Diese Arbeit wurde vorgelegt am
Lehrstuhl für Mathematik (MathCCES)

Didaktisch-methodische Ausarbeitung eines Lernmoduls zum Thema Shazam im Rahmen eines mathematischen Modellierungstages für Schülerinnen und Schüler der Sekundarstufe II

Bachelorarbeit
Mathematik

September 2016

Vorgelegt von Presented by	Nils Steffen [redacted]
Erstprüfer First examiner	Prof. Dr. Martin Frank Lehrstuhl für Mathematik (MathCCES) RWTH Aachen University
Zweitprüfer Second examiner	Prof. Dr. Johanna Heitzer Lehrstuhl A für Mathematik RWTH Aachen University
Koreferent Co-supervisor	Dr. Christina Roeckerath Lehrstuhl für Mathematik (MathCCES) RWTH Aachen University
Koreferent Co-supervisor	Jonas Kusch Lehrstuhl für Mathematik (MathCCES) RWTH Aachen University

Inhaltsverzeichnis

1. Einleitung

Schon seit Jahrzehnten spielen in der didaktischen Diskussion die Einbindung und Realisierung von Realbezügen sowie Modellierung im Mathematikunterricht eine bedeutende Rolle. Mit dem Einsatz von Realbezügen im Mathematikunterricht sollen die Schülerinnen und Schüler (SuS) erfahren, dass Mathematik Anwendung in ihrem Leben findet und nicht nur ein starres Konstrukt aus Formeln und Symbolen darstellt. Sie sollen entdecken, dass Mathematik alltäglich ist und in fast allen Lebens- und Berufssparten genutzt wird. Weiter soll den Lernenden mit Hilfe von Modellierungsaufgaben gezeigt werden, dass sie die in der Schule gelernten Kompetenzen und ihr erworbenes mathematisches Wissen auf realitätsnahe Fragestellungen und Problemen anwenden können, um diese zu lösen. Auch haben Studien gezeigt, dass sich realitätsbezogene Aufgaben positiv auf das Mathematikbild der SuS auswirken und die Motivation sowie das Interesse zum Mathematiklernen wecken [6, S. 1]. Denn die SuS erkennen, dass die Mathematik nicht nur zur abstrakten Lösung mathematischer Fragestellungen sondern auch als Werkzeug in realen Problemsituationen gewinnbringend und hilfreich angewendet werden kann. Diese Erkenntnis und das Trainieren des Modellierens als Kompetenz bereitet die SuS gleichzeitig auch auf die berufliche Zukunft vor, denn Modellierungen von Problemen sind elementarer Bestandteil diverser beruflicher Perspektiven [2, S. 357-358].
So hat auch das deutsche Bildungssystem in den letzten Jahren Anwendungsbezüge und mathematisches Modellieren im Mathematikunterricht als wichtiges und gewinnbringendes Werkzeug für die SuS erkannt, das zur Beschreibung realistischer Probleme heran gezogen werden kann und gleichzeitig einen nahen Alltagsbezug aufweist. Mittlerweile hat die Kultusministerkonferenz in ihren Bildungsstandarts Modellieren als eine zentrale Kompetenz verankert [4]. Auch im Kernlehrplan Nordrhein-Westfalen (Sek I / Sek II) wurde Modellieren als eine zentrale, prozessbezogene Kompetenz aufgenommen, sodass die SuS lernen sollen Mathematik als Werkzeug zu nutzen, um realistische Phänomene aus ihrer Umwelt zu beschreiben bzw. zu modellieren [8].
Jedoch stellt die Einbindung von Modellieren und problemorientierten Aufgaben in den Schulalltag eine Herausforderung dar. Denn das Einbinden außermathematischer Kontexte gestaltet den Mathematikunterricht komplexer und weniger vorhersehbar. Das heißt, einerseits bedeutet das Einbinden von Modellierungsaufgaben in den Mathematikunterricht einen erhöhten Vorbereitungsaufwand der betreffenden Unterrichtsstunde für die Lehrkraft, da sie sich außermathematisches Sachwissen aneignen muss. Anderseits können SuS zunächst Schwierigkeiten haben, Problemlösestrategien sowie -fähigkeiten zu entwickeln [1, S. 5]. Daher ist es wichtig, den SuS die Möglichkeit und Zeit zu geben, sich intensiv mit dem zu lösenden Problem zu beschäftigen. Zeit die der straffe Lehrplan jedoch nicht vorsieht bzw. einplant [2, S. 359]. Zudem gibt es unter den Lehrkräften noch eine gewissen Unsicherheit, wie sie die wichtige Kompetenz des Modellierens in ihren Unterricht einführen bzw. behandeln sollen. Es ist ihnen z.B. unklar, wie sie als Lehrperson intervenieren oder den Modellierungsprozess adäquat benoten können [10, S. 2]. Aus diesen Gründen findet in der Unterrichtsrealität trotz der positiven Bilanz in der didaktischen Diskussion nur selten realitätsnahe Modellie-

rung komplexer Probleme statt.

Das Schülerlabor CAMMP der RWTH Aachen hat sich zur Aufgabe gesetzt im Rahmen von CAMMP days oder CAMMP weeks, also einzelnen Modellierungstagen oder einer Modellierungswoche, die Entwicklung und Förderung der Modellierungskompetenz von SuS zu unterstützen. Dazu werden die SuS in unterschiedlichen Lernmodulen mit alltagsnahen Problemstellungen sowie realitätsnahen Fragestellungen konfrontiert, die sie mit geeigneten mathematischen Methoden und unter Verwendung von Computersimulationen lösen sollen. So lernen die SuS durch aktive Auseinandersetzung mit lebensnahen Problemen die Grundlagen der mathematischen Modellierung. Auch interessierte Lehrkräfte erhalten durch die vielfältigen Veranstaltungen von CAMMP die Möglichkeit, konkrete Materialien zu den Lernmodulen von CAMMP zu erhalten und sich somit Anregungen für die Einbindung außermathematischer Kontexte in ihren eigenen Unterricht zu holen.

Diese Arbeit beschäftigt sich mit der Gestaltung eines neuen Lernmoduls zum Thema Shazam, einer App, die Musikstücke in Sekundenbruchteilen erkennt. Ein Klick auf die Smartphone-App genügt und man weiß alles über den bis dato noch unbekannten Song. Shazam ist Marktführer im Bereich der Musikerkennung und unter den Jugendlichen weit verbreitet. Die SuS sollen in dem entwickelten Lernmodul die Idee hinter Shazam kennen lernen und den Prozess von der mathematischen Beschreibung eines Audiosignal bis hin zur Erstellung des akustischen Fingerabdrucks einer kurzen Aufnahme eines Musikstückes und der anschließenden Durchsuchung der Datenbank von Shazam nach dem passenden Song schrittweise durchlaufen.

Im ersten Kapitel dieser Arbeit werden zunächst die fachdidaktischen Hintergründe sowie Ziele des Modellierens im Mathematikunterricht erläutert. Weiter wird im diesen Kapitel auf die Philosophie von CAMMP eingegangen, die unter anderem didaktische Anhaltspunkte zum Aufbau des entwickelten Lernmoduls gibt. Nachfolgend werden die fachlichen und mathematischen Grundlagen von Shazam dargestellt.
Anschließend wird im dritten Kapitel das didaktisch-methodische Konzept näher erläutert, welches die Vorstellung der Materialien sowie die Erläuterung der didaktischen Vorgehensweise beinhaltet. Zum Schluss wird die Durchführung eines Modellierungstages im Rahmen einer Schüleruni der RWTH Aachen mit SuS aus der Oberstufe vorgestellt und evaluiert.

2. Didaktischer Hintergrund

In diesem Kapitel wird zunächst die Idee hinter CAMMP vorgestellt. Anschließend wird auf den didaktischen Hintergrund des Modellierens eingegangen, eins der zentralen Elemente des erstellten Lernmoduls sowie aller bestehenden CAMMP Projekte.

2.1. Das CAMMP Projekt

2.1.1. Zentrale Aspekte von CAMMP

Die Abkürzung CAMMP steht für „Computational and Mathematical Modeling Program“ und ist ein mathematisches Schülerlabor an der RWTH Aachen, welches vom Lehrstuhl Mathematik CCES und der Graduiertenschule AICES organisiert wird. CAMMP bietet mit seinen unterschiedlichen Lernmodulen SuS die Möglichkeit, sich mit realen, herausfordernden Problemen auseinanderzusetzen und diese mit geeigneten mathematischen Methoden und Computersimulationen zu lösen. Dabei wird an den aktuellen Lehrplänen angeknüpft und neue, für das Modul notwendige Mathematik vermittelt. CAMMP bietet dazu zwei Veranstaltungsformate an: Der CAMMP day und die CAMMP week.

Bei der CAMMP week arbeiten die SuS eine Woche lang in kleinen Gruppen und unter Betreuung eines Wissenschaftlers an der Lösung einer individuellen Problemstellung aus der aktuellen Forschung. Die Aufgabenstellungen werden dabei von Partnerfirmen und Universitätsinstituten gestellt. Am Ende der fünf Tage präsentieren die einzelnen Gruppen unter anderem auch den Auftraggebern des Problems im Rahmen einer Abschlussveranstaltung ihre Ergebnisse.

Im Rahmen des CAMMP days beschäftigen sich abhängig vom Lernmodul SuS aus der Mittel- und Oberstufe einen Tag lang an der RWTH Aachen mit einer lebensnahen Fragestellung und werden in die Grundlagen der mathematischen Modellierung eingeführt. In kleinen Gruppen arbeiten die SuS an der Lösung des realitätsnahen Problems, welches für die SuS didaktisch-methodisch ausgearbeitet wurde. Dazu werden sie von Wissenschaftlern unterstützt. Die SuS benutzen die Software *Matlab* als Unterstützung beim Lösen der mathematischen Problem. Das Angebot von CAMMP days umfasst derzeit die folgenden Lernmodule:

- Wie funktioniert eigentlich GPS und was hat das mit Mathe zu tun?
- Wie funktioniert eigentlich Google und was hat das mit Mathe zu tun?
- Spiegelaufstellung in einem Solarkraftwerk
- Von Mickey Mouse bis Buzz Light Year - wie Mathematik die Filmfiguren zum Leben erweckt
- Vom Lotfällen bis zum JPEG-Format

- Der Shazam-Algorithmus - Wie Mathematik Musik identifiziert

2.1.2. Ziele von CAMMP

Nachfolgend werden die verfolgten Ziele von CAMMP sowohl für die SuS als auch für die Lehrkräfte beschrieben. So soll CAMMP bei den SuS den Umgang mit mathematischer Modellierung fördern und die Kompetenzen in diesem Bereich ausbauen. Dazu werden abseits vom üblichen Unterrichtsgeschehen computergestützte, neue, forschungsgeleitete, didaktische Ideen ausprobiert und evaluiert. Weiter hat CAMMP das Ziel, den SuS eine Berufs- und Studienorientierung anzubieten sowie für ein Studium im Bereich CES oder Mathematik zu begeistern und zu ermutigen. Außerdem sollen die SuS Mathematik als Anwendung erfahren und an Hand der verschiedenen Lernmodule die gesellschaftliche Bedeutung sowie den zentralen Stellenwert der Mathematik in der Welt erkennen.
Ebenfalls können Mathematiklehrer/-innen von CAMMP profitieren. Einerseits werden die SuS durch das Lösen von lebensnahen Situationen motiviert und ihr Interesse am Mathematiklernen geweckt. Anderseits können die Lehrkräfte sich Anregungen holen, wie sie komplexe und anwendungsbezogene Probleme in ihren Unterricht integrieren und SuS zu selbstständigem Lernen motivieren können.

2.2. Mathematisches Modellieren

Modellieren und anwendungsorientierte Aufgaben im Mathematikunterricht sowie die Förderung von mathematischen Modellierungskompetenzen finden immer mehr Relevanz und Akzeptanz im deutschen Bildungssystem. Mittlerweile gehört die Kompetenz des Modellierens zu einer der sechs allgemeinen mathematischen Kompetenzen, die die SuS laut der Kultusministerkonferenz in ihrer Schullaufbahn durch aktive Auseinandersetzung mit mathematischen Inhalten erwerben sollen [4]. Auch in den Kernlehrplänen des Landes Nordreihn-Westfalen für die Sekundarstufe I und II findet man Modellieren unter den vier prozessbezogenen Kompetenzen wieder. So sollen die SuS Realbezüge mit mathematischen Modellen beschreiben sowie mathematische Modelle realen Problemen zuordnen können und somit den Wechsel zwischen Realität und Mathematik beherrschen [8, 7]. In den nächsten Kapiteln soll zunächst der Begriff Modellieren sowie Modell genauer definiert und anschließend der Modellierungsprozess sowie die Ziele und Vorteile von Modellieren und Modellierungsaufgaben im Unterricht genauer diskutiert werden.

2.2.1. Begriffsbildung: Modell und Modellierung

Ein mathematisches Modell ist eine vereinfachende Darstellung der Realität, auf das mathematische Methoden angewandt werden können, um mathematische Resultate zu erhalten [10, S. 12]. Das heißt, ein Modell stellt einen Bezug zwischen Realität und Mathematik her, und beschreibt ein reale Probleme mathematisch, um diese zu lösen. Vereinfachend bedeutet aber gleichzeitig ebenso, dass eine komplexe Realität nicht in

seiner Gänze anhand mathematischer Modellierungen beschrieben werden kann. Dies basiert auf der Tatsache, dass die Mathematik die Realität auf Grund ihrer Komplexität nur innerhalb bestimmter Grenzen abbilden kann. Jedoch ist die komplette Erfassung der Wirklichkeit in einem Modell auch gar nicht erwünscht, da Modelle das Ziel haben, reale Daten übersichtlich darzustellen und schnell zu verarbeiten [10, S.12-13] Denn schon Galileo Galilei erkannte: „Vereinfachung ist eine Tugend von Modellen, kein Nachteil“ [9, S. 10].
Kaiser unterscheidet je nach Verwendung zwischen zwei Arten mathematischer Modelle. Er nennt ein Modell *normativ*, wenn es Regeln vorschreiben und für bestimmte Realbezüge als Vorbild stehen sollen. Bildet ein Modell Aspekte der Realität unter bestimmten Berücksichtigungen möglichst genau ab, wird es als *depektiv* bezeichnet. Hierunter fallen Modelle, die erklären, beschreiben oder vorhersagen [10, S.13-14].

Der Begriff Modellieren bezeichnet den gesamten Problemlösungsprozess, der zum Lösen des aufgestellten Modells für ein Problem aus der Realität verwendet wird. Zur Beschreibung des Modellierungsprozesses wird der Prozess in einzelnen Phasen eingeteilt und anhand eines Modellierungskreislaufs dargestellt.

2.2.2. Der Modellierungskreislauf

Um den Bezug zwischen Realität und Mathematik auszuführen, wird auf den Modellierungskreislauf als strategisches Instrument zurückgegriffen, da er den typischen Lösungsprozess von Modellierungsaufgaben beschreibt. Der siebenschrittige Modellierungskreislauf von Blum und Leiß (vgl. Abbildung 1) hat sich dabei als besonders hilfreich erwiesen, einen realen Modellierungsprozess sowie auftretende kognitive Hürden zu beschreiben. Er geht dabei auf den wichtigen Übersetzungsprozess zwischen realer Situation und mathematisches Modell ein und beschreibt detailliert die einzelnen Schritte, die die Lernenden beim Lösen von Modellierungsaufgaben durchlaufen [6, S. 20-21].

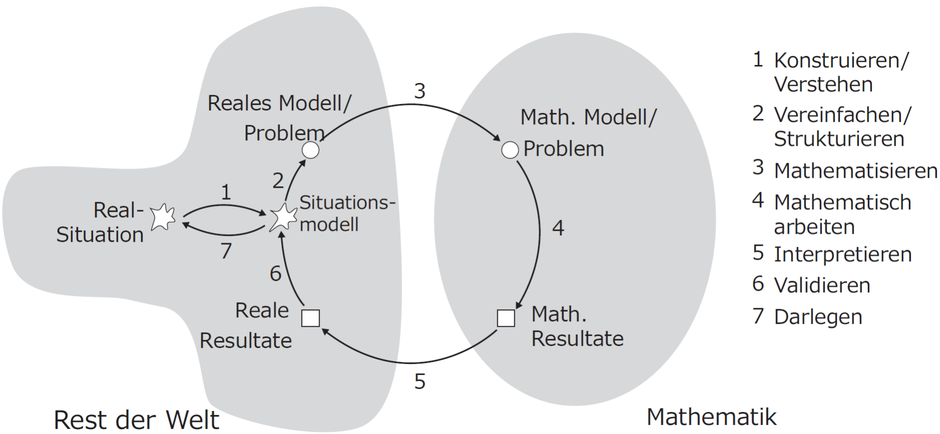


Abbildung 1: Modellierungskreislauf nach Blum und Leiß [2, S. 366]

Die einzelnen Phasen sollen nun genauer diskutiert werden:

1. **Konstruieren/ Verstehen**
   Der Modellierungskreislauf beginnt mit einer Realsituation, die die SuS im ersten Schritt zunächst verstehen müssen, um eine Vorstellung der Situation bilden zu können. Dieses mentale Modell, das sogenannte Situationsmodell, enthält meistens für jede Schülerin und jeden Schüler unterschiedliche und teilweise noch irrelevante Angaben für die Lösung des Problems [9, S. 20].

2. **Vereinfachen/ Strukturieren**
   Das nächste Ziel ist die Erstellung eines Realmodells, um die Komplexität der Realsituation zu minimieren. Für diesen Schritt müssen die wichtigsten Angaben identifiziert, strukturiert und mit Hilfe idealisierter Annahmen vereinfacht werden. Gefährlich ist dabei eine zu starke Reduktion der Annahmen, sodass die Realsituation zu sehr verfälscht und falsch abgebildet wird [9, S. 21-22].

3. **Mathematisieren**
   Hier findet nun durch Mathematisierung die Überführung in ein mathematisches Modell statt, auf das sich die Mathematik anwenden lässt. Somit stellt dieser Schritt einen Bezug zwischen Realität und Mathematik her [9, S. 22-24].

4. **Mathematisch arbeiten**
   Die Lernenden müssen nun mit dem erstellen Modell arbeiten, indem sie durch die Anwendung geeigneter, mathematischer Kenntnisse und Verfahren versuchen, ein mathematisches Resultat zu erzeugen. Neben dem Modellieren werden die Lernenden hier auch in den Kompetenzen *mathematische Probleme lösen* und *mit symbolischen, formalen und technischen Elementen der Mathematik umgehen* des Kernlehrplan NRWs [8] geschult. Je nach gegebener Realsituation kommen verschiedene Problemlösestrategien zum Einsatz, wie beispielsweise systematisches Probieren oder Analogienbildung auf bereits bekannte Probleme. Auch müssen die Lernenden situationsbedingt mit verschiedenen mathematischen Werkzeugen umgehen können. Häufig wird zum Lösen von Modellierungsaufgaben auf digitale Werkzeuge, speziell auf den Computer als Werkzeug zurückgegriffen. So auch bei CAMMP mit der Verwendung der Software *Matlab*. Auf die Vorteile digitaler Hilfsmittel für den Modellierungsprozess wird später noch explizit eingegangen [9, S. 24-25].

5. **Interpretieren**
   Die erhaltene mathematische Lösung wird nun wieder auf die Realität übertragen. Dort wird sie interpretiert, sodass die Lernenden ein reales Resultat erhalten [9, S. 25].

6. **Validieren**
   Das reale Resultat muss nun anhand des Situationsmodells überprüft werden. Dies kann beispielsweise durch Überprüfen der Größenordnung (Ist die Größenordnung plausibel?), durch Kontrolle der Stabilität des Modells (Ist das Modell

Anfällig für Veränderungen?), Kontrolle extremer Situationen (Wie ist das Verhalten des Modells an besonderen Werten?) oder durch Kontrolle der mathematischen Abgeschlossenheit und Widerspruchsfreiheit (Kommt man mit verschiedenen Modellen auf gleiche Resultate?) geschehen.
Meist führt die Validierung zu einer Verwerfung der erhaltenden Resultate und der Modellierungskreislauf (Schritt 2 bis 6) muss erneut durchlaufen werden. Dazu werden die getroffenen Annahmen auf ihre Gültigkeit überprüft. Das heißt bisherige Annahmen werden verworfen oder abgeändert sowie weitere Informationen und Bedingungen werden hinzugenommen [9, S. 25-28].

7. **Darlegen**
In diesem Schritt werden die aus dem Modellierungskreislauf gewonnenen, brauchbaren, sinnvollen Ergebnisse verständlich vermittelt bzw. kommuniziert [9, S. 28-29].

Obwohl der Modellierungskreislauf nach Blum und Leiß den SuS die Möglichkeit gibt, die einzelnen Schritte bei der Bearbeitung von Modellierungsaufgaben nachzuvollziehen und zu reflektieren, ist der siebenschrittige Kreislauf im Allgemeinen zu komplex. Studien haben dahingehend gezeigt, dass die SuS Schwierigkeiten haben, zwischen den sieben Schritten des Modelliierungskreislaufs zu unterscheiden [13]. Aus diesem Grund ist es ratsam, einen vereinfachten Kreislauf zu verwenden, der eine bessere Orientierungshilfe bei der Bearbeitung von Modellierungsaufgaben darstellt.
Ein Beispiel für ein didaktisch reduziertes Schema gibt der in CAMMP genutzte Modellierungskreislauf nach Ortlieb (siehe Abbildung 2), der sich an dem von Blum und Leiß orientiert.

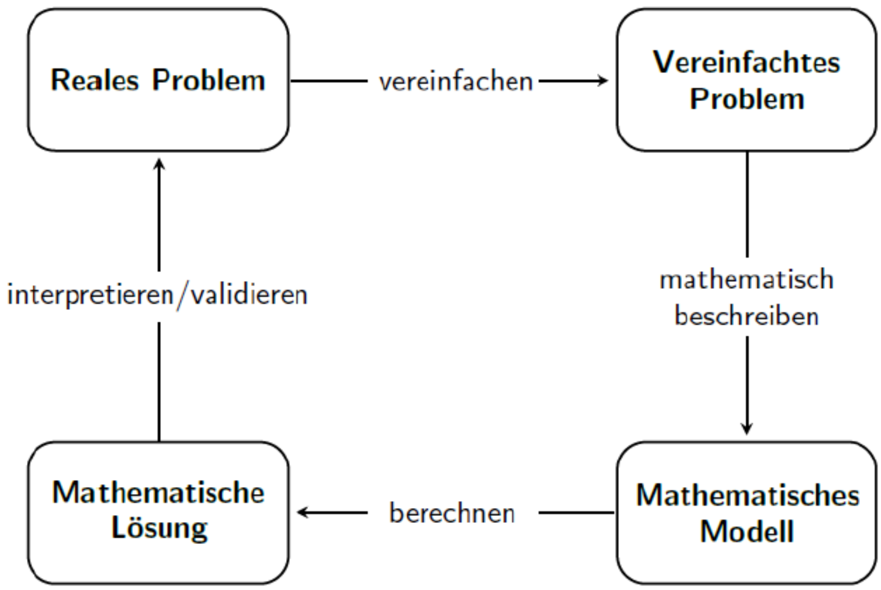


Abbildung 2: Vereinfachter, in CAMMP genutzter Modellierungskreislauf [10, S. 16]

Auf den ersten Blick ist zu erkennen, dass der Modellierungskreislauf von Ortlieb nur vier Schritte enthält. Auch hier besteht der erste Schritt (in Anlehnung an den Kreislauf von Blum und Leiß) darin, die reale Situation zu strukturieren und zu vereinfachen, um ein vereinfachtes Problem zu erhalten. Im zweiten Schritt wird das vereinfachte Problem mathematisiert bzw. mathematisch beschrieben und so zu einem mathematischen Modell überführt. Mit diesem Modell können die Lernenden im dritten Schritt mit mathematischen Methoden arbeiten und Lösungen für das Problem berechnen. Die daraus resultierende mathematische Lösung wird dann im letzten Schritt, bezogen auf das reale Problem, interpretiert und validiert. Hier wurden die im Kreislauf von Blum und Leiß genannten, einzelnen Schritte „Interpretieren“ und „Validieren“ in einem Schritt zusammengefasst. Auch beim Modellierungskreislauf von Ortlieb beginnt der Kreislauf von vorne, wenn die herausgefundene, mathematische Lösung kein zufriedenstellendes Ergebnis für das reale Ausgangsproblem darstellt.
Dieser reduzierte und vereinfachte Kreislauf kann den SuS dabei helfen, den Prozess des mathematischen Modellierens besser nachzuvollziehen und als Orientierung bei der Herangehensweise an die mathematische Lösung realer Probleme dienen.

2.2.3. Vorzüge digitaler Werkzeuge im Modellierungsprozess

Das Zeitalter der Medien schreitet immer weiter voran und die digitale Welt verändert das Lernen wie kaum eine gesellschaftliche Entwicklung zuvor. Besonders im Umgang mit realen Problemstellungen können die neuen, digitalen Werkzeuge, wie Computer oder grafikfähige Taschenrechner eine sinnvolle Unterstützung darstellen, da sie vor allem die Einführung komplexerer Anwendungen und Modellierungen ermöglichen. Weiter verringert der Einsatz digitaler Werkzeuge im Bereich Modellieren den Rechenaufwand, wodurch der Fokus auf den eigentlichen Modellierungprozess und das tiefgehende Verständnis dafür gelegt werden kann [2, S. 371]. Auch bieten digitale Werkzeuge den SuS die Möglichkeit, sich in angemessener Zeit anhand von Ausprobieren einem Problem zu näheren sowie ein realistisches Datenvolumen zu bearbeiten. Digitale Werkzeuge bieten außerdem die Möglichkeit ein reales Problem mit Hilfe einer Geometriesoftware oder einer Tabellenkalkulation in ein geometrisches Modell zu überführen. Mit diesen Modellen können sich die SuS die reale Situation veranschaulichen und experimentell der Lösung des Problems annähern. Ein weiterer Vorteil digitaler Werkzeuge ist im Bereich Kontrollieren und Überprüfen zu erkennen, dessen Prozess digitale Medien mit Hilfe von Überprüfungsfunktionen oder bereits vorhandenen Modellen unterstützen können. Werden zusätzlich digitale Geräte mit Internetanschluss genutzt, können diese auch zum Recherchieren von Informationen genutzt werden. Dies bietet den Lernenden gleich eine große Vielfalt an Wegen, das reale Problem zu verstehen und zu vereinfachen. Demnach ist der Einsatz von digitalen Medien in allen Phasen des Modellierungsprozesses sinnvoll und nützlich [2, S. 371-373].

2.2.4. Vorteile und Ziele von Modellieren

Mit Anwendungen und Modellierungsaufgaben im Mathematikunterricht werden unterschiedliche Ziele auf verschiedenen Ebenen verfolgt. Im Folgenden werden verschiedene Ziele erläutert:

- **Allgemeine Ziele**
  Die Verwendung von Mathematik im Kontext von Realbezügen soll den SuS ein ausgewogenes Bild von Mathematik als Wissenschaft vermitteln. Weiter sollen die SuS den Sinn und Nutzen der Mathematik an alltäglichen Problemstellungen erfahren sowie die Bedeutung der Mathematik in der Gesellschaft und der Technik erkennen. Die SuS sollen mit Modellieren in der Lage sein, den Stellenwert der Mathematik in der Welt zu erkennen [15, S. 16].

- **Inhaltsorientierte Ziele**
  Dieses Ziel beschreibt die pragmatische Sicht der SuS auf die Welt. Sie sollen sich durch die Beschäftigung mit Modellierungsaufgaben mit realen Situationen beschäftigen und ihre Umwelt mit mathematischen Mitteln erschließen. Das heißt, die SuS sollen in der Lage sein, Erscheinungen unserer Welt wahrzunehmen und zu verstehen. Dieses Ziel entspricht der ersten der drei Grunderfahrungen nach Winter, die jede Schülerin und jeder Schüler im Laufe seines Mathematikunterrichts erleben soll [10, S. 20].

- **Prozessbezogene Ziele**
  Die Beschäftigung mit anwendungsorientierten Aufgaben im Mathematikunterricht fördert allgemeine mathematische Kompetenzen, wie Problemlösefähigkeiten oder Kommunizieren und Argumentieren. Ebenso können bei der Bearbeitung von Modellierungsaufgaben heuristische Strategien wie beispielsweise Analogienbildung oder Rückwärtsarbeiten verwendet und gefördert werden [10, S. 20].

- **Lernpsychologische Ziele**
  Das Arbeiten an Modellierungsaufgaben fördert das Verstehen und Behalten mathematischer Inhalte, da Modellierungsaspekte motivierend wirken und den SuS Beispiele für die Relevanz der Mathematik im Alltag aufzeigen. Durch Anwendung und Transfer wird einerseits das gelernte Wissen besser behalten und anderseits das Interesse an Mathematik bei den SuS geweckt [15, S. 16].

- **Utilitaristische Ziele**
  Modellieren hilft den SuS in der Lage zu sein, Mathematik zu benutzen, um vielfältige Situation auch außerhalb der Mathematik zu bewältigen. So wird die Anwendung von Mathematik in außermathematischen Zusammenhängen durch die Beschäftigung mit Modellierungsaufgaben geübt [15, S. 16]. Den SuS soll mit Modellierungsaufgaben einerseits die nötigen Qualifikationen für die Bewältigung des Alltags vermittelt und andererseits die Entfaltung individueller Fähigkeiten ermöglicht werden [9, S. 38].

3. Mathematischer und theoretischer Hintergrund

Shazam ist eine App, die Musiktitel in Sekundenbruchteilen erkennt und dem Benutzer mitteilt. Ein einfacher Klick auf die Smartphone-App und der Nutzer weiß in Sekunden alles über den bis dato noch unbekannten Song und wird auch gleich via YouTube oder Spotify verlinkt, um diesen anhören zu können. Wie Shazam Musiktitel in seiner Datenbank speichert sowie mit den aufgenommenen Songs vergleicht und welche grundlegenden mathematischen Ideen hinter der App stecken, wird in den nachfolgenden Abschnitten erläutert.

3.1. Der Erfolg von Shazam

Shazam wurde 1999 gegründet und setzte sich zum Ziel, einen Algorithmus zu entwickeln, der Musikstücke erkennt. Bereits 2002 wurde dieses Ziel erreicht, jedoch blieb der Erfolg von Shazam zunächst aus. Dies hatte vor allem zwei Gründe: Zum einen war die Musikdatenbank von Shazam noch recht überschaubar und zum anderen gab es noch keine Smartphone-App, wie sie heute bekannt ist. Anfangs bestand die Abfrage eines Musiktitels im Anrufen einer Kurzwahlnummer, die kostenpflichtig war. Der Song wurde auf einem Band aufgezeichnet und der passende Eintrag dem Benutzer in Form einer SMS mitgeteilt [5]. Bekannt geworden ist Shazam erst im Jahr 2008, indem die Firma ihre App als einer der ersten Apps überhaupt im AppleAppStore veröffentlichte [5]. Heute ist Shazam eine der beliebtesten Apps weltweit. 2013 gehörte sie zu den Top 10 der Apps auf internationaler Ebene und ist damit Marktführer im Bereich der Musikerkennung. Nach Angaben von Shazam weist die App über 100 Millionen aktive Benutzer monatlich auf sowie eine Datenbank-Abfrage von mehr als 4 Millionen pro Tag. Vor allem die Schnelligkeit, mit der Songs gefunden werden, und die mittlerweile enorm große Datenbank an vorhanden Musikstücken zeichnet Shazam besonders aus. Auch hervorzuheben ist, dass die App trotz Störgeräuschen sowie schlechter Mikrofonqualität die richtigen Ergebnisse erzielt [14].
Was Shazam so erfolgreich machte, ist die Idee einen akustischen Fingerabdruck von einem Lied zu generieren. Dieser Fingerabdruck ist, genau wie bei Menschen, für jedes Musikstück einzigartig. Die hohe Effizienz ihres akustischen Fingerabdruck-Algorithmus sowie ein intelligentes Verfahren beim Durchsuchen der Datenbank haben Shazam zum Erfolg gebracht.
Im nachfolgenden Abschnitt werden die grundlegenden Schritte von Shazam erläutert. Dazu wird ein kurzer Überblick über die einzelnen Handlungsabläufe der App gegeben. Der Schwerpunkt liegt dann auf dem akustischen Fingerabdruck-Algorithmus sowie dem mathematischen Werkzeug zur Erstellung des Fingerabdrucks, der Fourieranalyse.

3.2. Wie funktioniert Shazam?

Zunächst besteht die Aufgabe von Shazam darin, akustische Fingerabdrücke von Songs aus großen Musikdatenbanken zu erstellen und diese Fingerabdrücke in einer eigenen

Datenbank zu speichern. Die zugrunde liegende Datenbank wird also ständig erweitert. Verwendet ein Benutzer die Shazam-App, wird ein kurzer Teil des zu hörenden Musikstückes mit dem Mikrophon des Smartphones aufgenommen. Nun werden mit Hilfe eines Filters zunächst störende Hintergrundgeräusche beseitigt. Anschließend wendet die App den gleichen Fingerabdruck-Algorithmus an, der dazu benutzt wird, um Musikstücke in der Datenbank zu speichern. Nachdem dieser Abdruck erstellt wurde, schickt die App die Daten zur Shazam-Datenbank und vergleicht das aufgenommenen Audiosignal mit allen gespeicherten Songs. Findet Shazam eine Übereinstimmung, schickt es dem Benutzer alle relevanten Daten zurück auf das Smartphone, wie Titel, Interpret, Amazon url oder YouTube url [12].

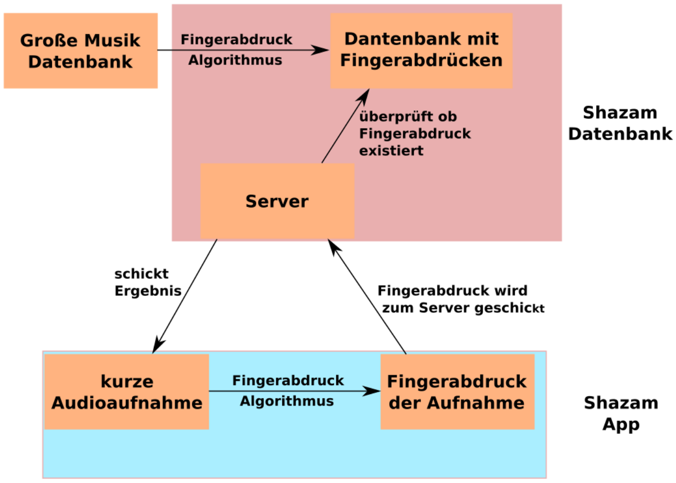


Abbildung 3: Dieses Modell von Shazam zeigt, wie die Shazam-App einen akustischen Fingerabdruck der Aufnahme erstellt und dieses an Shazam weiterleitet. Shazm sucht dann in seiner Datenbank nach einer Übereinstimmung und schickt das Ergebnis dem Benutzter der App zu.

Ziel der nachfolgenden Abschnitte ist es, die grundlegenden, mathematischen Ideen hinter dem akustischen Fingerabdruck-Algorithmus sowie hinter dem Durchsuchen der Datenbank zu verdeutlichen. Dazu soll jedoch zunächst diskutiert werden, wie man Töne bzw. Audiosignale mathematisch modellieren kann.

3.3. Töne mathematisch modellieren

Jedes Instrument und auch die menschliche Stimme verursacht eine gewisse Art von Schwingung, auch wenn diese für normale Menschen nicht wahrnehmbar ist und nur mit Hilfe eines Computers oder eines Oszilloskops sichtbar gemacht werden kann. So kann man sich mit diesen Mitteln ganz einfach das Audiosignal einer Klarinette (siehe Abbildung 4) anzeigen lassen. Anzumerken ist jedoch, dass der Computer oder das Oszilloskop ein Audiosignal nicht kontinuierlich aufnehmen kann. Stattdessen nimmt der Computer diskrete Messwerte, also einzelne Punkte des Signals, auf. Damit besteht in Wahrheit das zu beobachtende Bild am Computer nicht aus einer kontinuierlichen Kurve, sondern aus einer Aneinanderreihung von einzelnen Punkten. Die Anzahl der Punkte, die pro Sekunde aufgenommen werden, wird als *Sampling-Rate* bezeichnet. Für Musikaufnahmen liegt diese Rate typischerweise bei 44100 aufgenommenen Datenpunkten pro Sekunden, also bei $44100Hz$. Da so viele Punkte pro Sekunde aufgenommen werden, sieht es für das menschliche Auge so aus, als würde es eine kontinuierliche Kurve anschauen [12].

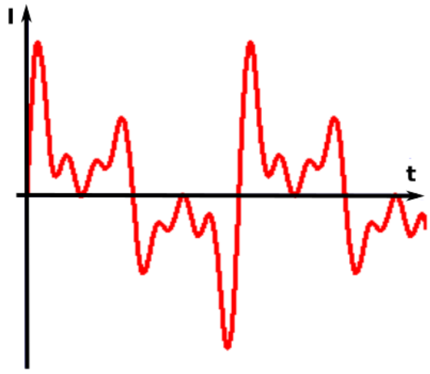


Abbildung 4: visualisiertes Audiosignal einer Klarinette

Komplexe Audiosignale, wie das einer Klarinette, bestehen aus verschiedenen Teiltönen (siehe Abbildung 5), die der Mensch einzeln nicht wahrnehmen kann, aber für jedes Signal einzigartig sind. Werden die einzelnen Teiltöne überlagert, erhält man wieder das ursprüngliche Audiosignal.

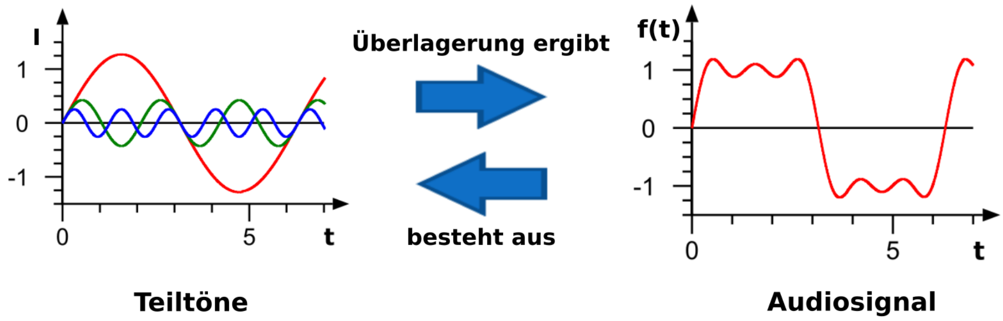


Abbildung 5: Audiosignale bestehen aus einzelnen Teiltönen, die mathematisch die Form einer Sinus- oder Cosinusschwingung haben. Die Überlagerung der einzelnen Teiltöne ergibt wieder das Audiosignal

Diese einzelnen Teiltöne können mathematisch ganz einfach beschrieben werden, da diese Töne die Gestalt von harmonischen Sinus- und Cosinusschwingung annehmen [12]. Als Beispiel wird der Ton einer Stimmgabel betrachtet, der eine fast perfekte Sinusschwingung darstellt, die mit dem Computer oder dem Oszilloskop sichtbar wird:

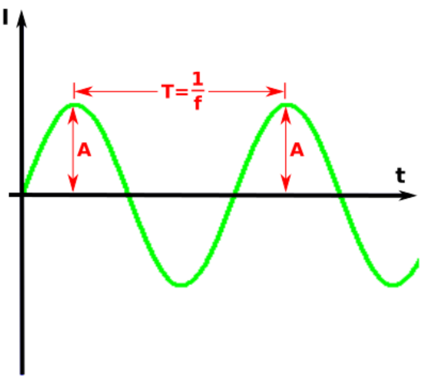


Abbildung 6: Sinusschwingung einer Stimmgabel mit A, der Amplitude, $T = \frac{1}{f}$, der Periode und f, der Frequenz

Das heißt Teiltöne können in der Form

$$f(t) = A \cdot sin(\omega t)$$

dargestellt werden (analog mit cos). Dabei steht

- t für die Zeit.
- A für die Amplitude und gibt die maximale Lautstärke des Tons an. Das heißt je höher die Amplitude, desto lauter wird der Ton wahrgenommen.
- ω für die Kreisfrequenz sowie $f = \frac{w}{2 \cdot \pi}$ für die Frequenz und gibt die Tonhöhe an. Das heißt je kleiner die Periode, also der Abstand zwischen zwei gleichen Maxima, desto höher wird der Ton wahrgenommen.

Mit dieser mathematischen Darstellung kann die Amplitude und die Frequenz der einzelnen Teiltöne gleich abgelesen und in ein Frequenzspektrum überführt werden, wie Abbildung 7 anhand des Sinustons der Stimmgabel zeigt.

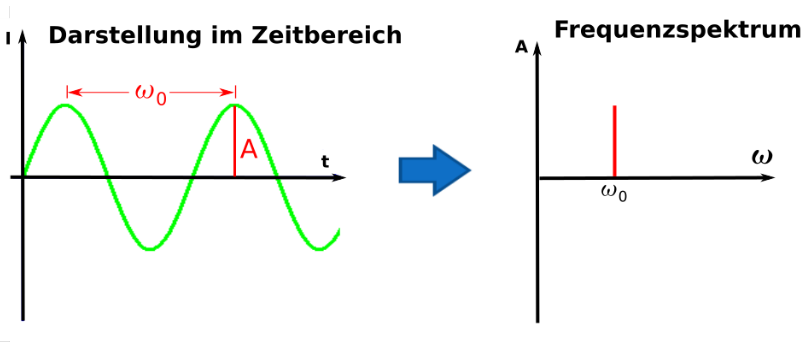


Abbildung 7: Links wird der Sinuston im Zeitbereich dargestellt, d.h. jedem Zeitpunkt wird ein Abtastwert zugeordnet. Die rechte Abbildung zeigt die Überführung in das Frequenzspektrum und die Frequenz mit zugehöriger Amplitude angibt.

Shazam ist später für die Erstellung des akustischen Fingerabdrucks an den enthaltenen Informationen im Frequenzspektrum interessiert, da das Frequenzspektrum für jedes Musikstück bzw. Audiosignal einzigartig ist.
Die Frage ist nun, wie aus den komplexen Audiosignalen die einzelnen, harmonischen Schwingungen (Sinus- und Cosinustöne) mit jeweils unterschiedlicher Amplitude und Frequenz dargestellt werden können, um die einzelnen Frequenzen sowie Amplituden der Sinus- und Cosinusfunktionen ablesen zu können und damit das Frequenzspektrum des ursprünglichen Audiosignals zu bestimmen. Die Lösung fand Joseph Fourier, der Anfang des 19. Jahrhunderts Folgendes erkannte [16]:

Satz von Fourier
Jede periodische Funktion lässt sich als Summe von Sinus- und Cosinusfunktionen darstellen.

Doch Fourier erkannte nicht nur, dass periodische Funktionen wie z.B. Audiosignale durch Überlagerungen von Sinus- und Cosinusfunktionen dargestellt werden können. Er entwarf auch ein mathematisches Werkzeug, die Fourieranalyse, mit dem die einzelnen Sinus- sowie Cosinusbestandteile des Signals herausgefunden werden können. Eine Fourieranalyse macht nämlich nichts anderes, als ein periodisches Eingangssignal (Abb. 8 a)) in Sinus- und Cosinusfunktionen zu zerlegen, die aufaddiert wieder das Eingangsignal ergeben bzw. approximieren (Abb. 8 b)). Damit können die einzelnen Frequenzen und Amplituden der Sinus- und Cosinustöne bestimmt und das Audiosignal in seinem Frequenzbereich dargestellt werden (Abb. 8 c)). So können mit der Fourieranalyse die Frequenzen und zugehörigen Amplituden der einzelnen Teiltöne des Audiosignals bestimmt werden [16].

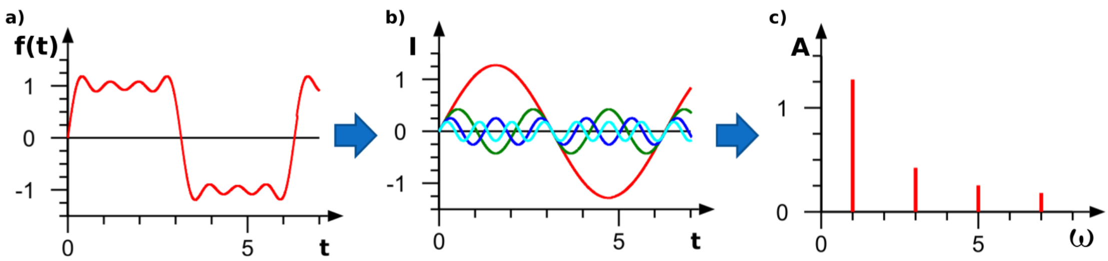


Abbildung 8: Beispiel für eine Frequenzzerlegung eines Audiosignals: a) Eingangssignal; b) Zerlegung in einzelne Sinus- und Cosinusbestandteile; c) Frequenzspektrum [1]

Wie die Fourieranalyse mathematisch genau aussieht und wie man die einzelnen Frequenzen sowie Amplituden eines Audiosignals damit berechnet, wird in Abschnitt 3.6 genauer thematisiert.

Shazam unterteilt die Audiosignale in verschiedene Zeitintervalle, um ein besseres Er-

[1] Teile der Abbildung entnommen von `https://upload.wikimedia.org/wikipedia/commons/thumb/f/fa/Fourier_synthesis.svg/2000px-Fourier_synthesis.svg.png` (Stand 28.05.2016)

gebnis bei der Durchführung der Fourieranalyse zu erzielen. Denn je größer das zu untersuchende Intervall gewählt wird, desto schlechter können kleinere Frequenzänderungen wahrgenommen werden. Das heißt, eine Fourieranalyse über das gesamte Audiosignal hätte zur Folge, dass viele, wichtige Frequenzen, die das Signal ausmachen, verloren gehen würden. Normalerweise sind komplizierte Audiosignale, wie Teile von Musikstücken, aber nicht periodisch. Um die Fourieranalyse trotzdem durchführen zu können, wird einfach implizit angenommen, dass jedes Signal innerhalb eines Zeitintervalls periodisch fortgesetzt wird, also durch Aneinanderreihung des Signals unendlich oft wiederholt wird [11].

3.4. Der Akustische Fingerabdruck-Algorithmus

3.4.1. Charakteristische Punkte finden

Der erste Schritt für den akustischen Fingerabdruck besteht darin, charakteristische Punkte des Musikstücks zu identifizieren. Charakteristische Punkte sind im Wesentlichen die Frequenzen mit den zugehörigen Amplituden der einzelnen Teiltönen zu einem gewissen Zeitpunkt. Um die Datenmenge zu reduzieren sowie die charakteristischen Punkte effizienter zu finden, unterteilt Shazam das Audiosignal in verschiedene Zeitintervalle und bildet für jedes Intervall mit der Fourieranalyse ein Frequenzspektrum, das die Frequenzen und Amplituden der einzelnen Teiltöne enthält. In diesen Frequenzspektren sucht Shazam nach der Frequenz mit der maximalen Amplitude und speichert davon nur die Frequenz zusammen mit der Zeit in ein Spektogramm ab. Das heißt, im Spektrogramm werden nur die Frequenzen der charakteristischen Punkte von dem in den Zeitintervallen lautesten Teilton gespeichert. Mit diesen Punkten erhält man ein für das jeweilige Lied charakteristisches Spektogramm [11]. Ein Beispiel zeigt folgende Abbildung:

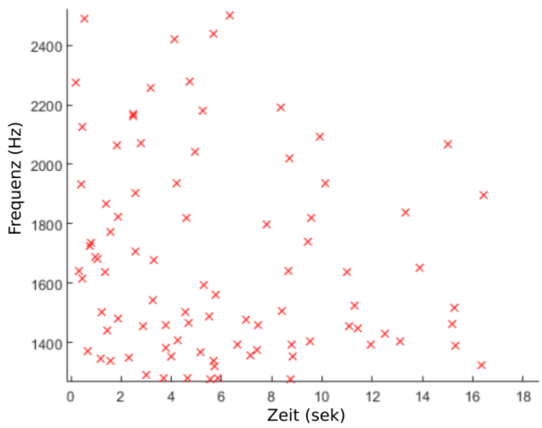


Abbildung 9: Das Spektogramm eines Audiosignals mit charakteristische Frequenzen, die im Signal enthalten sind [11].

Dieses charakteristische Spektogramm ist für jedes Audiosignal einzigartig und erstellt einerseits Shazam für jedes Musikstück in seiner Datenbank sowie andererseits auch die Smartphone-App für das kurze, aufgenommene Audiosignal.

3.4.2. Target Zone

Die Frage, die sich nun stellt, ist, wie es Shazam schafft, das Spektogramm des zehnsekundigen Audiosignals mit einem 200 sekündigen Musikstück zu vergleichen. Veranschaulicht und vereinfacht kann sich Folgendes vorgestellt werden:

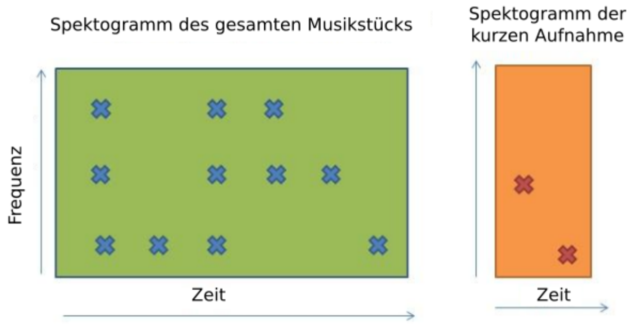


Abbildung 10: Spektogramm vom Original-Song (links) und von der zehnsekündigen Aufnahme (rechts) [11]

Shazam hat von jedem Musikstück in seiner Datenbank ein Spektrum, wie in Abbildung 10 links ersichtlich, gespeichert und erhält nun ein kleines Spektrum des aufgenommenen Audiosignals der Shazam-App (Abbildung 10 rechts). Nun überprüft Shazam im ganzen Musikstück durch Überlagerung der beiden Spektogramme, ob das kleine Spektogramm der aufgenommen Aufnahme mit einem Teil des Spektogramms des gesamten Musikstücks übereinstimmt.

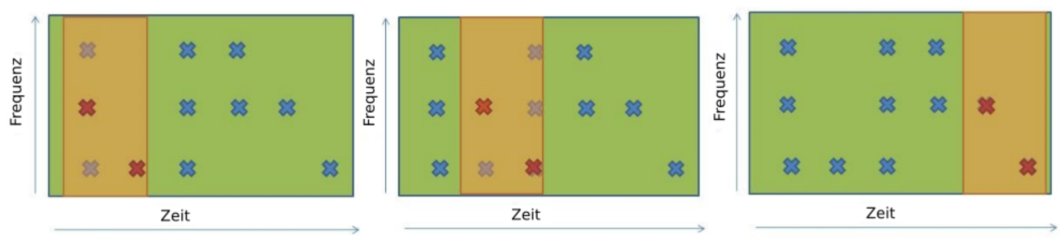


Abbildung 11: Überlagerung der beiden Spektogramme zum Finden einer Übereinstimmung [11]

In diesem Beispiel wurde eine perfekte Übereinstimmung zwischen der zehnsekündigen Aufnahme und dem Ende des Musikstücks gefunden. Dieser Vergleich muss mit jedem gespeicherten Song in der Datenbank von Shazam geschehen, bis man eine Übereinstimmung wie im obigen Beispiel erzielt. Wird auf Grund von störenden Einflüssen keine perfekte Übereinstimmung in allen gespeicherten Songs der Datenbank gefunden, nimmt man das Musikstück mit der besten Übereinstimmung. So kann man beispielsweise bei einer Übereinstimmung von über 90% davon ausgehen, den richtigen Song gefunden zu haben. Die restlichen Prozent lassen sich durch externe Geräusche oder die schlechte Mikrofonqualität des Smartphones erklären [11].

Um die Suche nach dem passenden Musikstück effizienter sowie schneller zu betreiben, vergleicht Shazam jedoch nicht jeden Datenpunkt, sondern mehrere Punkte zur gleichen Zeit. Diese Gruppe von Datenpunkten nennt man ***Target Zone***. Dazu nummeriert Shazam jeden einzelnen Frequenz-Datenpunkt durch, wie die folgende Abbildung verdeutlicht:

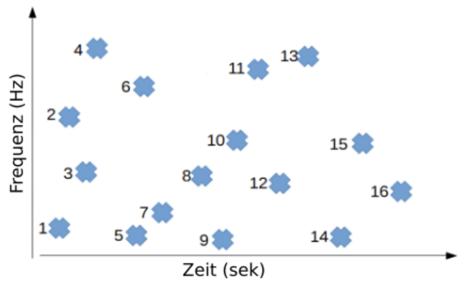


Abbildung 12: Nummerierte Zeit-Frequenz-Punkte

Da Shazam das genaue Vorgehen beim Bilden der *Target Zones* unter Verschluss hält, wird nachfolgend eine mögliche Vorgehensweise dargestellt:
Da pro Zeitintervall nur ein Datenpunkt gespeichert wird, werden die Datenpunkte einfach nach chronologischer Abfolge durchnummeriert, wie Abbildung 12 zeigt. Das heißt, der Datenpunkt vom ersten Zeitintervall bekommt die Nummer eins zugeordnet. Somit ist gewährleistet, dass die Aufnahme und der ganze Song die gleichen *Target Zones* bilden. Eine *Target Zone* besteht aus acht Datenpunkten, wobei sich im Optimalfall jeweils immer 7 Punkte mit einer anderen *Target Zone* überschneiden. Von den acht Punkten wird immer der erste einer *Target Zone* als ***Anchor Point*** definiert. Auf Grund der optimal angenommenen Überschneidung von 7 Punkten ist jeder Punkt damit ein *Anchor Point* für eine bestimmte *Target Zone* [11]. Da eine so große Anzahl an Überschneidung mit denen im CAMMP-Day zu Verfügung stehenden Methoden zu viel Rechenzeit beansprucht, reduziert der im CAMMP-Day verwendete Algorithmus diese Anzahl, sodass nur jeder fünfte Datenpunkt als *Anchor Point* definiert wird. Im Basic Paper wird aber weiterhin der Optimalfall, dass jeder Punkt einmal ein *Anchor Point* ist, betrachtet.

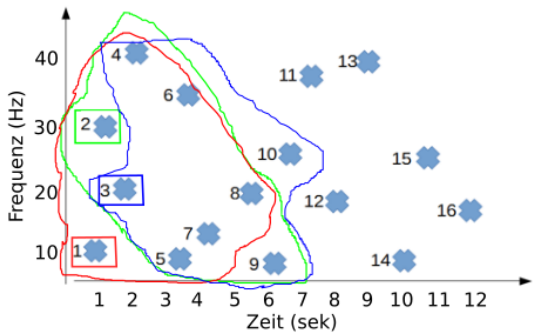


Abbildung 13: Andeutung der Bildung von Target Zones

3.4.3. Adressen bilden

Nachdem die *Targes Zones* gebildet wurden, kreiert Shazam für jeden Punkt eine sogenannte Adresse und speichert diese ab. Für die Abspeicherung der Adresse werden folgende Daten erfasst [12]:

1. Die Frequenz des *Anchor Points*, in dessen *Target Zone* sich der Datenpunkt befindet
2. Die Frequenz des Datenpunkts
3. Die Zeitdifferenz zwischen *Anchor Point* und Datenpunkt

Hier ein kleines Beispiel:

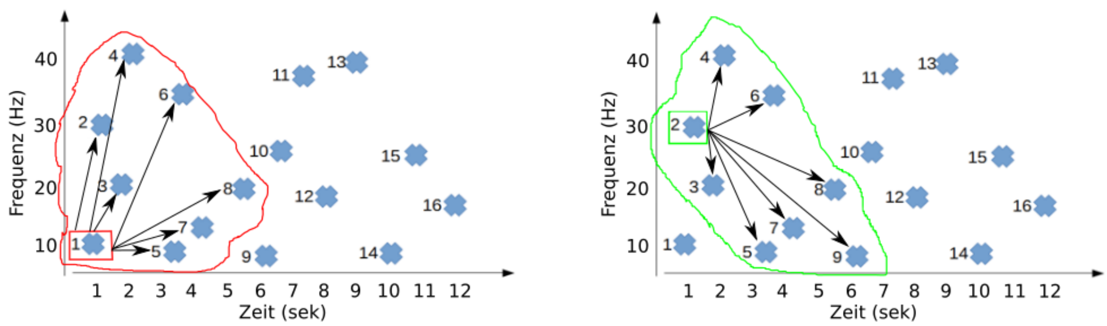


Abbildung 14: Beispiel zur Bestimmung der Adressen

So lautet beispielsweise bezogen auf die erste *Target Zone* (rot) die Adresse für den dritten Punkt [10, 20, 1] und für den sechsten [10, 35, 3]. Natürlich existieren für diese beiden Punkte auch noch weiter Adressen. Z.B. lautet die Adresse bezogen auf die zweite *Target Zone* (grün) [30, 20, 0.5] für den dritten und [30, 35, 2.5] für den sechsten Datenpunkt.
Diese Adressen werden in der Datenbank von Shazam zu einem Paar mit folgenden Informationen verlinkt [11]:

1. Der Zeitpunkt des *Anchor Points* und
2. Die Identität vom Musikstück (Titel, Interpret)

Wäre Abbildung 14 das Spektogramm des Musikstückes *Haus am See* von Peter Fox, würde man für die erste *Target Zone* die Adressen wie folgt verlinken:

$$[10, 20, 1] \rightarrow [1, \text{Haus am See von Peter Fox}]$$

$$[10, 35, 3] \rightarrow [1, \text{Haus am See von Peter Fox}]$$

Verfährt man mit diesem Schema mit allen Datenpunkten in allen *Target Zones*, so erhält man eine große Tabelle mit zwei Spalten:

i) Die Adresse (Frequenz *Anchor Point*, Frequenz Datenpunkt, Zeitdifferenz zwischen den beiden Punkten)

ii) Das Paar (Zeit vom *Anchor Point*, Songidentität)

Diese Tabelle ist der akustische Fingerabdruck eines Musikstücks und wird in der Datenbank von Shazam gespeichert.

3.5. Die Datenbank durchsuchen

Hat die Shazam-App ein zu erkennendes Audiosignal aufgenommen, erstellt sie zunächst von der kurzen Aufnahme nach dem gleichen Prinzip wie in den Liedern der Datenbank einen Fingerabdruck. Mit dem einzigen Unterschied, dass hier der Adresse nur der Zeitpunkt des *Anchor Points* zugeordnet wird, da die Songidentität nicht bekannt ist. Jede erstellte Adresse der Datenpunkte in der Aufnahme wird zur Datenbank weitergeleitet und mit den vorhandenen Adressen verglichen, um das dazugehörige Paar (Zeit vom *Anchor Point*, Songidentität) zu finden.
Anschließend werden die gefunden Paare gefiltert, indem nur die Lieder mit den meisten Übereinstimmungen beibehalten werden. Passen zum jetzigen Zeitpunkt noch mehrere Musikstücke zur Aufnahme, also wurde bei mehreren Liedern eine Übereinstimmung von bzw. nahe 100% mit den Adressen der Aufnahme gefunden, muss die Zeitkohärenz zwischen Aufnahme und möglichen Musikstücken betrachtet werden [11]. Denn ohne die Betrachtung der Zeitdifferenz kann es passieren, dass die gleichen Adressen der Aufnahme in einem Musikstück gefunden werden, jedoch die Reihenfolge der Adressen und damit der Datenpunkte nicht die gleiche ist wie in der Aufnahme. So kann es beispielsweise passieren, dass zwei unterschiedliche Songs die gleichen *Target Zones* haben wie die Aufnahme, nur zu unterschiedlichen Zeiten, wie Abbildung 15 zeigt. Während in Song 1 die *Target Zones* wie bei der Aufnahme hintereinander folgen, liegen die *Target Zones* bei Song 2 zeitlich auseinander. Das heißt, bei beiden Songs würde man ohne Berücksichtigung der Zeitdifferenz eine Übereinstimmung von 100% erhalten, obwohl die Reihenfolge der Adressen in Song 2 sich von der Aufnahme unterscheidet.

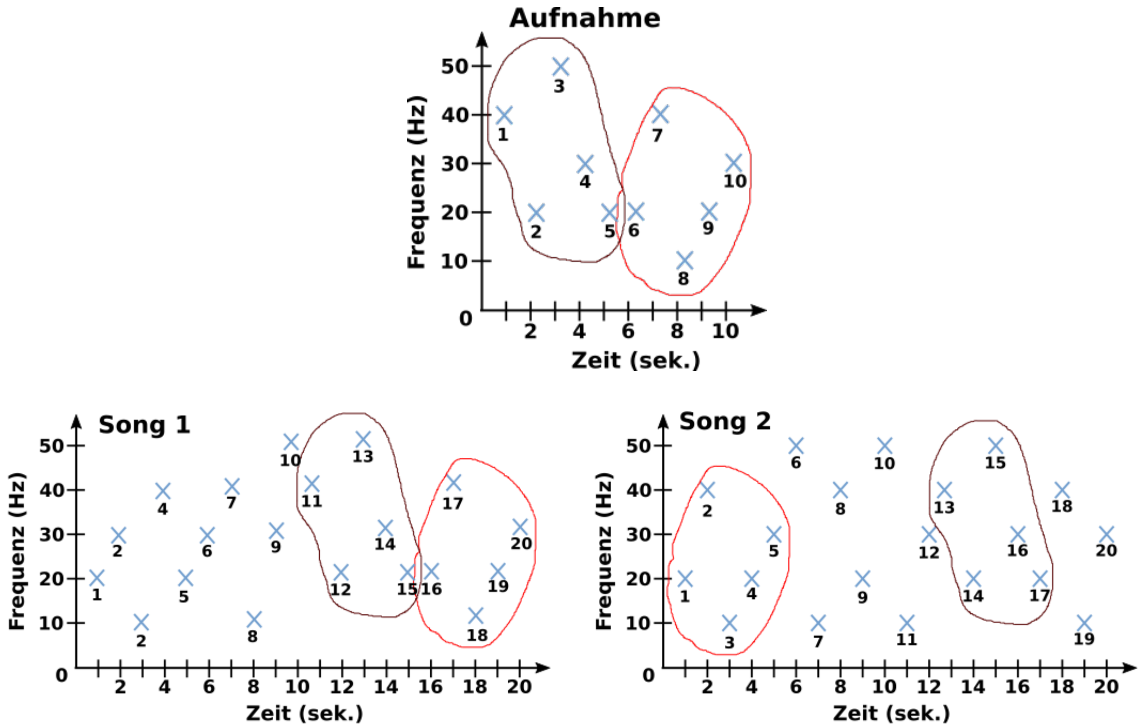


Abbildung 15: Zwei Songs, die die gleichen Target Zones (hier vereinfacht dargestellt nur 5 Punkte in einer *Target Zone*) haben wie die Aufnahme, jedoch zu unterschieldlichen Zeiten.

Natürlich müssen nicht immer gleich ganze *Target Zones* mit der Aufnahme übereinstimmen. So können auch die gleichen Adressen der Aufnahme in unterschiedlichen *Target Zones* im Song auftauchen, wie Abbildung 16 zeigt. Auch hier hätte man eine hohe prozentuale Übereinstimmung der Adressen, jedoch stimmt wieder die Reihenfolge der Datenpunkten bzw. Adressen des Songs nicht mit der Reihenfolge der Aufnahme überein.

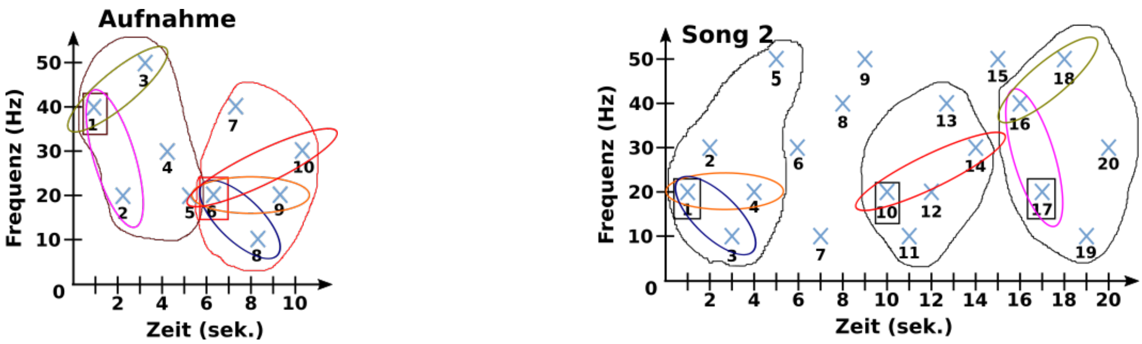


Abbildung 16: Die Adressen der Aufnhame finden sich in unterschiedlichen *Target Zones* im Song 2 wieder.

Somit ist es wichtig, die Zeitkohärenz zwischen Aufnahme und möglichen Musikstücken zu betrachten, um zu prüfen, ob alle Datenpunkten und damit alle Adressen in der Aufnahme in genau der gleichen Reihenfolge im Musikstück wieder gefunden werden. Um den zeitlichen Versatz, sprich die Zeitdifferenz, zu betrachten, vergleicht Shazam jeweils den Zeitpunkt des *Anchor Points* der losgeschickten Adresse und den Zeitpunkt des *Anchor Points*, der für diese Adresse im jeweiligen Song in der Datenbank gespeichert ist:

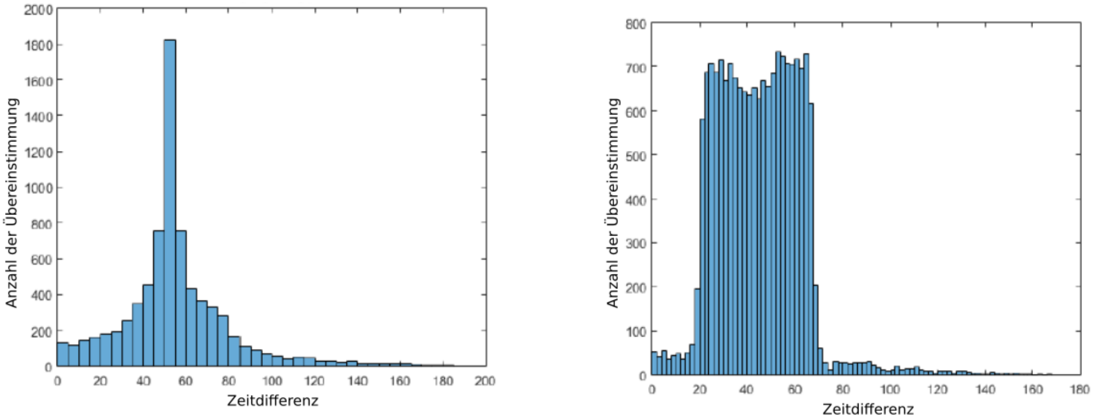
$$\text{Zeitpunkt } \textit{Anchor Point} \text{ im Song} - \text{Zeitpunkt } \textit{Anchor Point} \text{ der Aufnahme} = \Delta t$$

Diesen zeitlichen Vergleich zwischen den *Anchor Points* führt Shazam für jede Adresse durch [11]. Hier ein kleines Beispiel:

Tabelle 1: Berechnung der Zeitverschiebung zwischen Aufnahme und Musikstück in der Datenbank. Dazu wird bei einer gefundenen Übereinstimmung der Adresse in der Datenbank die Zeitverschiebung zwischen dem Anchor Point, der zur Adresse gehört, und dem Anchor Point, den die Datenbank rausgibt, gebildet.

Zeitpunkt des Anchor Points in der Aufnahme	**Adressen der Aufnahme**	**Ausgabe der Datenbank bei Übereinstimmungen mit einem Musikstück**	**Zeitdifferenz Δt**
1	$(20, 40, 1)$	[51, Haus am See von Peter Fox]	$51 - 1 = 50$
1	$(20, 10, 2)$	[51, Haus am See von Peter Fox]	$51 - 1 = 50$
1	$(20, 20, 3)$	[51, Haus am See von Peter Fox]	$51 - 1 = 50$
2	$(10, 50, 1)$	[52, Haus am See von Peter Fox]	$52 - 2 = 50$
2	$(10, 20, 2)$	[52, Haus am See von Peter Fox]	$52 - 2 = 50$
2	$(10, 40, 3)$	[52, Haus am See von Peter Fox]	$52 - 2 = 50$
3	$(40, 50, 1)$	[53, Haus am See von Peter Fox]	$53 - 3 = 50$
3	$(40, 10, 2)$	[53, Haus am See von Peter Fox]	$53 - 3 = 50$
3	$(40, 30, 3)$	[53, Haus am See von Peter Fox]	$53 - 3 = 50$

Das Musikstück, welches die meisten Übereinstimmungen in den Zeitdifferenzen besitzt, wird dem Benutzter der Shazam-App weitergeleitet. Das heißt bei einer Übereinstimmung mit der Aufnahme wird ein klarer Peak bei einer Zeitdifferenz zu beobachten sein, wie Abbildung 17 (links) zeigt. Tauchen andauernd verschiedene Zeitdifferenzen auf, wie in Abbildung 17 (rechts) zu beobachten, kann eine Übereinstimmung mit der Aufnahme ausgeschlossen werden.



Musikstück 1 Musikstück 2

Abbildung 17: Überprüfung der Zeitdifferenz zwischen zwei verschiedenen Musikstück und der Aufnahme. Beim ersten Musikstück (links) ist ein deutlicher Peak bei einer Zeitdifferenz von etwa 50 zu erkennen, während im zweiten Musikstück die Zeitdifferenz stark variiert. Musikstück 1 passt somit zu der Aufnahme.

In diesem Fall wäre beim Vergleich dieser beiden Musikstücke der linke Song passend, da dort die meisten Δt übereinstimmen.

3.6. Fourieranalyse

Die Fourieranalyse ist eins der wichtigsten, mathematischen Werkzeuge für Shazam, da es das Audiosignal in seine Sinus- und Cosinusfunktionen zerlegt und dadurch die einzelnen Frequenzen sowie Amplituden des Signals bestimmt werden können, wie Abbildung 18 nochmal zeigt. So soll in diesem Abschnitt die Fourieranalyse mathematisch thematisiert werden.

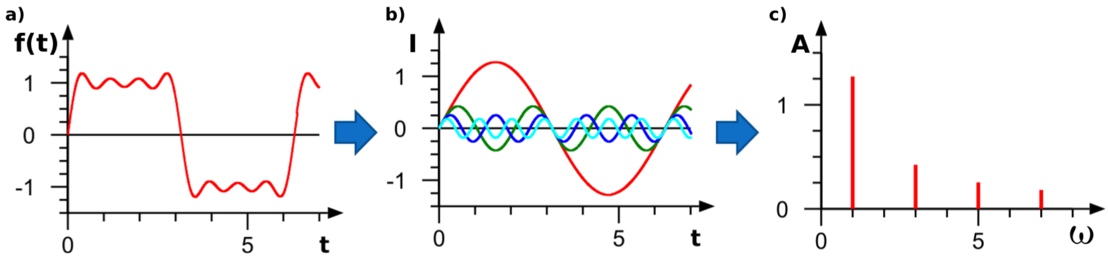


Abbildung 18: Beispiel Frequenzzerlegung eines Audiosignals: a) Eingangssignal; b) Zerlegung in einzelne Sinus- und Cosinusbestandteile; c) Frequenzspektrum [2]

[2]Teile der Abbildung entnommen von `https://upload.wikimedia.org/wikipedia/commons/thumb/f/fa/Fourier_synthesis.svg/2000px-Fourier_synthesis.svg.png` (Stand 28.05.2016)

Das Ergebnis einer solchen Zerlegung ist eine unendliche Reihe, die sogenannte *Fourier-Reihe*, mit folgender Gestalt [16]:

Fourier-Reihe

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t)) \tag{1}$$

Dabei bezeichnet ω_0 die Grundkreisfrequenz zu der Grundschwingung. Die Signale mit der Kreisfrequenz $n \cdot \omega_0 = \omega_n$ mit $n \in \mathbb{N}$ werden als Oberschwingungen bezeichnet. Die Koeffizienten a_0, a_n und b_n nennt man *Fourierkoeffizienten.* Diese müssen jeweils für das zu zerlegende Audiosignal rechnerisch bestimmt werden.

Bei der Fourieranalyse wird das Eingangssignal wie aus (1) ersichtlich durch eine andere Basis dargestellt und zwar mit den trigonometrischen Basisfunktionen zur Periode 2π. Diese Basisfunktionen können zusammengefasst werden als [16]:

$$\left.\begin{array}{lr} 1 \quad \text{(der konstanten Funktion)} & \\ \cos(n \cdot t) & \text{für } n \in \mathbb{N} \\ \sin(n \cdot t) & \text{für } n \in \mathbb{N} \end{array}\right\} \text{ Basisfunktionen} \tag{2}$$

Eine praktische Eigenschaft dieser Basis ist, dass je zwei verschiedene trigonometrische Basisfunktionen bezüglich des Skalarprodukts zueinander orthogonal sind sowie das Skalarprodukt mit sich selbst als Ergebnis die Länge bzw. die Norm liefert. Die Basisfunktionen in (2) bilden also eine Orthogonalbasis und für $n, m \in \mathbb{N}$ gelten somit folgende Beziehungen [16]:

$$\int_{-\pi}^{\pi} 1 \cdot \sin(nt)\ dt = 0 = \int_{-\pi}^{\pi} 1 \cdot \cos(nt)\ dt \tag{3}$$

$$\int_{-\pi}^{\pi} \cos(nt) \cdot \sin(nt)\ dt = 0 \tag{4}$$

$$\int_{-\pi}^{\pi} \cos(mt) \cdot \cos(nt)\ dt = 0 = \int_{-\pi}^{\pi} \sin(mt) \cdot \sin(nt)\ dt, \qquad \textit{falls } n \neq m \tag{5}$$

$$\int_{-\pi}^{\pi} \cos^2(nt)\ dt = \pi = \int_{-\pi}^{\pi} \sin^2(nt)\ dt, \qquad \textit{falls } n = m \tag{6}$$

$$\int_{-\pi}^{\pi} 1\ dt = 2\pi \tag{7}$$

Der einzelne mathematische Beweis dieser Relationen würde den Rahmen dieses Basic Papers sprengen, jedoch soll hier an der Beziehung $\int_{-\pi}^{\pi} \cos(nt) \cdot \sin(nt)\ dt = 0$ gezeigt werden, wie man sich die oben erwähnten Beziehungen geometrisch verdeutlichen kann. Wird die Funktion $f(t) = \cos(nt) \cdot \sin(nt)$ im 2π-periodischen Intervall graphisch dargestellt, wird wie in Abbildung 19 ersichtlich, dass sich die Flächen oberhalb sowie unterhalb der x-Achse aufheben, sodass das Integral über die Funktion $f(t)$ gerade Null ergibt.

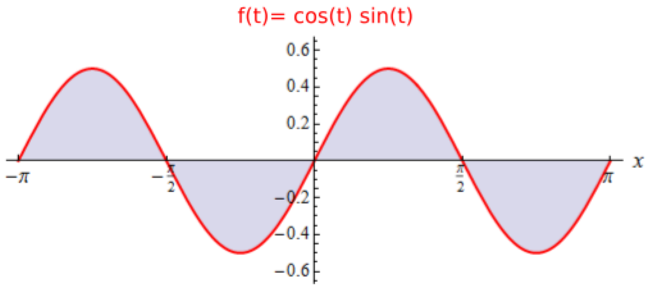


Abbildung 19: Graphische Darstellung der Funktion $f(t) = \cos(nt) \cdot \sin(nt)$ über die Periode 2π. Zu erkennen ist, dass sich die Flächen oberhalb der x-Achse und unterhalb der x-Achse aufheben [16].

Warum der Koeffizient in der Fourier-Reihe (1) als $\frac{a_0}{2}$ und nicht als a_0 geschrieben wird, ist nun auch aus der Norm der konstanten Funktion (7) ersichtlich.
Die Fourierkoeffizienten a_0, a_n und b_n können auf Grund der Orthogonalitätabeziehung der Basisfunktionen nun durch Integration über den 2π-periodischen Raum von (1), also

$$\begin{aligned} f(t) &= \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t)) \\ &= \frac{a_0}{2} + a_1 \cdot \cos(\omega_0 t) + a_2 \cdot \cos(2\omega_0 t) + ... + b_1 \cdot \sin(\omega_0 t) + b_2 \cdot \sin(2\omega_0 t) + ... \end{aligned} \tag{8}$$

einfach hergeleitet werden (siehe [16]).

- **Berechnung von a_0:**
Wird das Integral auf beiden Seite von (8) über das Intervall $[-\pi, \pi]$ gebildet, so verschwinden auf der rechten Seite die Integrale über alle auftretenden Cosinus- und Sinusfunktionen auf Grund der Orthogonalitätsbeziehung (3). Es trägt nur der konstante Term $\frac{a_0}{2}$ bei, dessen Wert des Integrals gleich $a_0\pi$ ist. Damit ist

$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t)\, dt. \tag{9}$$

- **Berechnung von a_n für $n \in \mathbb{N}$:**
Werden beide Seite von (8) mit der trigonometrischen Cosinus-Basisfunktion multipliziert und anschließend das Integral über das Intervall $[-\pi, \pi]$ gebildet, so verschwinden auf der rechten Seite wegen (4) und (5) alle Produkte von verschiedenen Basisfunktionen. Hier trägt nur der Term bei, in dem $\cos(nt)$ mit sich selbst multipliziert wird. Der Wert dieses Integrals ist mit (5) gleich $a_n\pi$ und damit ist

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos(nt)\, dt. \tag{10}$$

- **Berechnung von b_n für $n \in \mathbb{N}$:**
Für die Berechnung der b_n ist das Vorgehen identisch zu der Berechnung der a_n, nur dass hier beide Seiten mit der trigonometrischen Sinus-Basisfunktion multipliziert werden. Hier trägt dann nur der Term bei, in dem $\sin(nt)$ mit sich selbst multipliziert wird. Der Wert dieses Integrals ist mit (5) gleich $b_n\pi$ und damit ist

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)sin(nt)\,dt. \tag{11}$$

Zusammengefasst werden die drei Fourierkoeffizienten also wie folgt berechnet [16]:

$$\boxed{a_0 = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\,dt, \quad a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\cos(nt)\,dt, \quad b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)sin(nt)\,dt}$$

3.7. Windowing

Um Lieder besser vergleichen zu können, benötigt Shazam, wie bereits oben erwähnt, beim Erstellen des akustischen Fingerabdrucks lokale, maximale Frequenzen, die von der Zeit t abhängen. Dazu teilt Shazam das Audiosignal in Zeitintervalle ein, um dann mit Hilfe der Fourieranalyse die maximalen Frequenzen der Intervalle zu lokalisieren. Zum Schluss soll deshalb kurz diskutiert werden, wie Shazam die Audiosignale beim Erstellen des akustischen Fingerabdrucks in Zeitintervalle einteilt.
Zur Unterteilung des Audiosignals in verschieden Zeitblöcke wird die Funktion des Signals $f(t)$ mit einer *Fensterfunktion* (bzw. Window-function) $g(t)$ multipliziert. Damit wird nur ein kleines Fenster bzw. *Window* des Audiosignals betrachtet. Die Fourier Analyse wird dann anschließend über die resultierende Funktion $\tilde{f}(t) = g(t) \cdot f(t)$ durchgeführt [11, 3]. Die einfachste Variante, ein Signal in verschieden Blöcke bzw. *Windows* zu zerlegen, stellt die Rechteckfunktion (siehe Abbildung 20) dar.

$$g_R(t) = \begin{cases} 1 & t_i \leq t \leq t_j \\ 0 & \text{sonst} \end{cases} \qquad \text{mit } t_i < t_j \text{ Zeitpunkte im Audiosignal} \tag{12}$$

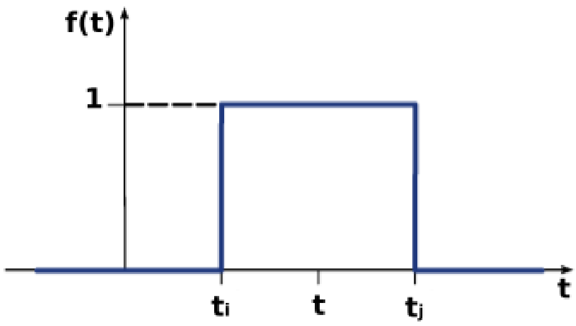


Abbildung 20: Darstellung einer Rechteckfunktion [3]

Jedoch treten beim Unterteilen des Audiosignals in Zeitintervalle auch Nachteile auf. Denn, wie bereits oben erwähnt, ist der Algorithmus für die Fourieranalyse nur für periodische Funktionen anwendbar. Das heißt, bei der Berechnung der Fourieranalyse wird implizit angenommen, dass das Signal innerhalb des eingeteilten Zeitintervalls bzw. *Windows* periodisch ist und dieses Signal durch Aneinanderreihung des Intervalls unendlich oft wiederholt wird. Das heißt, der Anfangs- und Endpunkt im *Window* werden so interpretiert, als würden sie zusammengehören. Ist das Audiosignal im betrachteten *Window* nicht periodisch, kann es an den Übergängen zu scharfen Diskontinuitäten kommen, wie Abbildung 21 zeigt. Diese Diskontinuitäten führen im Frequenzspektrum dazu, dass es zu einer Aufweitung der Spektrallinien sowie zum Auftauchen neuer Spektrallinien kommt. Es tauchen also Frequenzen auf, die im eigentlichen Audiosignal gar nicht enthalten sind. Diese Phänomen wird als *Leck-Effekt* bezeichnet [11]. Dieser Effekt kann dazu führen, dass im Frequenzspektrum nun Frequenzen auftauchen, die eine stärkere Intensität aufweisen, als die Frequenzanteile des ursprünglichen Audiosignals. So könnte es beim Erstellen des akustischen Fingerabdrucks passieren, dass Shazam fälschlicherweise Frequenzen abspeichert, die im Musikstück oder im aufgenommenen Audiosignal nicht enthalten sind.

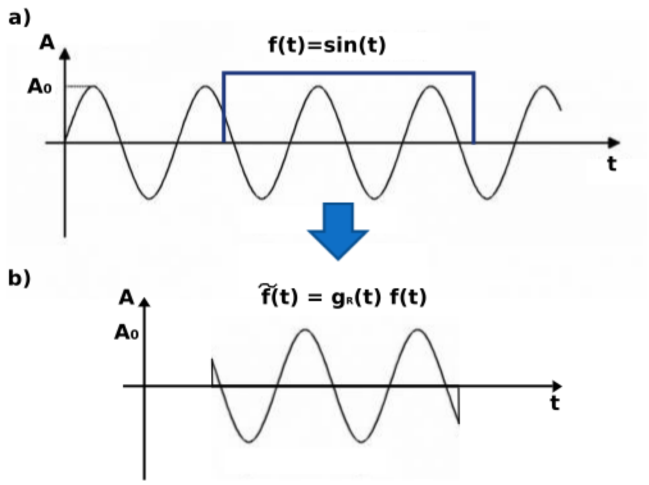


Abbildung 21: a) Anwendung der Rechteckfunktion $g_R(t)$ auf das Audiosignal $f(t)$; b) resultierende Funktion $\tilde{f}(t)$ mit den Unstetigkeiten an den Rändern der Funktion

Um den *Leck-Effekt* einzudämmen, werden effektivere Fensterfunktionen eingesetzt. Der für den CAMMP day entworfene Algorithmus nutzt beispielsweise das *Hamming Window* eine Funktion, die folgendermaßen aussieht [12]:

[3] Abbildung entnommen von `https://elearning.physik.uni-frankfurt.de/data/FB13-PhysikOnline/lm_data/lm_5563/daten/auto/bilder/s673a.gif` (Stand 25.06.2016)

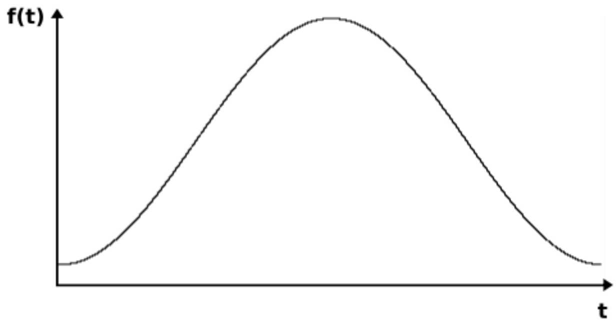


Abbildung 22: Darstellung des Hamming Windows

Durch die Form des *Hamming Windows* wird die Diskontinuität im Gegensatz zur Rechteckfunktion nahezu aufgehoben, was den *Leck-Effekt* deutlich verringert. Wie Abbildung 23 zeigt, ist mit dem *Hamming Window* eine deutlich geringere Aufweitung des Frequenzspektrum zu beobachten.

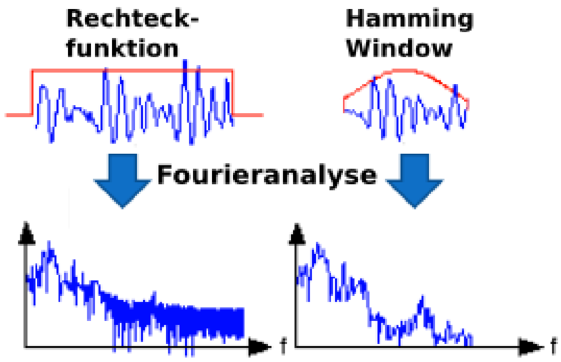


Abbildung 23: Vergleich des Frequenzspektrum nach Anwendung des Rechteckfensters (links) und des Hamming Window (rechts). Zu erkennen ist die geringe Aufweitung des Frequenzspektrums bei der Anwendung des Hamming Windows [4].

[4]Abbildung entnommen von `http://archive.cnx.org/resources/b3c2814076f9f3613b0625d25dae42b6578f0d2f/hanning1.png` (Stand 26.06.2016)

4. Didaktisch-methodisches Konzept

In diesem Kapitel werden die Ziele, der Aufbau sowie das didaktische Konzept hinter dem Lernmodul und den entwickelten Materialien erläutert.

4.1. Ziele des entwickelten Lernmoduls

Ein wichtiges Ziel dieses Lernmoduls ist es, den SuS die Relevanz des mathematischen Modellierens im alltäglichen Leben zu zeigen und bei ihnen die Motivation, das Interesse sowie den Spaß an der Mathematik zu wecken. Sie sollen den Sinn und auch die Nutzbarkeit von Mathematik in der Welt erkennen, um ihre Vorstellung über die Mathematik und deren Praxisrelevanz zu verbessern. Dazu bearbeiten die SuS ein reales Problem, das sich an ihrer Lebenswelt orientiert und deren Technik sie im Alltag oft benutzen. Das Lernmodul bietet den SuS so die Möglichkeit, eigene Erfahrungen über den praktischen Nutzen der Mathematik zu sammeln. Zum Lösen der Problemstellung sollen sie SuS die einzelnen Schritte des Modellierungsprozesses soweit wie möglich eigenständig durchlaufen. Somit soll das entwickelte Lernmodul den Aufbau der mathematischen Modellierungs- und Problemlösekompetenz fördern. Am Ende des Lernmoduls sollen die SuS die grundlegende Idee hinter Shazam sowie dem neuen mathematischen Inhalt, die Fourieranalyse, kennen und die Ansätze zum Bilden des akustischen Fingerabdruck und dem Durchsuchen der Datenbank verstanden haben. Weiter sollen sie durch den Umgang mit der computergestützten Software Matlab die Bedeutung von digitalen Werkzeugen zum Lösen anwendungsorientierter Aufgaben erfahren sowie den Umgang mit ihnen üben. Ebenfalls ein Ziel des Lernmoduls ist es, die Heterogenität innerhalb einer Lerngruppe zu fördern und das Lernmodul so aufzubauen, dass dieses von Lerngruppen mit unterschiedlichen Voraussetzungen (Lerntempo, Leistungsniveau) durchgeführt werden kann.

4.2. Struktur des CAMMP days

Für die komplette Durchführung des Modellierungstags werden etwa 4 Zeitstunden eingeplant. Aufgeteilt wird diese Zeit in eine 45 minutige Einführung sowie 15 minutige Nachbesprechung mit Evaluation und einer 3 stündigen aktiven Phase, in der die SuS die konzipierten Materialien des Lernmoduls bearbeiten. Folgender Ablauf ist für den CAMMP day vorgesehen:

1. **Begrüßung** - 10 Minuten
   Die SuS erhalten das anstehende Programm für den Modellierungstag und werden von dem CAMMP Team begrüßt. Die Betreuer stellen sich kurz vor und stellen anhand einer kleinen Präsentation das Konzept sowie die unterschiedlichen Veranstaltungsangebote von CAMMP vor.

2. **Modellierungsvortrag** - 20 Minuten
   Der anschließende Modellierungsvortrag, der von einem Professor oder Doktoranden des Instituts gehalten wird, soll den SuS einen Einblick ins mathematische

Modellieren geben und ihnen eine Anwendungsmöglichkeit (z.B. Strahlentherapie oder erneuerbare Energien) für Modellieren zeigen.

3. **Einführungsvortrag** - 15 Minuten
Im Einführungsvortrag, der von den Betreuern gehalten wird, wird die Smartphone-App Shazam und die Problemstellung für das Lernmodul vorgestellt. Die SuS erhalten einen Überblick über den Erfolg von Shazam sowie über die grundlegenden Ideen, die hinter der Smartphone-App stecken. Außerdem wird ein Experiment zur Darstellung eines Stimmgabeltons vorgeführt. Am Ende des Vortrags wird mit den SuS das Tagesprogramm besprochen.

4. **Erstes Arbeitsblatt: Töne mathematisch modellieren** - 75 Minuten
Im ersten Arbeitsblatt sollen die SuS lernen, wie Töne mathematisch beschrieben werden können. Anhand einer gegebenen Sinusschwingung beobachten die SuS, wie sich die Veränderung der Amplitude und der Frequenz als Variablen, die aufgabenbezogen von den SuS variiert werden können, akustisch und graphisch auf den gegebenen Sinuston auswirkt. Am Ende des Aufgabenblattes erstellen die SuS ihr eigenes Audiosignal. Ziel am Ende des Lernmoduls wird es sein, die erstellten Audiosignale der einzelnen Gruppen unter Zuhilfenahme des Shazam-Algorithmus zu identifizieren.

5. **Zweites Arbeitsblatt: Fourieranalyse** - 45 Minuten
In diesem Arbeitsblatt werden die SuS an die Fourieranalyse herangeführt, welche das mathematische Werkzeug zur Bestimmung des Frequenzspektrums des Audiosignals darstellt. Dazu sollen die SuS die Frequenzen der Teiltöne aus den erstellten Dreiklängen der anderen Gruppen heraus finden und später erkennen, dass die Überlagerung der Teiltöne wieder exakt den gleichen Dreiklang ergibt.

6. **Mittagspause**

7. **Drittes Arbeitsblatt: Modellentwicklung für Shazam** - 30 Minuten
Anhand eines Beispiel-Spektogramms sollen sich die SuS die einzelnen, mathematischen Modellschritte (Erstellung des genetischen Fingerabdrucks und Durchsuchung der Datenbank) hinter Shazam verdeutlichen und ihr erhaltenes Spektogramm einem Song in der programmierten Datenbank zuordnen. Die SuS werden jedoch kein eindeutiges Ergebnis erhalten.

8. **Zwischenvortrag** - 10 Minuten
An dieser Stelle erfolgt eine kurze Rekapitulation bzw. Zusammenfassung der bisherigen Modellschritte. Anschließend diskutieren die SuS, warum sie mit ihrem bisherigen Modell kein eindeutiges Ergebnis erzielen und nennen mögliche Modellverbesserungen. Eine wichtige Verbesserung, die Betrachtung der Zeitdifferenz zwischen Aufnahme und gespeicherten Songs in der Datenbank, wird den SuS genauer vorgestellt.

9. **Viertes Arbeitsblatt: Modellverbesserung** - 30 Minuten
   Im letzten Arbeitsblatt betrachten die SuS die Zeitdifferenz zwischen ihrer Beispielaufnahme vom Arbeitsblatt drei und den zwei in Frage kommenden Songs. Damit sollen sie ihre Aufnahme eindeutig zuordnen können. Zum Schluss überprüfen die SuS das entwickelte Modell, indem sie die erstellten Audiosignale anderer Gruppen (von Aufgabenblatt eins) identifizieren.

10. **Verabschiedung** - 15 Minuten
    Die Betreuer verabschieden die SuS und stellen ihnen Informationen zum Studiengang CES und zu weiteren Angeboten von CAMMP vor. Zum Schluss werden den SuS die Evaluationsbögen ausgeteilt.

4.3. Vorstellung der erstellten Materialien

Für das entwickelte Lernmodul wurden zwei Präsentationen und vier Arbeitsblätter mit zugehörigen, arbeitsprozessbegleitenden Matlab-Skripten entwickelt. Die Präsentationen gliedern sich in eine Einführungspräsentation zur Problembeschreibung und eine Zwischenpräsentation zur Sicherung und Erweiterung der Modellschritte. Um das unterschiedliche Leistungsniveau der SuS zu berücksichtigen sowie eigenständiges Lernen zu fördern, wurden für ausgewählte Aufgaben Hilfekarten erstellt. Den Betreuern stehen für die Einarbeitung in die Thematik weiterführende Materialien (Basic Paper, Methodisches Konzept) sowie Lösungen zu den einzelnen Arbeitsblättern zu Verfügung. Das methodischen Konzept beschreibt den genauen Ablauf des Modellierungstages, erläutert den Einsatz der Schülermaterialien und gibt den Betreuern Handlungsanweisungen. Weiter erhalten die Betreuer vorgefertigte Folien zur Besprechung und Ergebnissicherung. Darüber hinaus benötigen die SuS Kopfhörer für die Bearbeitung der Arbeitsblätter.
Alle erstellten Materialien zum entwickelten Lernmodul sind, bis auf das Basic Paper, im Anhang zu finden. Dabei wird zwischen der Version vor der ersten Durchführung und einer überarbeiteten Version nach der Evaluation unterschieden. Im Folgenden soll die erste Version der Schülermaterialien vorgestellt werden. Auf die Überarbeitung der Materialien wird später in der Evaluation genauer eingegangen.

4.3.1. Modellierungsvortrag

Der Modellierungsvortrag wurde bereits erstellt und musste nicht mehr für das Lernmodul entwickelt werden. Da er jedoch ein wichtiger Bestandteil des Modellierungstags ist, soll er hier kurz beschrieben werden. Der Vortrag wird von einem Professor oder Doktoranden, der an dem CAMMP Projekt beteiligt ist, vorgestellt und soll den SuS einen ersten Eindruck vermitteln, was mathematisches Modellieren ist. Dazu stellt der Vortragende ein aktuelles Forschungsthema vor und erläutert die einzelnen Schritte des Modellierungsprozesses am Modellierungskreislauf. Somit sehen die SuS gleichzeitig eine wichtige Anwendungsmöglichkeit von mathematischem Modellieren in der Forschung und werden so auf die zentrale Rolle, die Modellierung im Bereich Forschung, Industrie und Wissenschaft spielt, aufmerksam gemacht.

4.3.2. Einführungsvortrag

Zu Beginn der Einführungspräsentation wird den SuS zunächst die Smartphone App Shazam vorgestellt. Dazu führt der Betreuer ihnen vor Ort die App vor, indem er mit der App auf seinem Smartphone ein Musikstück identifiziert. Somit wird sichergestellt, dass auch die SuS, die zuvor noch nicht mit Shazam in Berührung gekommen sind, einen Eindruck von der Musikerkennungs-App erhalten. Anschließend bekommen die SuS Informationen über die Gründung und den Erfolg von Shazam. So wird ihnen zum Beispiel erzählt, das mehr als vier Millionen mal am Tag die Datenbank abgefragt wird sowie mehr als 100 Millionen Benutzer die App monatlich benutzen und Shazam somit Marktführer im Bereich Musikerkennung ist. Diese Informationen sollen das Interesse und die Motivation der SuS für das Thema des Lernmoduls wecken.
Anschließend werden die grundlegenden Schritte zum Bestimmen eines Musikstückes erläutert sowie das Lernziel, die Erstellung des akustischer Fingerabdrucks und anschließende Durchsuchung der Datenbank zu verstehen, definiert. Dazu erhalten die SuS zunächst Informationen über den Aufbau eines Audiosignals. Sie erfahren, dass jedes Audiosignal aus Teiltönen besteht, die für jedes Signal charakteristisch sind und deren Überlagerung wieder das ursprüngliche Audiosignal ergibt. Hier werden nun die SuS in den Vortrag mit eingebunden. Sie sollen gemeinsam überlegen, wie die Teiltöne mathematisch beschrieben werden können und welche charakteristischen Eigenschaften ein Teilton hat. Nachdem der Betreuer die Ideen der SuS gesammelt hat, hält er die charakteristischen Eigenschaften eines Teiltons (Amplitude, Frequenz, Zeit) fest und verdeutlicht den SuS, dass Shazam genau diese Informationen benötigt, um den akustischen Fingerabdruck eines Musikstücks zu erstellen. Anschließend zeigt der Betreuer den SuS, wie mit diesen Eigenschaften der Teilton mathematisch beschrieben werden kann.
Ein zentrales Element des Vortrags ist die Vorführung eines Experiments, welches auf das erste Arbeitsblatt vorbereiten und den SuS die Auswirkung der Veränderung von Amplitude und Frequenz auf den Ton demonstrieren soll. Dazu nimmt der Betreuer mit der Smartphone-App *phyphox* verschiedene Töne zweier Stimmgabeln auf. Das resultierende, aufgenommene Schwingungsbild der App kann gleich auf den Computer übertragen und somit den SuS ganz einfach über den Beamer gezeigt werden. Als Erstes stimmt der Betreuer die gleiche Stimmgabel einmal laut und einmal leise an, um die Veränderung der Amplitude zu beobachten. Anschließend werden nacheinander zwei unterschiedliche Stimmgabeln angestimmt, um zu erkennen, dass sich bei einer Erhöhung der Frequenz die Periode, also der Abstand zwischen zwei gleichen Maxima, verringert. Optional kann hier auch mit beiden Stimmgabel die Überlagerung zweier Sinustönen gezeigt werden.
Nach der Durchführung des Experiments wird den SuS kurz die Fourieranalyse vorgestellt, als das mathematisches Werkzeug, welches die einzelnen Teiltöne mit verschiedenen Frequenzen und zugehörigen Amplituden aus einem Audiosignal herausfindet. Zum

besseren Verständnis, was die Fourieranalyse bewirkt, wird den SuS eine Analogie zur Zerlegung des Sonnenlichts in seine Spektralfarben aufgezeigt. Es wird ihnen gezeigt, dass ein Prisma das Sonnenlicht in seine einzelnen Farbbestandteile zerlegen kann, genau wie die Fourieranalyse ein Eingangssignal in seine einzelnen Teiltöne zerlegt und deren Frequenzen mit zugehörigen Amplituden in ein Frequenzspektrum einträgt.
Der Abschluss der Präsentation bildet der Modellierungskreislauf, der den SuS als Orientierungshilfe im Verlauf des Modellierungstages dienen soll und gleichzeitig den Tagesablauf des Lernmoduls festlegt. Dabei wird das Erkennen von Musikstücken als reales Problem festgehalten und das Untersuchen einer kleinen Beispielaufnahme als vereinfachtes Problem. Weiter wird den SuS mitgeteilt, dass für die Aufstellung des mathematischen Modells Vorwissen über die Modellierung von Teiltönen sowie der Fourieranalyse benötigt wird, weshalb diese Themen als Erstes bearbeitet werden.

4.3.3. Arbeitsblatt 1 - Töne mathematisch modellieren

Ziel dieses Arbeitsblattes ist es, ein Verständnis für die Amplitude und die Frequenz sowie für ihre Auswirkung auf das Klangbild und den graphischen Verlauf der Schwingung zu bekommen. Dazu erhalten die SuS zu Beginn eine kurze Zusammenstellung, wie ein Sinuston mathematisch aufgebaut ist. Somit haben die SuS den Aufbau einer Sinusschwingung jederzeit vor Augen und können sich bei aufkommenden Fragen während des ersten Arbeitsblattes daran orientieren.
In einer anschließenden kurzen Aufgabenbeschreibung wird das Lernziel definiert und die Rahmenbedingungen vorgestellt. Für die Bearbeitung der Aufgaben arbeiten die SuS mit dem *Matlab*-Skript *Tonmodell.m*. Dabei dient das *Matlab*-Skript den SuS hauptsächlich zur Überprüfung ihrer Ergebnisse. So können sie sich die vorgenommenen Änderungen in der Amplitude oder der Frequenz gleich akustisch abspielen und graphisch anzeigen lassen. Dazu geben sie nach der Eingabe ihres Ergebnis im Command Window den Befehl $ckeck\_A_i$ bzw. $check\_F_i$ ein.
In der ersten Aufgabe des Arbeitsblattes variieren die SuS die Amplitude des vorgegeben Sinustons. Dazu hören sie sich zunächst mit dem Befehl *Sinuston* den gegebenen Sinuston an und lassen sich ihn graphisch anzeigen. Die SuS sollen sich nun überlegen, wie sie die Amplitude wählen müssen, damit der Ton lauter bzw. leiser klingt oder sich die Lautstärke innerhalb eines vorgegebenen Zeitfensters ändert. Für SuS, die beim Wählen einer zeitabhängigen Amplitude Schwierigkeiten haben, wurde eine Hilfekarte entwickelt, die bei Bedarf von dem Betreuer ausgeteilt wird. Auf dieser Hilfekarte finden die SuS in Form von gezeichneten, linearen Funktionen Anregungen zum Lösen dieser Aufgaben. Mit der Hilfekarte wird gewährleistet, dass die SuS auch bei Schwierigkeiten die Möglichkeit haben weiterhin selbstständig zu arbeiten.
Im zweiten Teil des Arbeitsblattes variieren die SuS anlog zum Ersten die Frequenz des vorgegeben Sinustons. Hier sollen sie sich zunächst überlegen, mit welchem Faktor die vorgegeben Frequenz multipliziert werden muss, damit der Ton höher bzw. tiefer klingt. Mit dem Befehl $check\_F1$ erhalten die SuS in einer Abbildung den Graphen des vorgegebenen Sinustons und den des neuen Tons. Sie sollen beobachten, dass sich die Anzahl an Maxima in einem festen Intervall um den eingegeben Faktor ändert.

Anschließend überlegen sich die SuS, wie sich die Frequenz zeitlich verändern lässt.

Die dritte Aufgabe des Arbeitsblatts bildet das Grundgerüst bzw. den didaktischen Rahmen für das entwickelte Lernmodul. Hier erstellen die SuS in ihren Zweiergruppen ein eigenes Audiosignal. Den Songtitel und Interpreten können sie frei wählen. Anschließend speichern sie das Audiosignal unter ihrer Gruppennummer ab, die sie auf ihrem Platz vorfinden, und kopieren die Datei in den Ordner *DataBase* in die *Dropbox*. Die Einbindung von *Dropbox* ermöglicht einen schnellen Austausch zwischen den einzelnen Gruppen sowie den Betreuern und wird in diesem Lernmodul des Öfteren verwendet. Der Betreuer fügt anschließend alle erstellten Audiosignale der Datenbank hinzu. Die genaue Anweisung dafür findet der Betreuer in dem methodische Konzept. Das Ziel am Ende des Lernmoduls wird es sein, dass die SuS den Songtitel und Interpreten anderer Gruppen mit Hilfe des entwickelten Modells für Shazam herausfinden.

Die Sicherung der erarbeiteten Ergebnisse erfolgt durch eine Besprechung der Aufgaben im Plenum. Dazu wird das *Matlab*-Skript *Tonmodell.m* über den Beamer angezeigt und die Lösungen der SuS gesammelt. Die jeweiligen Ergebnisse trägt der Betreuer in das *Matlab*-Skript ein und spielt sie dem Plenum vor. Das abgespielte Signal und der erscheinende Graph wird anschließend mit den SuS diskutiert.

4.3.4. Arbeitsblatt 2 - Fourieranalyse

Das zweite Arbeitsblatt soll den SuS die grundlegende Idee hinter der Fourieranalyse vermitteln sowie ihnen aufzeigen, dass ein Audiosignal aus einzelnen Teiltönen besteht, welche überlagert wieder das Ausgangssignal ergeben. Es startet mit einem kleinen Infotext, der die Frage klärt, warum Shazam die Fourieranalyse als mathematisches Werkzeug zur Musikerkennung überhaupt braucht. Dieser Infotext verdeutlicht den SuS die Relevanz, die die Fourieranalyse für das Lösen des Problems Musikstücke zu erkennen hat. Auch werden die einzelnen Schritte der Fourieranalyse anhand einer Abbildung nochmals verdeutlicht. So wird gewährleistet, dass jede Schülerin und jeder Schüler den Ablauf der Fourieranalyse vor Augen hat. Für die Bearbeitung des Arbeitsblatts arbeiten die SuS mit dem *Matlab*-Skript *Akkordtest.m*, welches später die Fourieranalyse für sie durchführen wird.
Bevor die SuS jedoch die Fourieranalyse anwenden, erstellen sie sich zunächst in ihren Zweiergruppen einen eigenen Dreiklang. Dazu erhalten sie ein Infoblatt mit zwei Tabellen. Eine Tabelle gibt den SuS verschiedene Dur- und Moll-Dreiklänge an. Die andere Tabelle enthält Information über die Frequenzen der einzelnen Töne. So kann sich jede Gruppe einen Dreiklang mit den benötigten drei Frequenzen aussuchen. Anschließend geben sie ihren Dreiklang in der Form $\sin(f_1 \cdot 2\pi \cdot t) + \sin(f_2 \cdot 2\pi \cdot t) + \sin(f_3 \cdot 2\pi \cdot t)$ in das *Matlab*-Skript ein und speichern den erstellten Dreiklang unter ihrer Gruppennummer ab. Die Datei kopieren die SuS wieder in die *Dropbox*, diesmal in den Ordner *Dreiklaenge*. Haben alle Gruppen ihre Dreiklänge in den Ordner geladen, kopieren sich

alle Gruppen den Ordner in den Ordner *Arbeitsblatt 2* auf ihren Desktop. So können die SuS auf die einzelnen, erstellten Dreiklänge der anderen Gruppen zurückgreifen.
Ziel der zweiten Aufgabe ist es nun, mit Hilfe der Fourieranalyse die Frequenzen der Dreiklänge anderer Gruppen herauszufinden. Dazu können die SuS im *Matlab*-Skript die Gruppennummer eintragen, deren Dreiklang sie gerne untersuchen möchten. Zunächst hören sie sich den Dreiklang an und lassen sich ihn in einem Graph anzeigen. Mit der Eingabe *Fourieranalyse* im Command Window erhalten die SuS das Frequenzspektrum des Dreiklangs und bekommen die drei Frequenzen im Command Window angezeigt. Mit diesen Frequenzen können sie dann auf die Töne und auf den erstellten Dreiklang der Gruppe schließen. Die erhaltenen Ergebnisse tragen die SuS in ihrer Tabelle auf den Arbeitsblatt ein. Auch sollen die SuS ihre Ergebnisse in einer Tabelle auf dem Overheadprojektor eintragen, sodass am Ende die Dreiklänge jeder Gruppe identifiziert werden. Dies spart Arbeitszeit und stellt sicher, dass am Ende alle SuS ihre gefunden Dreiklänge überprüfen können.
In der dritten Aufgabe sollen die SuS dann den umgekehrten Weg gehen. Für einen identifizierten Dreiklang bilden sie jeweils für die einzelnen, bestimmten Frequenzen die Sinustöne und schauen sich die drei Sinusschwingungen in einem Graphen an. Anschließend bilden sie die Überlagerung der drei Sinustöne. Mit der Eingabe des Befehls *Summe* wird den SuS zunächst der identifizierte Dreiklang abgespielt und graphisch angezeigt. Anschließend wird der Ton, der sich aus den drei Sinustönen zusammensetzt, abgespielt und der passende Graph ebenfalls angezeigt. Bei einem Vergleich der beiden Graphen sollen die SuS erkennen, dass beide identisch sind.
Die SuS haben am Ende folgenden Kreislauf einmal durchlaufen: Ein Audiosignal, wie das eines Dreiklangs, besteht aus einzelnen Teiltönen, deren Frequenzen mit der Fourieranalyse bestimmt werden können, und die Überlagerung der einzelnen Teiltönen ergibt wieder das Eingangsaudiosignal.
In einer Zusatzaufgabe können schnelle SuS beobachten, dass eine Änderung der Amplitude der einzelnen Teiltöne keine Auswirkung auf die Frequenzen im Frequenzspektrum der Fourieranalyse hat.
Die Besprechung der Ergebnisse erfolgt analog zum ersten Arbeitsplatt mit einer Diskussion im Plenum. Dabei führt der Betreuer die einzelnen Schritte zum zweiten Arbeitsblatt im *Matlab*-Skript vor. Diese Eingabe wird mit dem Beamer übertragen.

4.3.5. Arbeitsblatt 3 - Ein Modell für Shazam

In einem Einführungstext wird zunächst die grundlegende Idee für die Musikerkennung vorstellt. Denn in diesem Arbeitsblatt entwickeln die SuS ein erstes Modell zum Erkennen von Musikstücken. Sie sollen einmal per Hand die einzelnen Schritte zur Musikerkennung nachvollziehen, die sonst Shazam unsichtbar in Sekundenbruchteilen vollzieht. Dazu erhalten die SuS ein ausgedachtes Spektogramm einer Beispielaufnahme mit 20 Datenpunkten, zu dem sie den passenden Song in einer bereits programmierten Datenbank finden sollen. Mit diesem Spektogramm durchlaufen die SuS die einzelnen Schritte zum Erstellen des akustischen Fingerabdrucks und Durchsuchen der Datenbank.

Dazu nummerieren die SuS im ersten Schritt die Datenpunkte im Spektogramm durch und zeichnen die einzelnen *Target Zones* sowie die *Anchor Points* ein. Der Aufbau der *Target Zones* weicht hier bewusst von der beschriebenen Darstellung im theoretischen Hintergrund (Basic Paper) ab. Durch die Reduktion der *Target Zones* wird sichergestellt, dass die Anzahl der Adressen für die SuS überschaubar bleibt und die Aufgabe nicht als monoton empfunden wird. Während der Bearbeitung wird eine Schülerin bzw. ein Schüler gebeten, die *Target Zones* mit *Anchor Points* im Spektogramm auf dem Overheadprojektor einzuzeichnen. So wird sichergestellt, dass für die Bearbeitung des zweiten Schritts alle SuS den gleichen Ausgangspunkt haben.
Im zweiten Schritt erhalten die SuS die Information, wie Shazam die Adresse für einen Datenpunkt abspeichert. In einer lückenhaften Tabelle finden die SuS die einzelnen Adressen für die Datenpunkten im Spektogramm aufgelistet. Sie sollen die fehlenden Felder in der Tabelle ausfüllen und diese Werte ebenfalls im *Matlab*-Skript *Shazam_Modell.m* eingeben. Mit dem Befehl *Datenbank_Durchsuchen* können sie anschließend *Matlab* die Datenbank durchsuchen lassen. Die SuS erhalten kein eindeutiges Ergebnis, da zwei Songs (Song02 und Song05) eine Übereinstimmung von 100 % mit den losgeschickten Adressen der Beispielaufnahme aufweisen, wie Abbildung 24 zeigt. Dieses Ergebnis wird im anschließenden Zwischenvortrag diskutiert.

[100]	'Song05,'	'Artist05'
[100]	'Song02,'	'Artist02'
[73.3333]	'Song04,'	'Artist04'
[23.3333]	'Song01,'	'Artist01'

Abbildung 24: Das resultierende Ergebnis der Datenbankdurchsuchung, welches *Matlab* ausgibt. Song02 und Song05 haben jeweils eine Übereinstimmung von 100 % mit der Beispielaufnahme.

4.3.6. Zwischenvortrag

Der Zwischenvortrag verfolgt zwei Ziele. Einerseits soll er die bisherigen Modellschritte sichern und andererseits das resultierende Ergebnis aus Arbeitsblatt 3 sowie mögliche Modellverbesserungen diskutieren. Zunächst werden die einzelnen Modellschritte, welche die SuS auf ihre Beispielaufgabe angewendet haben, aufgezählt und anschließend ihr resultierendes Ergebnis angezeigt. Es folgt die Einblendung des Modellierungskreislaufs, an dem den SuS aufgezeigt wird, dass sie nun an dem Punkt angekommen sind, an dem ihr Ergebnis interpretiert und validiert werden muss. Sie stellen fest, dass ihre Modellannahmen zu einfach waren, sie eine Modellverbesserung anstreben und mit dieser den Modellierungskreis erneut durchlaufen müssen. Die Verbesserungsvorschläge der SuS werden vom Betreuer gesammelt, an der Tafel festgehalten und diskutiert. Anschließend stellt der Betreuer den SuS eine wichtige Modellverbesserung, die Betrachtung der Zeitdifferenz zwischen Aufnahme und Musikstücken in der Datenbank, genauer vor. Er zeigt ihnen anhand von zwei Beispielen, dass eine 100%ige Übereinstimmung nicht bedeuten muss, dass die Reihenfolge der Adressen bzw. Datenpunkte der Musikstücke in der Datenbank mit der Reihenfolge in der Aufnahme übereinstimmt. Zur Vorbereitung auf das letzte Arbeitsblatt wird den SuS die Idee vorgestellt, wie Shazam die Zeitdifferenz und damit die Reihenfolge der Datenpunkte überprüft.

4.3.7. Arbeitsblatt 4 - Modellverbesserung

Im letzten Arbeitsblatt sollen die SuS mit Hilfe der Modellverbesserung den passenden Song zu ihrer Beispielaufgabe herausfinden. Dazu arbeiten sie weiter mit dem *Matlab*-Skript *Shazam_Modell.m*. Mit den Eingaben *Treffer_Song05* und *Treffer_Song02* im Command Window erhalten die SuS zu ihren losgeschickten Adressen die zugehörigen Zeitpunkte der *Anchor Points* in Song05 und Song02. In der Tabelle vom dritten Aufgabenblatt stehen die Zeiten der Anchor Points zu den Adressen der Aufnahme. Mit diesen Informationen können die SuS die Zeitdifferenzen für die einzelnen Adressen überprüfen und herausfinden, bei welchem der beiden in Frage kommenden Songs die Zeitdifferenz konstant bleibt. Auf diese Weise können die SuS am Ende den passenden Song zu ihrer Beispielaufnahme herausfinden. Ihr Ergebnis können sie mit der Eingabe *RTS* im *Matlab*-Skript überprüfen, die ihnen den richtigen Song anzeigt.
Die letzte Aufgabe nimmt Bezug auf das erste Arbeitsblatt und soll so das Lernmodul didaktisch abrunden bzw. abschließen. Hier sollen die SuS das entwickelte Modell überprüfen, indem sie die Songtitel und Interpreten der erstellten Audiosignale der anderen Gruppe vom ersten Arbeitsblatt identifizieren. Dazu kopieren sie sich den Ordner *DataBase* von der *Dropbox* in den Ordner *Arbeitblatt 3 & 4* auf ihren Desktop, sodass sie auf die vom Betreuer erstellte Datenbank zugreifen können. Die SuS können nun im *Matlab*-Skript *Shazam_Modell.m* die Gruppennummer eingeben, deren Songtitel sowie Interpret sie herausfinden wollen, und mit dem Befehl *Ton_Suchen* die Suche in der Datenbank starten. *Matlab* gibt dann im Command Window den passenden Songtitel und Interpreten heraus. Diese Informationen tragen die SuS in ihrer Tabelle auf dem Arbeitsblatt und in die Tabelle auf dem Overheadprojektor ein. Diese letzte Aufgabe soll dem Lernmodul einen didaktischen Rahmen verleihen, indem die SuS zum Schluss genau wie die Shazam-App, Audiosignale bestimmen können.

4.4. Matlab als digitales Werkzeug

Die computergestützte Software *Matlab* begleitet die SuS durch das gesamte Lernmodul und soll sie bei der Bearbeitung des Problems unterstützen. Dazu wurde für jede Aufgabe ein *Matlab*-Code entwickelt.
Durch den Einsatz der Computersoftware wird der Fokus des Lernmoduls auf den Aufbau der Modellierungskomopetenz sowie auf das problemorientierte Bearbeiten der Aufgaben gesetzt, da die SuS von der Rechenarbeit entlastet werden. Denn die SuS sollen in dem Lernmodul die Mathematik als Werkzeug betrachten, mit der sie die Problemstellung lösen können. Weiter würde der Rechenaufwand, eine Fourierreihe per Hand zu lösen oder die Durchsuchung der Datenbank nach dem passenden Song

zur Beispielaufnahme, ohne den Computereinsatz zeitlich den Rahmen des CAMMP days sprengen.
Weiterführend dient *Matlab* für die SuS als eine Kontrollinstanz (siehe Abbildung 25) und zur Illustration (siehe Abbildung 26). Eine wichtige Funktion von *Matlab* in diesem Lernmodul ist die Fähigkeit, erstellte Sinusschwingungen bzw. auch komplexere Audiosignale akustisch sowie graphisch darzustellen, wie beispielsweise der Einsatz der Check-Befehle im *Matlab*-Code zum ersten Arbeitsblatt.

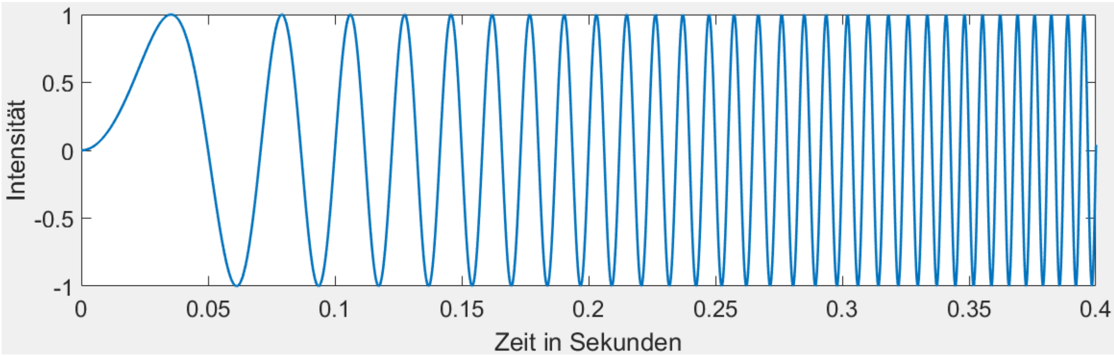


Abbildung 25: *Matlab* als Kontrollinstanz zur Überprüfung von Ergebnissen. Hier die visuelle Bestätigung aus Arbeitsblatt 1, dass der Ton mit der Zeit höher wird.

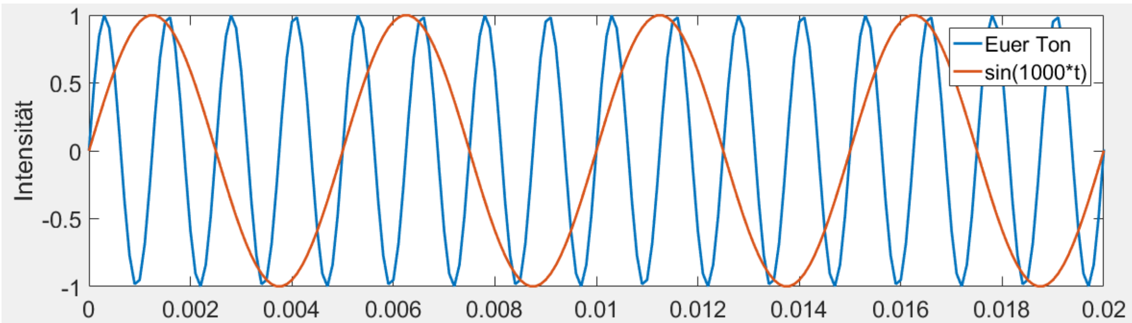


Abbildung 26: *Matlab* als Illustration von Eigenschaften. Hier aus Arbeitsblatt 1 das Verhalten der Periode bei Multiplikation der Frequenz um den Faktor 4.

Da die wenigsten SuS im Umgang mit *Matlab* geübt sind, wurde darauf geachtet, den Code so einfach, übersichtlich und intuitiv wie möglich zu gestalten. Dazu wird mit Hilfe von Kommentaren jeder Abschnitt im Code einer Aufgabe auf dem Arbeitsblatt zugeordnet. Auch dokumentiert das *Matlab*-Skript, an welchen Stellen die SuS im Code ihre Eingaben für die Lösungen tätigen sollen. An den Stellen im Code, an denen die SuS eigene Formeln eingeben müssen, steht der Ausdruck NaN (*Not a Number*), den sie durch ihre Lösung ersetzen. Im Folgenden wird ein kleiner Auszug aus dem Skript *Tonmodell.m*, welches die SuS zur Bearbeitung des ersten Arbeitsblatts verwenden, als Beispiel zum Aufbau der einzelnen *Matlab*-Skripte gezeigt:

```
%% Aufgabe 1     Wie kann ich die Lautstärke eines Tons verändern?

% c)       Lautstärke lauter werden lassen

% Amplitude für lineare Zunahme der Lautstärke:
A2_linear = NaN;


% Amplitude für schnellere Zunahme der Lautstärke:
A2_fix = NaN;

% d)       Lautstärke leiser werden lassen

% Intervall [a,b] in dem der Graph angezeigt wird:
a = NaN;
b = NaN;

% Amplitude:
A3 = NaN;
```

Anhand dieses Beispiels wird der einem Lückentext ähnliche Aufbau deutlich, der den SuS den Umgang mit *Matlab* erleichtern soll. Zudem erhalten die SuS auf den Arbeitsblättern detaillierte Informationen (teilweise mit Zeilenangaben) wie sie ihre Lösung in den *Matlab*-Code eingeben und wie sie die Ausführung von Befehlen starten.

5. Durchführung und Evaluation des Lernmoduls

Im Rahmen dieser Bachelorarbeit wurde das Lernmodul mit einer Schülergruppe durchgeführt, um einerseits die Reaktion der SuS auf den entwickelten CAMMP day zu testen und anderseits notwendig erscheinende Verbesserungen vorzunehmen. Im Folgenden werden die Ergebnisse der Evaluation und die daraus resultierenden Verbesserungen vorgestellt.

5.1. Die Schülergruppe

Der CAMMP day wurde am 15. Juli 2016 von der Schüleruni der RWTH Aachen durchgeführt mit SuS in der Qualifikationsphase 1 und 2. Die Schülergruppe bestand aus 18 Teilnehmern, aufgeteilt in 10 männliche und 8 weibliche Schüler. Von den 18 Teilnehmern haben 9 eines von den Fächern Mathematik, Informatik oder Physik als Leistungskurs in der Schule belegt.

5.2. Leitende Gesichtspunkte der Evaluation

Zur Durchführung der Evaluation wurde nach dem Ablauf des CAMMP days den SuS ein für das Lernmodul entwickelter Evaluationsbogen ausgeteilt (siehe Anhang). Sowohl die Fragen des Evaluationsbogens als auch meine Beobachtungen orientieren sich dabei an verschiedenen inhaltlichen Themenschwerpunkten. Diese werden nachfolgend benannt und kurz erläutert.

- **Interesse**:
  Hat die Thematik des Lernmoduls das Interesse bei den SuS geweckt und sie für den CAMMP day begeistern können?
- **Verständnis**:
  Haben die SuS am Ende des Lernmoduls die grundlegende Idee hinter Shazam verstanden und kennen die einzelnen Schritte der Musikerkennung?
- **Umgang mit Matlab**:
  Hat der entwickelte *Matlab*-Code zu den einzelnen Aufgaben den Umgang mit *Matlab* erleichtert? Hatten die SuS Schwierigkeiten, die Aufgaben mit *Matlab* zu bearbeiten?
- **Aufgaben**:
  Ist der Schwierigkeitsgrad unter Berücksichtigung der Heterogenität der Gruppe angemessen?
- **Modellieren**:
  Leistet das entwickelte Lernmodul einen Beitrag zum Aufbau der Modellierungskompetenz sowie zum allgemeinen Verständnis von Modellieren und ihrer Bedeutung in der Welt?

5.3. Beobachtung und Evaluation

In diesem Abschnitt soll kurz der Verlauf vorgestellt sowie die Ergebnisse der ersten Durchführung des Lernmoduls anhand der leitenden Gesichtspunkte evaluiert werden. Dabei wird sowohl auf meine persönlichen Beobachtungen während des CAMMP days als auch auf die Auswertung der von den SuS ausgefüllten Evaluationsbögen eingegangen.
Der Modellierungstag begann um 9:00 Uhr und endete um 14:20 Uhr, sodass der geplante zeitliche Rahmen von 4 Stunden ohne Berücksichtigung der Pausen eingehalten werden konnte. Der allgemeine Ablauf des CAMMP days lief reibungslos und der entwickelte *Matlab*-Code funktionierte bis auf eine Ausnahme einwandfrei. Zu Beginn des Lernmoduls erwies sich der erste *Matlab*-Befehl *Sinuston* als fehlerhaft, sodass den SuS der Sinuston bei der erste Aufgabe nicht abgespielt wurde. Jedoch konnte das Problem im Code vor Ort noch gelöst und der korrigierte Code mit Hilfe der *Dropbox* schnell an die SuS verteilt werden. Eine weitere Hürde stellte die Besprechung des zweiten Arbeitsblattes dar. Auf Grund einer älteren Version des *Matlab*-Codes auf dem Laptop des Betreuers konnte die Fourieranalyse zunächst nicht vorgeführt werden. Da dieses Problem kurz vor der Mittagspause auftrat, konnte dieses während der Pause behoben und anschließend das zweite Aufgabenblatt besprochen werden. Darüber hinaus waren keine weiteren Zwischenfälle zu vermerken, sodass man mit der ersten Durchführung des Lernmoduls sehr zufrieden sein kann.
Im Folgenden wird das entwickelte Lernmodul anhand der leitenden Gesichtspunkten evaluiert:

Interesse:
Zu beobachten war ein großes Interesse sowie eine hohe Motivation bei der Bearbeitung des Lernmoduls. Die SuS haben aufmerksam und interessiert zugehört und mit großem Eifer die Aufgaben bearbeitet. Eine Schülerin kommentierte am Ende des Modellierungstags, dass sie es super fand, sich mit einem Thema wie Shazam auseinanderzusetzen, welches man selber fast täglich benutzt. Auch die Auswertung der Evaluationsbögen zeigt das Interesse der SuS an der Thematik des Lernmoduls. So gaben 12 der 18 ein sehr hohes Interesse und 4 ein hohes Interesse an. Lediglich zwei Teilnehmer setzten ihr Kreuz auf ein niedriges Interesse am Thema Shazam. Generell war ein hohes Interesse am CAMMP Projekt zu beobachten, sodass am Ende des Tages einige SuS sich über weitere Angebote von CAMMP informiert haben.

Verständnis:
Ziel des Lernmoduls ist es, den SuS das Verständnis für Shazam zu vermitteln und ihnen die zentralen Ideen hinter dem Musikerkennungsprogramm nahezubringen. So ist es erfreulich, dass in der Evaluation 17 der 18 Teilnehmer vollkommen zustimmen, die Idee hinter Shazam vollständig verstanden zu haben. Ähnliche Resultate wurden bei der Frage nach dem Verständnis des akustischen Fingerabdrucks erzielt. Hier stimmen 16 SuS vollkommen zu, am Ende des Lernmoduls verstanden zu haben, was ein akustischer Fingerabdruck ist. Die anderen beiden Teilnehmer haben die Stufe davor

angekreuzt.
Zusammenfassend kann anhand der Evaluation festgehalten werden, dass das zentrale Ziel, den SuS die zentralen Ideen hinter Shazam zu vermitteln, als erfüllt angesehen werden kann.

Umgang mit *Matlab*:
Für die SuS ist es natürlich schwierig, Aufgaben mit einem Programm zu lösen, das sie am CAMMP day zum ersten Mal benutzen. Aus diesem Grund ist es wichtig, SuS den Umgang mit *Matlab* so einfach wie möglich zu gestalten. Denn sie sollen in *Matlab* ein Werkzeug sehen, dass ihnen das Lösen von Modellierungsaufgaben erleichtert. Aus der Evaluation ergibt sich, dass 7 SuS vollkommen zustimmen, beim entwickelten Lernmodul keine Probleme gehabt zu haben, die Aufgaben mit *Matlab* umzusetzen, sodass ihnen der Umgang mit diesem Programm nicht schwer viel. Weitere 8 Teilnehmer stimmen der Aussage zu. Nur ein Schüler bzw. Schülerin berichtet von Schwierigkeiten mit *Matlab* und mit der Lösung der Aufgaben mit dem Programm. In Anbetracht der Tatsache, dass die SuS keine Einführung und Erfahrungen in *Matlab* hatten, kann dieses Ergebnis als positiv bewertet werden. Das Ergebnis aus der Evaluation deckt sich auch mit den Beobachtungen im CAMMP day. Verständnisfragen zu *Matlab* oder zur Eingabe von Lösungen in das Programm traten nur selten auf. Trat jedoch ein Problem mit *Matlab* auf, lag es meistens daran, dass die SuS sich nicht mehr im richtigen Ordner befanden oder sie in der ersten Aufgabe den Punkt vor dem Operator vergessen hatten. Diese Probleme könnten reduziert werden, indem den SuS zu Beginn des Lernmoduls zum einen erklärt wird, warum *Matlab* beim Rechnen mit der Variabel t einen Punkt vor den Operator braucht. Zum anderen könnte den SuS ebenfalls im Vorhinein gezeigt werden, wie sie auf die Fehlermeldung, dass sich der Code nicht im Ordner befindet, umgehen.

Aufgaben:
Die Aufgaben wurden anhand mehrerer Aussagen evaluiert, die den Schülern vorgelegt wurden. Besonders positiv fällt auf, dass die SuS die Aufgaben, die Formulierungen der Aufgaben und die Lösungen als verständlich beschrieben haben, da es in diesen beiden Aussagen keine negativen Bewertungen gibt. Beiden Aussagen wird von 16 der 18 Teilnehmer voll zugestimmt. Zudem sollte der Schwierigkeitsgrad der Aufgaben im Lernmodul eingeschätzt werden. Dabei findet die Mehrheit der SuS (16 von 18) die Aufgaben nicht zu schwierig, wobei zwei der SuS der Aussage „Die Aufgaben waren zu schwierig“ zugestimmt haben. Die Aussage „Die Aufgaben waren zu einfach“ wurde dahingegen nicht so homogen beantwortet. 7 der SuS stimmen der Aussage voll zu und 4 Teilnehmer kreuzten die darunter liegende Stufe an. Weitere 7 SuS haben der Aussage nicht zugestimmt, jedoch bewertet damit die Mehrheit der SuS die Aufgaben als zu einfach. Schon während des CAMMP days fiel auf, dass einige SuS die Aufgaben schneller bearbeiten konnten als andere und es somit für diese Teilnehmer zu Leerphasen während des Lernmoduls kam. Denn bevor mit dem neuen Arbeitsblatt und damit mit dem neuen Thema begonnen werden konnte, mussten die schnellen SuS auf die Teilnehmer warten, die noch mit dem Arbeitsblatt beschäftigt waren. So schrieben

auch einige SuS unter den offenen Kritikpunkten, dass die Aufgaben zu einfach seien. Aus diesem Grund wurden weitere Zusatzaufgaben entwickelt, um die erwähnten Leerphasen für die nächsten CAMMP days zu vermeiden und die Heterogenität der Lerngruppe stärker zu berücksichtigen (siehe nächster Abschnitt).

Modellieren:
Ein weiteres Ziel des entwickelten Lernmoduls sowie von CAMMP generell ist es, den SuS ein Verständnis für die Bedeutung von Modellieren in der Welt zu geben sowie einen Beitrag zum Aufbau der Modellierungskompetenz zu leisten. In der Evaluation stimmen 13 SuS zu, durch den entwickelten Workshop ein besseres Verständnis von mathematischem Modellieren erhalten zu haben. Nur 3 Teilnehmer stimmen dieser Aussage nicht zu. Damit konnte die Mehrheit der Teilnehmer durch das Lernmodul eine Vorstellung darüber bilden, was hinter dem abstrakten Begriff des Modellierens steht.

Auf den Evaluationsbögen hatten die SuS ebenfalls die Möglichkeit, dem Betreuer sowie dem Lernmodul eine Gesamtnote in Form von Schulnoten zu geben. Die von den SuS gegeben Durchschnittsnote für den Betreuer liegt bei $1,1$ und die für das entwickelte, durchgeführte Lernmodul bei $1,4$. Insgesamt kann die erste Durchführung des Lernmoduls damit als gelungen angesehen werden.

5.4. Aus der Evaluation resultierende Verbesserungen zum entwickelten Lernmodul

Nach der Durchführung des CAMMP days ist besonders aufgefallen, dass die Heterogenität, vor allem in den ersten beiden Arbeitsblättern, nicht optimal berücksichtigt wurde. So kam es für schnelle SuS in diesen beiden Blättern zu Leerläufen, da sie auf die Fertigstellung der anderen Teilnehmern warten mussten. Aus diesem Grund wurden für diese beiden Arbeitsblätter Zusatzaufgaben entwickelt, die sich auch vom Schwierigkeitsgrad von den anderen Aufgaben abheben. So soll gewährleistet werden, dass einerseits jeder SuS das Lernminimum erreichen kann sowie anderseits schnelle SuS keine Leerphasen zwischen den einzelnen Arbeitsblättern haben und weiter gefördert werden. Die Zusatzaufgaben befinden sich nicht mit auf den Arbeitsblättern, sondern werden den SuS nach Fertigstellung der Aufgaben ausgeteilt. Zu diesen Aufgaben wurden ebenfalls Hilfekarten erstellt, um das eigenständige Arbeiten zu fördern. Im Folgenden werden die entwickelten Zusatzaufgaben vorgestellt. Die überarbeiteten Arbeitsblätter sowie erstellten Hilfekarten sind im Anhang zu finden.
Auf dem ersten Arbeitsblatt wurde die bereits bestehende Zusatzaufgabe vom Arbeitsblatt entfernt und zu den anderen Zusatzaufgaben hinzugefügt, sodass sich sie SuS zunächst auf die Aufgaben zum Erreichen des Lernziels konzentrieren. Die ersten beiden Zusatzaufgaben konzentrieren sich auf die Veränderung der Amplitude. Hier soll die Amplitude so gewählt werden, dass die Lautstärke des Tons erst abnimmt und dann wieder zunimmt. Anschließend wird der umgekehrte Fall, also erst eine Zunahme

und danach eine Abnahme der Lautstärke, betrachtet. Zur Lösung dieses Problems müssen die SuS für die Amplitude eine quadratische Gleichung ansetzen, die jeweils bestimmte Bedingungen erfüllt. Für jede der beiden Aufgaben steht den SuS eine Hilfekarte zur Verfügung. Auf diesen Hilfekarten sehen die SuS eine graphische Lösung des Problems sowie die allgemeine Scheitelpunktsformel für eine quadratische Funktion. In einer weiteren Zusatzaufgabe sollen sich die SuS überlegen, mit welchem Faktor bzw. mit welchem Term sie die Frequenz multiplizieren müssen, damit der Ton mit der Zeit ausschließlich tiefer wird. Die Schwierigkeit liegt hier darin, die Punktsymmetrie des Sinus zu betrachten. So reicht es nicht aus, eine negative Steigung oder eine umgedrehte Parabel zu wählen, da $\sin(-x) = -\sin(x)$ gilt. Die SuS müssen sich demnach eine andere Funktion überlegen, die das Problem löst. In der letzten Zusatzaufgabe sollen die SuS den Faktor so wählen, dass der Ton zunächst tiefer und anschließend wieder höher klingt.
Auf dem zweiten Arbeitsblatt wurde die bestehende Zusatzaufgabe entfernt und durch eine anspruchsvollere ersetzt. Nun besteht die Aufgabe darin, eine Fourieranalyse per Hand durchzuführen. Dazu erhalten die SuS zunächst einen kleinen Infotext der Informationen über die Fourierreihe und zur Berechnung der Koeffizienten a_0, a_n und b_n enthält. Anschließend wird den SuS die Funktionsvorschrift sowie der zugehörige Graph einer Sägezahnfunktion vorgestellt, zu der sie die Fourieranalyse durchführen sollen. Die SuS sollen die Fourierkoeffizienten ausrechnen und die Fourierreihe für die ersten fünf Summanden aufstellen. Haben die SuS die ersten fünf Summanden der Fourierreihe berechnet, können sie diese in *Matlab* eingeben. Mit der Eingabe des Befehls *Fourier* können sie dann anschließend ihr Ergebnis überprüfen. Dazu wird den SuS in einer Abbildung der Graph der Sägezahnfunktion und der Graph ihrer aufgestellten Fourierreihe angezeigt. So können die SuS einerseits beobachten, ob sich ihre gebildete Fourierreihe der Sägezahnfunktion annähert und anderseits schauen, wie zufriedenstellend das Ergebnis für fünf Summanden ist. Um ein genaueres Ergebnis zu erzielen, können die SuS weitere Summanden der Fourierreihe berechnen. Für diejenigen SuS, die Probleme bei der partiellen Integration haben bzw. mit dieser Technik in der Schule noch nicht in Berührung gekommen sind, wurde eine Hilfekarte entwickelt, die die partielle Integration vorstellt. Auch gibt es für die Werte der Cosinusfunktion zur Berechnung der ersten fünf Summanden der Fourierreihe eine Hilfekarte.

Eine weitere Verbesserung wurde bei der Erkennung der erstellten Musikstücke der SuS vorgenommen. Nachdem der Betreuer die erstellen Songs der SuS zur Datenbank hinzugefügt hat, können die SuS sich diesen Ordner aus der *Dropbox* in ihren Ordner auf den Desktop kopieren. Mit *Matlab* können die SuS anschließend den Songtitel und Interpreten der anderen Gruppen herausfinden. Jedoch ist zwei SuS während des CAMMP days aufgefallen, dass diese Informationen bereits in den *txt.-Dateien* der einzelnen Songs in der Datenbank (in dem Ordner *DataBase*) enthalten sind. Werden diese Dateien mit dem Editor geöffnet, wird einem zunächst der Songtitel sowie Interpret und anschließend die Informationen für den Song angezeigt. Aus diesem Grund muss der Betreuer die *txt.-Dateien* löschen, nachdem er die Datenbank erstellt hat. Diese Anweisung wurde in das Methodische Konzept hinzugefügt.

Auch beim Zwischenvortrag wurde in der Phase, in der die SuS Verbesserungsvorschläge nennen, eine weiter Handlungsanweisung in das methodische Konzept hinzugenommen. Hier soll der Betreuer die einzelnen Vorschläge auf einer Folie auf dem Overheadprojektor sammeln und mit den SuS gemeinsam diskutieren, bevor die Betrachtung der Zeitdifferenz vorgestellt wird. Damit soll gewährleistet werden, dass die SuS nicht das Gefühl haben, ihre Verbesserungsvorschläge werden nicht ernst genommen bzw. übergangen.
Eine weitere Verbesserung besteht in der Kürzung des zweiten Arbeitsblatts. So wurde die Tabelle zur Eintragung der herausgefundenen Dreiklänge gekürzt und den SuS die Anweisung gegeben, nur drei fremde Dreiklänge zu identifizieren. So soll monotones Arbeiten vermieden werden, zumal schon während der ersten Durchführung des CAMMP days aufgefallen ist, dass die SuS sich beim Herausfinden der einzelnen Dreiklänge mit anderen Gruppen untereinander abgesprochen haben. Außerdem werden alle herausgefundenen Dreiklänge auf der Folie auf dem Overheadprojektor gesichert, sodass sich jede Gruppe dort wiederfinden kann. Die gleiche Kürzung wurde auf dem vierten Arbeitsblatt vorgenommen, auf dem die SuS die Songtitel sowie Interpreten der einzelnen, erstellten Songs der Gruppen identifizieren sollen. Weiter wurde zur Entlastung des zweiten Arbeitsblatts die Tabelle *Übersicht der Dreiklänge* entfernt und auf das Blatt, auf dem auch die Frequenzen der einzelnen Töne stehen, transferiert. Einerseits besteht das zweite Arbeitsblatt so nur noch aus zwei Blättern, was für die SuS motivierender wirkt und die Umwelt schont. Anderseits kann so die Überlegung umgesetzt werden, bestimmte Materialien zu laminieren und dadurch ebenfalls die Umwelt durch eine geringere Menge an gedruckten Blättern zu schonen. Diese beiden Tabellen können so einlaminiert werden und dadurch in weiteren CAMMP days wieder verwendet werden. Die gleiche Überlegung könnte mit dem Frequenzspektrum umgesetzt werden. Die SuS könnten so das einlaminierte Frequenzspektrum mit Folienstiften bearbeiten, um auch dieses Material für weitere CAMMP days benutzen zu können.

5.5. Fazit

Alles in allem kann das entwickelte Lernmodul und seine erste Durchführung als gelungen angesehen werden. Der einzige Kritikpunkt, der anzuführen wäre, besteht in der Verbesserungswürdigkeit der Heterogenität und teilweise des Schwierigkeitsgrades. Dieser Kritikpunkt wurde nach der Evaluation durch das Einführen weiterer Zusatzaufgaben in Ansätzen verbessert. Positiv hervorzuheben ist die hohe Interaktivität des Lernmoduls. Einerseits durch die Einbindung der Smartphone-App *phyphox*, die den SuS anhand der Stimmgabel einen Ton und seine Eigenschaften (Amplitude und Frequenz) visualisiert und dadurch den SuS einen guten Einstieg in das Lernmodul ermöglicht. Andererseits ermöglicht *Matlab* den SuS ihre Lösungen gleich anzuhören und visuell darzustellen. Dadurch erhalten sie gleich eine Rückmeldung und können auch durch gezieltes Ausprobieren zur Lösung finden. Diese beiden Punkte sowie die Einbindung verschiedener Sinnesmodalitäten erhöht die Motivation der SuS. Weiter hat sich die Einbindung der *Dropbox* in den CAMMP day als hilfreich erwiesen. Durch

ihren Einsatz kann flexibel auf Fehler reagiert werden (z.B. Verbesserungen im Code können direkt mit allen SuS unkompliziert geteilt werden). Aber auch der Austausch von Materialien und Lösungen unter den SuS und mit dem Betreuer wird wesentlich vereinfacht.
Der entwickelte CAMMP day zeigt darüber hinaus ein hohes Potenzial für den Einsatz in niedrigeren Altersklassen, wie der Mittelstufe. Durch Weiterentwicklungen und Anpassungen kann dieses Lernmodul auf andere Altersgruppen mit niedrigerem Leistungsniveau ausgeweitet werden, sodass aus dem Lernmodul einer der ersten CAMMP days für die Mittelstufe entstehen könnte.
Insgesamt zeigt der CAMMP day viele positive Eindrücke, die sich ebenso in der Bewertung der SuS bei der Evaluation widerspiegeln.

6. Ausblick

Im letzten Abschnitt dieser Bachelorarbeit sollen Möglichkeiten zur Weiterentwicklung und Optimierung des Lernmoduls aufgezeigt werden.
Die erste Möglichkeit zur Weiterentwicklung besteht darin, das Lernmodul für die Durchführung in der Mittelstufe zu konzipieren und anzupassen. Hierzu wurden bereits die ersten beiden Arbeitsblätter im Rahmen des CAMMP days MP3 mit einer Mittelstufe erfolgreich durchgeführt. Trotzdem müssen auch die ersten beiden sowie die weiteren Arbeitsblätter im Schwierigkeitsgrad für die Mittelstufe angepasst werden. Diese Anpassung betrifft beispielsweise den Sprachgebrauch, die zeitliche Dimension, den Umfang der Arbeitsblätter und die mathematischen Vorkenntnisse. So sollte zum Beispiel über eine detailliertere Einführung der Sinusfunktion oder eine anschaulichere Methode zur Durchsuchung der Datenbank nachgedacht werden. Eine Aufgabe zur Veranschaulichung der Datenbankdurchsuchung könnte wie folgt aussehen: Die SuS erhalten einlaminierte Spektogramme von verschiedenen Songs sowie ein einlaminiertes Spektrogramm einer kurzen Aufnahme. Durch Überlagerung der Aufnahme mit den einzelnen Songs können die SuS dann den passenden Song zur Aufnahme herausfinden.
Für die Durchführung mit der Oberstufe sollte hingegen die Heterogenität stärker berücksichtigt werden sowie der Schwierigkeitsgrad etwas angehoben werden. Dies ließe sich zum Beispiel über weitere Zusatzaufgaben realisieren. Denkbar wäre ebenfalls die SuS versuchen zu lassen, Teile des Codes selber zu programmieren (Unterteilung der Aufnahme in Zeitintervalle, Finden der maximalen Amplitude, etc.), und damit die Selbstständigkeit sowie Modellierungskompetenzen der SuS weiter zu fördern.
Weiter wurde in dem entwickelten Lernmodul ein vereinfachtes Modell von Shazam betrachtet. Möglich wäre hier den SuS ein Modell vorzustellen, was sich näher an der Realität orientiert. In realen Spektogrammen befinden sich die einzelnen Frequenzen auf verschiedenen Stufen, wie Abbildung 27 zeigt.

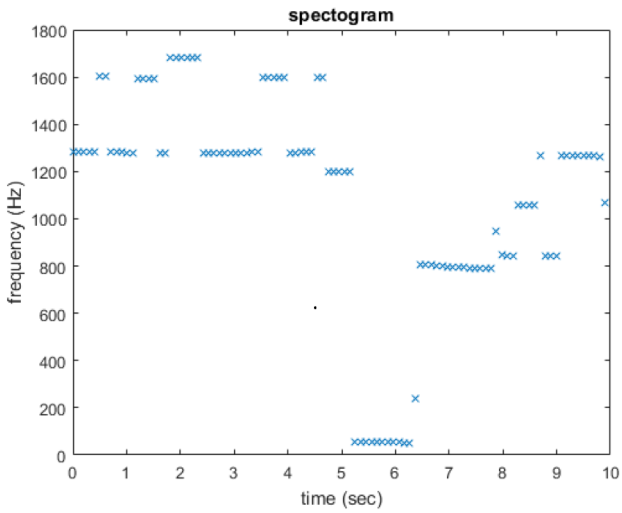


Abbildung 27: Auszug eines Orginalspektogramms eines Musikstücks, welches mit dem programmierten CAMMP day Code erstellt wurde.

Dieses Modell rechtfertigt auch die Vorgehensweise des für den CAMMP day programmierten Code, nur jeden dritten Datenpunkt als *Anchor Point* zu betrachten. Da die Aufnahme meistens nicht zeitgleich mit dem Song startet, kann man mit dem programmierten Code somit andere Anchor Points erhalten als im Song. Mit dem vereinfachten Modell, in dem die Datenpunkte als Punktwolke angeordnet sind, würden die *Anchor Points* in der Aufnahme andere Frequenzen haben und dadurch würden die Adressen nicht mehr mit dem Song übereinstimmen. Die stufenförmige Anordnung der Frequenzen gewährleistet aufgrund einer höheren Schnittmenge der gleichen Frequenzen zu einer hohen Wahrscheinlichkeit, dass die Frequenzen der *Anchor Points* in der Aufnahme ebenso in dem Song in der Datenbank wieder gefunden werden, obwohl die Aufnahme zeitlich versetzt zum Song startet. Demnach würde man trotz anderer Anchor Points viele, gleiche Adressen erhalten.

7. Anhang

A. Erste Version der Arbeitsblätter

A.1. Arbeitsblatt 1

CAMMP day

Wie funktioniert eigentlich -Shazam

-Arbeitsblatt 1-

Einen Ton mathematisch modellieren

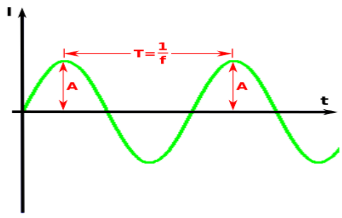


Bild 1: Sinusschwingung einer Stimmgabel

Jedes Instrument und auch die menschliche Stimme verursacht eine gewisse Art von Schwingung, auch wenn diese für normale Menschen nicht wahrnehmbar ist und nur mit Hilfe eines Computers oder eines Oszilloskops sichtbar gemacht werden kann. So ist beispielsweise der Ton einer Stimmgabel mathematisch bzw. physikalisch nichts anderes als eine harmonische Schwingung in Form einer Sinuskurve und kann mit der Sinusfunktion

$$g(t) = A \cdot \sin(f \cdot 2 \cdot \pi \cdot t)$$

modelliert werden. Dabei steht

- t für die Zeit.
- A für die **Amplitude** und gibt die maximale **Lautstärke** des Tons an. Das heißt je höher die Amplitude, desto lauter wird der Ton wahrgenommen.
- f für die **Frequenz** und gibt die **Tonhöhe** an. Der Abstand zwischen zwei gleichen Maxima wird als Periode $T = \frac{1}{f}$ bezeichnet. D.h. je größer die Frequenz f ist, desto näher liegen die die Maxima nebeneinander und umso höher wird der Ton wahrgenommen. Die Frequenz wird in Hertz [Hz] angegeben.

Aufgabenbeschreibung | Töne selber modellieren

Euer Ziel wird es sein, in die Rolle des Computers zu schlüpfen und Töne selber mathematisch zu modellieren. Das dafür benötigte Matlab-Skript `Tonmodell.m` ist schon geöffnet. Ein Ton der Form $g(t) = \sin(200 \cdot 2 \cdot \pi \cdot t)$ ist bereits in Matlab gespeichert. Die Abspieldauer des Tons ist so programmiert, dass sie $6s$ beträgt.

Aufgabe 1 | Wie kann ich die Lautstärke eines Tons verändern?

a) Höre und schaue dir zunächst den Sinuston an, indem du dazu in das Command Window `Sinuston` eingibst und auf Enter drückst. Es wird dir auch gleich der passende Graph angezeigt. Um die Sinusschwingung besser zu erkennen, wird nur das kleine Intervall $[0, 0.1]$ betrachtet. Welchen Wert hat die Amplitude dieses Tons?

Wir halten zunächst den Wert für die Frequenz konstant und wollen nur die Lautstärke, also die Amplitude A des Tons verändern. Betrachte dazu folgende Hinweise bezüglich Matlab:

Eingaben im MATLAB-Skripts `Tonmodell.m`

In Matlab muss vor jeder Operation ein Punkt gesetzt werden. Möchte man also beispielsweise eine Variabel, wie die Zeit t, quadrieren oder $\frac{1}{t}$ rechnen, muss man in Matlab $t.\hat{}2$ bzw. $1./t$ eingeben. Bei der Addition und Subtraktion gilt das gleiche. Zum Beispiel gibt man für $3 - t$ in Matlab $3. - t$ ein. **Ausnahme**: Für Eingaben der Art $t \pm 3$ darf kein Punkt gesetzt werden.

b) Je größer die Amplitude gewählt wird, desto lauter wird der Ton wahrgenommen. Überlege dir wie du die Konstante A_1 beim Ton wählen musst, damit der Ton lauter bzw. leiser klingt. Trage dazu in Matlab in Zeile 21 verschiedene Werte für $A1$ ein und überprüfe jeweils mit der Eingabe `check_A1` im Command Window dein Ergebnis. Dir wird anschließend die Schwingung des Ausgangston $g(t) = \sin(200 \cdot 2 \cdot \pi \cdot t)$ und die des neuen Tons im Intervall $[0, 0.1]$ graphisch angezeigt.

Überlege dir nun, wie du die Amplitude wählen musst, damit:

c) Die Lautstärke kontinuierlich, also linear, mit der Zeit t ausschließlich ansteigt. Trage dazu deine Term in Matlab für A2_linear ein und überprüfe wieder dein Ergebnis, indem du im Command Window den Aufruf `check_A2_linear` eingibst. Überlege dir, wie du einen schnelleren Anstieg der Lautstärke erzeugen kannst. Trage dazu deinen Term für A2_fix ein und überprüfe mit `check_A2_fix` dein Ergebnis.
Tipp: Zu dieser Aufgabe liegt die Hilfekarte 1 aus, die du bei Bedarf nutzen kannst.

d) Die Lautstärke kontinuierlich mit der Zeit ausschließlich abfällt, sodass nach 6 Sekunden der Ton nicht mehr zu hören ist. Trage dein Ergebnis für $A3$ ein. Hier kannst du nun selber das Intervall bestimmen, indem du dir den Graph des Tons anschauen willst. Trage dazu in Matlab für a den Startwert und für b den Endwert des Intervalls ein. Überprüfe nun mit `check_A3` dein Ergebnis.
Tipp: Zu dieser Aufgabe liegt die Hilfekarte 1 aus, die du bei Bedarf nutzen kannst.

Zusatzaufgabe: Die Lautstärke erst abnimmt und danach wieder ansteigt. Trage dein Ergebnis für $A4$ ein und bestimme wieder mit und $a2$ und $b2$ das Intervall, in dem ihr die Schwingung des Tons betrachten wollt. Überprüfe mit `check_A4` dein Ergebnis.

Aufgabe 2 | Wie kann ich die Frequenz eines Tons verändern?

Als nächstes wird die Lautstärke konstant gehalten und die Frequenz verändert.

a) Höre dir zunächst nochmal den Sinuston $g(t) = \sin(200 \cdot 2 \cdot \pi \cdot t)$ an, indem du im Command Window den Befehl `Sinuston` eingibst.

b) Je kleiner der Abstand zwischen zwei gleichen Maxima in der Sinusschwingung ist, desto höher wird der Ton wahrgenommen. Betrachte nun $g(t) = \sin(200 \cdot 2 \cdot \pi \cdot F1 \cdot t)$ und überlege dir wie du die Konstante $F1$ wählen musst, damit der Ton höher bzw. tiefer klingt. Trage in Matlab verschiedene Werte für $F1$, die zwischen 0 und 6 liegen, ein und überprüfe jeweils mit `check_F1` dein Ergebnis. Dir wird anschließend die Schwingung des Ausgangston $g(t) = \sin(200 \cdot 2 \cdot \pi \cdot t)$ und die des neuen Tons angezeigt.

c) Betrachte nun $g(t) = \sin(200 \cdot 2 \cdot \pi \cdot F2 \cdot t)$ und überlege dir, wie du den Faktor $F2$ wählen musst, damit der Ton mit der Zeit ausschließlich höher wird. Trage deinen Term in Matlab für $F2$ ein und überprüfe dein Ergebnis mit `check_F2`

Aufgabe 3 | Erstellt euer eigenes Audiosignal

Am Ende des Tages wird es das Ziel sein, ein erstelltes Audiosignal von einer anderen Gruppe mit Hilfe des Shazam-Algorithmus zu identifizieren. Dazu könnt ihr jetzt euer eigenes Audiosignal erstellen. In Matlab könnt ihr in Zeile 88 für y_3 den Term für euer Audiosignal eintragen. Auch die Addition von mehreren Sinusschwingungen ist hier möglich, z.B.

$$y\_3 = 3. * \sin(2000 \cdot 2 \cdot \pi. * t) + 0.5. * \sin(4100 \cdot 2 \cdot . * t.^2) + \sin(1000 \cdot 2 \cdot . * (2. - t).^2).$$

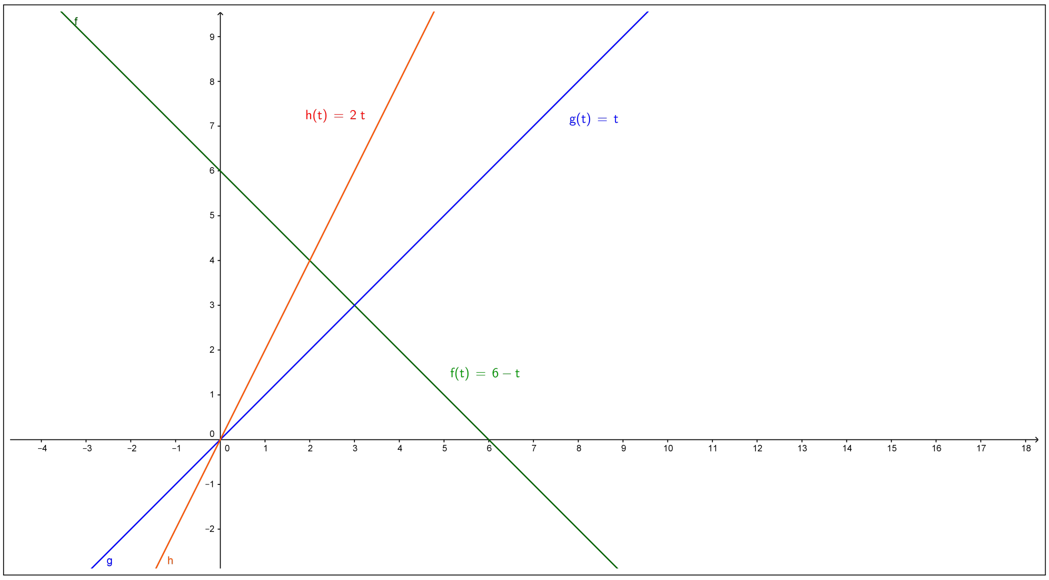
Um euer Audiosignal zu speichern, tragt ihr unter `file` in Zeile 91 für 00 eure Gruppennummer ein, die ihr auf eurem Platz vorfindet. Denkt euch nun für euer erstelltes Audiosignal noch einen Songtitel und den Namen des Interpreten aus, die ihr unter `title` und `artist` eingeben könnt. Die Intervalllänge, in der ihr euer eigenes kleines Musikstück betrachten wollt, könnt ihr wieder mit $a5$ und $b5$ festlegen. Mit der Eingabe `Eigener_Sound` wird euer Audiosignal abgespielt, graphisch angezeigt und als .txt-Datei im Ordner `Arbeitsblatt 1` unter dem Ordner `Euer_Sound` gespeichert. Ladet nun euren erstellten Ton unter dem Ordner `DataBase` in die *Dropbox*. Der Betreuer wird dann mit Hilfe des Fingerabdruck-Algorithmus euer Audiosignal zu einer Datenbank hinzufügen.

A.2. Zusatzmaterial zu Arbeitsblatt 1

A.2.1. Hilfekarten

Hilfekarte 1

Tipp: Damit sich die Amplitude mit der Zeit ändert, muss die Amplitude zeitabhängig gewählt werden.



A.3. Arbeitsblatt 2

CAMMP day

Wie funktioniert eigentlich -Shazam

-Arbeitsblatt 2-

Fourieranalyse

Bild 1: Joseph Fourier

Im ersten Arbeitsblatt haben wir bereits gesehen, dass die Amplitude und die Frequenz einen Ton mathematisch beschreiben bzw. charakterisieren. Zum Erstellen des akustischen Fingerabdrucks ist Shazam vor allem an den Frequenzen interessiert, die im Musikstück oder in der kurzen Aufnahme enthalten sind, da die Abfolge der Frequenzen für jeden Song einzigartig sind. Aus den Frequenzen erstellt Shazam mit einem bestimmten Algorithmus ein für jeden Song charakteristisches Spektogramm, aus dem anschließend der Fingerabdruck mathematisch gebildet wird. Die Frage, die sich jedoch zunächst stellt, ist, wie Shazam aus den Musikstücken und den Aufnahmen, die aus ganz vielen, unterschiedlichen Tönen bestehen, die einzelnen Frequenzen zum Erstellen der Spektogramme erhält.

Problembeschreibung | Überlagerung von Tönen

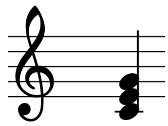


Bild 2: C-Dur Dreiklang

Dazu schauen wir uns ganz einfache Audiosignale an, nämlich verschiedene Dreiklänge. In der Musik kann man einen sogenannten Dreiklang erzeugen, indem man drei Töne gleichzeitig abspielt. Somit kann man einen Dreiklang mit der Funktion $g(t) = \sin(f_1 \cdot 2 \cdot \pi \cdot t) + \sin(f_2 \cdot 2 \cdot \pi \cdot t) + \sin(f_3 \cdot 2 \cdot \pi \cdot t)$ modellieren. Das Ziel wird es sein, einen eigenen Dreiklang zu erstellen und die Frequenzen fremder Dreiklänge mit Hilfe der Fourieranalyse herauszufinden. Die kommenden Aufgaben bearbeitet ihr im Matlab-Skript *Akkordtest.m* , welches bereits für euch geöffnet ist.

Aufgabe 1 | Einen eigenen Dreiklang modellieren

Tabelle 1: Übersicht der Dreiklänge

Dreiklang	Dur	Moll
As	as, c, es	as, ces, es
A	a, cis, e	a, c, e
B	b, d, f	b, des, f
H	h, dis, fis	h. d, fis
C	c, e, g	c, es, g
Des	des, f, as	des, e, as
D	d, fis, a	d, f, a
Es	es, g, b	es, ges, b
E	e, gis, h	e, g, h
F	f, a, c	f, as, c
Fis	fis, ais, cis	fis, a, cis
G	g, h, d	g, b, d

Erstellt zunächst mit Tabelle 1 euren eigenen Dreiklang. Die Frequenzen für die einzelnen Töne findet ihr in der beiliegenden *Tontabelle*. Tragt euren Dreiklang als Summe von drei Sinusschwingungen in Matlab für g ein. Gebt anschließend eure Gruppennummer, die ihr auf eurem Platz vorfindet, in Zeile 17 für 00 ein.
Mit der Eingabe des Befehls `Eigener_Dreiklang` im Command Window wird euch der Dreiklang graphisch angezeigt und der zugehörige Sound abgespielt. Außerdem wird euer erstellter Dreiklang im Ordner `Aufgabenblatt 2` als csv.-Datei gespeichert. Kopiert diese Datei in die *Dropbox* in den Ordner `Dreiklaenge`.

Aufgabe 2 | Frequenzen eines fremden Dreiklangs finden

Zur Erinnerung: Eine Fourieranalyse zerlegt ein Audiosignal in seine einzelnen Sinus- und Cosinusfunktionen. Aus diesen Funktionen können dann die einzelnen Frequenzen mit zugehöriger Amplitude abgelesen und in ein Frequenzspektrum eingetragen werden.

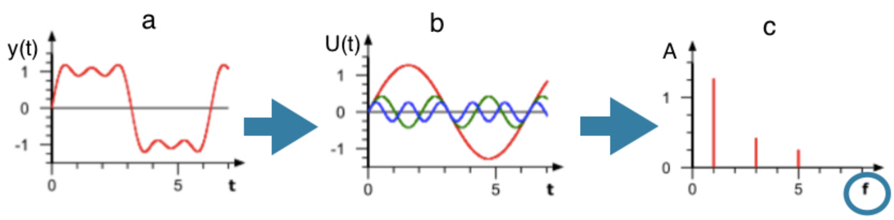


Bild 3: Aus einem Audiosignal (a) können die einzelnen Sinus- und Cosinusfunktionen (b) mit der Fourieranalyse dargestellt werden. In einem weiteren Schritt werden die Amplituden und Frequenzen der einzelnen Schwingungen im Frequenzspektrum (c) eingetragen.

Sobald sich die Dreiklänge der anderen Gruppen ebenfalls im Ordner `Dreiklaenge` befinden, kopiert ihr euch den Ordner `Dreiklaenge` in den Order `Arbeitsblatt 2` auf euren Desktop. Ihr sollt nun die Frequenzen eines Dreiklangs einer anderen Gruppe herausfinden. Geht dazu wie folgt vor:

a) Gebt im Editor in Zeile 25 für 00 die Gruppennummer ein, von der ihr die Frequenzen des unbekannten Dreiklangs herausfinden wollt. Anschließend gebt ihr im Command Window den Befehl `Fremder_Dreiklang` ein, wodurch euch der unbekannte Dreiklang abgespielt und graphisch dargestellt wird.

b) Versucht nun die Frequenzen der drei Töne zu bestimmen, aus denen die unbekannten Dreiklänge der einzelnen Gruppen bestehen. Gebt dazu im Command Window den Befehl `Fourieranalyse` ein. Auf dem erscheinenden Bild seht ihr nun die einzelnen drei Frequenzen (x-Werte der drei Peaks) und deren zugehörige Amplituden (dargestellt durch die Höhe des jeweiligen Peaks). Außerdem werden euch die Frequenzen im Command Window angezeigt. Rundet die angezeigten Werde auf drei Nachkommastellen und tragt die gerundeten Werte in die beiliegende Tabelle ein. Bestimmt mit der obigen Internetseite sowie Tabelle 2 die Töne zu den Frequenzen sowie den resultierenden Dreiklang.

Aufgabe 3 | Identifizierte Frequenzen des Dreiklangs überprüfen

a) Wir wollen nun für einen Dreiklang überprüfen, ob die gefunden drei Frequenzen stimmen. Gebt dazu die Frequenzen der einzelnen Töne von dem zuletzt bestimmten Dreiklang im Matlab-Skript in den Zeilen 35-37 für f1-f3 ein. Tragt in Matlab in Zeile 45-48 für y1-y2 die Sinusfunktionen unter Verwendung der Frequenzen f1-f3 für die einzelnen Töne ein. Mit der Eingabe des Befehls

Einzelne_Toene im Command Window werden euch die einzelnen drei Sinusschwingungen graphisch dargestellt. Sollten die drei Schwingungen nicht optimal zu erkennen sein, könnt ihr das Intervall analog zum 1. Aufgabenblatt mit a und b variieren.

b) Bilde nun die Summe der drei Sinusterme in Zeile 51. Gebt anschließend den Befehl Summe im Command Window ein und euch wird die Überlagerung der drei Töne abgespielt sowie graphisch angezeigt. Mit Betätigung der Enter Taste wird der unbekannte Dreiklang erneut abgespielt und ebenfalls graphisch angezeigt.

c) Vergleicht die beiden Bilder. Habt ihr die Frequenzen des fremden Dreiklang korrekt erkannt?

Zusatzaufgaben

a) Variiert nun die Amplituden der Sinusschwingungen aus Aufgabenteil c). Gebt dazu in den Zeilen 59-61 beliebige Werte für die Amplituden a1-a3 ein. Ihr könnt nun auch Amplituden ausprobieren, die kleiner als 1 sind. Geht dabei jedoch **nicht über** einen Wert von **10** hinaus.

b) Gebt nun im Command Window Fourieranalyse2 ein. Ihr erhaltet nun das Frequenzspektrum von Aufgabenteil b) sowie das Frequenzspektrum für die Überlagerung der drei Sinustöne mit geänderter Amplitude von Aufgabenteil f). Was fällt euch auf?

Tabelle 2: herausgefundene Frequenzen, zugehörige Tonnamen und Name des Dreiklangs

Gruppenname	Frequenzen			Töne	Dreiklang
Beispiel	f1: 523	f2: 659	f3: 784	c, e, g	C-Dur
	f1:	f2:	f3:		
	f1:	f2:	f3:		
	f1:	f2:	f3:		
	f1:	f2:	f3:		
	f1:	f2:	f3:		
	f1:	f2:	f3:		
	f1:	f2:	f3:		
	f1:	f2:	f3:		
	f1:	f2:	f3:		
	f1:	f2:	f3:		
	f1:	f2:	f3:		
	f1:	f2:	f3:		
	f1:	f2:	f3:		

A.4. Zusatzmaterial zu Arbeitsblatt 2

A.4.1. Tontabelle

Tontabelle

Tonname	Frequenz [Hz]
as	415
a	440
b=ais	466
h	493
c	523
des=cis	554
d	587
es=dis	622
e	659
f	698
fis=ges	740
g	784

Übersicht über die Töne, aus denen die Dreiklänge bestehen.

A.5. Arbeitsblatt 3

CAMMP day

Wie funktioniert eigentlich
-Shazam

-Arbeitsblatt 3-

Modell zum Finden des richtigen Musikstückes

Shazam ist eine App, die Musikstücke in Sekundenbruchteilen erkennt und dem Benutzer mitteilt. Ein Klick auf die Smartphone-App und der Nutzer weiß alles über den bis dato noch unbekannten Song. Aber wie funktioniert das?

Bild 1: Shazam App

Die Idee hinter Shazam

Die Grundidee hinter Shazam besteht darin, markante Frequenzen der Teiltöne einer kleinen Aufnahme mit den gespeicherten, markanten Frequenzen in der Datenbank zu vergleichen. Um an die Frequenzen zu gelangen, unterteilt Shazam die Musikstücke in der Datenbank und die Aufnahme in kleine Zeitintervalle und bildet von jedem Intervall eine Fourieranalyse, wie ihr sie bereits kennengelernt habt. Mit Hilfe der Fourieranalyse wird für jedes Zeitintervall ein Frequenzspektrum erstellt. In diesen Frequenzspektren sucht Shazam jeweils nach der Frequenz mit der maximalen Amplitude, also die Frequenz des lautesten Teiltons, und speichert diese mit dem dazugehörigen Zeitpunkt im Spektogramm ab. Dieses Spektogramm ist für jedes Musikstück einzigartig, genau wie der Fingerabdruck eines Menschen. Bildlich gesprochen vergleicht Shazam nun das kleine Spektogramm der Aufnahme mit allen Spektogrammen aus der Datenbank.

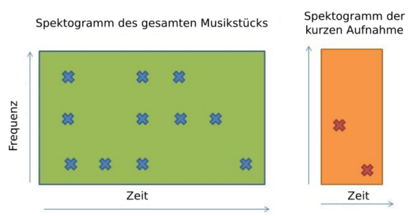


Bild 2: Spektogramm vom Original-Song (links) und von der kurzen Aufnahme (rechts) [1]

Nun überprüft Shazam im ganzen Musikstück durch Überlagerung der beiden Spektogramme, ob das kleine Spektogramm der Aufnahme mit einem Teil des Spektogramms eines beliebigen Musikstücks aus der Datenbank übereinstimmt.

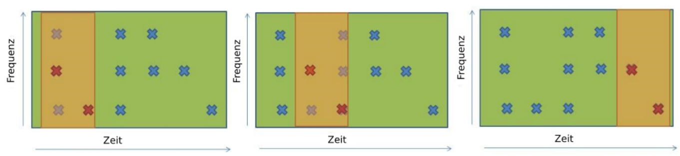


Bild 3: Überlagerung der beiden Spektogramme zum Finden einer Übereinstimmung

In diesem Beispiel wurde eine perfekte Übereinstimmung zwischen der zehnsekündigen Aufnahme und dem Ende des Musikstücks gefunden. Dieser Vergleich muss mit jedem gespeicherten Song in der Datenbank von Shazam geschehen, bis man eine Übereinstimmung wie im obigen Beispiel erzielt.

[1] Dies ist nur eine vereinfachte Darstellung. In der Realität enthält ein Spektogramm wesentlich mehr Punkte.

Problembeschreibung | Eine Aufnahme in der Datenbank finden

Wir wollen uns ein vereinfachtest Modell anschauen, wie Shazam die Datenpunkte in dem Spektogramm speichert und die Datenbank nach dem richtigen Musikstück durchsucht. Dazu erhaltet ihr das beiliegende Spektogramm einer kurzen Audioaufnahme, zu dem ihr den passenden Song in einer bereits programmierten Datenbank finden sollt. Das Matlab-Skript `Shazam_Modell.m` ist bereits für euch geöffnet. Bearbeitet nun die folgenden Schritte:

Schritt 1 | Target Zones bilden

Um den Fingerabdruck der Aufnahme zu speichern, unterteilt Shazam die Datenpunkte im Spektogramm in Gruppen ein, den sogenannten *Target Zones*. Eine Target Zone besteht dabei aus jeweils fünf, zeitlich aufeinander folgenden Datenpunkten, wobei sich zwei benachbarte *Target Zones* immer genau zwei Datenpunkte teilen.

a) Nummeriert die Datenpunkte im Spektogramm auf dem beiliegendem Blatt nach der Zeit durch.

b) Zeichnet die *Target Zones* wie oben beschrieben in euer Spektogramm ein.

c) Jede Target Zone braucht einen Referenzpunkt, den sogenannten *Anchor Point*. Der *Anchor Point* ist dabei der erste Punkt in einer Target Zone. Umkreist die *Anchor Points* in eurem Frequenzspektrum.

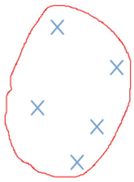


Bild 4: *Target Zone*

Schritt 2 | Den Fingerabdruck mathematisch speichern

Für den Fingerabdruck kreiert Shazam für jeden Datenpunkt eine sogenannte Adresse und speichert diese ab. Die Liste aus den einzelnen Adressen bildet dann den Fingerabdruck. Die Adresse eines Datenpunktes besteht dabei aus folgenden Daten:

1. Die Frequenz des *Anchor Points*, in dessen *Target Zone* sich der Datenpunkt befindet
2. Die Frequenz des Datenpunkts
3. Die Zeitdifferenz zwischen *Anchor Point* und Datenpunkt

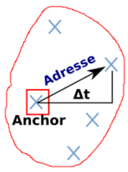


Bild 5: *Adresse*

$$\Rightarrow \text{Adresse: } (\text{Frequenz } \textit{Anchor Point}, \text{ Frequenz Datenpunkt}, \ \Delta t)$$

Bestimmt in der beiliegenden Tabelle die fehlenden Felder der Adressen. Anschließend ergänzt ihr die fehlenden Daten für die jeweiligen Datenpunkte im Matlab-Skript `Shazam_Modell.m` für NaN ersetzen.

Schritt 3 | Die Datenbank durchsuchen

Die Adressen jedes Musikstückes sind in der Datenbank von Shazam gespeichert. Darüber hinaus wird in der Datenbank jede Adresse mit der Songidentität (Interpret, Songtitel) verknüpft. Hier ein kleines Beispiel:

$$(10, 20, 1) \rightarrow \text{[Haus am See von Peter Fox]}$$

Die Adressen der Audioaufnahme werden zur Datenbank geschickt und mit den dortigen Adressen überprüft. Bei einer Übereinstimmung der Adressen speichert Shazam die möglichen Verknüpfungen (Songidentität) und zählt am Ende die Anzahl der Übereinstimmungen mit den möglichen Musikstücken.

Gebt nun in Matlab den Befehl `Datenbank_Durchsuchen` ein. Euch wird eine Matrix mit drei Spalten angezeigt. Der erste Eintrag gibt euch an, zu wie viel Prozent die Adressen eure Aufnahme mit den Adressen der einzelnen, gespeicherten Songs in der Datenbank übereinstimmen. Der zweite und dritte Eintrag geben euch Auskunft über den Songtitel und den Interpreten. Interpretiert das resultierende Ergebnis, das im Command Window erscheint.

Tabelle 1: Adressen der einzelnen Datenpunkte

Datenpunkt	Zeit *Anchor Point*	Adresse		
		Frequenz *Anchor Point*	**Frequenz Datenpunkt**	Δt
1	1	20	20	0
2	1	20	40	1
3	1	20	10	2
4	1	20	20	3
5	1	20	30	4
4	4	20	20	0
5	4	20	30	1
6	4		50	2
7	4	20	10	3
8	4	20		
7	7	10	10	0
8	7	10	40	1
9	7	10	20	2
10	7			3
11	7	10	10	4
10	10	50	50	0
11	10	50	10	1
12	10	50	30	2
13	10	50	40	3
14	10	50	20	4
13	13	40	40	0
14	13	40	20	1
15	13	40	50	2
16	13	40	30	3
17	13	40	20	4
16	16			
17	16			
18	16			
19	16			
20	16			

A.6. Zusatzmaterial zu Arbeitsblatt 3

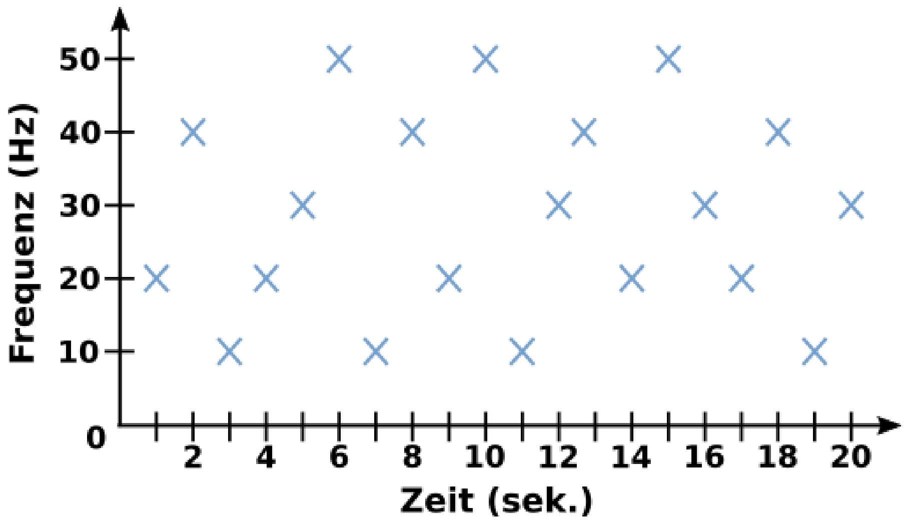
A.6.1. Spektogramm

CAMMP day

Wie funktioniert eigentlich -Shazam

- zu Arbeitsblatt 3-

Spektogramm einer Audioaufnahme



1/1

A.7. Arbeitsblatt 4

CAMMP day

Wie funktioniert eigentlich -Shazam

-Arbeitsblatt 4-

Modellverbesserung

Nach Bearbeitung der drei Schritte habt ihr als Ergebnis zwei mögliche Musikstücke erhalten, die anscheinend mit eurer Aufnahme zu 100% übereinstimmen. Ziel ist es nun das Modell zu erweitern, sodass wir zwischen den beiden Songs den Passenden identifizieren können.

Schritt 4 | Zeitdifferenz betrachten

Um aus den zwei möglichen Songs den Richtigen auszuwählen, müssen wir die Zeitdifferenz zwischen unserer Aufnahme und den beiden Songs untersuchen. Findet Shazam in der Datenbank einen Treffer, heißt das nämlich auch, dass alle Datenpunkte in der Aufnahme in genau der gleichen Reihenfolge auch im Musikstück wieder gefunden werden. Das bedeutet, dass die Reihenfolge und damit auch der zeitliche Versatz zwischen den aufeinander folgenden Datenpunkten in der Aufnahme und den Datenpunkten im Musikstück der Datenbank immer gleich bleiben muss. Um den zeitlichen Versatz, sprich die Zeitdifferenz, zu betrachten, vergleichen wir jeweils den Zeitpunkt des *Anchor Points* unserer losgeschickten Adresse und den Zeitpunkt des *Anchor Points*, der für diese Adresse im jeweiligen Song in der Datenbank gespeichert ist. Diesen zeitlichen Vergleich zwischen den *Anchor Points* führt Shazam für jede Adresse durch.

a) Gebt im Matlab-Skript `Shazam_Modell.m` im Command Window den Befehl `Treffer_Song05` ein, worauf ihr eine Matrix mit 4 Spalten erhaltet. Die ersten drei Einträge in jeder Zeile stehen für die losgeschickten Adressen von eurer Aufnahme (Frequenz *Anchor Point*, Frequenz Datenpunkt, Zeitdifferenz), die mit Song05 übereinstimmen. Der letzte Eintrag gibt euch den Zeitpunkt des *Anchor Points* an, der zu der jeweiligen Adresse im Spektogramm von Song05 gehört. Bildet nun exemplarisch die Zeitdifferenzen für einige Adresse wie folgt:

$$\Delta t = \text{Zeitpunkt } \textit{Anchor Point} \text{ im Song05} - \text{Zeitpunkt } \textit{Anchor Point} \text{ in eurer Aufnahme}$$

Hinweis: Den zugehörigen Zeitpunkt des *Anchor Points* eurer bestimmten Adresse findet ihr in der Tabelle von Arbeitsblatt 3.

b) Gebt nun im Command Window den Befehl `Treffer_Song02` ein und bestimmt analog die Zeitdifferenzen für ausgewählte Adressen.

c) Welcher Song passt nun zu der Aufnahme? Mit dem Befehl `RTS` führt Matlab die Suche nach dem richtigen Song mit Betrachtung der Zeitdifferenz durch und zeigt euch den passenden Song im Command Window an. Gebt den Befehl `RTS` im Command Window ein und überprüft euer Ergebnis.

Modellanwendung | Audiosignale der andren Gruppen bestimmen

Zum Schluss sollt ihr nun das neue Modell testen, indem ihr versucht die erstellten Audiosignale von den anderen Gruppen in der Datenbank zu suchen und den Songtitel sowie Interpreten zu den einzelnen Gruppen herauszufinden. Ladet euch den Ordner `DataBase` aus der *Dropbox* herunter und fügt ihn in den Ordner `Arbeitsblatt 3 & 4` hinzu. Nun gebt ihr im Matlab-Skript `Shazam_Modell` in Zeile 154 unter `file` für 00 die Zahl der Gruppe ein, dessen Audiosignal ihr in der Datenbank suchen möchtet. Gebt anschließend im Command Window den Befehl `Ton_Suchen` ein. Im Command Window wird euch dann der passende Song der jeweiligen Gruppe angezeigt. Notiert euch in der beiliegenden Tabelle zu den Gruppen den Songtitel und den Interpret.

Gruppennummer	Songtitel	Interpret

B. Überarbeitete Version der Arbeitsblätter mit Lösung

B.1. Arbeitsblatt 1

CAMMP day

Wie funktioniert eigentlich -Shazam

-Arbeitsblatt 1-

Einen Ton mathematisch modellieren

Jedes Instrument und auch die menschliche Stimme verursacht eine gewisse Art von Schwingung, auch wenn diese für normale Menschen nicht wahrnehmbar ist und nur mit Hilfe eines Computers oder eines Oszilloskops sichtbar gemacht werden kann. So ist beispielsweise der Ton einer Stimmgabel mathematisch bzw. physikalisch nichts anderes als eine harmonische Schwingung in Form einer Sinuskurve und kann mit der Sinusfunktion

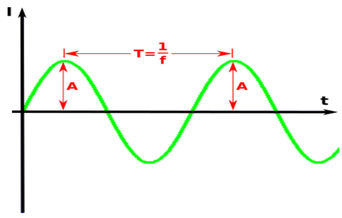


Bild 1: Sinusschwingung einer Stimmgabel

$$g(t) = A \cdot \sin(f \cdot 2 \cdot \pi \cdot t)$$

modelliert werden. Dabei steht

- t für die Zeit.
- A für die **Amplitude** und gibt die maximale **Lautstärke** des Tons an. Das heißt je höher die Amplitude, desto lauter wird der Ton wahrgenommen.
- f für die **Frequenz** und gibt die **Tonhöhe** an. Der Abstand zwischen zwei gleichen Maxima wird als Periode $T = \frac{1}{f}$ bezeichnet. D.h. je größer die Frequenz f ist, desto näher liegen die die Maxima nebeneinander und umso höher wird der Ton wahrgenommen. Die Frequenz wird in Hertz [Hz] angegeben.

Aufgabenbeschreibung | Töne selber modellieren

Euer Ziel wird es sein, in die Rolle des Computers zu schlüpfen und Töne selber mathematisch zu modellieren. Das dafür benötigte Matlab-Skript `Tonmodell.m` ist schon geöffnet. Ein Ton der Form $g(t) = \sin(200 \cdot 2\pi \cdot t)$ ist bereits in Matlab gespeichert. Die Abspieldauer des Tons ist so programmiert, dass sie $6s$ beträgt.

Aufgabe 1 | Wie kann ich die Lautstärke eines Tons verändern?

a) Höre und schaue dir zunächst den Sinuston an, indem du dazu in das Command Window `Sinuston` eingibst und auf Enter drückst. Es wird dir auch gleich der passende Graph angezeigt. Um die Sinusschwingung besser zu erkennen, wird nur das kleine Intervall $[0, 0.1]$ betrachtet. Welchen Wert hat die Amplitude dieses Tons?

Wir halten zunächst den Wert für die Frequenz konstant und wollen nur die Lautstärke, also die Amplitude A des Tons verändern. Betrachte dazu folgende Hinweise bezüglich Matlab:

Eingaben im MATLAB-Skripts `Tonmodell.m`

In Matlab muss vor jeder Operation ein Punkt gesetzt werden. Möchte man also beispielsweise eine Variabel, wie die Zeit t, quadrieren oder $\frac{1}{t}$ rechnen, muss man in Matlab $t.\hat{}2$ bzw. $1./t$ eingeben. Bei der Addition und Subtraktion gilt das gleiche. Zum Beispiel gibt man für $3 - t$ in Matlab $3. - t$ ein. **Ausnahme**: Für Eingaben der Art $t \pm 3$ darf kein Punkt gesetzt werden.

b) Je größer die Amplitude gewählt wird, desto lauter wird der Ton wahrgenommen. Überlege dir wie du die Konstante A_1 beim Ton wählen musst, damit der Ton lauter bzw. leiser klingt. Trage dazu in Matlab in Zeile 21 verschiedene Werte für $A1$ ein und überprüfe jeweils mit der Eingabe `check_A1` im Command Window dein Ergebnis. Dir wird anschließend die Schwingung des Ausgangston $g(t) = \sin(200 \cdot 2 \cdot \pi \cdot t)$ und die des neuen Tons im Intervall $[0, 0.1]$ graphisch angezeigt.

1/2

Überlege dir nun, wie du die Amplitude wählen musst, damit:

c) Die Lautstärke kontinuierlich, also linear, mit der Zeit t ausschließlich ansteigt. Trage dazu deine Term in Matlab für A2_linear ein und überprüfe wieder dein Ergebnis, indem du im Command Window den Aufruf `check_A2_linear` eingibst. Überlege dir, wie du einen schnelleren Anstieg der Lautstärke erzeugen kannst. Trage dazu deinen Term für A2_fix ein und überprüfe mit `check_A2_fix` dein Ergebnis.
Tipp: Zu dieser Aufgabe liegt die Hilfekarte 1 aus, die du bei Bedarf nutzen kannst.

d) Die Lautstärke kontinuierlich mit der Zeit ausschließlich abfällt, sodass nach 6 Sekunden der Ton nicht mehr zu hören ist. Trage dein Ergebnis für $A3$ ein. Hier kannst du nun selber das Intervall bestimmen, indem du dir den Graph des Tons anschauen willst. Trage dazu in Matlab für a den Startwert und für b den Endwert des Intervalls ein. Überprüfe nun mit `check_A3` dein Ergebnis.
Tipp: Zu dieser Aufgabe liegt die Hilfekarte 1 aus, die du bei Bedarf nutzen kannst.

Aufgabe 2 | Wie kann ich die Frequenz eines Tons verändern?

Als nächstes wird die Lautstärke konstant gehalten und die Frequenz verändert.

a) Höre dir zunächst nochmal den Sinuston $g(t) = \sin(200 \cdot 2\pi \cdot t)$ an, indem du im Command Window den Befehl `Sinuston` eingibst.

b) Je kleiner der Abstand zwischen zwei gleichen Maxima in der Sinusschwingung ist, desto höher wird der Ton wahrgenommen. Betrachte nun $g(t) = \sin(200 \cdot 2\pi \cdot F1 \cdot t)$ und überlege dir wie du die Konstante $F1$ wählen musst, damit der Ton höher bzw. tiefer klingt. Trage in Matlab verschiedene Werte für $F1$, die zwischen 0 und 6 liegen, ein und überprüfe jeweils mit `check_F1` dein Ergebnis. Dir wird anschließend die Schwingung des Ausgangston $g(t) = \sin(200 \cdot 2\pi \cdot t)$ und die des neuen Tons angezeigt.

c) Betrachte nun $g(t) = \sin(200 \cdot 2\pi \cdot F2 \cdot t)$ und überlege dir, wie du den Faktor $F2$ wählen musst, damit der Ton mit der Zeit ausschließlich höher wird. Trage deinen Term in Matlab für $F2$ ein und überprüfe dein Ergebnis mit `check_F2`

Aufgabe 3 | Erstellt euer eigenes Audiosignal

a) Am Ende des Tages wird es das Ziel sein, ein erstelltes Audiosignal von einer anderen Gruppe mit Hilfe des Shazam-Algorithmus zu identifizieren. Dazu könnt ihr jetzt euer eigenes Audiosignal erstellen. In Matlab könnt ihr in Zeile 80 für y_3 den Term für euer Audiosignal eintragen. Auch die Addition von mehreren Sinusschwingungen ist hier möglich, z.B.

$$y\_3 = 3. \cdot \sin(2000 \cdot 2\pi. \cdot t) + 0.5. \cdot \sin(4100 \cdot 2\pi. \cdot t.^2) + \sin(1000 \cdot 2\pi. \cdot (2. - t).^2).$$

Um euer Audiosignal zu speichern, tragt ihr unter `file` in Zeile 83 für 00 eure Gruppennummer ein, die ihr auf eurem Platz vorfindet. Denkt euch nun für euer erstelltes Audiosignal noch einen Songtitel und den Namen des Interpreten aus, die ihr unter `title` und `artist` eingeben könnt. Die Intervalllänge, in der ihr euer eigenes kleines Musikstück betrachten wollt, könnt ihr wieder mit $a2$ und $b2$ festlegen. Mit der Eingabe `Eigener_Sound` wird euer Audiosignal abgespielt, graphisch angezeigt und als .txt-Datei im Ordner `Arbeitsblatt 1` unter dem Ordner `Euer_Sound` gespeichert.

b) Ladet nun euren erstellten Ton unter dem Ordner `DataBase` in die *Dropbox*. Der Betreuer wird dann mit Hilfe des Fingerabdruck-Algorithmus euer Audiosignal zu einer Datenbank hinzufügen.

B.2. Zusatzmaterial zu Arbeitsblatt 1

B.2.1. Zusatzaufgaben

CAMMP day

Wie funktioniert eigentlich -Shazam

-Zusatzaufgaben zu Arbeitsblatt 1-

Zusatzaufgaben

a) Wähle die Amplitude so, dass die Lautstärke erst abnimmt und danach wieder ansteigt. Trage dein Ergebnis für $A4$ ein und bestimme wieder mit $a3$ und $b3$ das Intervall, in dem ihr die Schwingung des Tons betrachten wollt. Überprüfe mit `check_A4` dein Ergebnis.
Tipp: Zu dieser Aufgabe liegt die Hilfekarte 2 aus, die du bei Bedarf nutzen kannst.

b) Wähle die Amplitude so, dass die Lautstärke erst zunimmt und danach wieder abnimmt. Um den Effekt besser hören zu können wird die Frequenz auf $f\_zb = 50Hz$ gesetzt. Trage dein Ergebnis für $A5$ ein. Das Intervall indem der Graph angezeigt wird kannst du wieder mit $a4$ und $b4$ bestimmen. überprüfe mit `check_A5` dein Ergebnis.
Tipp: Zu dieser Aufgabe liegt die Hilfekarte 3 aus, die du bei Bedarf nutzen kannst.

c) Betrachte nun $g(t) = \sin(200 \cdot F3)$ und überlege dir, wie du den Faktor $F3$ wählen musst, damit der Ton mit der Zeit ausschließlich tiefer wird. Trage dein Ergebnis für $F3$ ein und wähle mit $a5$ und $b5$ das Intervall, indem der Graph angezeigt werden soll. Überprüfe mit `check_F3` dein Ergebnis. Wird der Ton so schnell tiefer, dass man diesen Prozess nicht wirklich hören kann, erhöhe die Frequenz unter f_zc.
Hinweis: Beachtet die Punktsymmetrie vom Sinus, sodass gilt: $\sin(-x) = -\sin(x)$.

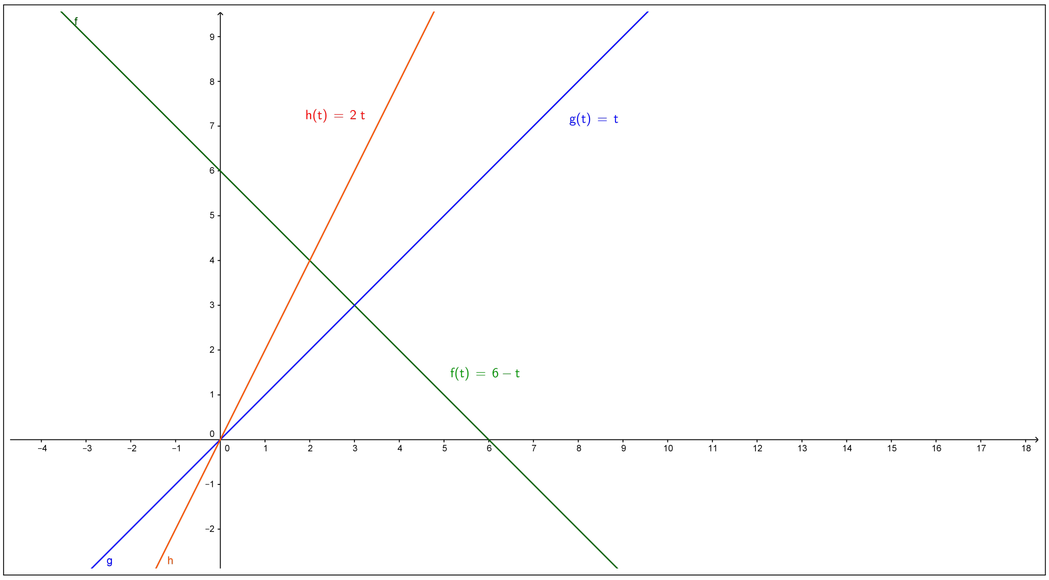
d) Betrachte nun $g(t) = \sin(200 \cdot F4)$ und überlege dir, wie du den Faktor $F4$ wählen musst, damit der Ton mit der Zeit erst tiefer und dann wieder höher wird. Trage dein Ergebnis für $F4$ ein und bestimme auch hier selbst mit $a6$ und $b6$ das Intervall, indem der Graph angezeigt werden soll. überprüfe mit `check_F4` dein Ergebnis.

1/1

B.2.2. Hilfekarten

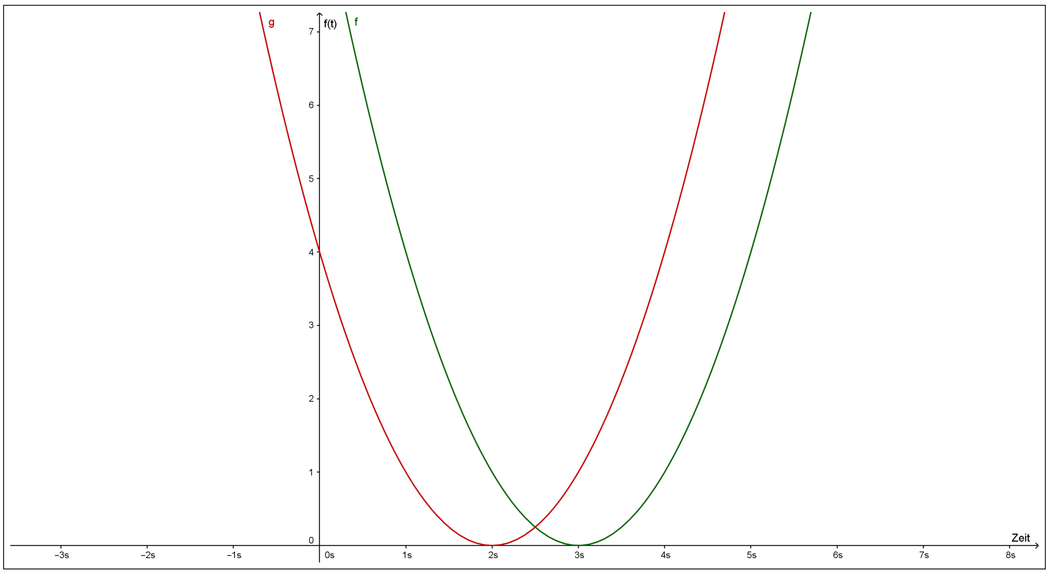
Hilfekarte 1

Tipp: Damit sich die Amplitude mit der Zeit ändert, muss die Amplitude zeitabhängig gewählt werden.



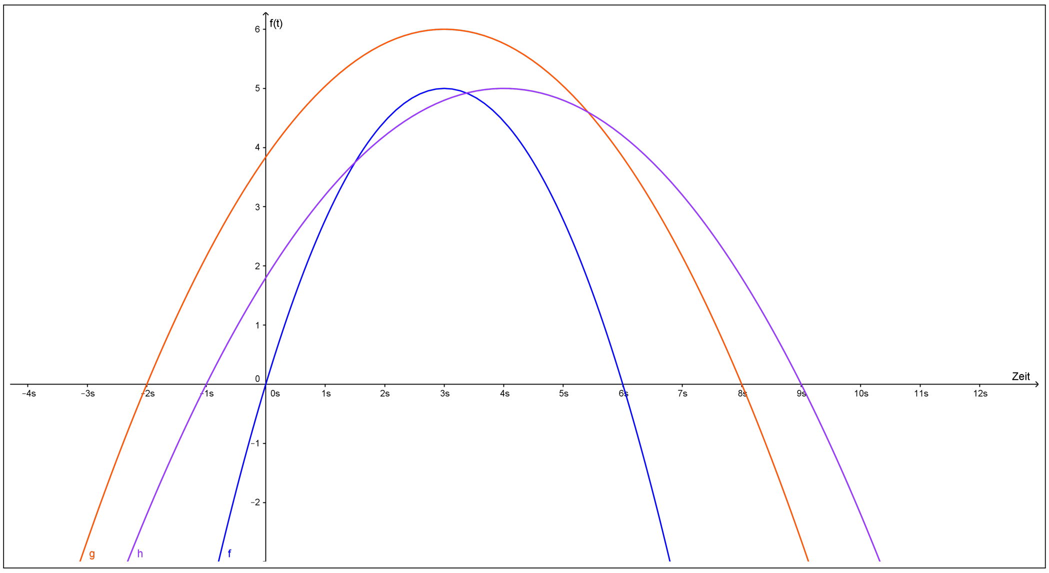
Hilfekarte 2

Tipp: Wähle eine Parabel, deren Scheitelpunkt zwischen $0\ s$ und $6\ s$ liegt. Die Scheitelpunktsformel einer Parabel lautet $f(t) = a \cdot (t + b)^2 + c$ mit Scheitelpunkt $S = (-b, c)$.



Hilfekarte 3

Tipp: Wähle eine Parabel, die nach unten geöffnet ist und deren eine Nullstelle ≤ 0 und deren andere Nullstelle ≥ 6 ist. Die Scheitelpunktsformel einer Parabel lautet $f(t) = a \cdot (t + b)^2 + c$ mit Scheitelpunkt $S = (-b, c)$.



CAMMP day

Wie funktioniert eigentlich -Shazam

Musterlösung Arbeitsblatt 1

Anmerkung: Da t ein Vektor ist, muss vor den Operatoren, die sich auf den Vektor t bezieht, ein Punkt gesetzt werden.

Aufgabe 1 | Wie kann ich die Lautstärke eines Tons verändern?

a) Mit dem Befehl `Sinuston` den vorgegeben Ton abspielen und sich graphisch anzeigen lassen. Der Wert der Amplitude beträgt 1.

b) Damit der vorgegebene Ton lauter bzw. leiser klingt, muss die Amplitude vergrößert bzw. verkleinert werden. Dazu wird für A1 eine Konstante gewählt, für die folgendes gilt:

$A1 > 1$	Der Ton klingt lauter als der Ton in a)
$A1 < 1$	Der Ton klingt leise als der Ton in a)

Mit `check_A1` wird sowohl der Ton von Aufgabenteil a) als auch der neue Ton graphisch dargestellt, um die Auswirkung der Änderung der Amplitude zu beobachten.

c) Um den Ton linear mit Zeit lauter werden zu lassen, muss die Amplitude zeitabhängig gewählt werden. Für einen linearen Anstieg wird für A2_linear die Geradengleichung $m \cdot t + c$ angesetzt, wobei m und c Konstanten sind und beliebig gewählt werden können. Mögliche Lösungen wären beispielsweise

$$t, \quad 2. \cdot t, \quad \text{oder} \quad 4. \cdot t + 1.$$

Einen schnelleren Anstieg der Lautstärke wird beispielsweise mit $t.\hat{}2$ oder $t.\hat{}3$ erreicht.

d) Für einen kontinuierlichen Abfall kann beispielsweise eine Geradengleichung angesetzt werden, die die x-Achse bei $t = 6\ s$ schneidet. Eine Lösung wäre dann z.B.

$$A3 = x - 6.$$

Aufgabe 2 | Wie kann ich die Frequenz eines Tons verändern?

a) Analog zu Aufgabe 1 mit dem Befehl `Sinuston` den vorgegeben Ton lediglich anhören und sich graphisch anzeigen lassen.

b) Es wird der Ton $\sin(f \cdot 2\pi \cdot F1 \cdot t)$ betrachtet. Damit der vorgegeben Ton höher bzw. tiefer klingt, muss ω vergrößert bzw. verkleinert werden. Dazu wird ω mit dem konstanten Faktor F1 multipliziert, der folgendes bewirkt:

$F1 > 1$	Der Ton klingt höher als der Ton in a)
$F1 < 1$	Der Ton klingt tiefer als der Ton in a)

Mit `check_F1` wird sowohl der Ton von Aufgabenteil a) als auch der neue Ton graphisch dargestellt, um die Auswirkung der Veränderung der Frequenz auf die Sinusschwingung zu beobachten. So fällt auf, dass sich die Anzahl der Maxima in einem Intervall um den Faktor F1 verändert.

c) Es wird der Ton $\sin(f \cdot 2\pi \cdot F2 \cdot t)$ betrachtet. Damit der Ton mit der Zeit höher klingt muss die Frequenz zeitabhängig werden. Für einen kontinuierlichen Anstieg der Frequenz kann demnach F2 als t gewählt werden.

1/2

Zusatzaufgaben

a) Um den Ton erst leiser und dann wieder lauter werden zu lassen, wird eine Parabel benötigt, die ihren Scheitelpunkt zwischen 0 s und 6 s hat. Mit der allgemeinen Scheitelpunktformel $f(t) = a \cdot (t + d)^2 + e$ mit Scheitelpunkt $S = (-d/e)$ erhält man z.B. folgende zwei Lösungen:

$$A4 = (t - 3).^2 \text{ oder } A4 = (t - 4).^2$$

b) Wird der umgekehrte Fall betrachtet, also erst steigt die Lautstärke und anschließend fällt sie wieder, wird ebenfalls eine Parabel benötigt. Diese Parabel muss jedoch nach unten geöffnet sein und ihre erste Nullstelle muss ≤ 0 und ihre zweite Nullstelle ≥ 6 sein. Formt man die Scheitelpunktsformel entsprechend um erhält man beispielsweise als Lösung

$$A5 = -\frac{5}{9} \cdot (t - 3).^2 + 5 \text{ oder } A5 = -\frac{5}{25} \cdot (t - 4).^2 + 5.$$

c) Es wird der Ton $\sin(f \cdot 2\pi \cdot F3)$ betrachtet. Damit der Ton mit der Zeit tiefer wird reicht es nicht aus eine Gerade mit negative Steigung oder eine nach unten geöffnete Parabel zu benutzen. Denn die Sinusfunktion ist punktsymmetrisch und es gilt $\sin(-x) = -\sin(x)$. Fallende Funktionen, die das Problem läsen wären z.B

$$F3 = 1./t \text{ oder } F3 = exp(-t).$$

d) Es wird der Ton $\sin(f \cdot 2\pi \cdot F4)$ betrachtet. Für diese Aufgabe wird für $F4$ eine Parabel gewählt, die ihren Scheitelpunkt zwischen 0 und 6 hat. Eine Lösung wäre z.B.

$$F4 = (x - 3).^2.$$

B.4. Arbeitsblatt 2

CAMMP day

Wie funktioniert eigentlich -Shazam

-Arbeitsblatt 2-

Fourieranalyse

Bild 1: Joseph Fourier

Im ersten Arbeitsblatt haben wir bereits gesehen, dass die Amplitude und die Frequenz einen Ton mathematisch beschreiben bzw. charakterisieren. Zum Erstellen des akustischen Fingerabdrucks ist Shazam vor allem an den Frequenzen interessiert, die im Musikstück oder in der kurzen Aufnahme enthalten sind, da die Abfolge der Frequenzen für jeden Song einzigartig sind. Aus den Frequenzen erstellt Shazam mit einem bestimmten Algorithmus ein für jeden Song charakteristisches Spektogramm, aus dem anschließend der Fingerabdruck mathematisch gebildet wird. Die Frage, die sich jedoch zunächst stellt, ist, wie Shazam aus den Musikstücken und den Aufnahmen, die aus ganz vielen, unterschiedlichen Tönen bestehen, die einzelnen Frequenzen zum Erstellen der Spektogramme erhält.

Problembeschreibung | Überlagerung von Tönen

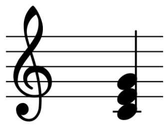


Bild 2: C-Dur Dreiklang

Dazu schauen wir uns ganz einfache Audiosignale an, nämlich verschiedene Dreiklänge. In der Musik kann man einen sogenannten Dreiklang erzeugen, indem man drei Töne gleichzeitig abspielt. Somit kann man mathematisch einen Dreiklang mit der Funktion $g(t) = \sin(f_1 \cdot 2 \cdot \pi \cdot t) + \sin(f_2 \cdot 2 \cdot \pi \cdot t) + \sin(f_3 \cdot 2 \cdot \pi \cdot t)$ modellieren. Das Ziel wird es sein, einen eigenen Dreiklang zu erstellen und die Frequenzen fremder Dreiklänge mit Hilfe der Fourieranalyse herauszufinden. Die kommenden Aufgaben bearbeitet ihr im Matlab-Skript *Akkordtest.m* , welches bereits für euch geöffnet ist.

Aufgabe 1 | Eigenen Dreiklang erstellen

Erstellt zunächst mit Hilfe der beiliegenden Tontabelle euren eigenen Dreiklang. Tragt euren Dreiklang als Summe von drei Sinusschwingungen in Matlab für g ein. Gebt anschließend eure Gruppennummer, die ihr auf eurem Platz vorfindet, in Zeile 17 für 00 ein.

Mit der Eingabe des Befehls `Eigener_Dreiklang` im Command Window wird euch der Dreiklang graphisch angezeigt und der zugehörige Sound abgespielt. Außerdem wird euer erstellter Dreiklang im Ordner `Aufgabenblatt 2` als csv.-Datei gespeichert. Kopiert diese Datei in die *Dropbox* in den Ordner `Dreiklaenge`.

Aufgabe 2 | Frequenzen eines fremden Dreiklangs finden

Zur Erinnerung: Eine Fourieranalyse zerlegt ein Audiosignal in seine einzelnen Sinus- und Cosinusfunktionen. Aus diesen Funktionen können dann die einzelnen Frequenzen mit zugehöriger Amplitude abgelesen und in ein Frequenzspektrum eingetragen werden.

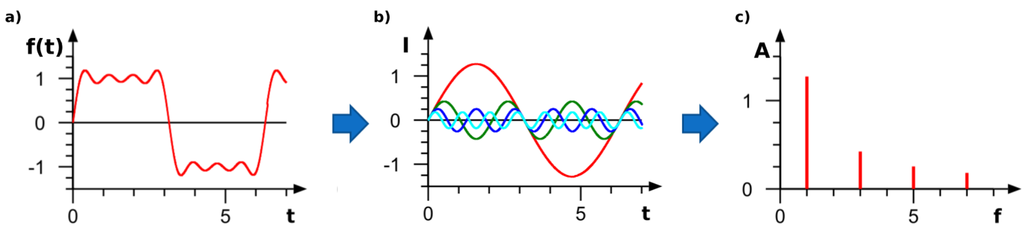


Bild 3: Aus einem Audiosignal (a) können die einzelnen Sinus- und Cosinusfunktionen (b) mit der Fourieranalyse dargestellt werden. In einem weiteren Schritt werden die Amplituden und Frequenzen der einzelnen Schwingungen im Frequenzspektrum (c) eingetragen.

Sobald sich die Dreiklänge der anderen Gruppen ebenfalls im Ordner `Dreiklaenge` befinden, kopiert ihr euch den Ordner `Dreiklaenge` in den Order `Arbeitsblatt 2` auf euren Desktop. Ihr sollt nun die Frequenzen eines Dreiklangs einer anderen Gruppe herausfinden. Geht dazu wie folgt vor:

a) Gebt im Editor in Zeile 25 für 00 die Gruppennummer ein, von der ihr die Frequenzen des unbekannten Dreiklangs herausfinden wollt. Anschließend gebt ihr im Command Window den Befehl `Fremder_Dreiklang` ein, wodurch der unbekannte Dreiklang abgespielt und graphisch dargestellt wird.

b) Versucht nun für drei Gruppen die Frequenzen der drei Töne zu bestimmen, aus denen die unbekannten Dreiklänge der einzelnen Gruppen bestehen. Gebt dazu im Command Window den Befehl `Fourieranalyse` ein. Auf dem erscheinenden Bild seht ihr nun die einzelnen drei Frequenzen (x-Werte der drei Peaks) und deren zugehörige Amplituden (dargestellt durch die Höhe des jeweiligen Peaks). Außerdem werden euch die Frequenzen im Command Window angezeigt. Rundet die angezeigten Werte auf drei Nachkommastellen und **tragt die gerundeten Werte in Tabelle 1 ein**. Bestimmt mit den beiden beiliegenden Tabellen die Töne zu den Frequenzen sowie den resultierenden Dreiklang.

Aufgabe 3 | Identifizierte Frequenzen des Dreiklangs überprüfen

a) Wir wollen nun für einen Dreiklang überprüfen, ob die gefundenen drei Frequenzen stimmen. Gebt dazu die Frequenzen der einzelnen Töne von dem zuletzt bestimmten Dreiklang im Matlab-Skript in den Zeilen 35-37 für f1-f3 ein. Tragt in Matlab in Zeile 45-48 für y1-y3 die Sinusfunktionen unter Verwendung der Frequenzen f1-f3 für die einzelnen Töne ein. Mit der Eingabe des Befehls `Einzelne_Toene` im Command Window werden euch die einzelnen drei Sinusschwingungen graphisch dargestellt. Sollten die drei Schwingungen nicht optimal zu erkennen sein, könnt ihr das Intervall analog zum 1. Aufgabenblatt mit a und b variieren.

b) Bildet nun die Summe der drei Sinusterme in Zeile 51. Gebt anschließend den Befehl `Summe` im Command Window ein und euch wird die Überlagerung der drei Töne abgespielt sowie graphisch angezeigt.

c) Mit Betätigung der Enter Taste wird der unbekannte Dreiklang erneut abgespielt und ebenfalls graphisch angezeigt. Vergleicht die beiden Bilder. Habt ihr die Frequenzen des fremden Dreiklang korrekt erkannt?

Tabelle 1: herausgefundene Frequenzen, zugehörige Tonnamen und Name des Dreiklangs

Gruppennummer	Frequenzen	Töne	Dreiklang
Beispiel: 00	f_1: 523 f_2: 659 f_3: 784	c, e, g	C-Dur
	f_1: f_2: f_3:		
	f_1: f_2: f_3:		
	f_1: f_2: f_3:		

B.5. Zusatzmaterial zu Arbeitsblatt 2

B.5.1. Tontabelle

Tontabellen

Tonname	Frequenz [Hz]	Dreiklang	Dur	Moll
as	415	**As**	as, c, es	as, ces, es
a	440	**A**	a, cis, e	a, c, e
b=ais	466	**B**	b, d, f	b, des, f
h	493	**H**	h, dis, fis	h. d, fis
c	523	**C**	c, e, g	c, es, g
des=cis	554	**Des**	des, f, as	des, e, as
d	587	**D**	d, fis, a	d, f, a
es=dis	622	**Es**	es, g, b	es, ges, b
e	659	**E**	e, gis, h	e, g, h
f	698	**F**	f, a, c	f, as, c
fis=ges	740	**Fis**	fis, ais, cis	fis, a, cis
g	784	**G**	g, h, d	g, b, d

Übersicht über die Töne, aus denen die Dreiklänge bestehen.

B.5.2. Zusatzaufgaben

CAMMP day

Wie funktioniert eigentlich -Shazam

-Zusatzaufgaben zu Arbeitsblatt 2-

Zusatzaufgabe | Fourieranalyse per Hand

Im Folgenden wollen wir die Fourieranalyse einmal per Hand durchführen. Ließ dir dazu folgenden Infotext aufmerksam durch:

Die Fourieranalyse zerlegt ein Signal in seine einzelne Sinus- und Cosinusfunktionen. Mathematisch gesehen ist das Ergebnis einer solchen Zerlegung eine unendliche Reihe, die sogenannte *Fourier-Reihe* mit folgender Gestalt:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t))$$

Das Summenzeichen $\sum_{n=1}^{\infty}$ summiert dabei die einzelnen Einträge von $(a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t))$ auf. Weiter ist $\omega_0 = \frac{2\pi}{T}$, wobei T die Periode der zu zerlegenden Funktion angibt, also die Länge bis sich die Funktion wiederholt. Die Koeffizienten a_0, a_n und b_n lassen sich wie folgt berechnen:

$$a_0 = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\, dt, \quad a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\cos(nt)\, dt, \quad b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t) sin(nt)\, dt$$

Ist die zu untersuchende Funktion **achsensymmetrisch**, sind alle Koeffizienten $b_n = 0$. Ist die Funktion **punktsymmetrisch**, sind alle Koeffizienten $a_n = 0$.

Das Ziel wird es sein, die Fourieranalyse für die Sägezahnfunktion durchzuführen d.h. die Funktion mit Sinus-Cosinusfunktionen auszudrücken. Die Sägezahnfunktion ist die 2π-periodische Fortsetzung $(y(t+2\pi))$ der Funktion

$$y(t) = \begin{cases} 0 & \text{falls } t = -\pi,\ t = \pi \\ t & \text{falls } -\pi < t < \pi \end{cases}$$

und hat folgende Gestalt:

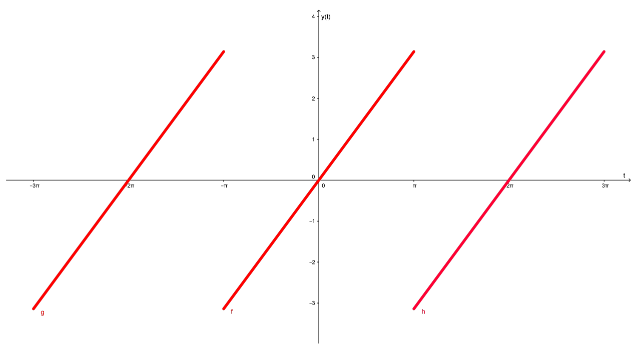


Abbildung 1: Graph der Sägezahnfunktion

a) Bestimme die Periode der Sägezahnfunktion und damit den Wert für ω_0.

b) Berechne die Koeffizienten a_0, a_n und b_n.
Tipp: Beachte die Symmetrie der Sägezahnfunktion.
Hinweis: Für die Berechnung der Integrale liegt die Hilfekarte 4 aus, die bei Bedarf benutzt werden kann.

c) Setze nun deine berechneten Koeffizienten in die Fourierreihe ein und berechne die Reihe für die ersten fünf Koeffizienten, d.h. die Summe läuft von 1 bis 5 ($\sum_{n=1}^{5}$). Tragt die resultierende Funktion in Matlab in Zeile 60 für f ein. Mit dem Befehl `Fourier` wird dir die Sägezahnfuntkion und deine berechneten Funktion angezeigt. Deine Funktion sollte sich der Sägezahnfunktion nun annähern. Für eine bessere Approximation kannst du weitere Summanden der Fourierreihe berechnen.
Hinweis: Für die Werte des Cosinus liegt die Hilfekarte 5 aus, die bei Bedarf benutzt werden kann.

B.5.3. Hilfekarten

Hilfekarte 4

Hinweis: Zur Lösung eures Integrals wird folgende Integrationsregel verwendet:

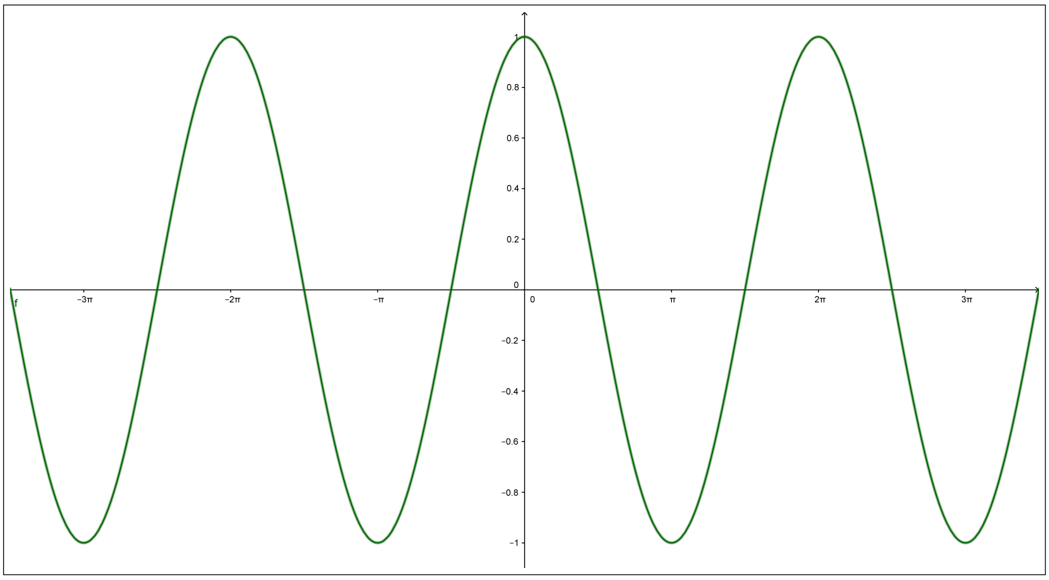
Partielle Integration

$$\int f'(x)g(x) = f(x)g(x) - \int f(x)g'(x)$$

Wählt $\cos(t)$ bzw. $\sin(t)$ als $f'(t)$ und t als $g(t)$. Beachtet beim Integrieren auch das n in $\sin(n \cdot t)$ bzw. $\cos(n \cdot t)$.

Hilfekarte 5

Hinweis: Anhand des Funktionsgraphen der Cosinusfunktion können die benötigten Werte des Cosinus zur Berechnung der Fourierreihe abgelesen werden.



CAMMP day

Wie funktioniert eigentlich -Shazam

Musterlösung Arbeitsblatt 2

Aufgabe 1 | Eigenen Dreiklang erstellen

In der ersten Aufgabe erstellen die Schülerinnen und Schüler ihren eigenen Dreiklang, indem sie sich einen Dreiklang und die dazugehörigen Frequenzen aus den beiden Tabellen auf dem Blatt *Tontabelle.pdf* heraussuchen. Den Dreiklang können sie anschließend in Matlab in der Form

$$g(t) = \sin(f_1 \cdot 2 \cdot \pi \cdot t) + \sin(f_2 \cdot 2 \cdot \pi \cdot t) + \sin(f_3 \cdot 2 \cdot \pi \cdot t)$$

eingeben. Mit der Eingabe `Eigener_Dreiklang` im Command Window von Matlab wird der Dreiklang als *csv.-Datei* gespeichert.
Es muss darauf geachtet werden, dass alle Gruppen ihre Dateien in den Ordner Dreiklaenge in der Dropbox kopieren, bevor mit der zweiten Aufgabe begonnen wird.

Aufgabe 2 | Frequenzen eines fremden Dreiklangs finden

Hier bestimmen die Schülerinnen und Schüler mit dem Befehl Fourieranalyse die Frequenzen der Dreiklänge dreier, verschiedener Gruppen. Mit den beiden Tabellen auf dem Blatt *Tontabelle.pdf* können sie den Frequenzen einen Dreiklang zuordnen.

Aufgabe 3 | Identifizierte Frequenzen eines Dreiklangs überprüfen

Für den zuletzt bestimmten Dreiklang bilden die Schülerinnen und Schüler mit den herausgefundenen drei Frequenzen wieder die Überlagerung von drei Sinustönen. Dazu geben sie die Summe der drei Sinustöne mit den entsprechenden Frequenzen in Matlab ein. Mit dem Befehl `Summe` wird ihnen ihr überlagerter Ton abgespielt und als Graph angezeigt. Mit **Drücken der Enter-Taste** wird ihnen erneut der fremde Dreiklang abgespielt und ebenfalls graphisch angezeigt. Ein Vergleich der beiden Bilder soll zeigen, dass die Dreiklänge identisch sind.

Zusatzaufgabe

a) Die Periode der Sägezahnfunktion beträgt $T = 2\pi$ und damit ist $\omega_0 = \frac{2\pi}{T} = \frac{2\pi}{2\pi} = 1$.

b) Da die Funktion punktsymmetrisch ist, gilt $a_n = 0$. Weiter gilt:

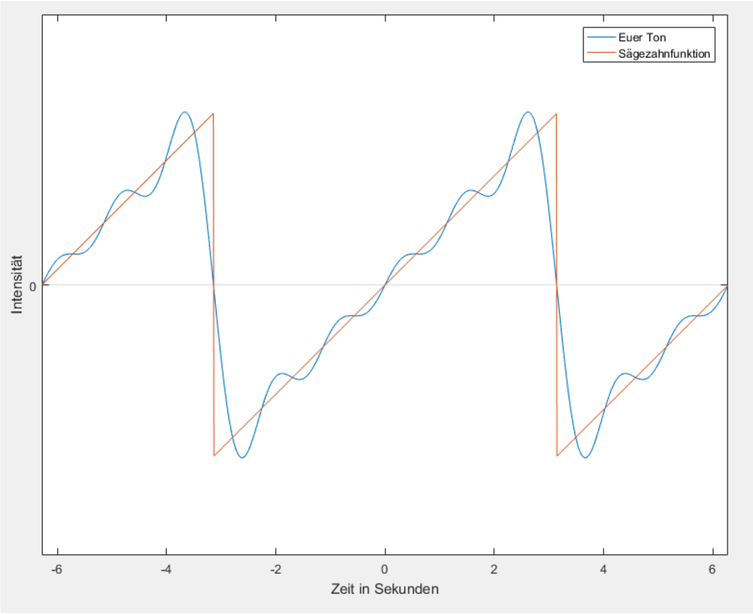
$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} t \; dt = \frac{1}{2\pi} \cdot t^2 \Big|_{-\pi}^{\pi} = 0$$

$$\begin{aligned} b_n &= \frac{1}{\pi} \int_{-\pi}^{\pi} t \cdot \sin(nt) \; dt \\ &= \frac{1}{\pi} \cdot [- \frac{t}{n} \cdot \cos(nt) \Big|_{-\pi}^{\pi} + \frac{1}{n} \cdot \int_{-\pi}^{\pi} \cos(nt) \; dt] \\ &= \frac{1}{\pi} \cdot (-\frac{\pi}{n} \cdot \cos(n\pi) - \frac{\pi}{n} \cdot \cos(-n\pi) = -\frac{2}{n} \cdot \cos(n\pi) \\ &= (-1)^{n+1} \cdot \frac{2}{n} \end{aligned}$$

c) Die ersten fünf Summanden der Fourierreihe ergeben sich aus den in b) herausgefundenen Fourierkoeffizienten:

$$
\begin{aligned}
f_5(t) &= \frac{a_0}{2}\sum_{n=1}^{5}(a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t)) \\
&= \sum_{n=1}^{5}(-1)^{n+1}\cdot\frac{2}{n}\sin(nt) \\
&= 2\sin(t) - \sin(2t) + \frac{2}{3}\sin(3t) - \frac{1}{2}\sin(4t) + \frac{2}{5}\sin(5t).
\end{aligned}
$$

Gibt man dieses Ergebnis in Matlab ein, erhält man mit dem Befehl `Fourier` folgendes Bild:



Zu erkennen ist eine relativ gute Annäherung an die Sägezahnfunktion.

2/2

B.7. Arbeitsblatt 3

CAMMP day

Wie funktioniert eigentlich -Shazam

-Arbeitsblatt 3-

Modell zum Finden des richtigen Musikstückes

Shazam ist eine App, die Musikstücke in Sekundenbruchteilen erkennt und dem Benutzer mitteilt. Ein Klick auf die Smartphone-App und der Nutzer weiß alles über den bis dato noch unbekannten Song. Aber wie funktioniert das?

Bild 1: Shazam App

Die Idee hinter Shazam

Die Grundidee hinter Shazam besteht darin, markante Frequenzen der Teiltöne einer kleinen Aufnahme mit den gespeicherten, markanten Frequenzen in der Datenbank zu vergleichen. Um an die Frequenzen zu gelangen, unterteilt Shazam die Musikstücke in der Datenbank und die Aufnahme in kleine Zeitintervalle und bildet von jedem Intervall eine Fourieranalyse, wie ihr sie bereits kennengelernt habt. Mit Hilfe der Fourieranalyse wird für jedes Zeitintervall ein Frequenzspektrum erstellt. In diesen Frequenzspektren sucht Shazam jeweils nach der Frequenz mit der maximalen Amplitude, also die Frequenz des lautesten Teiltons, und speichert diese mit dem dazugehörigen Zeitpunkt im Spektogramm ab. Dieses Spektogramm ist für jedes Musikstück einzigartig, genau wie der Fingerabdruck eines Menschen. Bildlich gesprochen vergleicht Shazam nun das kleine Spektogramm der Aufnahme mit allen Spektogrammen aus der Datenbank.

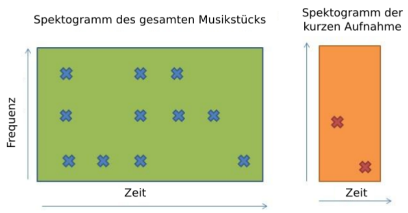


Bild 2: Spektogramm vom Original-Song (links) und von der kurzen Aufnahme (rechts) [1]

Nun überprüft Shazam im ganzen Musikstück durch Überlagerung der beiden Spektogramme, ob das kleine Spektogramm der Aufnahme mit einem Teil des Spektogramms eines beliebigen Musikstücks aus der Datenbank übereinstimmt.

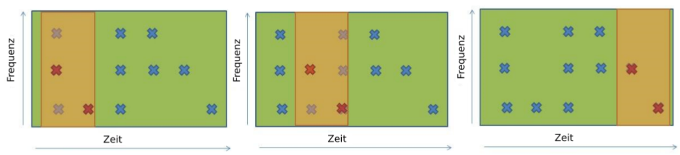


Bild 3: Überlagerung der beiden Spektogramme zum Finden einer Übereinstimmung

In diesem Beispiel wurde eine perfekte Übereinstimmung zwischen der kurzen Aufnahme und dem Ende des Musikstücks gefunden. Dieser Vergleich muss mit jedem gespeicherten Song in der Datenbank von Shazam geschehen, bis man eine Übereinstimmung wie im obigen Beispiel erzielt.

Problembeschreibung | Eine Aufnahme in der Datenbank finden

Wir wollen uns ein vereinfachtest Modell anschauen, wie Shazam die Datenpunkte in dem Spektogramm speichert und die Datenbank nach dem richtigen Musikstück durchsucht. Dazu erhaltet ihr

[1] Dies ist nur eine vereinfachte Darstellung. In der Realität enthält ein Spektogramm wesentlich mehr Punkte.

das beiliegende Spektogramm einer kurzen Audioaufnahme, zu dem ihr den passenden Song in einer bereits programmierten Datenbank finden sollt. Das Matlab-Skript `Shazam_Modell.m` ist bereits für euch geöffnet. Bearbeitet nun die folgenden Schritte:

Schritt 1 | Target Zones bilden

Um den Fingerabdruck der Aufnahme zu speichern, unterteilt Shazam die Datenpunkte im Spektogramm in Gruppen ein, den sogenannten *Target Zones*. Eine Target Zone besteht dabei aus jeweils fünf, zeitlich aufeinander folgenden Datenpunkten, wobei sich zwei benachbarte *Target Zones* immer genau zwei Datenpunkte teilen.

a) Nummeriert die Datenpunkte im Spektogramm auf dem beiliegendem Blatt nach der Zeit durch.

b) Zeichnet die *Target Zones* wie oben beschrieben in euer Spektogramm ein.

c) Jede Target Zone braucht einen Referenzpunkt, den sogenannten *Anchor Point*. Der *Anchor Point* ist dabei der erste Punkt in einer Target Zone. Umkreist die *Anchor Points* in eurem Frequenzspektrum.

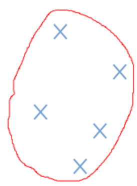


Bild 4: *Target Zone*

Schritt 2 | Den Fingerabdruck mathematisch speichern

Für den Fingerabdruck kreiert Shazam für jeden Datenpunkt eine sogenannte Adresse und speichert diese ab. Die Liste aus den einzelnen Adressen bildet dann den Fingerabdruck. Die Adresse eines Datenpunktes besteht dabei aus folgenden Daten:

1. Die Frequenz des *Anchor Points*, in dessen *Target Zone* sich der Datenpunkt befindet
2. Die Frequenz des Datenpunkts
3. Die Zeitdifferenz zwischen *Anchor Point* und Datenpunkt

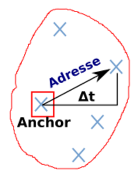


Bild 5: *Adresse*

$$\Rightarrow \text{Adresse: } (\text{Frequenz } \textit{Anchor Point}, \text{ Frequenz Datenpunkt}, \ \Delta t)$$

Bestimmt in der beiliegenden Tabelle die fehlenden Felder der Adressen. Anschließend ergänzt ihr die fehlenden Daten für die jeweiligen Datenpunkte im Matlab-Skript `Shazam_Modell.m` für NaN ersetzen.

Schritt 3 | Die Datenbank durchsuchen

Die Adressen jedes Musikstückes sind in der Datenbank von Shazam gespeichert. Darüber hinaus wird in der Datenbank jede Adresse mit der Songidentität (Interpret, Songtitel) verknüpft. Hier ein kleines Beispiel:

$$(10, 20, 1) \rightarrow \text{[Haus am See von Peter Fox]}$$

Die Adressen der Audioaufnahme werden zur Datenbank geschickt und mit den dortigen Adressen überprüft. Bei einer Übereinstimmung der Adressen speichert Shazam die möglichen Verknüpfungen (Songidentität) und zählt am Ende die Anzahl der Übereinstimmungen mit den möglichen Musikstücken.
Gebt nun in Matlab den Befehl `Datenbank_Durchsuchen` ein. Euch wird eine Matrix mit drei Spalten angezeigt. Der erste Eintrag gibt euch an, zu wie viel Prozent die Adressen eure Aufnahme mit den Adressen der einzelnen, gespeicherten Songs in der Datenbank übereinstimmen. Der zweite und dritte Eintrag geben euch Auskunft über den Songtitel und den Interpreten. Interpretiert das resultierende Ergebnis, das im Command Window erscheint.

Tabelle 1: Adressen der einzelnen Datenpunkte

		Adresse		
Datenpunkt	Zeit *Anchor Point*	Frequenz *Anchor Point*	Frequenz Datenpunkt	Δt
1	1	20	20	0
2	1	20	40	1
3	1	20	10	2
4	1	20	20	3
5	1	20	30	4
4	4	20	20	0
5	4	20	30	1
6	4	20	50	2
7	4	20	10	3
8	4	20		
7	7	10	10	0
8	7	10	40	1
9	7	10	20	2
10	7			3
11	7	10	10	4
10	10	50	50	0
11	10	50	10	1
12	10	50	30	2
13	10	50	40	3
14	10	50	20	4
13	13	40	40	0
14	13	40	20	1
15	13	40	50	2
16	13	40	30	3
17	13	40	20	4
16	16			
17	16			
18	16			
19	16			
20	16			

3/3

B.8. Zusatzmaterial zu Arbeitsblatt 3

B.8.1. Spektogramm

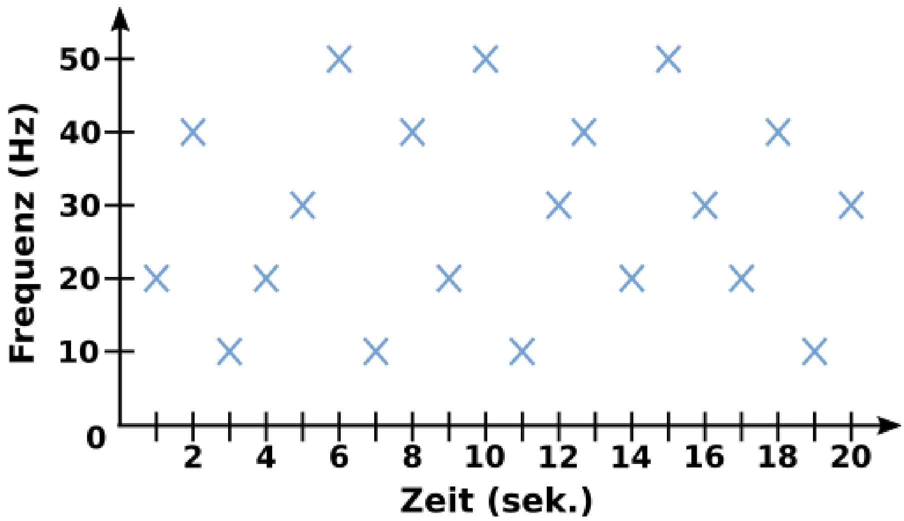
CAMMP day

Wie funktioniert eigentlich -Shazam

CAMMP RWTH AACHEN UNIVERSITY

- zu Arbeitsblatt 3-

Spektogramm einer Audioaufnahme



1/1

CAMMP day

Wie funktioniert eigentlich -Shazam

-Arbeitsblatt 4-

Modellverbesserung

Nach Bearbeitung der drei Schritte habt ihr als Ergebnis zwei mögliche Musikstücke erhalten, die anscheinend mit eurer Aufnahme zu 100% übereinstimmen. Ziel ist es nun das Modell zu erweitern, sodass wir zwischen den beiden Songs den Passenden identifizieren können.

Schritt 4 | Zeitdifferenz betrachten

Um aus den zwei möglichen Songs den Richtigen auszuwählen, müssen wir die Zeitdifferenz zwischen unserer Aufnahme und den beiden Songs untersuchen. Findet Shazam in der Datenbank einen Treffer, heißt das nämlich auch, dass alle Datenpunkte in der Aufnahme in genau der gleichen Reihenfolge auch im Musikstück wieder gefunden werden. Das bedeutet, dass die Reihenfolge und damit auch der zeitliche Versatz zwischen den aufeinander folgenden Datenpunkten in der Aufnahme und den Datenpunkten im Musikstück der Datenbank immer gleich bleiben muss. Um den zeitlichen Versatz, sprich die Zeitdifferenz, zu betrachten, vergleichen wir jeweils den Zeitpunkt des *Anchor Points* unserer losgeschickten Adresse und den Zeitpunkt des *Anchor Points*, der für diese Adresse im jeweiligen Song in der Datenbank gespeichert ist. Diesen zeitlichen Vergleich zwischen den *Anchor Points* führt Shazam für jede Adresse durch.

a) Gebt im Matlab-Skript `Shazam_Modell.m` im Command Window den Befehl `Treffer_Song05` ein, worauf ihr eine Matrix mit 4 Spalten erhaltet. Die ersten drei Einträge in jeder Zeile stehen für die losgeschickten Adressen von eurer Aufnahme (Frequenz *Anchor Point*, Frequenz Datenpunkt, Zeitdifferenz), die mit Song05 übereinstimmen. Der letzte Eintrag gibt euch den Zeitpunkt des *Anchor Points* an, der zu der jeweiligen Adresse im Spektogramm von Song05 gehört. Bildet nun exemplarisch die Zeitdifferenzen für einige Adresse wie folgt:

$$\Delta t = \text{Zeitpunkt } \textit{Anchor Point} \text{ im Song05} - \text{Zeitpunkt } \textit{Anchor Point} \text{ in eurer Aufnahme}$$

Hinweis: Den zugehörigen Zeitpunkt des *Anchor Points* eurer bestimmten Adresse findet ihr in der Tabelle von Arbeitsblatt 3.

b) Gebt nun im Command Window den Befehl `Treffer_Song02` ein und bestimmt analog die Zeitdifferenzen für ausgewählte Adressen.

c) Welcher Song passt nun zu der Aufnahme? Mit dem Befehl `RTS` führt Matlab die Suche nach dem richtigen Song mit Betrachtung der Zeitdifferenz durch und zeigt euch den passenden Song im Command Window an. Gebt den Befehl `RTS` im Command Window ein und überprüft euer Ergebnis.

Modellanwendung | Audiosignale der andren Gruppen bestimmen

Zum Schluss sollt ihr nun das neue Modell testen, indem ihr versucht die erstellten Audiosignale von drei anderen Gruppen in der Datenbank zu suchen und den Songtitel sowie Interpreten zu diesen Gruppen herauszufinden. Ladet euch den Ordner `DataBase` aus der *Dropbox* herunter und fügt ihn in den Ordner `Arbeitsblatt 3 & 4` hinzu. Nun gebt ihr im Matlab-Skript `Shazam_Modell` in Zeile 198 unter `file` für 00 die Zahl der Gruppe ein, dessen Audiosignal ihr in der Datenbank suchen möchtet. Gebt anschließend im Command Window den Befehl `Ton_Suchen` ein. Im Command Window wird euch dann der passende Song der jeweiligen Gruppe angezeigt. Notiert euch in 1 auf der Rückseite zu den drei Gruppen den Songtitel und den Interpret.

Tabelle 1: Songtitel und Interpret der Songs

Gruppennummer	Songtitel	Interpret

2/2

B.10. Lösung zu Arbeitsblatt 3 & 4

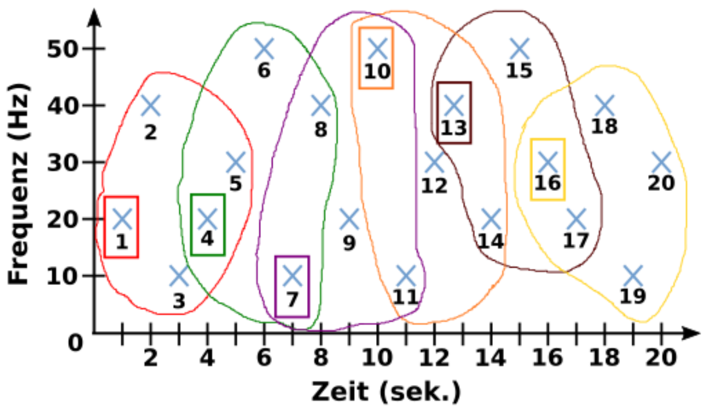
CAMMP day

Wie funktioniert eigentlich -Shazam

Musterlösung Arbeitsblatt 3 & 4

Schritt 1 | Target Zones bilden

Werden die Datenpunkten im Spektogramm durchnummeriert und die *Target Zones* sowie *Anchor Points* eingezeichnet, erhält man folgendes Bild:



Schritt 2 | Den Fingerabdruck mathematisch speichern

Die fehlenden Felder (rot markiert) in der Tabelle müssen wie folgt ausgefüllt werden:

Tabelle 1: Fehlende Adressen der einzelnen Datenpunkte

		Adresse		
Datenpunkt	Zeit *Anchor Point*	Frequenz *Anchor Point*	Frequenz Datenpunkt	Δt
8	4	20	40	4
10	7	10	50	3
16	16	30	30	0
17	16	30	20	1
18	16	30	40	2
19	16	30	10	3
20	16	30	30	4

Die Adressen der Datenpunkte werden in Matlab wie folgt eingetragen. Für jede *Target Zone* wird zunächst der Zeitpunkt und die Frequenz des *Anchor Points* eingetragen. Anschließend werden für jeden der fünf Datenpunkte die Frequenz und die Zeitdifferenz bezüglich des *Anchor Points* notiert. Die fehlenden Tabelleneinträge werden in Matlab in der entsprechenden Zeile für NaN eingefügt. Hier sind die fehlenden Einträge in Matlab aufgelistet:

```
%—— Zweite Targetzone —— %

% Zeitpunkt Anchor Point 2:
timeAnchor_2 = 4;

% Datenpunkte 8:
anchorFreq_2 = 20;
pointFreq_8 = 40;
T_delta_8 = 4;
```

```
%—— Dritte Targetzone —— %

% Zeitpunkt Anchor Point 3:
timeAnchor_3 = 7;

% Datenpunkte 10:
anchorFreq_3 = 10;
pointFreq_10 = 50;
T_delta_10 = 3;



%—— Sechste Targetzone —— %

% Zeitpunkt Anchor Point 6:
timeAnchor_6 = 16;

% Datenpunkte 16:
anchorFreq_6 = 30;
pointFreq_16_2 = 30;
T_delta_16_2 = 0;

% Datenpunkte 17:
anchorFreq_6 = 30;
pointFreq_17_2 = 20;
T_delta_17_2 = 1;

% Datenpunkte 18:
anchorFreq_6 = 30;
pointFreq_18 = 40;
T_delta_18 = 2;

% Datenpunkte 19:
anchorFreq_6 = 30;
pointFreq_19 = 10;
T_delta_19 = 3;

% Datenpunkte 20:
anchorFreq_6 = 30;
pointFreq_20 = 30;
T_delta_20 = 4;
```

Schritt 3 | Die Datenbank durchsuchen

Mit dem Befehl `Datenbank_Durchsuchen` und den in Schritt 2 in Matlab eingetragenen Adressen erhält man im Command Window folgendes Ergebnis:

```
[    100]    'Song05,'    'Artist05'
[    100]    'Song02,'    'Artist02'
[73.3333]    'Song04,'    'Artist04'
[23.3333]    'Song01,'    'Artist01'
```

In der linken Spalten steht in Prozentangaben die Übereinstimmung der bestimmten Adressen mit den Adressen der einzelnen Songs, die in der Datenbank gespeichert sind. Auffällig ist, dass sowohl für Song05, als auch für Song02 eine Übereinstimmung von 100% gefunden wurde. Der Grund dafür ist die fehlende Betrachtung der Zeitdifferenz zwischen den einzelnen Songs in der Datenbank und der vorliegenden Aufnahme, die erste in Schritt 4 zur Modellverbesserung erfolgt.

Schritt 4 | Zeitdifferenz betrachten

Mit den Eingaben `Treffer_Song02` und `Treffer_Song05` erhält man im Command Window von Matlab eine Matrix, die in den ersten drei Einträgen die zur Datenbank geschickten Adresse zu den einzelnen Datenpunkte anzeigt. In der letzten Spalte steht jeweils der Zeitpunkt des *Anchor Points*, der in der Datenbank im jeweiligen Song der losgeschickten Adresse zugeordnet wird.

(losgeschickte Adresse) [[zugehöriger *Anchor Point* im Song]

Die Zeitdifferenz wird zwischen den Zeitpunkten des Anchor Points der losgeschickten Adresse und dem der Adresse zugeordneten Anchor Point im Song gebildet.

a) Zeitdifferenzen für Song05

1. Adresse: $\Delta t = 22 - 1 = 21$
2. Adresse: $\Delta t = 22 - 1 = 21$
3. Adresse: $\Delta t = 22 - 1 = 21$
4. Adresse: $\Delta t = 49 - 1 = 48$
5. Adresse: $\Delta t = 49 - 1 = 48$
6. Adresse: $\Delta t = 34 - 4 = 30$
7. Adresse: $\Delta t = 34 - 4 = 30$
8. Adresse: $\Delta t = 34 - 4 = 30$
9. Adresse: $\Delta t = 85 - 4 = 81$
10. Adresse: $\Delta t = 49 - 4 = 45$
11. Adresse: $\Delta t = 58 - 7 = 51$
12. Adresse: $\Delta t = 58 - 7 = 51$
13. Adresse: $\Delta t = 58 - 7 = 51$
14. Adresse: $\Delta t = 58 - 7 = 51$
15. Adresse: $\Delta t = 58 - 7 = 51$

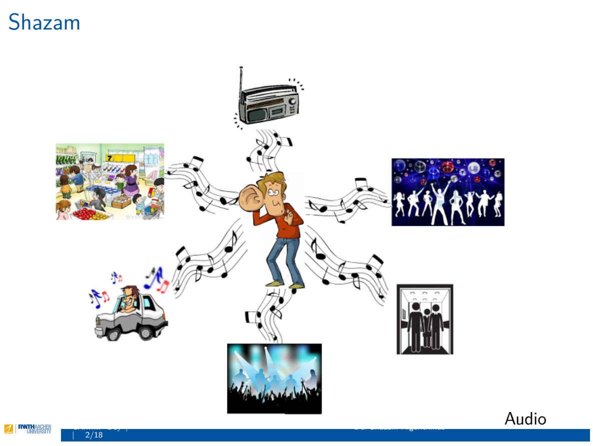
b) Zeitdifferenzen für Song02

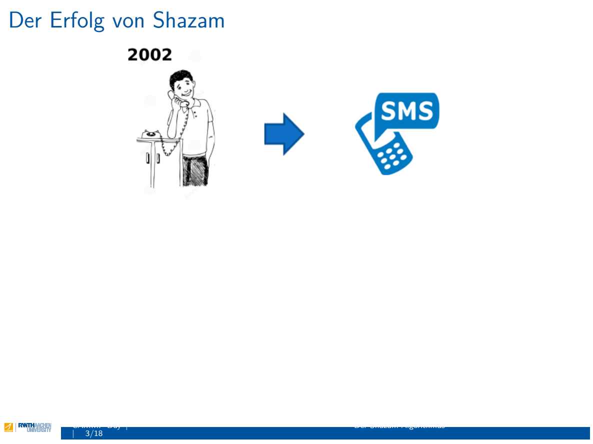
1. Adresse: $\Delta t = 52 - 1 = 51$
2. Adresse: $\Delta t = 52 - 1 = 51$
3. Adresse: $\Delta t = 52 - 1 = 51$
4. Adresse: $\Delta t = 52 - 1 = 51$
5. Adresse: $\Delta t = 52 - 1 = 51$
6. Adresse: $\Delta t = 55 - 4 = 51$
7. Adresse: $\Delta t = 55 - 4 = 51$
8. Adresse: $\Delta t = 55 - 4 = 51$
9. Adresse: $\Delta t = 55 - 4 = 51$
10. Adresse: $\Delta t = 55 - 4 = 51$
11. Adresse: $\Delta t = 58 - 7 = 51$
12. Adresse: $\Delta t = 58 - 7 = 51$
13. Adresse: $\Delta t = 58 - 7 = 51$
14. Adresse: $\Delta t = 58 - 7 = 51$
15. Adresse: $\Delta t = 58 - 7 = 51$

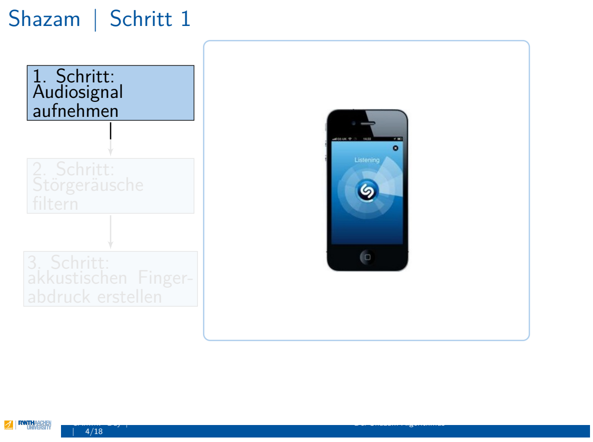
Es fällt schnell auf, dass die Zeitdifferenz zwischen unserer Aufnahme und Song02 immer $51s$ beträgt, während sie für Song05 ständig variiert. **Damit kann der Aufnahme nun eindeutig Song02 zugeordnet werden.**

C. Vorträge

C.1. Einführungsvortrag







Shazam | Schritt 1

1. Schritt: Audiosignal aufnehmen

2. Schritt: Störgeräusche filtern

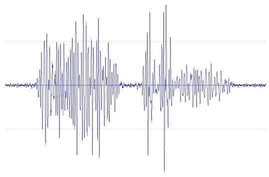
3. Schritt: akkustischen Fingerabdruck erstellen

Shazam | Schritt 2

1. Schritt: Audiosignal aufnehmen

2. Schritt: Störgeräusche filtern

3. Schritt: akkustischen Fingerabdruck erstellen



Shazam | Schritt 3

1. Schritt: Audiosignal aufnehmen

2. Schritt: Störgeräusche filtern

3. Schritt: akkustischen Fingerabdruck erstellen

Shazam | Schritt 4

3. Schritt: akkustischen Fingerabdruck erstellen

4. Schritt: Datenbank durchsuchen

5. Schritt: Ergbnis dem User schicken

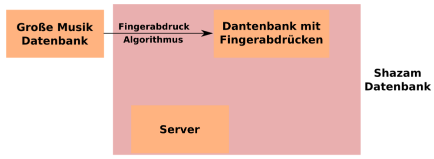
Shazam | Schritt 5

3. Schritt: akkustischen Fingerabdruck erstellen

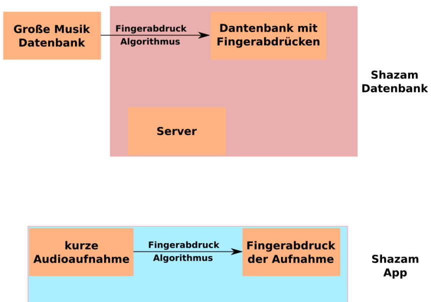
4. Schritt: Datenbank durchsuchen

5. Schritt: Ergbnis dem User schicken

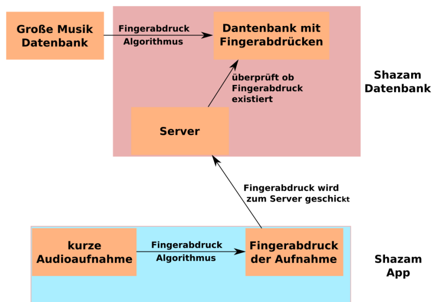
Shazam | Zusammenfassung



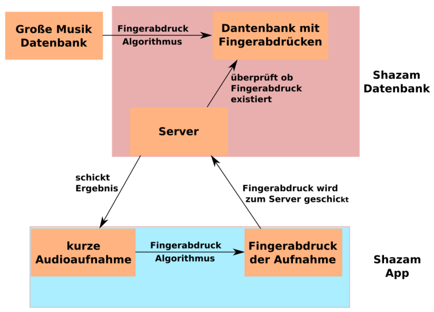
Shazam | Zusammenfassung



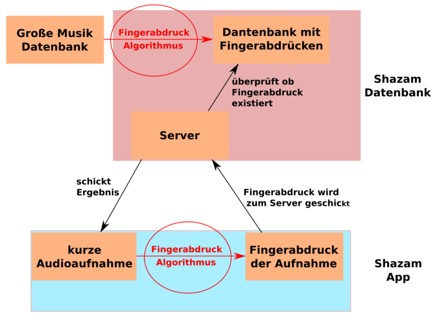
Shazam | Zusammenfassung



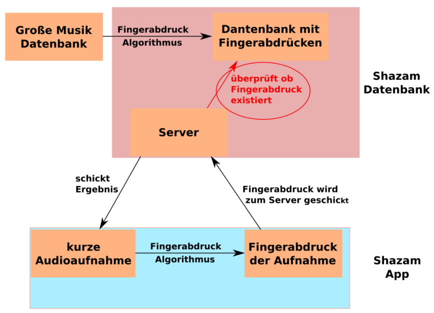
Shazam | Zusammenfassung



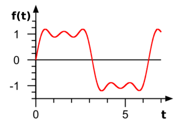
Erstes Ziel



Zweites Ziel

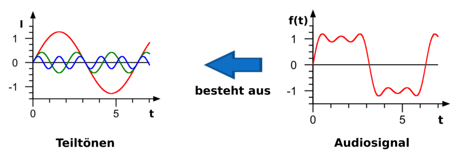


Wie ist ein Audiosignal aufgebaut?



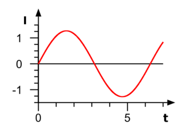
Audiosignal

Wie ist ein Audiosignal aufgebaut?



Wie ist ein Audiosignal aufgebaut?

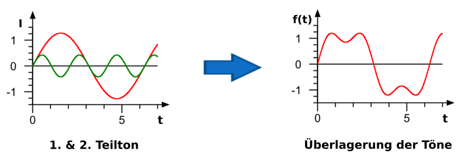
Überlagerung der Teiltöne



1. Teilton

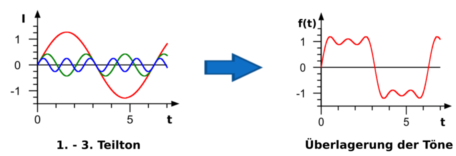
Wie ist ein Audiosignal aufgebaut?

Überlagerung der Teiltöne



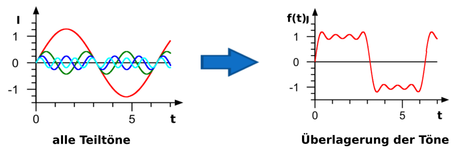
Wie ist ein Audiosignal aufgebaut?

Überlagerung der Teiltöne



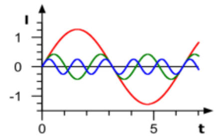
Wie ist ein Audiosignal aufgebaut?

Überlagerung der Teiltöne



ergibt das Ausgangssignal

Wie können wir mathematisch die Teiltöne beschreiben?

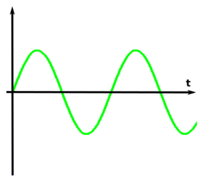


Wie können wir mathematisch die Teiltöne beschreiben?

- Mathematisch gesehen Teiltöne nichts anderes als eine periodische Sinus- oder Cosiunusschwingung.

Wie können wir mathematisch die Teiltöne beschreiben?

- Mathematisch gesehen Teiltöne nichts anderes als eine periodische Sinus- oder Cosiunusschwingung.
- Die Stimmgabel erzeugt z.B. einen fast perfekten Sinuston

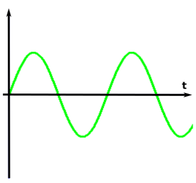


Was sind charakteristische Eigenschaften eines Teiltons?

Was sind charakteristische Eigenschaften eines Teiltons?

- Amplitude
- Frequenz
- Zeit

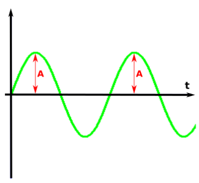
Einen Teilton mathematisch modellieren



$f(t) = A \cdot \sin(f \cdot 2 \cdot \pi \cdot t)$

Sinusschwingung einer Stimmgabel

Einen Teilton mathematisch modellieren

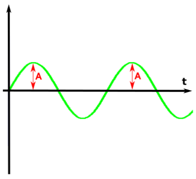


$f(t) = A \cdot \sin(f \cdot 2 \cdot \pi \cdot t)$

- A ist die Amplitude und gibt die maximale Auslenkung bzw. die Lautstärke des Tons an;

Amplitude des Sinustons

Einen Teilton mathematisch modellieren

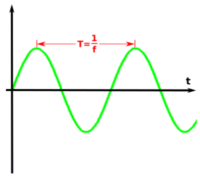


leiserer Ton

$$f(t) = A \cdot \sin(f \cdot 2 \cdot \pi \cdot t)$$

- A ist die Amplitude und gibt die maximale Auslenkung bzw. die Lautstärke des Tons an;

Einen Teilton mathematisch modellieren

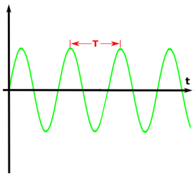


Frequenz des Sinustons

$$f(t) = A \cdot \sin(f \cdot 2 \cdot \pi \cdot t)$$

- A ist die Amplitude und gibt die maximale Auslenkung bzw. die Lautstärke des Tons an;
- f für die Frequenz und gibt die Tonhöhe an und kann indirekt über die Periode $T = \frac{1}{f}$, also dem Abstand zwischen zwei Maxima abgelesen werden.

Einen Teilton mathematisch modellieren

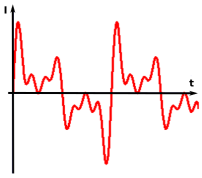


höherer Ton

$$f(t) = A \cdot \sin(f \cdot 2 \cdot \pi \cdot t)$$

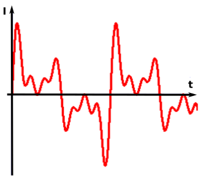
- A ist die Amplitude und gibt die maximale Auslenkung bzw. die Lautstärke des Tons an;
- f für die Frequenz und gibt die Tonhöhe an und kann indirekt über die Periode $T = \frac{1}{f}$, also dem Abstand zwischen zwei Maxima abgelesen werden.

Wie erhalten wir die einzelnen Teiltöne eines Klangs/ Tongemisches?



Klang einer Klarinette

Wie erhalten wir die einzelnen Teiltöne eines Klangs/ Tongemisches?



Klang einer Klarinette

- Wie stellen wir das Audiosignal als Sinus und Cosinusfunktionen dar?

Satz von Fourier

Joseph Fourier (1768 – 1830)

Satz von Fourier
Jede periodische Funktion lässt sich als Summe von Sinus- und Cosinusfunktionen darstellen.

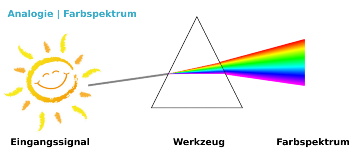
Satz von Fourier

Joseph Fourier (1768 – 1830)

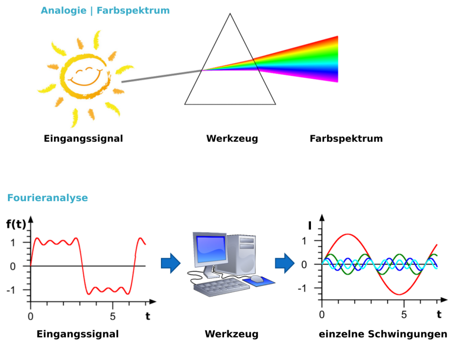
Satz von Fourier
Jede periodische Funktion lässt sich als Summe von Sinus- und Cosinusfunktionen darstellen.

- Entwickelte die Fourieranalyse

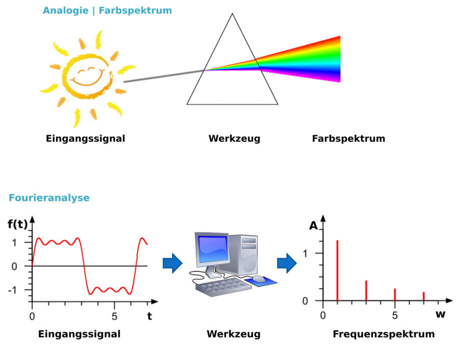
Fourieranalyse



Fourieranalyse



Fourieranalyse



Modellierungskreislauf

Reales Problem (Erkennung von tausenden Songs) — vereinfachen → **Vereinfachtes Problem**

Vereinfachtes Problem — mathematisch beschreiben → **Mathematisches Modell**

Mathematisches Modell — berechnen → **Mathematische Lösung**

Mathematische Lösung — interpretieren → **Reales Problem**

Modellierungskreislauf

Reales Problem (Erkennung von tausenden Songs) — vereinfachen → **Vereinfachtes Problem** (Erkennung einer Beispielaufnahme)

Vereinfachtes Problem — mathematisch beschreiben → **Mathematisches Modell**

Mathematisches Modell — berechnen → **Mathematische Lösung**

Mathematische Lösung — interpretieren → **Reales Problem**

Modellierungskreislauf

Reales Problem (Erkennung von tausenden Songs) — vereinfachen → **Vereinfachtes Problem** (Erkennung einer Beispielaufgabe)

Vereinfachtes Problem — mathematisch beschreiben → **Mathematisches Modell**

Mathematisches Modell — berechnen → **Mathematische Lösung**

Mathematische Lösung — interpretieren → **Reales Problem**

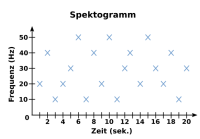
Modellierungskreislauf

Reales Problem (Erkennung von tausenden Songs) — vereinfachen → **Vereinfachtes Problem** (Erkennung einer Beispielaufgabe)

Vereinfachtes Problem — mathematisch beschreiben → **Mathematisches Modell** (Spektogramm einer Aufnahme/ eines Songs)

Mathematisches Modell — berechnen → **Mathematische Lösung**

Mathematische Lösung — interpretieren → **Reales Problem**



Modellierungskreislauf

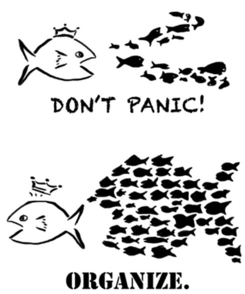
Reales Problem (Erkennung von tausenden Songs) — vereinfachen → **Vereinfachtes Problem** (Erkennung einer Beispielaufgabe)

Vereinfachtes Problem — mathematisch beschreiben → **Mathematisches Modell** (Spektogramm einer Aufnahme/ eines Songs)

Mathematisches Modell — berechnen → **Mathematische Lösung**

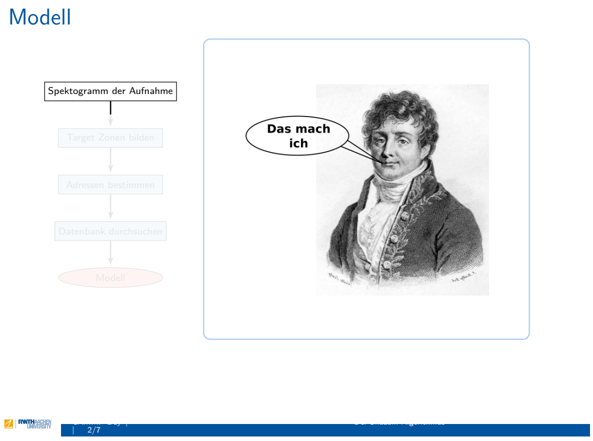
Mathematische Lösung — interpretieren → **Reales Problem**

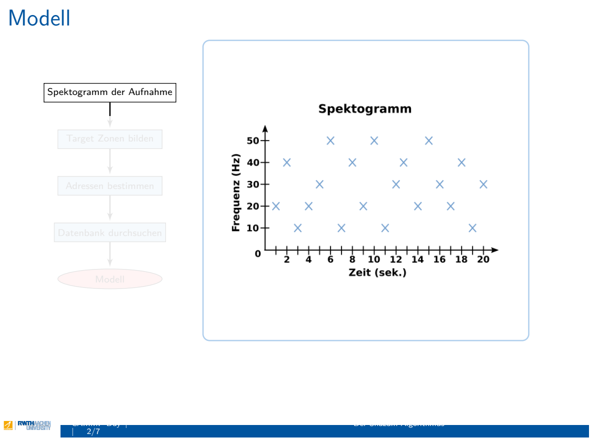
Wie findest du Antworten auf all diese Fragen?

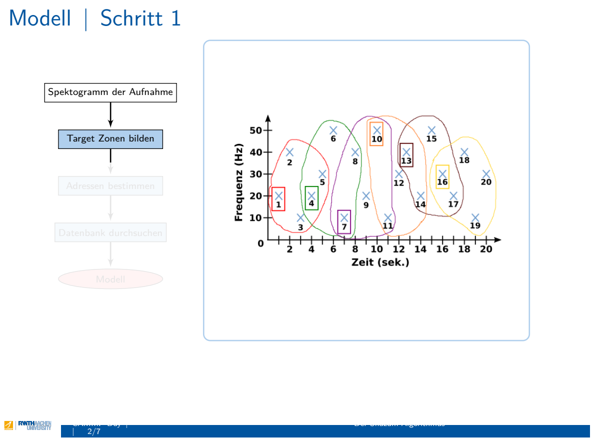


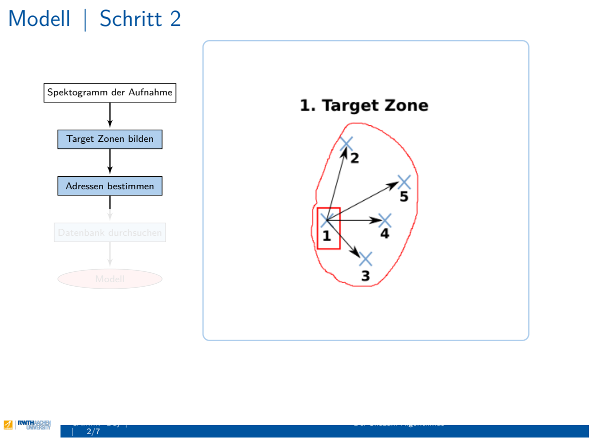
- Bearbeite die Arbeitsblätter!
- Nutze MATLAB!
- Arbeite im Team mit deinem Partner!
- Nutze die Hilfekarten!
- Frag' die Betreuer!
- Nutze das Internet!
- Präsentiere Ergebnisse deinen Mitschülern!
- Hilf deinen Mitschülern!

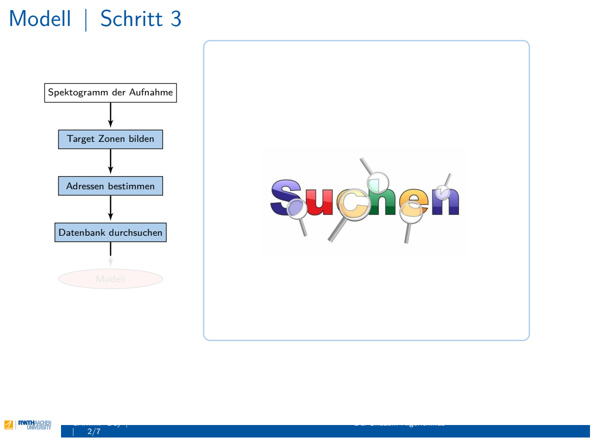
C.2. Zwischenvortrag











Modell | Schritt 3

Spektogramm der Aufnahme → Target Zonen bilden → Adressen bestimmen → Datenbank durchsuchen → Modell

Modell

Spektogramm der Aufnahme → Target Zonen bilden → Adressen bestimmen → Datenbank durchsuchen → Modell

[100]	'Song05,'	'Artist05'
[100]	'Song02,'	'Artist02'
[73.3333]	'Song04,'	'Artist04'
[23.3333]	'Song01,'	'Artist01'

Modellierungskreislauf

Reales Problem Erkennung von tausenden Songs — vereinfachen → **Vereinfachtes Problem** Erkennung einer Beispielaufgabe

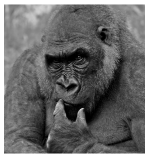
mathematisch beschreiben → **Mathematisches Modell** Spektogramm einer Aufnahme/ eines Songs

berechnen → **Mathematische Lösung**

interpretieren → **Reales Problem**

Modellierungskreislauf

Reales Problem Erkennung von tausenden Songs — vereinfachen → **Vereinfachtes Problem** Erkennung einer Beispielaufgabe



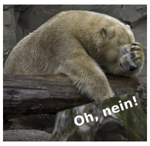
mathematisch beschreiben → **Mathematisches Modell** Spektogramm einer Aufnahme/ eines Songs

berechnen → **Mathematische Lösung**

interpretieren → **Reales Problem**

Modellierungskreislauf

Reales Problem Erkennung von tausenden Songs — vereinfachen → **Vereinfachtes Problem** Erkennung einer Beispielaufgabe



mathematisch beschreiben → **Mathematisches Modell** Spektogramm einer Aufnahme/ eines Songs

berechnen → **Mathematische Lösung** 2 Songs passen zur Aufnahme

interpretieren → **Reales Problem**

Modellverbesserung

Reales Problem Erkennung von tausenden Songs — vereinfachen → **Vereinfachtes Problem** Erkennung einer Beispielaufgabe

mathematisch beschreiben → **Mathematisches Modell** Spektogramm einer Aufnahme/ eines Songs

berechnen → **Mathematische Lösung**

interpretieren → **Reales Problem**

Modellverbesserung

Reales Problem Erkennung von tausenden Songs — vereinfachen → **Vereinfachtes Problem** Erkennung einer Beispielaufgabe

mathematisch beschreiben → **Mathematisches Modell** Spektogramm einer Aufnahme/ eines Songs

berechnen → **Mathematische Lösung** 2 Songs passen zur Aufnahme

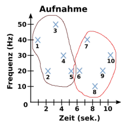
interpretieren → **Reales Problem**

Modellverbesserung | Reihenfolge der Datenpunkte

- Kommen die Datenpunkte in den Songs in der Datenbank genau in der gleichen Reihenfolge vor wie die Datenpunkte in der Aufnahme?

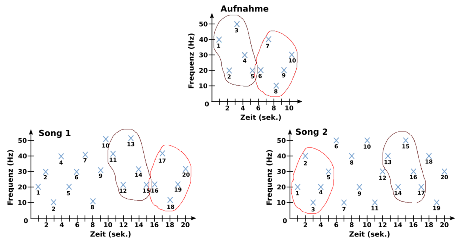
Modellverbesserung | Reihenfolge der Datenpunkte

- Kommen die Datenpunkte in den Songs in der Datenbank genau in der gleichen Reihenfolge vor wie die Datenpunkte in der Aufnahme?



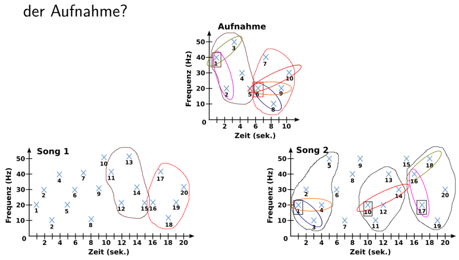
Modellverbesserung | Reihenfolge der Datenpunkte

- Kommen die Datenpunkte in den Songs in der Datenbank genau in der gleichen Reihenfolge vor wie die Datenpunkte in der Aufnahme?



Modellverbesserung | Reihenfolge der Datenpunkte

- Kommen die Datenpunkte in den Songs in der Datenbank genau in der gleichen Reihenfolge vor wie die Datenpunkte in der Aufnahme?



Modellverbesserung | Reihenfolge der Datenpunkte

- Kommen die Datenpunkte in den Songs in der Datenbank genau in der gleichen Reihenfolge vor wie die Datenpunkte in der Aufnahme?

- gleiche Adressen können in verschiedenen Songs zu unterschiedlichen Zeiten vorkommen

Modellverbesserung | Reihenfolge der Datenpunkte

- Kommen die Datenpunkte in den Songs in der Datenbank genau in der gleichen Reihenfolge vor wie die Datenpunkte in der Aufnahme?

- gleiche Adressen können in verschiedenen Songs zu unterschiedlichen Zeiten vorkommen

Wichtig: Zeitdifferenz zwischen Aufnahme und Musikstücken in der Datenbank betrachten

Modellverbesserung | Zeitdifferenz betrachten

Idee: Bestimme die Zeitdifferenz zwischen dem Anchor Point der losgeschickten Adresse und dem Anchor Point, den die Datenbank zur Adresse zurückgibt.

Modellverbesserung | Zeitdifferenz betrachten

Idee: Bestimme die Zeitdifferenz zwischen dem Anchor Point der losgeschickten Adresse und dem Anchor Point, den die Datenbank zur Adresse zurückgibt.

Adresse	Übereinstimmung in der Datenbank
(40, 10, 2) →	[78, Haus am See]

Modellverbesserung | Zeitdifferenz betrachten

Idee: Bestimme die Zeitdifferenz zwischen dem Anchor Point der losgeschickten Adresse und dem Anchor Point, den die Datenbank zur Adresse zurückgibt.

Adresse	Übereinstimmung in der Datenbank
(40, 10, 2) →	[78, Haus am See]

Anchor im Musikstück

Modellverbesserung | Zeitdifferenz betrachten

Idee: Bestimme die Zeitdifferenz zwischen dem Anchor Point der losgeschickten Adresse und dem Anchor Point, den die Datenbank zur Adresse zurückgibt.

Anchor Point zu der Aufnahme	Adresse		Übereinstimmung in der Datenbank
[3]	(40, 10, 2)	→	[78, Haus am See]

Modellverbesserung | Zeitdifferenz betrachten

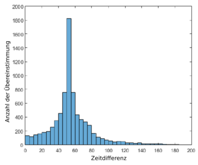
Idee: Bestimme die Zeitdifferenz zwischen dem Anchor Point der losgeschickten Adresse und dem Anchor Point, den die Datenbank zur Adresse zurückgibt.

Anchor Point zu der Aufnahme	Adresse		Übereinstimmung in der Datenbank
[3]	(40, 10, 2)	→	[78, Haus am See]

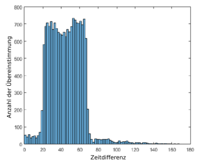
Δt = 75

Modellverbesserung | Zeitdifferenz betrachten

Idee: Bestimme die Zeitdifferenz zwischen dem Anchor Point der losgeschickten Adresse und dem Anchor Point, den die Datenbank zur Adresse zurückgibt.



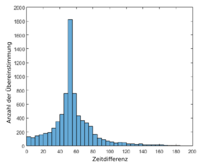
Hohe Übereinstimmung bei einer Zeitdifferenz



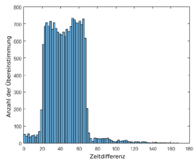
Keine hohe Übereinstimmung bei einer Zeitdifferenz

Modellverbesserung | Zeitdifferenz betrachten

Idee: Bestimme die Zeitdifferenz zwischen dem Anchor Point der losgeschickten Adresse und dem Anchor Point, den die Datenbank zur Adresse zurückgibt.



⇒ Song passt zur Aufnahme



⇒ Song passt nicht zur Aufnahme

D. Programm

CAMMP Day

Wie funktioniert eigentlich -Shazam

Programm

Information für Schülerinnen und Schüler

Wie funktioniert eigentlich Shazam?

9:00	Vortrag	Vortrag zu CAMMP science und Einführungsvortrag Shazam
9:30	Partnerarbeit	Modellierung von Tönen
10:30	Pause	
10:45	Partnerarbeit	Fourieranalyse
12:00	Pause	
13:00	Partnerarbeit	Akustischer Fingerabdruck
14:00	Vortrag/ Diskussion	Modellverbesserung
14:10	Partnerarbeit	Modellverbesserung und Anwendung
14:50	Evaluation	

1/1

CAMMP Day

Wie funktioniert eigentlich -Shazam

Methodisches Konzept

für die Dozenten

Material

Blätter:

- ☐ Programm.pdf
- ☐ Evaluationsbogen.pdf
- ☐ ZDI-Unterschriftenl.
- ☐ Gruppennummern.pdf
- ☐ Arbeitsblatt 1.pdf
- ☐ Arbeitsblatt 2.pdf
- ☐ Tontabelle.pdf
- ☐ Arbeitsblatt 3.pdf
- ☐ Spektogramm.pdf
- ☐ Arbeitsblatt 4.pdf
- ☐ Zusatzaufgabe AB 1.pdf
- ☐ Zusatzaufgabe AB 2.pdf
- ☐ Hilfekarte 1.pdf
- ☐ Hilfekarte 2.pdf
- ☐ Hilfekarte 3.pdf
- ☐ Hilfekarte 4.pdf
- ☐ Hilfekarte 5.pdf

Präsentationen:

- ☐ OpeningPresentation.pdf
- ☐ Modellierungsvortrag.pdf
- ☐ Einführungsvortrag.pdf
- ☐ Shazam Zwischenvortrag.pdf

Folien:

- ☐ Loesung_Spek.pdf

Experiment:

- ☐ Stimmgabeln mit Resonanzkörper
- ☐ Anschlagshammer
- ☐ Smartphone mit phyphox-App

Ablauf

8:15	Vorbereitung	☐ Beginn der Vorbereitungen: Laptops in den Raum bringen Wlan aktivieren, Material in den Raum bringen, Experiment vorbereiten (siehe unten) und Beamer anschließen ☐ Gruppennummern austeilen ☐ Programm austeilen ☐ Ordner `Arbeitsblatt 1`, `Arbeitsblatt 2` und `Arbeitsblatt 3 & 4` mit dem passenden Code als Inhalt auf den Desktops erstellen ☐ Matlab-Skript `Tonmodell.m` auf den Laptops öffnen und den Ordner Shazam als Subfolder hinzufügen
9:00	Einführung	☐ `ZDI-Unterschriftenliste` ausfüllen lassen und Unterschriften der Eltern einammeln ☐ `Vortrag_OpeningPresentation.pdf` halten ☐ `Modellierungsvortrag.pdf` halten ☐ `Einführungsvortrag.pdf` halten ☐ Beim Einführungsvortrag Experiment durchführen
9:40	Partnerarbeit (Teil I)	☐ AB `Arbeitsblatt 1.pdf` austeilen ☐ Hinweis auf Fragen an Betreuer während der Partnerarbeit
10:25		☐ Klären, ob alle Songdateien in die Dropbox in den Ordner DataBase kopiert wurden
10:30	Pause	
10:45		☐ Lösungen anhand des Matlab-Skripts *Tonmodell.m* besprechen, indem für jede Aufgabe eine Lösung der SuS eingetragen und mit Matlab angezeigt wird

10:55	Partnerarbeit (Teil II)	☐ Matlab-Skript `Akkordtest.m` auf den Laptops öffnen
		☐ AB `Aufgabenblatt 2.pdf` austeilen
		☐ Einführung und Problembeschreibung gemeinsam lesen
		☐ Verständnisfragen klären
		☐ Folie zum Eintragen der Dreiklänge auf den OHP legen
11:50		☐ Ergebnisse der SuS analog zum ersten Arbeitsblatt anhand des Matlab-Skripts besprechen.
12:00	Pause	☐ Erstellte Töne der SuS in die Datenbank hinzufügen (siehe unten)
13:00	Partnerarbeit (Teil III)	☐ Matlab-Skript `Shazam_Modell.m` auf den Laptops öffnen und den Ordner Shazam als Subfolder hinzufügen
		☐ AB `Arbeitsblatt 3.pdf` und `Spektogramm.pdf` austeilen
		☐ Einführung und Problemstellung gemeinsam lesen
		☐ Verständnisfragen klären
		☐ Wenn SuS mit Aufgabe 1 fertig auf der Folie `Loesung_Spek` gemeinsam mit den SuS die *Target Zones* einzeichnen
parallel	Powerwall	☐ 3D-Simulation in der Gesundheitsforschung
13:30	Zwischenvortrag	☐ `Shazam Zwischenvortrag.pdf` halten und dadurch bisherige Ergebnisse sichern und Modellverbesserung diskutieren
		☐ Auf einer Folie die einzelnen Modellverbesserungen der SuS sammeln und diskutieren bevor auf die Modellverbesserung der Zeitdifferenz eingegangen wird
13:45	Partnerarbeit (Teil IV)	☐ AB `Arbeitsblatt 4.pdf` austeilen
		☐ Folie zum Eintragen der Songtitel und Interpreten auf den OHP legen
14:15	Evaluation	☐ `Evaluationsbogen.pdf` austeilen

Experiment

Das Experiment wird während des Einführungsvortrags bei der Einführung der Amplitude und der Frequenz (**Folie 10**) vorgeführt und wird wie folgt aufgebaut:

1. Öffnet die App phyphox und wählt unter der Überschrift Akustik das Programm „Audio Autokorrelation“ aus.
2. Wählt unter Einstellung (oben rechts die drei Punkte) „ Fernzugriff erlauben“ aus.
3. Bestätigt den anschließenden Aufruf mit ok und gibt die unten erscheinende URL in den Internetbrowser im Laptop ein.

Anschließend kann die App über den Laptop bedient werden. Zur Aufnahme eines Stimmgabeltons muss der „Play-Button“ gedrückt werden. Zur besseren Betrachtung der Schwingungskurve sollte nach dem Anstimmen der Stimmgabel das Experiment und damit die Aufnahme pausiert werden.

Datenbank der SuS erstellen

Zum Erstellen der Datenbank mit den erstellen Audiosignalen der SuS folgende Schritte befolgen:

1. Den Ornder `DataBase` aus der *Dropbox* in den Ordner `Datenbank der SuS erstellen` ziehen.
2. Matlab-Skript `Datenbank_Erstellen` öffnen und ausführen.
3. Anschließend die *Gruppe.txt*-Dateien löschen (nicht die *SongDatabase.txt*-Datei)
4. Ordner `DataBase` mit den *Gruppe.dat*-Dateien sowie der Datei *SongDatabase.txt* wieder in die *Dropbox* hinzufügen.

F. Evaluation

F.1. Evaluationsbogen

CAMMP day

Wie funktioniert eigentlich -Shazam

Evaluation

Schüleruni – 15. Juli 2016

Es besteht immer die Möglichkeit, unsere Programme zu verbessern, und wir würden gerne deine Meinung erfahren. Vielen Dank für deine Rückmeldung.

1. Persönliche Angaben:

Jahrgangsstufe: _____ Geschlecht: ☐ weiblich ☐ männlich

Leistungskurse: ____

2. Bewertung des Workshops Shazam

	Trifft gar nicht zu (--)	Trifft eher nicht zu (-)	Trifft zum Teil zu (+)	Trifft voll zu (++)
Mich hat das Thema Shazam interessiert.	○	○	○	○
Ich habe die Idee hinter Shazam verstanden.	○	○	○	○
Ich weiß nun, was ein akustischer Fingerabdruck ist.	○	○	○	○
Der Umgang mit MATLAB fiel mir schwer.	○	○	○	○
Ich hatte keine Probleme die Aufgaben mit Matlab umzusetzen.	○	○	○	○
Die Aufgabenstellungen waren für mich verständlich formuliert.	○	○	○	○
Die Aufgaben waren zu einfach.	○	○	○	○
Die Aufgaben waren zu schwierig.	○	○	○	○
Ich habe die Aufgaben und ihre Lösung verstanden.	○	○	○	○
Ich konnte eigenständig arbeiten.	○	○	○	○
Durch den Workshop habe ich mathematisches Modellieren besser begriffen.	○	○	○	○
Der CAMMP day hat mir Spaß gemacht.	○	○	○	○

Kommentar:

Ich gebe dem CAMMP day die Schulnote __________

Ich gebe dem Betreuer die Schulnote __________

Bitte wenden...

1/2

Was würdest du am Workshop verändern bzw. verbessern wollen?

3. Weiterführende Fragen:

	Trifft gar nicht zu (--)	Trifft eher nicht zu (-)	Trifft zum Teil zu (+)	Trifft voll zu (++)
1. Der Workshop hat mein Interesse an den Naturwissenschaften und Technik gesteigert.	○	○	○	○
2. Die Anleitungen zu den Aufgaben waren zufriedenstellend und gut verständlich.	○	○	○	○
3. Die Dozenten haben die Inhalte plausibel und klar dargestellt.	○	○	○	○
4. Die Betreuer haben das selbstständige Arbeiten gefördert.	○	○	○	○
5. Es wurden interessante Berufe und Studiengänge vorgestellt.	○	○	○	○
6. Ich kann mir gut vorstellen, später ein Studium oder eine Ausbildung im Bereich Technik oder Naturwissenschaften aufzunehmen.	○	○	○	○
7. Ich würde einen solchen Workshop gerne noch einmal besuchen.	○	○	○	○

Kommentar:

Abschließender persönlicher Kommentar (z.B. Lob, Kritik, Verbesserungsvorschläge):

F.2. Auswertung der Evaluation

CAMMP Day

Wie funktioniert eigentlich GPS, und was hat das mit Mathe zu tun?

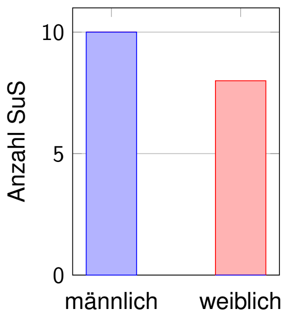
Evaluation

CAMMP day Schüleruni

15 Juli 16

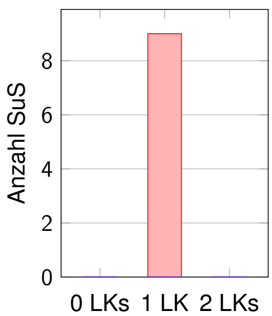
1 Persönliche Angaben

1.1 Geschlecht



m: 10 w: 8

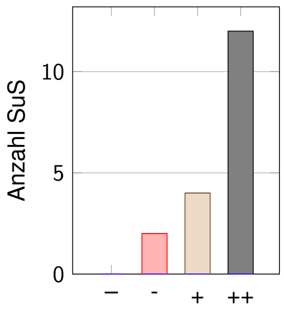
1.2 Anzahl der LKs in Mathematik, Informatik und Physik



0 LK: 0 1 LK: 9 2 LK: 0

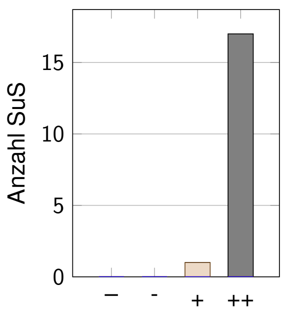
2 Bewertung des Workshops

2.1 Mich hat das Thema Shazam interessiert



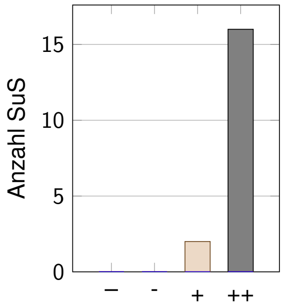
–: 0 -: 2 +: 4 ++: 12

2.2 Ich habe die Idee hinter Shazam verstanden



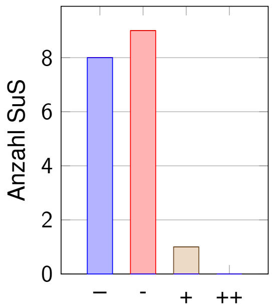
–: 0 -: 0 +: 1 ++: 17

2.3 Ich weißnun, was ein akustischer Fingerabdruck ist



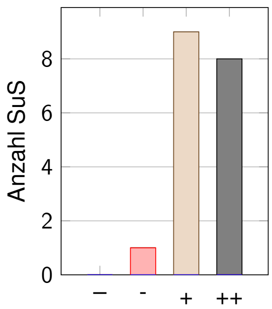
–: 0 -: 0 +: 2 ++: 16

2.4 Der Umgang mit MATLAB viel mir schwer



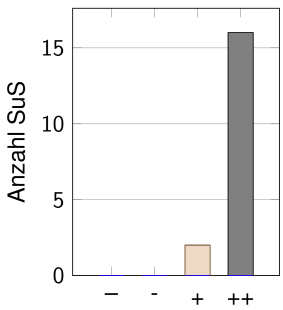
–: 8 -: 9 +: 1 ++: 0

2.5 Ich hatte keine Probleme, die Aufgaben mit MATLAB umzusetzen



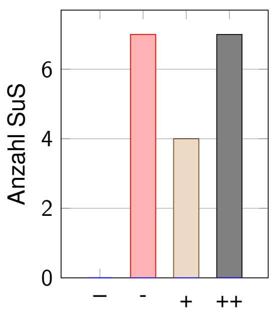
–: 0 -: 1 +: 9 ++: 8

2.6 Die Aufgabenstellungen waren für mich verständlich formuliert



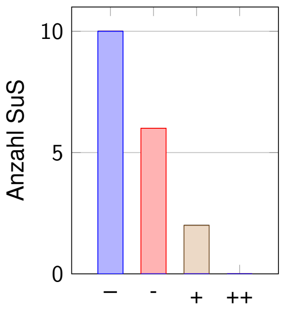
–: 0 -: 0 +: 2 ++: 16

2.7 Die Aufgaben waren zu einfach.



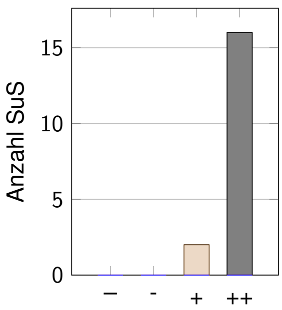
–: 0 -: 7 +: 4 ++: 7

2.8 Die Aufgaben waren zu schwierig.



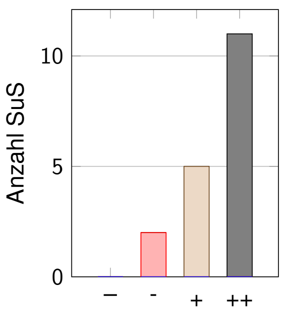
–: 10 -: 6 +: 2 ++: 0

2.9 Ich habe die Aufgaben und ihre Lösung verstanden



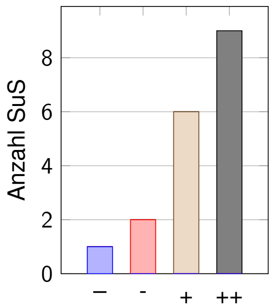
–: 0 -: 0 +: 2 ++: 16

2.10 Ich konnte eigenständig arbeiten



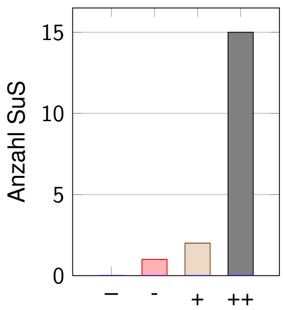
–: 0   -: 2   +: 5   ++: 11

2.11 Durch den Workshop habe ich mathematisches Modellieren besser begriffen



–: 1   -: 2   +: 6   ++: 9

2.12 Der CAMMP day hat mir Spaßgemacht.



–: 0   -: 1   +: 2   ++: 15

2.13 Kommentar

2.14 Ich gebe dem CAMMP day die Schulnote

m (Jgst.:): 1.

m (Jgst.:): 1.

m (Jgst.:): 2.

w (Jgst.:): 3.

m (Jgst.:): 2.

m (Jgst.:): 1.

m (Jgst.:): 1.

m (Jgst.:): 1.

m (Jgst.:): 1.

m (Jgst.:): 1.

w (Jgst.:): 1.

w (Jgst.:): 1.

w (Jgst.:): 2.

m (Jgst.:): 11.

w (Jgst.:): 1.

w (Jgst.:): 1.

w (Jgst.:): 1.

w (Jgst.:): 9.

2.15 Ich gebe dem Betreuer die Schulnote

m (Jgst.:): 1.

m (Jgst.:): 2.

m (Jgst.:): 2.

w (Jgst.:): 1.

m (Jgst.:): 1.

m (Jgst.:): 1.

m (Jgst.:): 1.

m (Jgst.:): 1.

m (Jgst.:): 1.

m (Jgst.:): 1.

w (Jgst.:): 1.

w (Jgst.:): 1.

w (Jgst.:): 1.

m (Jgst.:): 13.

w (Jgst.:): 1.

w (Jgst.:): 1.

w (Jgst.:): 1.

w (Jgst.:): 13.

2.16 Was würdest du am Workshop verändern bzw. verbessern wollen?

m (Jgst.:): Alles super.

m (Jgst.:): weniger technische Probleme.

w (Jgst.:): mehr eigenständige Arbeit.

m (Jgst.:): schwierigere Aufgaben.

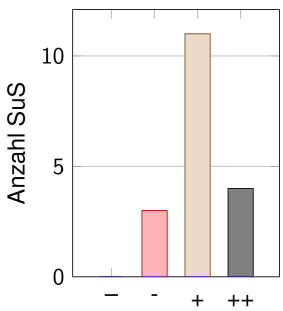
m (Jgst.:): gut so.

w (Jgst.:): komplexere Aufgaben.

w (Jgst.:): technische Probleme lösen, MATLAB auf allen Rechnern gleich.

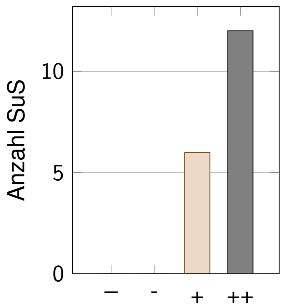
3 Weiterführende Fragen

3.1 Der Workshop hat mein Interesse an den Naturwissenschaften und Technik gesteigert.



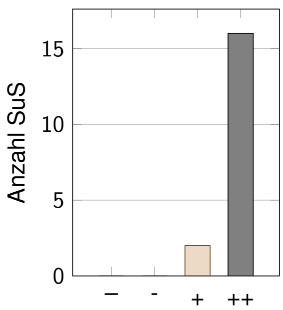
–: 0 -: 3 +: 11 ++: 4

3.2 Die Anleitungen zu den Aufgaben waren zufriedenstellend und gut verständlich.



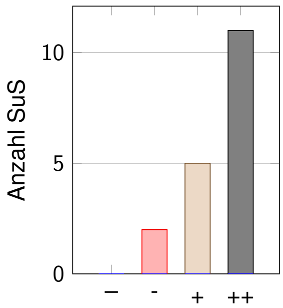
–: 0 -: 0 +: 6 ++: 12

3.3 Die Dozenten haben die Inhalte plausibel und klar dargestellt.



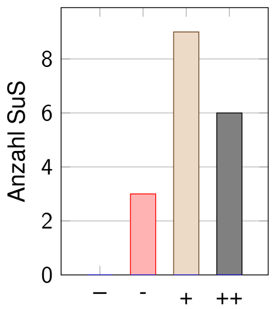
–: 0 -: 0 +: 2 ++: 16

3.4 Die Betreuer haben das selbstständige Arbeiten gefördert.



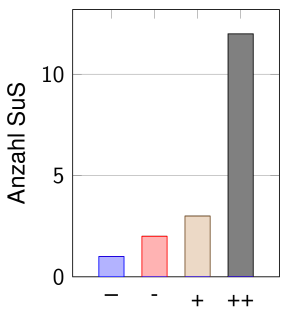
–: 0 -: 2 +: 5 ++: 11

3.5 Es wurden interessante Berufe und Studiengänge vorgestellt.



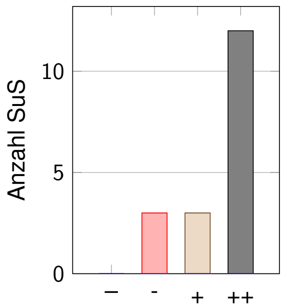
–: 0 -: 3 +: 9 ++: 6

3.6 Ich kann mir gut vorstellen, später ein Studium oder eine Ausbildung im Bereich Technik oder Naturwissenschaften aufzunehmen.



–: 1 -: 2 +: 3 ++: 12

3.7 Ich würde einen solchen Workshop gerne noch einmal besuchen.



–: 0 -: 3 +: 3 ++: 12

3.8 Kommentar

3.9 Abschließender persönlicher Kommentar

m (Jgst.:): hat Spaßgemacht.

w (Jgst.:): eher für Mittelstufe.

m (Jgst.:): sehr gut umgesetzt.

m (Jgst.:): toll.

m (Jgst.:): schöner Tag.

w (Jgst.:): hat Spaßgemacht.

w (Jgst.:): Betreuung gut, war interessant.

w (Jgst.:): hat Spaßgemacht.

Literatur

[1] W. Blum. *Beiträge zum Mathematikunterricht*. Franzbecker, Berlin, Hildesheim, 2006.

[2] R. Bruder. *Handbuch der der Mathematikdidaktik*. Springer Spektrum, Berlin, Heidelberg, 2015.

[3] National Instruments Corporation. *Schnelle Fourier-Transformation (FFT) und Fensterung*. URL `http://www.ni.com/white-paper/4844/de/`. Eingesehen am 13.06.2016.

[4] Kultusministerkonferenz der Länder in der Bundesrepublik Deutschland. *Bildungsstandards im Fach Mathematik für die Allgemeine Hochschulreife*. Carl Link, München, 2015.

[5] Shazam Entertainment. Shazam - company information. URL: `http://www.shazam.com/de/company`. Eingesehen am 13.06.2016.

[6] R. B. Ferri. *Wege zur Innenwelt des mathematischen Modellierens - Kognitive Analysen zu Modellierungsprozessen im Mathematikunterricht*. Vieweg+Taubner Verlag, Springer Fachmedien, Wiesbaden, 2011.

[7] Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen. *Kernlehrplan für das Gymnasium - Sekundarstufe II Gymnasium / Gesamtschule in Nordrhein-Westfalen - Mathematik*. URL `http://www.schulentwicklung.nrw.de/lehrplaene/upload/klp_SII/m/KLP_GOSt_Mathematik.pdf`.

[8] Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen. *Kernlehrplan für das Gymnasium - Sekundarstufe I (G8) in Nordrhein-Westfalen - Mathematik*. Ritterbach Verlag, Frechen, 2007.

[9] G. Hinrichs. *Modellierung im Mathematikunterricht*. Spektrum Akademischer Verlag, Heidelberg, 2008.

[10] G. Kaiser. *Mathematisches Modellieren für Schule und Hochschulen*. Vieweg+Taubner Verlag, Springer Fachmedien, Wiesbaden, 2011.

[11] C. Kalenzaga. How does shazam work. URL: `http://coding-geek.com/how-shazam-works/`. Eingesehen am 21.05.2016.

[12] R. Kataoka. A little about how shazam works. URL: `https://infosimples.com/en/articles/how-shazam-works`. Eingesehen am 15.06.2016.

[13] K. Maaß. *Mathematisches Modellieren im Unterricht*. Franzbecker, Hildesheim, 2004.

[14] J. A. Pearson. Shazam announces faster recognition time, in-app search, and over 2 billion followers. URL: `http://news.shazam.com/pressreleases/shazam-announces-faster-recognition-time-in-app-search-and-over-2-billion-followers-1264334`. Eingesehen am 13.06.2016.

[15] P. Stender. *Wirkungsvolle Lehrerinterventionsformen*. Springer Spektrum, Wiesbaden, 2016.

[16] Fachhochschule Technikum Wien. *Fourieranalyse*. URL `http://www.mathe-online.at/mathint/fourier/i.html`. Eingesehen am 25.05.2016.