [Arg13]. Besides that, good posture in Salsa has another meaning. Dancing with the wrong posture influences healthiness [DWMZ96]. Looking to the ground and bending the neck while dancing can put pressure on the spine. This can result into back problems. A common beginners mistake is exactly looking down at their feet to check if they are doing the steps correctly. The DT gives feedback if the user is not looking straight to prevent health issues due to a bent neck while dancing. The angle between the *Head*, *Neck* and *SpineShoulder* joints is calculated (equation 4.4) and if it is not between 170 and 190 degrees the feedback is triggered. The calculation contains the directional vectors from the neck to the head, and from the neck to the *SpineShoulder* joint (equation 4.1).

7 Evaluation

7.1 Recognition of Feedback

In this work detectors and classifiers are built to communicate feedback to the user. They are evaluated to give an insight how good they work. These are classifiers for the recognition of basic Salsa steps, *Move Body*, *Smile* and *Look Straight* feedback.

Discrete gestures were evaluated with the VGB Analyzer, an analyzing function of the tool. Unfortunately, the analyzer was erroneous for continuous gestures and had unexplainable NaN values. That is why continuous gestures are evaluated with the final basic Salsa steps recognitions, one for forth and back, and one for side steps. The data for the evaluation sets was collected by performing the gestures with the DT and writing down if the recognition appeared or not. The tutorial mode of the DT was used especially for recognizing steps, because it waits until the step is performed correctly. For the other feedback, a small adaption that visualized the triggering of the recognition was added. The setup to collect the data is the same as in figure 7.1. The steps are recorded from 1 to 8 meaning that the evaluation assumes the previous step to be done.

Gesture Name	#Frames	Average Root Mean Square (RMS)
FootTapping_Left	1405	0,2586579
FootTapping_Right	1375	0,6125951
SideFootTapping_Left	1393	0,5851721
SideFootTapping_Right	1389	0,3082733

Table 7.1: Evaluation of the discrete gestures with the VGB Analyzer.

Table 7.1 shows the output of the VGB Analyzer for the discrete gestures. The analyzing tool provides average RMS, FPs and FNs. However, the FPs and FNs are not displayed in the evaluation because of the lack of documentation to understand its values, e.g. *FootTapping_Right* has a FP of 1.6. The RMS indicates the delay to detect the gesture. You can see that *FootTapping_Left* and *SideFootTapping_Right* has a RMS less than 0.31. The other two gestures have a higher RMS with 0.61 for *FootTapping_Right* and 0.59 for *SideFootTapping_Left*. You can assume that one gesture can be faked easier if it has an higher RMS. The delay is so high such that it can be interpreted as a new gesture. Potentially, this could lead to a higher FP. There is no specific rule for the RMS that determines a gesture as good. Ideally the RMS should be 0 but it is unclear if it is possible to achieve or not. Observations and experiences show that the RMS for the discrete gestures do not interfere with the overall Salsa step recognition. One reason is the exclusion of gestures when recognizing steps described in subsection 6.3.2. Another reason is the more loosely defined thresholds in table 6.5 to recognize the dancing steps.

Table 7.3 shows the confusion matrix and scores of the overall forth and back basic Salsa steps recognition. The evaluation set has $n=160$ instances with 82 being positive and 78 negative. Each of the steps are represented by $n=20$ instances. Table 7.2 shows the confusion matrices and scores for every individual forth and back Salsa step. Positive instances are produced by a *good* user that tries to perform each step correctly. Negative instances are produced by a *malicious* user that tries to trick the classification. He does false steps on purpose e.g. performing step 5 or sliding instead of stepping if step 1 is asked. Notice that in table 7.2 step 4 and 8 always have a perfect Precision, Recall and F-Score of 1. You can notice the same in table 7.4 for side steps. The reason is that steps 3 and 4, or 7 and 8 end in the same position such that the next step is recognized instantly. This means that producing other instances is harder because of physical boundaries. However, it does not mean that steps 4 and 8 can not be faked at all. The gestures *ForthAndBackProgress_Left* and *SideStepProgress_Left* have a value of around 0.5 if there is no movement or movements the gestures does not know. E.g. a side steps does not change the value of *ForthAndBackProgress_Left*. But, it can cause a slight increase. The same goes for *SideStepProgress_Left* with a forward step.

Most F-Scores in table 7.2 are greater or equal than 0.7. Except for step 3 and 7 where the

Step 1	Negative	Positive	
Predicted Negative	11	0	11
Predicted Positive	2	7	9
	13	7	20
Precision	Recall	F-Score	
0.78	1	0.88	

(a) Confusion matrix and scores for step 1.

Step 3	Neg.	Pos.	
Pred. Neg.	5	0	5
Pred. Pos.	8	7	15
	13	7	20
Precision	Recall	F-Score	
0.47	1	0.64	

(c) Confusion matrix and scores for step 3.

Step 5	Neg.	Positive	
Pred. Neg.	12	0	12
Pred. Pos.	1	7	8
	13	7	20
Precision	Recall	F-Score	
0.88	1	0.93	

(e) Confusion matrix and scores for step 5.

Step 7	Neg.	Pos.	
Pred. Neg.	6	0	6
Pred. Pos.	7	7	14
	13	7	20
Precision	Recall	F-Score	
0.5	1	0.67	

(g) Confusion matrix and scores for step 7.

Step 2	Neg.	Pos.	
Pred. Neg.	11	2	13
Pred. Pos.	2	5	7
	13	7	20
Precision	Recall	F-Score	
0.71	0.71	0.71	

(b) Confusion matrix and scores for step 2.

Step 4	Neg.	Pos.	
Pred. Neg.	0	0	0
Pred. Pos.	0	20	20
	0	0	20
Precision	Recall	F-Score	
1	1	1	

(d) Confusion matrix and scores for step 4.

Step 6	Neg.	Pos.	
Pred. Neg.	12	1	13
Pred. Pos.	1	6	7
	13	7	20
Precision	Recall	F-Score	
0.86	0.86	0.86	

(f) Confusion matrix and scores for step 6.

Step 8	Neg.	Pos.	
Pred. Neg.	0	0	0
Pred. Pos.	0	20	20
	0	20	20
Precision	Recall	F-Score	
1	1	1	

(h) Confusion matrix and scores for step 8

	Precision	Recall	F-Score
μ	0.78	0.95	0.84
σ	0.19	0.1	0.14

(i) Average μ and standard deviation σ for all the steps.

Table 7.2: Confusion matrices and scores of the basic forth and back Salsa steps recognition. Each step is represented by n=20 instances.

All Steps	Negative	Positive		Precision	0.79
Predicted Negative	57	3	60	Recall	0.96
Predicted Positive	21	79	100	F-Score	0.87
	78	82	160		

Table 7.3: Confusion matrix and scores of the overall basic forth and back Salsa steps recognition.
n=160

F-Score is 0.64 for step 3 and 0.67 for step 7. In fact, step 3 and 7 have the lowest Precision values with 0.47 for step 3 and 0.5 for step 7. Looking at the Recall you can see that the steps are mostly recognized if they are performed correctly. Even if the Precision of step 3 and 7 is low, they have a perfect Recall of 1. This is not the case for steps 2 and 6, where the Recall is 0.71 for step 2 and 0.86 for step 6. Although Precision, Recall and F-Scores might not be enough to evaluate classification problems, the tendency can be assumed that the Salsa step recognition for forth and back steps works if the steps are performed correctly. However, a malicious user is able to fake the classification. E.g. sliding instead of stepping triggers the recognition every time. In every case the Recall is at least as high as the Precision. You can observe it in table 7.2 for separate steps and in table 7.3 for the final Salsa recognition for forth and back steps. The standard deviation in table 7.2 is at most 0.19 and could support that the scores of the individual steps are stable.

Table 7.5 shows the confusion matrix and scores for the overall basic side Salsa steps recognition. The evaluation set has n=160 instances with 82 being positive and 78 negative. Table 7.4 shows the confusion matrices and scores of each individual side step. Each of the steps has n=20 instances. The data is collected the same way as for the evaluation set of forth and back Salsa steps. The lowest F-Scores in table 7.4 are 0.64 for step 7 and 0.67 for step 3. The Precision is again lower than the Recall for all steps. The lowest Precision values are 0.47 for step 7 and 0.5 for step 3. All steps have a perfect Recall of 1. Both Salsa step recognitions show the same pattern that correctly performed steps are recognized. The Precision has the highest standard deviation of 0.18 for the individual side steps. However, the tendency can again be assumed that steps are recognized if they are done correctly. Experience with the Salsa step recognition underlines this assumption. Table 7.5 shows the scores of the final Salsa step recognition for side steps. It has a Precision of 0.77, a Recall of 1 and an F-Score of 0.87. The standard deviations of table 7.4i range from 0 to 0.18. The Precision has the highest

Step 1	Negative	Positive	
Predicted Negative	11	0	11
Predicted Positive	2	7	9
	13	7	20
Precision	Recall	F-Score	
0.78	1	0.88	

(a) Confusion matrix and scores for step 1.

Step 3	Neg.	Pos.	
Pred. Neg.	6	0	6
Pred. Pos.	7	7	14
	13	7	20
Precision	Recall	F-Score	
0.5	1	0.67	

(c) Confusion matrix and scores for step 3.

Step 5	Neg.	Pos.	
Pred. Neg.	11	0	11
Pred. Pos.	2	7	9
	13	7	20
Precision	Recall	F-Score	
0.78	1	0.88	

(e) Confusion matrix and scores for step 5.

Step 7	Neg.	Pos.	
Pred. Neg.	5	0	5
Pred. Pos.	8	7	15
	13	7	20
Precision	Recall	F-Score	
0.47	1	0.64	

(g) Confusion matrix and scores for step 7.

Step 2	Neg.	Pos.	
Pred. Neg.	11	0	11
Pred. Pos.	2	7	9
	13	7	20
Precision	Recall	F-Score	
0.78	1	0.88	

(b) Confusion matrix and scores for step 2.

Step 4	Neg.	Pos.	
Pred. Neg.	0	0	0
Pred. Pos.	0	20	20
	0	0	20
Precision	Recall	F-Score	
1	1	1	

(d) Confusion matrix and scores for step 4.

Step 6	Neg.	Pos.	
Pred. Neg.	10	0	10
Pred. Pos.	3	7	10
	13	7	20
Precision	Recall	F-Score	
0.7	1	0.82	

(f) Confusion matrix and scores for step 6.

Step 8	Neg.	Pos.	
Pred. Neg.	0	0	0
Pred. Pos.	0	20	20
	0	20	20
Precision	Recall	F-Score	
1	1	1	

(h) Confusion matrix and scores for step 8

	Precision	Recall	F-Score
μ	0.75	1	0.85
σ	0.18	0	0.13

(i) Average μ and standard deviation σ for all the steps.

Table 7.4: Confusion matrices and scores of the basic side Salsa steps recognition. Each step is represented by n=20 instances.

All Steps	Negative	Positive		Precision	0.77
Predicted Negative	54	0	54	Recall	1
Predicted Positive	24	82	106	F-Score	0.87
	78	82	160		

Table 7.5: Confusion matrix and scores of the overall side Salsa steps recognition. n=160

value with 0.18 and the Recall the lowest with 0. The F-Score has a standard deviation of 0.13. You can argue that the standard deviation could support that each individual side step is stable in this case, too.

	Negative	Positive		Precision	0.79
Predicted Negative	40	12	52	Recall	0.76
Predicted Positive	10	38	48	F-Score	0.78
	50	50	100		

Table 7.6: Confusion matrix (left) and scores (right) of *Move Body* feedback. n=100

Table 7.6 shows the confusion matrix and scores of the evaluation of the *Move Body* feedback. The evaluation set has 50 positive and 50 negative instances that regard motions like arm lifting, body rotation, moving, standing still and a combination of those. The evaluation has a Precision of 0.79, a Recall of 0.76 and an F-Score of 0.78. Here, Precision and Recall are more or less equal.

	Negative	Positive		Precision	1
Predicted Negative	5	0	5	Recall	1
Predicted Positive	0	10	10	F-Score	1
	5	10	15		

Table 7.7: Confusion matrix (left) and scores (right) of *Smile* feedback. n=15.

Table 7.7 shows the confusion matrix and scores of the evaluation of the *Smile* feedback. The evaluation set has 10 positive and 5 negative instances. Positive instances are represented by smiling and maybe smiling. Negative instances are represented by no smiling. The confusion matrix yields a Precision, Recall and F-Score of 1. Perfect scores seem too optimistic. In fact, the evaluation set contained only 15 instances. But note that small smiles are considered as enough. Further, the detection of smiling is done by the Face Basics of the Kinect v2 and an explicit evaluation of it could not be found.

Table 7.8 shows the confusion matrix and scores of the evaluation of the *Look Straight* feed-

	Negative	Positive		Precision	1
Predicted Negative	12	0	12	Recall	1
Predicted Positive	0	20	20	F-Score	1
	12	20	32		

Table 7.8: Confusion matrix (left) and scores (right) of *Look Straight* feedback. n=32

back. The evaluation set has 20 positive and 12 negative instances. Positive and negative instances are produced by looking straight, turning and bending your head slightly and fully in different directions. Slight motions are considered as positive because they are likely to happen while shifting weight when stepping. The confusion matrix gives a Precision, Recall and F-Score of 1.

To sum up, the evaluation shows that the recognition of Salsa steps and feedback works. In fact, it is good enough because they all have an F-Score greater than 0.8 as defined in the non functional requirements (subsection 5.3.2). The *Move Body* feedback is the only exception with an F-Score of 0.78. There is the possibility to spoof the system by faking the steps. Sliding instead of stepping always tricks the classification for forth and back, and side steps. However, users generally have a serious intent such that they probably do not trick the system because ultimately they trick themselves. Generally, the tapping movements on 2 and 6 are harder to recognize than the other steps. This is indicated by an overall lower Precision. This can be said for step 3 and 7, too. But have in mind that the continuous gestures usually have a neutral value of 0.5 if you stand or do wrong movements. Thus, it is not surprising that they have a lower Precision, too.

7.2 Experiment

A pilot study is conducted to investigate the quality of the DT that is built with this work. It helps finding current strengths to emphasize on and weaknesses to reduce. Plus, it gives insight about the acceptance and user experience of the DT. Figure 7.1 visualizes the procedure of the pilot study described by the following phases:

1. Pre-Questionnaire
2. Salsa Introduction
3. Salsa Training
4. Dancing Session
5. Performance Review
6. Post-Questionnaire

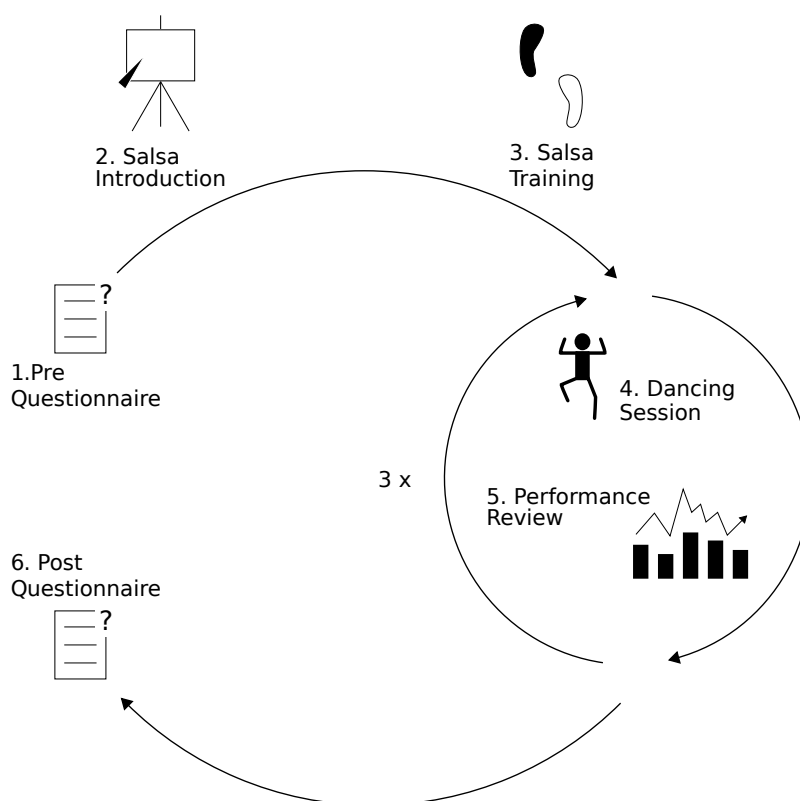


Figure 7.1: Procedure of the experiment for the DT.

At the beginning the participant is asked to fill in a pre-questionnaire that focuses on the knowledge, skills and experiences of the participants before the usage of the DT. The introduction to Salsa and the tutorial of the DT ensures the acquisition of a base knowledge about the DT and Salsa among the participants. After practicing the steps with the tutorial mode of the DT the participant is ready to enter dancing sessions that in contrast provide feedback. For the experiment three songs are chosen with different levels of difficulty denoted by the BPM. These levels are easy (BPM = 70), intermediate (BPM = 90) and hard (BPM = 110). The cycle starts with the easiest song but the participant is given the choice to change it between the cycles. In total there are three cycles that start with a dancing session of one minute and end with a performance review. The participant can self reflect and decide how he wants to adapt his behavior. After all the cycles are completed the participant is asked to fill in a post-questionnaire that focuses on the user experience of the DT.

The experiment was set up as in figure 6.2. Devices used are a big screen that the participants were facing, the Kinect v2 to recognize the behavior of participants and a *Windows 10* PC from where the experiment was controlled by the experimenter.

In total, 25 participants participated in the experiment, 48% were male and 52% female. The age of the participants ranged from 19 to 49 with an average of 29.16 and a standard deviation of 7.22. Although the participants provided feedback with the questionnaires, it is important to mention observations that happened during the experiment, too.

Participants found the usage of the DT vitalizing and refreshing. Many liked to change the song between each cycle. Generally, 70 BPM were perceived too slow. Unexpectedly, a higher BPM turned out to yield better results. Not changing a song and attending it in a second cycle resulted into a better performance, too. Participants wanted to try out different songs to have fun. But changing the song every cycle, usually did not improve the performance. This can be explained by participants having to adapt their behavior to a new and unknown song. This is certainly not the case if participants stayed with a song at least twice. They were more used to it, thus the performance improved.

The perception of 70 BPM being too slow could be due to the tutorial that participants attended before the cycle. The tutorial was perceived as fast because the next step is shown instantly. Although telling the participants that they decide the tempo, the observation was that

they were driven by the images. Thus, participants increased their tempo unconsciously. Another downside of the instant display of the next step was that the pauses on 4 and 8 were skipped. The end position of step 3 and 7 is the same as 4 and 8. This results into immediately recognizing the pause and showing step 5 or 1.

Other remarks regarded the coloring of the images and the images themselves. The coloring could be a problem for people who have a dyschromatopsia and can not distinguish green from red. However, this is not an evidence that those people would suffer substantial disadvantages recognizing the steps. On the contrary, people that have a right-left disorientation are likely to have more problems, like they would in a dancing course. Other participants reported that the images of the steps (figure 5.1) were not intuitive. Participants did not know how exactly to move their feet even if the darker color indicated the foot that has to be moved. This means that images might not be enough to show how Salsa steps are done. That is why the DT provides an additional description when entering the tutorial mode. Besides, there were no comments of participants on the description. However, a video that shows how to perform basic Salsa steps was suggested by participants.

Other observations were about the recognition of Salsa steps. The recognition did not work every time right from the beginning. This could be due to the erroneous recognition or the user performing the step wrong. However, it can be frustrating when you believe that you correctly performed a step and that the system does not react at all. This is especially the case in the tutorial mode because the DT waits for the user until he performs the step correctly. For normal dancing sessions the user has only insight about this information in the post analysis. But after some practice participants got used to the behavior of the Salsa step recognition.

It was also observed that participants unconsciously stepped forward while dancing. However, this did not hinder the recognition substantially. But one participant wore a dress that occluded his legs and that hindered the recognition of Salsa steps.

Participants questioned the *Online Feedback* and did not perceive the same events as the system. But, it was observed that participants liked the *Online Feedback* and that it was perceived according to their performance. This was also the case for *Offline Feedback*. However, the plot between the recognized steps of the participants and the suggested steps had to be explained often. This means that the DT lacks in explanation.

Pre-Questionnaire	μ	σ
Age	29.16	7.22
How confident do you feel in your dancing skills?	3.56	2.18
How much experience do you have in dancing?	3.38	2.16
How motivated are you using this application?	7	1.96
Post-Questionnaire	μ	σ
How confident do you feel in your dancing skills?	4.88	1.86
How natural would you rate your experience with the application?	5.68	1.63
How motivated would you be to use this application again?	6.56	1.78
How helpful was the feedback displayed in real time?	6.36	1.58
How likely would it be that you use this application in your free time?	5.36	2.27
Do you feel like you learned something while interacting with the application?	6.92	1.61
How does using this application compare to how you would normally learn the same content?	5.52	1.92
I have experience with applications that are similar to the one that I just tested.	2.92	2.41
I felt like following the steps and instructions of a real dancing teacher.	4.84	2.25
Looking at the post analysis made me aware of my performance.	6.8	2.04
Looking at the post analysis helped me improve my skills.	6.44	2.33
I would recommend this application to a friend.	6.6	2.12
The application is fun to use.	7.6	2.04

Table 7.9: Closed Questions and average answers with standard deviation. Answers range from 1 to 10 with 10 being considered the most and 1 the least. Total observations of $n=25$ participants with 52% female and 48% male participants.

Table 7.9 shows the average answers and standard deviation of closed questions from the pre- and post-questionnaire. Closed questions could only be answered with a value from 1 to 10. For all questions 10 is considered as the highest and 1 the lowest value. Looking at the table, you can see that most participants learned something using the application and had fun using it. Participants did not have any or little experience with similar applications. They have little dancing experience, too. You can see that the average confidence before using the DT is much lower than the other average values. However, after using the DT participants felt an increase in confidence. On the contrary, the motivation drops a little bit. All the other average values are in the range of around 5 to 6, thus mediocre. The standard deviation is never greater than 3, except for the age. Therefore it can be assumed that all the answers lie near the average.

The pre- and post-questionnaire contained open questions, too. Open questions could be answers in text form and are evaluated separately. The answers were categorized and the categories counted to give a better overview over the answers. Note that the naming of categories

and the assignment of answers is subjective. Another person would probably name categories and assign answers differently. The assignment of answers to categories shall only give an overview about the answers and what they could aim for. Thus, the according figures that visualizes open questions has to be understood with caution. Figure 7.2 shows the histogram of the categorized answers in the pre-questionnaire. In this case, the question was multiple choice with the possibility of adding a custom answer. This means that categories partially correspond to the answers of the multiple choice. Figures 7.3, 7.4 and 7.5 show the histograms of the categorized answers of the post-questionnaire.

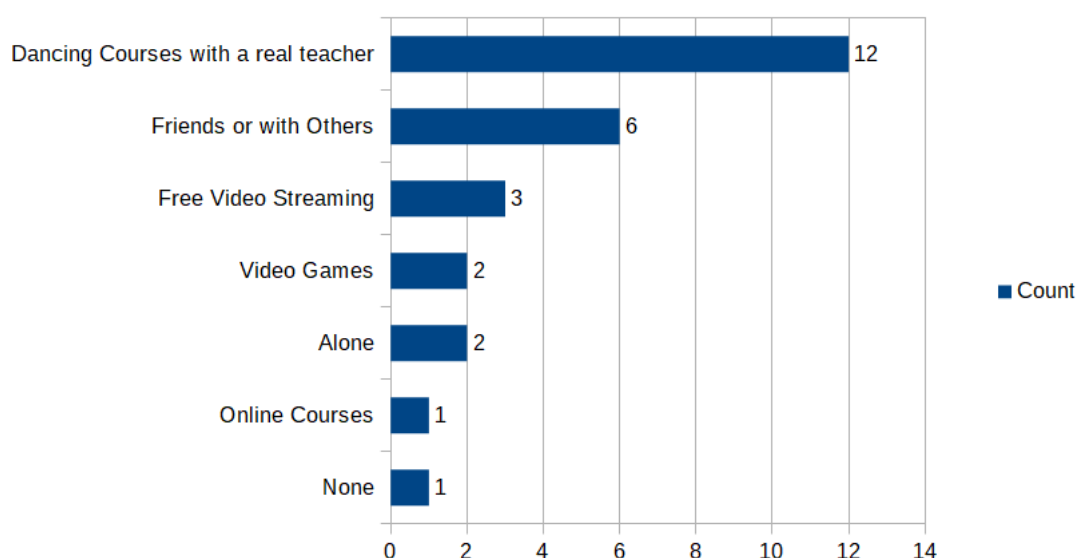


Figure 7.2: Histogram of the categorized answers of the question *"Where does your dancing experience come from?"* in the pre-questionnaire.

At the beginning of the experiment, participants had to fill in a pre-questionnaire that captured their knowledge and confidence before the usage of the DT. One question were about the dancing experience of participants to get an idea about their dancing skills. You can see in figure 7.2 that 12 participants have visited a dancing course with a real teacher. Note that the answers of this particular question are multiple choice and that the chart does not give any information about the duration of the dancing course and when it has been visited. The same goes for the other categories. However, it can be assumed that the participants did not visit a dancing course for long or that it was long ago because of the relatively low average value of 3.38 of the corresponding question *"How much experience do you have in dancing?"* in table 7.9. But there is no further data that supports this assumption. Other participants have their

dancing experience from video games or online courses. This means that participants do not have experience with similar applications like the DT. Table 7.9 could support this because of the low average value of 2.92 of question *"I have experience with applications that are similar to the one that I just tested."*

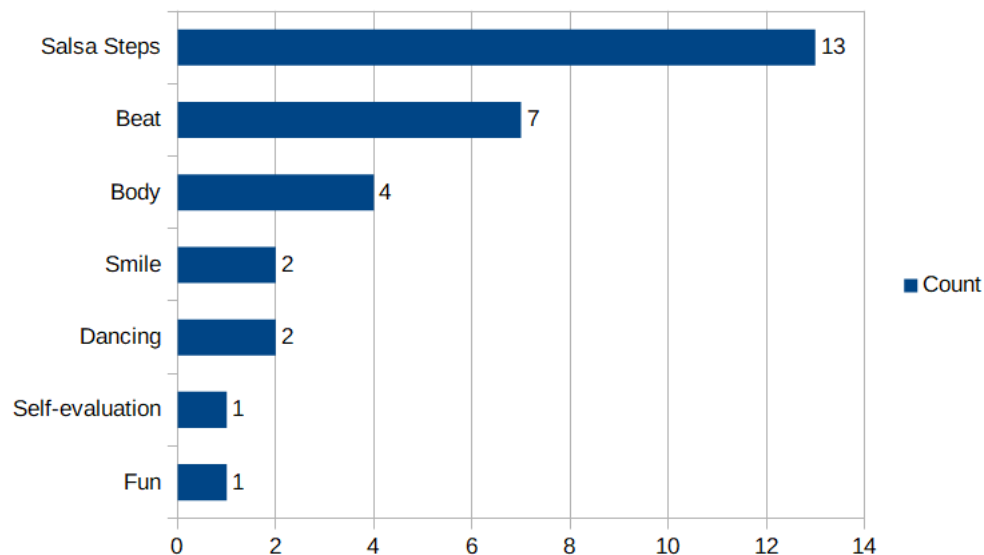


Figure 7.3: Histogram of the categorized answers of the question *"What do you feel like you learned from using this application?"* in the post questionnaire.

In the post-questionnaire participants had to describe what they learned after using the application. Figure 7.3 shows the categorized answers. Participants stated that they learned the most about *Salsa Steps* and *Beat*. The Salsa step pattern, staying to and knowing the beat or not were learned. This reflects the low average value for confidence and dancing skills of the participants in table 7.9. Moreover, participants learned about smiling more and how to control their body.

Figure 7.4 shows the histogram of the categories of the question *"List the most three positive aspects."* Participants experienced the *Online Feedback* and the simple usage of the DT as the most positive aspects. They had fun and liked the *Offline Feedback*, too. Participants felt positive that they improved and that they could learn alone. It was perceived as positive that basic Salsa steps could be learned and that different songs could be chosen. On the bottom of the chart participants state positive effects on the confidence and that they felt more active. Only one participant each found the experience realistic or interesting.

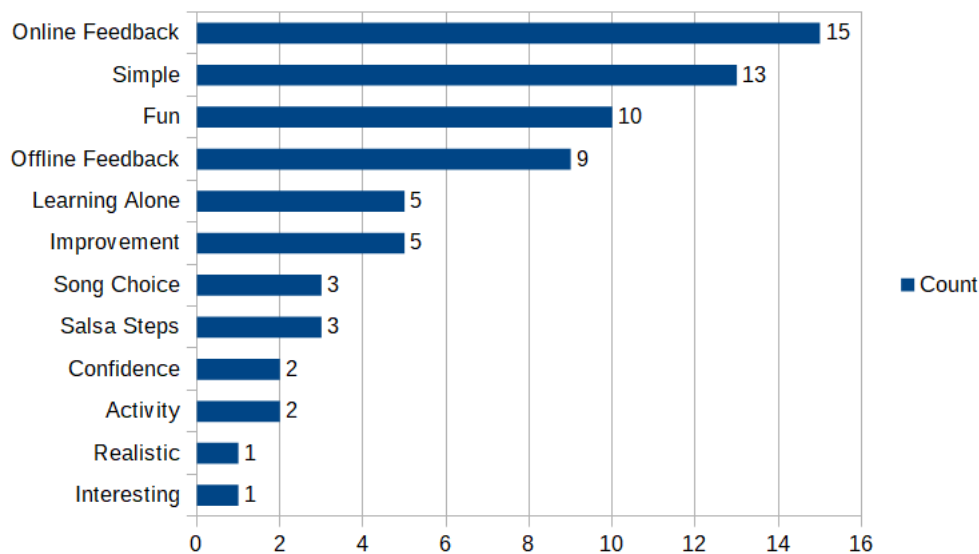


Figure 7.4: Histogram of the categorized answers of the question *"List the most three positive aspects."* in the post questionnaire.

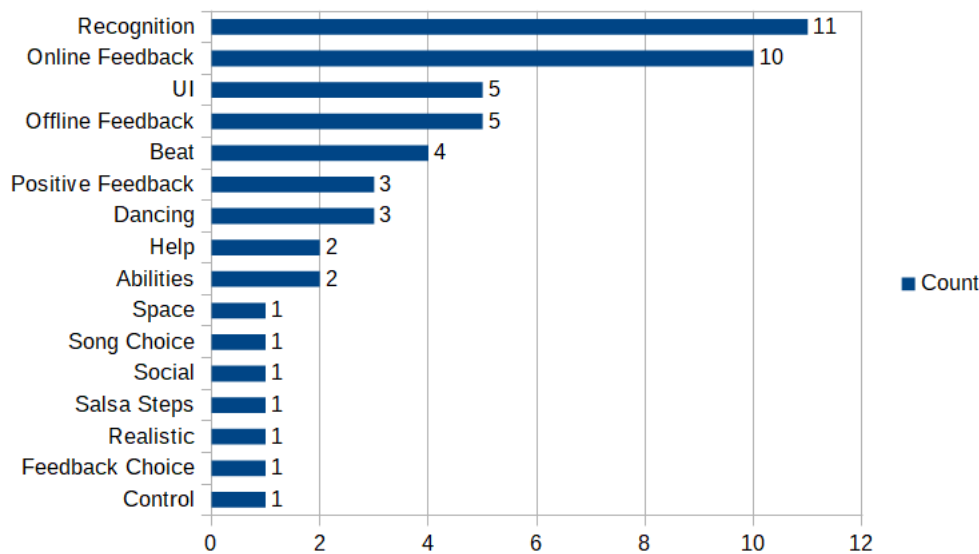


Figure 7.5: Histogram of the categorized answers of the question *"List the most three negative aspects."* in the post questionnaire.

Figure 7.5 shows the histogram of the categories of the question *"List the most three negative aspects."* Among all, the recognition of the Salsa steps and the *Online Feedback* stand out the most. Participants stated that the recognition does not work 100% and that they had the impression of performing the steps correctly. *Online Feedback* was either perceived as not completely agreeable, not detailed enough or confusing. Participants suggested a display of

the real time plot of their recognized steps such that they could compare it directly with the step suggestion. Others suggest that feedback could be more concrete, e.g. telling that you are too slow/fast instead of *Reset Dancing*. However, others found that the feedback and the step suggestion interfered with each other. Further problems of the UIs were about the readability and the brightness of the background. Participants did not agree with the *Online* or *Offline Feedback* every time. Further, *Offline Feedback* was too complicated to understand. However, observations state that participants were conform with the feedback they received online and offline. Other answers referred to the beat not being aligned to the song and that the step suggestion was increasingly getting off the beat while dancing. This can be an indicator that the different timing functions of the DT interfere with each other when dancing for a longer period of time. One participant stated that he did not understand why he received the *Reset Dancing* feedback. Participants desired more help and positive feedback from the system, too. The DT would also encourage a false dancing behavior, e.g. stomping instead of stepping. On the bottom of the chart participants complained about not having much control, no choice over the feedback or songs, and the realism of the DT. Two participants were concerned about the space you need and the lack of social interaction. Participants suggested other improvements like audio feedback, seeing themselves or the steps that are recognized while dancing and the addition of more constructive or positive feedback.

t-Test Confidence		
Alpha	0.05	
Hypothesized Mean	0	
	Post	Pre
Mean	4.88	3.56
Variance	3.44	4.76
Observations	25	25
Observed Mean Difference	1.32	
Variance of the Difference	2.06	
Degree of Freedom	24	
t Stat	4.6	
P(T≤t) two-tail	0.0001	
t Critical two-tail	2.06	

Table 7.10: T-Test to see if there is a statistical difference in confidence of dancing skills before and after the usage of the DT.

Table 7.10 shows the result of a two-tail t-Test on the confidence of the participants before and after the usage of the DT. You can see that there is an increase between the mean values for the confidence. The two-tail t-Test explains if the observed increase in the difference of mean values is significant. This is the case if the p value is lower than the alpha value, or if the critical value is lower than the *t stat* value. The alpha value (usually) is 0.05 and the p value is given by $P(T \leq t) = 0.0001$. The *t Stat* value of 4.6 is greater than the critical value of 2.06, too. You can see that with a Degree of Freedom of 24 the p value or critical value are lower than the alpha or the *t Stat* value. The Degree of Freedom is the number of values that can be changed without affecting the outcome. This means that the observed difference of mean values in the confidence is significant.

t-Test Motivation		
Alpha	0.05	
Hypothesized Mean	0	
	Post	Pre
Mean	6.56	7
Variance	3.17	3.83
Observations	25	25
Observed Mean Difference	-0.44	
Variance of the Difference	6.84	
Degree of Freedom	24	
t Stat	-0.84	
$P(T \leq t)$ two-tail	0.41	
t Critical two-tail	2.06	

Table 7.11: T-Test to see if there is a statistical difference in motivation of using the DT before and after.

Table 7.11 shows the outcome of a two-tail t-Test on the motivation participants felt before and after using the DT. With a Degree of Freedom of 24, the critical value and the p value are greater in this case. That means that the difference in mean values for motivation is not significant. The data does not support the observation of a difference in mean values.

Finally, you can say that the participants did not have much dancing experience but enjoyed using the DT. The aspects they liked the most were the ones they did not. Participants liked the *Online* and *Offline Feedback* but they stated how it could be improved. Put simply, it should be clearer, more detailed and less interfering. The recognition of Salsa steps were questioned by the participants, too. Thus, it has to be optimized. Although, participants had difficulties at the beginning, they adopted their behavior to the system quickly. Generally, more assistance and

constructiveness were desired. The outcome on the usage of the DT does not say anything about the change in motivation but as shown by the t-test the data supports the observation of an increase in confidence.

8 Result

This work proposes the DT as a generic MMLA application to rare or missing human interaction approaches to learn dancing. The framework presented by [GD12] was used in order to design the DT as a generic LA application. Therefore stakeholders, objectives, data, instruments, external constraints and internal limitations are defined. Stakeholders were divided into three different types of relationships to have with the DT: Business-to-Business, Business-to-Person and Person-to-Person. Despite the bigger objective of the DT being to assist the user to learn dancing, the objective could be broken down to *reflection* and *prediction*. These are worked out in the context of the defined relationships that categorize the stakeholders. A part of the objective is giving feedback to the user. Therefore, the DT needs to collect data to derive feedback from.

The Kinect v2 is used as the main instrument to capture this data with the user as its source. Note, that motion data could be abused by e.g. health insurances to determine the health condition of the patients in order to calculate individual payments [SS15; SDT17]. Other instruments that were used in this work were a computer and the tools for the Kinect v2 such as KS to record (raw) clips, *KSConvert.exe* to convert them into processed and VGB to create gestures that recognize Salsa steps based on machine learning approaches.

Data regulation laws as the GDPR, the necessity of devices, enough space to dance and the social acceptance towards digitalization were defined as external constraints. Internal limitations were the understanding of the system and the possible drawback of lacking social aspects that exist in dancing courses.

Furthermore, this work suggests a concept for the DT that was created based on the defined uses cases from which functional and non functional requirements were inferred. UIs, the

navigation between them and their interaction were presented, too. The architecture of the DT is defined to support the addition of other dance styles.

The BAML was created as a temporal solution such that Beat Tracking could be excluded in this work and for non available music libraries that contain an annotation of the beat. In this work the BPM was chosen to annotate the beat of songs. Users can choose songs from the BAML that they like to dance to. The decision to create the BAML was due to the exclusion of Beat Tracking for this work. The exclusion was based on the tasks complexity and the challenges that had to be faced with it.

The DT uses the Kinect v2 to collect data, process it and infer feedback from. Feedback is provided to assist users to learn dancing. Although, the general system is designed with a generic framework [GD12], the feedback is not. Due to the usage of technology and the ability of collecting multiple types of data with it, the model to design MMLA application presented by [Mit+18] was chosen to design the underlying feedback. The Kinect v2 is used as a sort of translator of modalities that describe methods how knowledge is exchanged. These are aural, gestural, linguistic, spatial and visual. The Kinect v2 decodes the users behavior and uses the modalities to encode feedback that is provided to the user. A Salsa teacher was interviewed to come up with feedback that beginners do when they learn Salsa. The feedback that is designed by this work tells the user if he is smiling, dancing to the beat, engaging his body while dancing and looking straight to maintain a good posture.

This work shows that basic Salsa steps could be defined as gestures. Basic Salsa steps are forth and back, and side steps. The recognition of basic Salsa steps is handled by smaller gestures that focus on individual parts of the whole stepping sequence. Brought together, the smaller gestures built the final recognition. The clips to build the gestures were recorded with a Salsa teacher. The evaluation shows that basic Salsa steps could be recognized well enough.

An experiment to capture the user experience, current strengths and weaknesses of the DT was conducted as a pilot study at the end of this work. It shows how the usage with the DT was perceived. Users had fun dancing with the DT but had difficulties with the recognition of Salsa steps at the beginning. After some practice, users could adopt their behavior. Users liked the *Online Feedback* but they had a lot of ideas how to improve it, e.g. showing a real time plot of the recognized steps at run time. In rare cases they did not perceive the same feedback as

the DT. However, too much *Online Feedback* was also perceived as distractive when paying attention on dancing. Overall, users stated that the DT is a good application to start with and that it appears to be good to learn basic Salsa steps, have fun and get active. The DT caused an increase of confidence in the users dancing skill.

Summing up, this work successfully presents the DT as a MMLA application to learn basic Salsa steps that is liked by users and that is designed to add other dance style. Thus, the DT is a generic system to learn dancing. Further, it is shown that motion data captured by the Kinect v2 can be used to define gestures that recognize basic dancing steps.

9 Discussion

This work presents the DT as a solution for different dance styles to establish a machine driven approach that supports humans to learn how to dance. This chapter discusses current problems on the Kinect v2 as a device to use, the BAML and Beat Tracking, the limitations of recognizing basic dancing steps, the particular recognition of Salsa steps, its evaluation, the designed feedback and the conducted experiment.

The Kinect v2 plays a main function because it captures important data of the user that is processed to provide users with feedback. A big problem for the development of the DT is that the Kinect v2 is no longer being sold.^{1,2} Thus, other alternatives that can follow the same approach of this work could be used to replace the Kinect v2.

Beat Tracking is valued as an important asset that enhances the DT. But as explained in subsection 5.2.1 it can be a challenging task e.g. due to tempo changes. This is especially true for Salsa because different instruments play at different rhythms that overlap. That is why Beat Tracking is excluded in this work and the BPM is used as a simple assumption for the beat in the BAML. However, the beat denoted by the BPM does not have to be the beat the instruments are playing to. In fact, the assumed BPM does not regard the downbeat. Hence, the DT suggests a false beat to the user. But, building the BAML is still considered as useful because it can be a starting point to develop similar applications and work around the challenges of Beat Tracking. Beat Tracking should be considered for future versions of the DT and the BAML should be built more sophisticatedly.

Establishing a dancing corpus with gestures that are created to classify basic Salsa steps is possible as shown by this work. However, there is a limitation that depends on the complexity

¹ <https://www.polygon.com/2017/10/25/16543192/kinect-discontinued-microsoft-announcement>

² <https://www.engadget.com/2015/04/02/rip-kinect-for-windows/>

and variety of dance movements. Dance styles where body parts get occluded are assumed to be problematic. That is why pair dances like Salsa are problematic, too. Additional cameras might be required that has to be synchronized. It is assumed that other basic dancing steps can be recognized by building gestures, too.

Basic Salsa steps could be recognized and built by combining discrete and continuous gestures. However, the recognition is not perfect, e.g. steps 2 and 6 are harder to perform. Plus, standing too long in step 1 or 5 made the recognition false. This is due to the ambiguity of both stances being equal, thus the system could not recognize or memorize if users did a left step forward or a right step backward. Besides, the whole evaluation process is done manually. Steps or gestures are performed and the reaction of the system is written down as true or false. Detectors for smiling and looking straight have perfect scores but users did not always understand why the respective feedback appeared. The number of instances of the respective evaluation sets might be too small. Plus, every person has a different understanding of smiling or looking straight. Both depend on individual facial and postural traits. The same goes for the evaluation about the upper body motion. Motion vectors seem to work well enough but it is problematic to capture every possible free motion. Besides, there will always be some kind of movement when stepping due to weight shifting. An attempt has been made to build a gesture that recognizes standing still such that the opposite output of the recognition can be taken to classify free motion. But it was without success. Speech recognition was implemented as method to control the DT but quick trials showed that it is very sensitive. The speech recognition was triggered by the sound of the dancing steps. However, every recognition task should be optimized, e.g. finding the best thresholds.

Online and *Offline Feedback* were designed to support users with their learning. The feedback was liked by users but improvements were desired. It helped learning basic Salsa steps but it is distractive at the same time. The moment feedback appeared, was the moment where users stopped for an instant and got off the beat. However, the desired improvements were about a more detailed *Online Feedback* e.g. showing a real time plot of the recognized steps while dancing or the view of the camera to see yourself. Another problem was that feedback was not always perceived as correct or users did not understand why it appeared. This could be due to the imperfect recognition, or the mechanism how feedback is selected to display or

blocked. Another reason could be missing information that the DT does not provide to users. *Offline Feedback* was one case where the DT did not provide enough information because it was hardly understood. The plot between the recognized Salsa steps of the user and the suggested ones might be too complicated. An aggregated information that shows the ratio of stepping too late/early might be easier to understand.

Further, it has to be noticed that the measurements to ensure the non functional requirement are neither tested nor evaluated properly in this work. The design process aims to provide basic functionalities users might desire to start and practice dancing. The time that could be spent is limited in the scope of this work. That is why the requirements and the design process might not be complete, but yield a good foundation for further development.

This is shown by the conducted experiment, too. It showed that users liked using the DT and that they gained confidence in their dancing skills while learning basic Salsa steps. However, the experiment does not show anything about the long term usage of the DT.

The main problems are about the imperfect recognition of basic dancing steps, the improvement of feedback and the limited state of the current DT. Users can only practice forth and back, and side Salsa steps and dance to songs in the BAML. Enhancing the DT with automated Beat Tracking and the addition of other dance styles is crucial for the development of the system.

10 Outlook

Although the DT left a good first impression on users, there are problems to be tackled. These are the decision to substitute the Kinect v2 with another device or not, the optimization of the recognition of dancing steps, a better evaluation, the redesign of feedback, the improvement of the BAML, the enhancement of the DT with Beat Tracking and the support of other dance styles.

The Kinect v2 is no longer sold and the decision to continue using it or to change to another similar device is inevitable. However, using the Kinect v2 for this work was not useless because it showed the potential of technology to learn dancing. Plus, it is not the end because there are similar devices or technology that could replace the Kinect v2 like VicoVR, Orbbec or Open-Pose.¹ There are reports for a new Kinect, too.^{2,3} The new Kinect would have the capability to capture the environment in 3D. This would be great for the DT because pair dances like Salsa could be possible. However, Microsoft plans to use the Kinect more as an Internet of Things (IoT) device. It is probable that the new Kinect is a lighter version of the Kinect v2 and that it could not be used to the same extent. Besides using the Kinect v2 or similar cameras, you can think of other devices to enhance the DT. Dance mats as in video games can be used, too. The problem relying only on dance mats is that feedback for upper body motion, looking straight and smiling can not be captured. Another disadvantage would be that the purchaseability of the DT decreases. Adding more devices means that the user has to purchase them all. This comes to a cost in comfortability and counts for all devices you want to add. Probably, a dancing mat can be built digitally and can be defined in the coordinate system of the camera.

¹ <https://pterneas.com/2017/10/25/kinect-dead/>

² <https://www.polygon.com/2018/5/7/17327982/project-kinect-azure-ai-sensor-microsoft>

³ <https://www.golem.de/news/azure-kinect-dk-xbox-kinect-ist-jetzt-ein-iot-sensor-1902-139600.html>

This could be also used to prevent users from unconsciously stepping forward while dancing and resolve the ambiguity if the user stepped forth or not. A calibration and giving the user feedback where to stand exactly could be needed. Other devices can be e.g. trackers put on the feet or a smartphone. Not every device has to support the recognition of dancing steps. Too many devices could be perceived as invasive and distractive, too. The smartphone could be used as an audio source that plays music recognized by the Beat Tracking module of the DT. Controlling the DT with a remote control instead of voice commands would be a possibility, too. Thus, future tasks could be about the inclusion of other devices and the finding of the minimum number of devices needed.

VGB is a quick way of recording and tagging clips to build classifiers. The problem is the lack of control of the training features. E.g. if you do not want acceleration affecting the outcome of the classifier, you have to add fast and slow training clips such that the feature gets ignored in the training process to some extent. This is a tedious process and it can cause the confidence value to drop down to gain robustness as observed for step 6 for forth and back Salsa steps. Despite that, the evaluation shows that it is possible to generate gestures to detect (Salsa) dancing steps even if not all gestures are robust. The gesture for step 5 triggers even if users step with their right foot back. Due to the exclusion from another more robust gesture this error did not affect the recognition of Salsa steps. However, users were concerned about the correctness of the recognition. Therefore, future work could aim towards the optimization of the thresholds. Other devices could support the recognition, but beware that purchaseability and comfortability of the DT can be affected negatively.

Moreover the evaluation of the feedback seems poor. The execution of gestures for the evaluation set was made by one person. Plus, the perfect scores for the *Smile* and *Looking Straight* feedback seem to optimistic because users did not always agree on the *Online Feedback*. The evaluation could be enhanced if more people contribute to the evaluation sets and if the evaluation sets would have more instances. Additionally, a long term study was not conducted but would capture the learning progress better.

Finding the right feedback that helps users such that they can improve is not trivial [Mit+18]. Especially when dancing with the DT, real time feedback can be very disruptive. The user tries to follow the beat and gets out of it as soon as the feedback is displayed. He has to process

new information of the feedback that causes the moment when the user forgets about the beat. Unexpectedly, users asked for more detailed information to be displayed in real time. Therefore, it would be necessary to determine the smallest set of real time feedback that has the best impact on a life performance of users. The same has to be done for feedback that is shown after a performance. Feedback should be negative to let users know about wrong behavior and positive to praise users in a constructive form. Be aware that feedback being negative or positive depends on the context e.g. speaking soft during a presentation generally would be bad. But if it is done intentionally to emphasize on something then it does not have to be bad. Regardless, positive feedback is not implemented by this work and was missed by users. Therefore, it should be added in a future work. The modalities visual, aural, gestural, linguistic and spacial has to be regarded in the process, too. E.g. more aural feedback was wished or another coloring of the UI was remarked by users. People who have a dyschromatopsia and have difficulties distinguishing between red and green could have problems while dancing. Linguistic and spatial feedback are not supported at all and their implementation can be topics for future works. An example of spacial feedback can be the usage of a digital dancing mat to tell users if they unconsciously move away from their starting position while dancing. *Online Feedback* is not the only feedback that can be improved in future works. There is *Offline Feedback*, too. Users had problems understanding it, especially the plot between the recognized Salsa steps and the suggested ones. Therefore, the information shall be transmitted differently such that users understand it intuitively. Elaborating feedback can result into the inclusion of a user related learning curve, user accounts or a ranking to playfully compete with each other in a social way.

Beat Tracking is a major component worth to be tackled in the future. It allows users to play their songs with e.g. a smartphone and dance to them because their beat is tracked automatically. If Beat Tracking would be supported the BAML could be useless. But that is not the case. The BAML was important to launch a first version of the DT and it will be for similar projects. Furthermore, it can be used for evaluation of Beat Tracking approaches, too. The standard ballroom set is such an evaluation set.⁴ Note that the standard ballroom set does provide the BPM for the songs, but the length of the songs is only around 20 seconds. Plus, the downbeat is not considered, too. However, the current BAML can not be left as it is. The BPM

⁴ <http://anasynth.ircam.fr/home/media/ExtendedBallroom>

is not enough to assume the correct Salsa beat because it does not provide any counting of the (down-) beats. A better BAML provides an exact annotation for the beat. A JSON annotation like JAMS can provide an exactly annotated beat and other meta data.⁵ Human experts are required to annotate each song manually. However, it allows a proper beat annotation with correct (down-) beats and counting for Salsa and other dance styles.

Adding other dance styles to the DT is valued as crucial for the development of the system. Non functional requirements has to be tested explicitly, too. Note that there are dance styles that are easier to add than others. E.g. break dancing is very fast and occludes many parts of the dancer's body while spinning or balancing in a position. This can cause the recognition to be harder or not feasible.

Summing up, the DT is a good starting application but it needs polishing. The feedback and the recognition has to be improved. Thresholds have to be optimized or other technologies can be added. However, two of the most important assets are the implementation of Beat Tracking and the addition of other dance styles. Adding the Beat Tracking system presented by [BKW14] is assumed to be a major step towards a more complete DT as a generic solution to learn different dance styles.

⁵ <https://jams.readthedocs.io/en/stable/index.html>

11 Conclusion

The DT is presented as a generic MMLA solution to learn dancing with technology. There were approaches for similar tutoring systems in the domain of dancing but all of them aimed for one specific dance style like Thai dance [MST17], Bharatanatyam (traditional Indian dance) [Aic+18], Ballet [Kya+15], K-Pop [KKK17] or Salsa [Ale+11; Don+17]. The DT is designed with the generic framework of [GD12] to position it among LA applications and with the MLeAM presented by [Mit+18] to process data captured by the Kinect v2 and provide users with according feedback. A low cost sensor like the Kinect v2 was suitable to recognize basic dancing steps. Therefore, dancing steps could be defined as gestures with VGB. It can be assumed that it is possible for similar sensors, too. The design of the DT used use cases to come up with functional and non functional requirements and regarded the addition of other dance styles in its architecture. The addition of Beat Tracking is valued as a major asset to improve the current DT. However, Beat tracking was discarded due its challenges. This allowed the creation of the BAML that was introduced by this work. Although being very simple, the BAML permitted to conduct first user test on the DT. The DT was rated as a good application to start learning basic dancing steps. Dancing courses offer the most classic way to learn dancing while socially interacting with others. Online courses provide users with material to learn dancing e.g. at home. The DT allows users to learn alone at home as well. Shy people can gain confidence and always chose other ways to learn dancing later. In contrast to video games, basic dancing steps could be learned with the DT while still having fun. Thus, the DT has its positions among the stated ways people can chose to learn dancing. Let us conclude this work with some citations of users. Some of them are translated from German into English. The original answers are marked with quotes:

"If you improve the system I would definitely use it to learn dancing :-)"

I would like a music option to dance the doubled speed. It was a lot of fun and I would do it again.

"Was nice to be here ;-)"

"Very positive feeling"

It would be good to have more songs you can dance to

"This is a lot of fun. I would buy your application if you add further steps and maybe you can make it more enjoyable for the eyes."

"It would be nice if you can dance to your own music."

Good job.

"It feels like accomplishing a task rather than enjoying dancing, maybe an approach for improving."

The application is very good to get started and too many features would distract from learning.

It is an interesting application to learn dancing for the first time.

Abbreviations

BAF	Beat Activation Function	19
BAML	Beat Annotated Music Library	2
BPM	Beats Per Minute	3
CCS	Cartesian Coordinate System	8
CNN	Convolutional Neural Network	19
DBN	Dynamic Bayesian Network	19
DT	Dancing Trainer	vii
FN	False Negative	13
FP	False Positive	13
FPS	Frames Per Second	46
GDPR	General Data Protection Regulation	18
ICMI	International Conference on Multimodal Interaction	39
IoT	Internet of Things	87
ITS	Intelligent Tutoring System	8
Kinect v2	Microsoft Kinect Camera for Windows v2	2
KS	Kinect Studio	49
LA	Learning Analytics	5
MLeAM	Multimodal Learning Analytics Model	39
MMLA	Multimodal Learning Analytics	39
MOOC	Massive Open Online Course	7
MSPB	Milliseconds Per Beat	36
NaN	not a number	44
NN	Neural Network	9

RMS	Root Mean Square	61
RNN	Recurrent Neural Network	19
TN	True Negative	13
TP	True Positive	13
UI	User Interface	5
VGB	Visual Gesture Builder	2

List of Tables

4.1	Binary Confusion Matrix.	13
5.1	Use cases with priority values from 1 (lowest) to 3 (highest). Use cases with a priority greater than 1 are considered.	23
5.2	Functional requirements that are inferred from the use cases of table 5.1. The number before the dot of the ID refers to the use case and the number after it to an implementation measure.	24
6.1	Content of the current <i>baml.csv</i> . The artist is <i>www.arudeeyah.com</i> . The last three columns are filenames and not included in this table.	38
6.2	Technical information about the Kinect v2. [Fan+15][Jia+17]	46
6.3	Recorded clips with KS. Every clip has a fast and slow version for training and testing. The total number of clips is 32.	53
6.4	Gestures to recognize basic Salsa steps that use joints in the lower body and regard left and right. (*) recorded much slower and with stances to target the independence of speed (**) bobbing caused false classification	54
6.5	Conditions to recognize basic Salsa steps.	56
7.1	Evaluation of the discrete gestures with the VGB Analyzer.	61
7.2	Confusion matrices and scores of the basic forth and back Salsa steps recognition. Each step is represented by n=20 instances.	63
7.3	Confusion matrix and scores of the overall basic forth and back Salsa steps recognition. n=160	64

7.4	Confusion matrices and scores of the basic side Salsa steps recognition. Each step is represented by n=20 instances.	65
7.5	Confusion matrix and scores of the overall side Salsa steps recognition. n=160	66
7.6	Confusion matrix (left) and scores (right) of <i>Move Body</i> feedback. n=100	66
7.7	Confusion matrix (left) and scores (right) of <i>Smile</i> feedback. n=15.	66
7.8	Confusion matrix (left) and scores (right) of <i>Look Straight</i> feedback. n=32	67
7.9	Closed Questions and average answers with standard deviation. Answers range from 1 to 10 with 10 being considered the most and 1 the least. Total observations of n=25 participants with 52% female and 48% male participants.	71
7.10	T-Test to see if there is a statistical difference in confidence of dancing skills before and after the usage of the DT.	75
7.11	T-Test to see if there is a statistical difference in motivation of using the DT before and after.	76

List of Figures

4.1	2D (left) and 3D (right) CCSs.	12
5.1	Salsa stepping order of the basic forth and back steps. The numbers indicate the beat/step with 8: neutral, 1: left forward, 2: right tap, 3: left back to neutral, 4: neutral, 5: right back, 6: left tap, 7: right back to neutral	21
5.2	Salsa stepping order of the basic side steps. The numbers indicate the beat/step with 8: neutral, 1: left side, 2: right tap, 3: left back to neutral, 4: neutral, 5: right side, 6: left tap, 7: right back to neutral	21
5.3	Concept of the <i>MainWindow</i> with the explanation of the UI elements in subsection 5.4.1.	28
5.4	Concept of the <i>SalsaWindow</i> with the explanation of the UI elements in subsection 5.4.2.	30
5.5	Concept of the <i>TutorialWindow</i> with the explanation of the UI elements in subsection 5.4.3.	32
5.6	Extract of the class diagram of the current developed DT. beatMan = SalsaBeatManager object, kinWin = KinectWindow object, mainWin = MainWindow object and salWin = SalsaWindow object. Arrows has to be read like: <i>ArrowStart</i> holds object named X of <i>ArrowEnd</i>	34
6.1	Examples of the dashboard during normal dancing sessions with <i>look Straight</i> feedback for step 2 (left) and <i>Reset Dancing</i> for step 5 (right).	36
6.2	Setting up the DT. Left: top view. Right: profile view.	37
6.3	Feedback icons and instructions.	42

6.4	An example of the <i>Offline Feedback</i> . Top: Summary of <i>Online Feedback</i> that distinguishes between good (green), recognized (light gray) and displayed (dark gray). Down: Plot of ms and Salsa steps between the suggested (orange, 110 BPM) and the recognized steps (green, Test).	43
6.5	The Kinect v2 with 1: Color Camera, 2: Microphone Array, 3: Infrared Camera, 4: Infrared Projector	47
6.6	Image of the human body and the joints of the skeleton that are captured by the Kinect v2.	48
6.7	Graphical visualization of linear separability. Left: linearly separable. Right: Not linearly separable.	50
6.8	Interaction of the gestures for step 5 of the forth and back Salsa steps. From left to right: user, <i>FootTapping_Left</i> , <i>FootTapping_Right</i> and <i>ForthAndBackProgress_Left</i>	55
7.1	Procedure of the experiment for the DT.	68
7.2	Histogram of the categorized answers of the question " <i>Where does your dancing experience come from?</i> " in the pre-questionnaire.	72
7.3	Histogram of the categorized answers of the question " <i>What do you feel like you learned from using this application?</i> " in the post questionnaire.	73
7.4	Histogram of the categorized answers of the question " <i>List the most three positive aspects.</i> " in the post questionnaire.	74
7.5	Histogram of the categorized answers of the question " <i>List the most three negative aspects.</i> " in the post questionnaire.	74

Listings

6.1 JSON example for *Offline Feedback*. 45

Bibliography

- [ABR85] John R. Anderson, C. Franklin Boyle, and Brian J. Reiser. “Intelligent Tutoring Systems”. In: *Science* 228.4698 (1985), pp. 456–462. DOI: 10.1126/science.228.4698.456. eprint: <https://science.sciencemag.org/content/228/4698/456.full.pdf>. URL: <https://science.sciencemag.org/content/228/4698/456>.
- [Agu+16] Jose Aguilar et al. “A General Framework for Learning Analytic in a Smart Classroom”. In: *Technologies and Innovation*. Ed. by Rafael Valencia-García et al. Cham: Springer International Publishing, 2016, pp. 214–225. ISBN: 978-3-319-48024-4.
- [Aic+18] Achyuta Aich et al. “NrityaGuru: A Dance Tutoring System for Bharatanatyam Using Kinect”. In: *Computer Vision, Pattern Recognition, Image Processing, and Graphics*. Ed. by Renu Rameshan, Chetan Arora, and Sumantra Dutta Roy. Singapore: Springer Singapore, 2018, pp. 481–493. ISBN: 978-981-13-0020-2.
- [Ale+11] Dimitrios S. Alexiadis et al. “Evaluating a Dancer’s Performance Using Kinect-based Skeleton Tracking”. In: *Proceedings of the 19th ACM International Conference on Multimedia*. MM ’11. Scottsdale, Arizona, USA: ACM, 2011, pp. 659–662. ISBN: 978-1-4503-0616-4. DOI: 10.1145/2072298.2072412. URL: <http://doi.acm.org/10.1145/2072298.2072412>.
- [Arg13] Michael Argyle. *Bodily Communication*. Routledge, Apr. 2013. DOI: 10.4324/9780203753835. URL: <https://doi.org/10.4324/9780203753835>.
- [BKW14] Sebastian Böck, Florian Krebs, and Gerhard Widmer. “A Multi-model Approach to Beat Tracking Considering Heterogeneous Music Styles”. In: *ISMIR*. 2014.
- [CK00] Bill Cope and Mary Kalantzis. *Multiliteracies : literacy learning and the design of social futures*. English (US). Routledge, 2000. ISBN: 9780415214209.

- [CL14] Maxine Campion and Liat Levita. “Enhancing positive affect and divergent thinking abilities: Play some music and dance”. In: *The Journal of Positive Psychology* 9.2 (2014), pp. 137–145. DOI: 10.1080/17439760.2013.848376. eprint: <https://doi.org/10.1080/17439760.2013.848376>. URL: <https://doi.org/10.1080/17439760.2013.848376>.
- [Don+17] Y. Dong et al. “SalsaAsst: Beat Counting System Empowered by Mobile Devices to Assist Salsa Dancers”. In: *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. 2017, pp. 81–89. DOI: 10.1109/MASS.2017.25.
- [DWMZ96] S David, MD Weiss, and PT Mimi Ziatkowski. “Rehabilitation of dance injuries to the shoulder, lumbar spine, pelvis, and hip”. In: *Orthopaedic physical therapy clinics of North America* 5.4 (1996).
- [Fan+15] P. Fankhauser et al. “Kinect v2 for mobile robot navigation: Evaluation and modeling”. In: *2015 International Conference on Advanced Robotics (ICAR)*. 2015, pp. 388–394. DOI: 10.1109/ICAR.2015.7251485.
- [Gal+14] Brook Galna et al. “Accuracy of the Microsoft Kinect sensor for measuring movement in people with Parkinson’s disease”. In: *Gait & Posture* 39.4 (2014), pp. 1062 –1068. ISSN: 0966-6362. DOI: <https://doi.org/10.1016/j.gaitpost.2014.01.008>. URL: <http://www.sciencedirect.com/science/article/pii/S0966636214000241>.
- [Gar17] Raul Garreta. *Machine learning simplified*. 2017. URL: <https://hds.hebis.de/ubffm/Record/HEB435348264>.
- [GD12] Wolfgang Greller and Hendrik Drachsler. “Translating Learning into Numbers: A Generic Framework for Learning Analytics”. In: *Educational Technology & Society* 15 (July 2012), pp. 42–57.
- [GK17] Aggelos Gkiokas and Vassilios Katsouros. “Convolutional Neural Networks for Real-Time Beat Tracking: A Dancing Robot Application.” In: *ISMIR*. 2017, pp. 286–293.
- [GM11] P. Grosche and M. Muller. “Extracting Predominant Local Pulse Information From Music Recordings”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.6 (2011), pp. 1688–1701. ISSN: 1558-7916. DOI: 10.1109/TASL.2010.2096216.

- [GY19] Yasemin Gulbahar and Denizer Yildirim. "Towards an Adaptive Learning Analytics Framework". In: *Proceedings of Society for Information Technology & Teacher Education International Conference 2019*. Ed. by Kevin Graziano. Las Vegas, NV, United States: Association for the Advancement of Computing in Education (AACE), 2019, pp. 1025–1032. URL: <https://www.learntechlib.org/p/208370>.
- [Han95] Judith Lynne Hanna. "The Power of Dance: Health and Healing". In: *The Journal of Alternative and Complementary Medicine* 1.4 (1995). PMID: 9395627, pp. 323–331. DOI: 10.1089/acm.1995.1.323. eprint: <https://doi.org/10.1089/acm.1995.1.323>. URL: <https://doi.org/10.1089/acm.1995.1.323>.
- [Har12] Linda Harasim. "Learning Theory and Online Technologies". In: *Learning Theory and Online Technologies* (Jan. 2012), pp. 1–192. DOI: 10.4324/9780203846933.
- [Hen08] Maximilian Hendler. "Was ist Salsa?" ger. In: *Beiträge zur Populärmusikforschung*. Justus-Liebig-Universität, 2008. URL: <http://geb.uni-giessen.de/geb/volltexte/2010/7568>.
- [HT07] John Hattie and Helen Timperley. "The Power of Feedback". In: *Review of Educational Research* 77.1 (2007), pp. 81–112. DOI: 10.3102/003465430298487. eprint: <https://doi.org/10.3102/003465430298487>. URL: <https://doi.org/10.3102/003465430298487>.
- [Jia+17] Jichao Jiao et al. "A Post-Rectification Approach of Depth Images of Kinect v2 for 3D Reconstruction of Indoor Scenes". In: *ISPRS International Journal of Geo-Information* 6.11 (2017). ISSN: 2220-9964. DOI: 10.3390/ijgi6110349. URL: <http://www.mdpi.com/2220-9964/6/11/349>.
- [Kit+14] A. Kitsikidis et al. "Dance analysis using multiple Kinect sensors". In: *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*. Vol. 2. 2014, pp. 789–795.
- [KK07] Sungwoon Kim and Jingu Kim. "Mood after Various Brief Exercise and Sport Modes: Aerobics, Hip-Hop Dancing, ICE Skating, and Body Conditioning". In: *Perceptual and Motor Skills* 104.3_suppl (June 2007), pp. 1265–1270. DOI: 10.2466/pms.104.4.1265-1270. URL: <https://doi.org/10.2466/pms.104.4.1265-1270>.

- [KKK17] Dohyung Kim, Dong-Hyeon Kim, and Keun-Chang Kwak. "Classification of K-Pop Dance Movements Based on Skeleton Information Obtained by a Kinect Sensor". In: *Sensors* 17.6 (2017). ISSN: 1424-8220. DOI: 10.3390/s17061261. URL: <http://www.mdpi.com/1424-8220/17/6/1261>.
- [Kos12] Robert J. Kosinski. "A Literature Review on Reaction Time Kinds of Reaction Time Experiments". In: 2012.
- [Kou+17] Maria Kourakli et al. "Towards the improvement of the cognitive, motoric and academic skills of students with special educational needs using Kinect learning games". In: *International Journal of Child-Computer Interaction* 11 (2017). Designing with and for Children with Special Needs, pp. 28–39. ISSN: 2212-8689. DOI: <https://doi.org/10.1016/j.ijcci.2016.10.009>. URL: <http://www.sciencedirect.com/science/article/pii/S2212868916300186>.
- [Kre03] Gunther Kress. *Literacy in the New Media Age*. Routledge, Sept. 2003. DOI: 10.4324/9780203299234. URL: <https://doi.org/10.4324/9780203299234>.
- [Kre+16] Florian Krebs et al. "Downbeat Tracking Using Beat-synchronous Features and Recurrent Neural Networks". EN. In: *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*. Aug. 2016.
- [Kya+15] Matthew Kyan et al. "An Approach to Ballet Dance Training Through MS Kinect and Visualization in a CAVE Virtual Reality Environment". In: *ACM Trans. Intell. Syst. Technol.* 6.2 (Mar. 2015), 23:1–23:37. ISSN: 2157-6904. DOI: 10.1145/2735951. URL: <http://doi.acm.org/10.1145/2735951>.
- [Luc14] Dorothy Lucardie. "The Impact of Fun and Enjoyment on Adult's Learning". In: *Procedia - Social and Behavioral Sciences* 142 (2014). The Fourth International Conference on Adult Education, Romania 2014, pp. 439–446. ISSN: 1877-0428. DOI: <https://doi.org/10.1016/j.sbspro.2014.07.696>. URL: <http://www.sciencedirect.com/science/article/pii/S1877042814046242>.
- [Mit+18] Daniele Di Mitri et al. "From signals to knowledge: A conceptual model for multimodal learning analytics". In: *Journal of Computer Assisted Learning* 34.4 (Aug. 2018), pp. 338–349. ISSN: 0266-4909. DOI: 10.1111/jcal.12288. URL: <https://doi.org/10.1111/jcal.12288>.

- [MST17] Ob-orm Muangmoon, Pradorn Sureephong, and Karim Tabia. "Dance Training Tool Using Kinect-Based Skeleton Tracking and Evaluating Dancer's Performance". In: *Advances in Artificial Intelligence: From Theory to Practice*. Ed. by Salem Benferhat, Karim Tabia, and Moonis Ali. Cham: Springer International Publishing, 2017, pp. 27–32. ISBN: 978-3-319-60045-1.
- [Ohk+15] M. Ohkita et al. "Audio-visual beat tracking based on a state-space model for a music robot dancing with humans". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 5555–5560. DOI: 10.1109/IR0S.2015.7354164.
- [Pai90] Allan Paivio. *Mental Representations*. Oxford University Press, Sept. 1990. DOI: 10.1093/acprof:oso/9780195066661.001.0001. URL: <https://doi.org/10.1093/acprof:oso/9780195066661.001.0001>.
- [Pro+17] Eftychios Protopapadakis et al. "FOLK DANCE PATTERN RECOGNITION OVER DEPTH IMAGES ACQUIRED VIA KINECT SENSOR". In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2/W3* (Feb. 2017), pp. 587–593. DOI: 10.5194/isprs-archives-XLII-2-W3-587-2017.
- [Ren04] Priscilla Renta. "Salsa Dance: Latino/ a History in Motion". In: vol. 16. Sept. 2004.
- [Ric67] C. E. Rice. "Human Echo Perception". In: *Science* 155.3763 (Feb. 1967), pp. 656–664. DOI: 10.1126/science.155.3763.656. URL: <https://doi.org/10.1126/science.155.3763.656>.
- [Sah+13] S. Saha et al. "Gesture Recognition from Indian Classical Dance Using Kinect Sensor". In: *2013 Fifth International Conference on Computational Intelligence, Communication Systems and Networks*. 2013, pp. 3–8. DOI: 10.1109/CICSYN.2013.11.
- [San+18] Augusto Santos et al. "You Are Off The Beat!: Is Accelerometer Data Enough for Measuring Dance Rhythm?" In: June 2018, pp. 1–8. DOI: 10.1145/3212721.3212724.
- [Sch+15] Jan Schneider et al. "Stand Tall and Raise Your Voice! A Study on the Presentation Trainer". In: *Design for Teaching and Learning in a Networked World*. Ed. by Gráinne Conole et al. Cham: Springer International Publishing, 2015, pp. 311–324. ISBN: 978-3-319-24258-3.

- [Sch+16] J. Schneider et al. "Can You Help Me with My Pitch? Studying a Tool for Real-Time Automated Feedback". In: *IEEE Transactions on Learning Technologies* 9.4 (2016), pp. 318–327. ISSN: 1939-1382.
- [SDT17] Bryant Seamon, Meredith DeFranco, and Mary Thigpen. "Use of the Xbox Kinect virtual gaming system to improve gait, postural control and cognitive awareness in an individual with Progressive Supranuclear Palsy". In: *Disability and Rehabilitation* 39.7 (2017). PMID: 27008348, pp. 721–726. DOI: 10.3109/09638288.2016.1160444. eprint: <https://doi.org/10.3109/09638288.2016.1160444>. URL: <https://doi.org/10.3109/09638288.2016.1160444>.
- [SNV06] Sabine Sonnentag, Cornelia Niessen, and Judith Volmer. "Expertise in Software Design". In: *Cambridge handbook of expertise and expert performance*. Ed. by K. Anders Ericsson et al. Cambridge: Cambridge University Press, 2006, pp. 373–387.
- [SS15] E. E. Stone and M. Skubic. "Fall Detection in Homes of Older Adults Using the Microsoft Kinect". In: *IEEE Journal of Biomedical and Health Informatics* 19.1 (2015), pp. 290–301. ISSN: 2168-2194. DOI: 10.1109/JBHI.2014.2312180.
- [Sun+17] Bingli Sun et al. "Self-esteem recognition based on gait pattern using Kinect". In: *Gait & Posture* 58 (2017), pp. 428–432. ISSN: 0966-6362. DOI: <https://doi.org/10.1016/j.gaitpost.2017.09.001>. URL: <http://www.sciencedirect.com/science/article/pii/S0966636217308974>.
- [SWM12] Stefan Scherer, Marcelo Worsley, and Louis-Philippe Morency. "1st International Workshop on Multimodal Learning Analytics: Extended Abstract". In: *Proceedings of the 14th ACM International Conference on Multimodal Interaction*. ICMI '12. Santa Monica, California, USA: ACM, 2012, pp. 609–610. ISBN: 978-1-4503-1467-1. DOI: 10.1145/2388676.2388803. URL: <http://doi.acm.org/10.1145/2388676.2388803>.
- [WHH16] Deborah West, David Heath, and Henk Huijser. "Let's Talk Learning Analytics: A Framework for Implementation in Relation to Student Retention". In: *Online Learning (Online Learning Consortium)* 20 (June 2016). DOI: 10.24059/olj.v20i2.792.
- [Wie13] Karl Eugene Wiegers. *Software requirements*. 2013. URL: <http://proquest.tech.safaribooksonline.de/9780735679658>.

- [Wor12] Marcelo Worsley. "Multimodal Learning Analytics: Enabling the Future of Learning Through Multimodal Data Analysis and Interfaces". In: *Proceedings of the 14th ACM International Conference on Multimodal Interaction*. ICMI '12. Santa Monica, California, USA: ACM, 2012, pp. 353–356. ISBN: 978-1-4503-1467-1. DOI: 10.1145/2388676.2388755. URL: <http://doi.acm.org/10.1145/2388676.2388755>.
- [Wor+16] Marcelo Worsley et al. "Situating Multimodal Learning Analytics". In: June 2016.
- [YON17] N. Yalta, T. Ogata, and K. Nakadai. "Sequential Deep Learning for Dancing Motion Generation". In: (2017).
- [Zha+18] Mingshao Zhang et al. "Recent Developments in Game-Based Virtual Reality Educational Laboratories Using the Microsoft Kinect". In: *International Journal of Emerging Technologies in Learning (iJET)* 13.01 (2018), pp. 138–159. ISSN: 1863-0383. URL: <http://online-journals.org/index.php/i-jet/article/view/7773>.
- [Zhu05] Hong Zhu. *Software design methodology : From principles to architectural styles*. Elsevier Science & Technology, 2005, pp. 33–39, 55–66. URL: <https://ebookcentral.proquest.com>.

