# Akustiska fingeravtryck

Kandidatavhandling i Datateknik

Fredrik Blomqvist

Handledare: Hannu Toivonen

Institutionen för informationsteknologi

Åbo Akademi

Våren 2015

## Referat

Ett akustiskt fingeravtryck är en kompakt beskrivning av ett ljudstycke. Denna beskrivning kan användas t.ex. för att identifiera ljudstycket eller för att hitta liknande stycken i en databas. Algoritmer som används för att beräkna akustiska fingeravtryck tittar inte på hur ljudfilerna representeras digitalt utan beaktar istället hur ljudet uppfattas av en lyssnare. Därför kan ljudfiler jämföras utifrån deras fingeravtryck även om de sparats i olika format. I denna avhandling presenteras de beståndsdelar som de flesta system för beräkning av fingeravtryck består av. Dessutom studeras några av dessa system mer grundligt.

Innehåll

# 1 Inledning

Det finns många olika tillämpningar där akustiska fingeravtryck används. Du kanske hör en bra låt på radion som du gärna vill veta mer om. En mobilapplikation som t.ex. Shazam kan då utifrån ett par sekunders lyssnande identifiera vilket musikstycke som spelas. Andra användningsändamål kan vara att hitta duplikat i ett musikbibliotek eller att automatiskt övervaka hur upphovsrättsskyddad musik används av t.ex. rundradion eller Youtube. Gemensamt för alla dessa är att ett så kallat fingeravtryck beräknas utgående från en ljudfil varefter en sökning görs för att hitta matchande fingeravtryck i en databas.

Ett sätt att avgöra om två digitala filer är lika att jämföra dem bit för bit. Denna metod jämför dock inte själva innehållet i filerna, utan samma ljudfil kan ha helt olika bitmönster beroende på bl.a. i vilket format den sparas eller hur den komprimeras. Ett bättre sätt att jämföra ljudfiler vore därför att också ta i beaktande hur ljudet uppfattas. Detta görs med hjälp av akustiska fingeravtryck.

Den största fördelen med akustiska fingeravtryck jämfört med t.ex. digital vattenmärkning är att ingen information behöver lagras i den ljudfil som skall identifieras. Dessutom kan ljudfiler identifieras även om de är komprimerade samt oberoende av vilket ljudformat som används. Förvrängt ljud eller brus utgör inte heller något problem för en bra fingeravtrycksalgoritm.

# 2 Krav

Ett idealt system för beräkning av fingeravtryck samt sökning i databas borde enligt Cano m.fl. [1] korrekt kunna identifiera ett ljudstycke utifrån endast ett par sekunder långt utdrag. Identifieringen borde ske fastän ljudkvaliteten är försämrad och även om t.ex. tempot i en låt har ändrats.

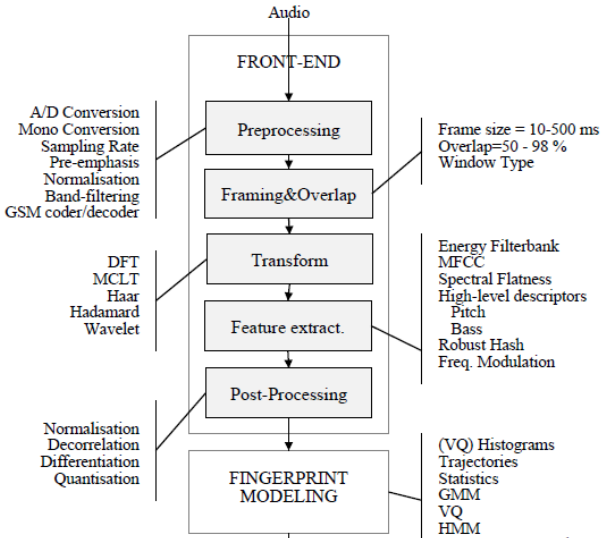
För att kunna avgöra hur bra ett fingeravtryckssystem fungerar använder sig Haitsma och Kalker [2] av ett antal olika parametrar. Dessa parametrar är:

- *Robusthet*: kan ett ljudklipp identifieras även om signalen är försämrad? För att systemet ska ha hög robusthet borde algoritmen använda egenskaper som inte förändras mycket även om signalen försämras.
- *Tillförlitlighe*t: hur ofta identifieras ljudklippet fel?
- *Storleken på fingeravtrycken*: hur mycket lagringsutrymme behövs per fingeravtryck?
- *Detaljrikedom*: hur många sekunder av ljudklippet behövs för identifiering?
- *Sökhastighet*: hur snabb är en sökning efter fingeravtryck i databasen?
- *Skalbarhet*: hur mycket försämras söktiden om antalet fingeravtryck i databasen ökas?

Dessa parametrar är alla beroende av varandra och kompromisser måste göras beroende på systemets användningsändamål. Haitsma och Kalker [2] använder som exempel storleken på fingeravtrycken, detaljrikedomen och tillförlitligheten. För att minska detaljrikedomen behövs ett större fingeravtryck för att fortfarande uppnå samma tillförlitlighet.

# 3 Modellering av fingeravtryck

Identifiering av ljud med hjälp av fingeravtryck kan enligt Cano m.fl. [1] uppdelas i två mindre problem: beräkning av fingeravtryck och databassökning. Figur 1 visar de olika steg som oftast ingår i modelleringen av fingeravtryck och de förklaras alla noggrannare längre fram i detta kapitel.




*Figur 1 Beräkning av fingeravtryck [1]*

## 3.1 Förbehandling

Under förbehandlingen digitaliseras signalen och konverteras till det format som används. Det är vanligt att signalen konverteras till monoljud [3] samt till en bestämd samplingsfrekvens [1]. Andra metoder som kan höra till förbehandlingen är t.ex. GSM- kodning och avkodning.

## 3.2 Inramning

Efter förbehandlingen delas signalen upp i mindre delar kallade ramar. Storleken av en ram är vanligen 10–500 millisekunder [1] och under den tiden antas signalen vara konstant.

Enligt Mapelli m.fl. [3] kan vissa signalbehandlingsmetoder leda till att ramarna i den modifierade ljudfilen förskjuts i förhållande till originalljudet. För att minska denna verkan låter man ramarna överlappa varandra [1] [3].

## 3.3 Linjär transformation

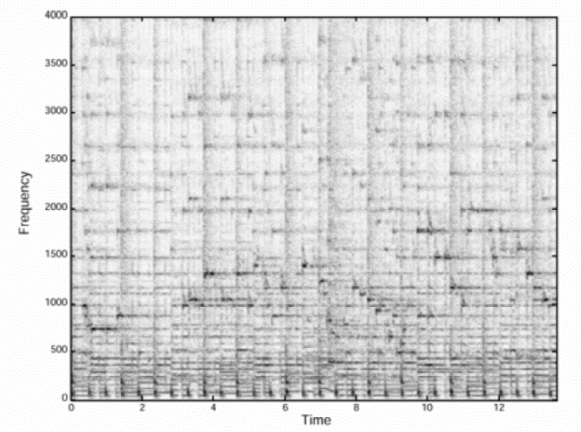
I detta steg transformeras signalen till frekvensplanet vilket medför lättare beräkning av de egenskaper man är intresserad av. Den mest använda transformmetoden är den snabba fouriertransformen [1], men även andra transformer används, t.ex. Walsh-Hadamardtransformen och diskreta fouriertransformen [4].

## 3.4 Extrahering av särdrag

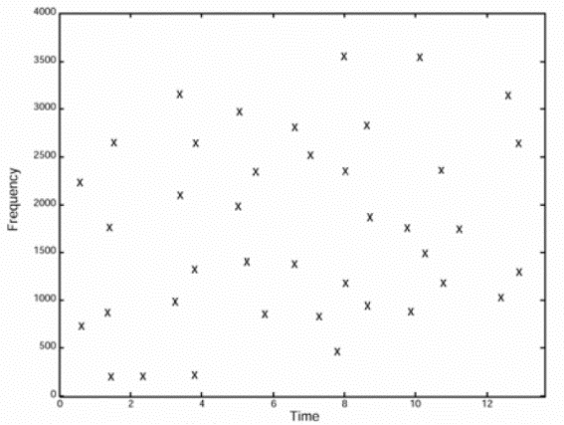
Efter transformationen extraheras relevant information som senare används för att beräkna ett fingeravtryck för ljudsignalen. I [3] använder Mapelli m.fl. sig av amplitudfaktorn, spektrumets planhet samt energin för varje ram. Andra egenskaper som enligt Cano m.fl. [1] visat sig fungera bra för jämförelse av musik är harmonik, bandbredd samt hastigheten på nollgenomgångarna (zero crossing rate).

Musiktjänsten Shazam använder sig av topparna i ett spektrogram, med tanken att de frekvenser som har högst intensitet också har störst sannolikhet att bestå vid olika

förvrängningar eller bakgrundsljud [5]. På detta sätt reduceras spektrogrammet i Figur 2 till en så kallad stjärnbild, se Figur 3, bestående av en mängd koordinater. Om man i databasen kan hitta en liknande stjärnbild kan man med stor sannolikhet korrekt identifiera musikstycket.




*Figur 2 Spektrogram [5]*




*Figur 3 Stjärnbild [5]*

## 3.5 Efterbehandling

För att bättre kunna beskriva förändringar i signalen kan tidsderivator av de egenskaper som extraherats läggas till i modellen. Cano m.fl. [6] använder t.ex. första och andra tidsderivatan av både energin och MFCC (Mel-frequency cepstral coefficients). Det finns också system som inte alls använder de egenskaper som extraherats utan endast deras derivator. Användningen av derivator har en tendens att förstärka störningar i signalen [1], men filtrerar samtidigt förvrängningar i linjära tidsinvarianta system [1].

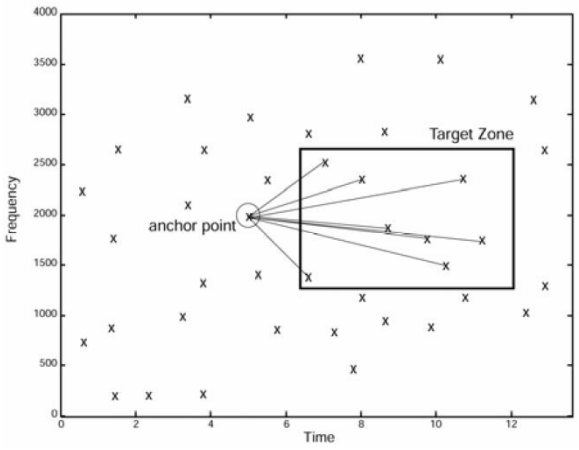
Det är också vanligt att kvantisera egenskaperna med en låg upplösning, t.ex. binär eller ternär [1]. Anledningen kan vara t.ex. ökad robusthet, enklare hårdvaruimplementation eller ökad slumpmässighet för att minska risken för konflikt mellan fingeravtryck [1].

## 3.6 Beräkning av fingeravtryck

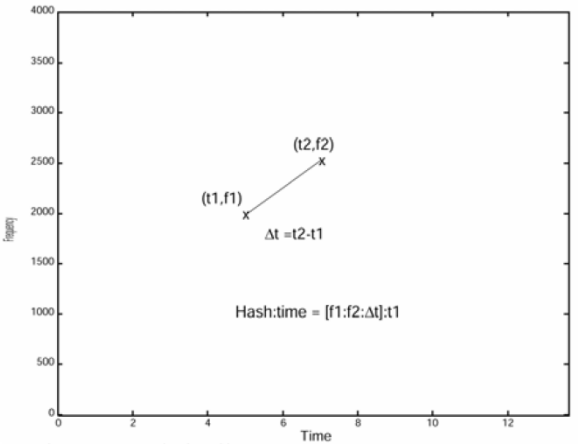
Fingeravtrycken beräknas vanligen utifrån en mängd vektorer innehållande egenskaper för de enskilda ramarna. Hur beräkningen sker påverkar både storleken på fingeravtrycken och precisionen vid en jämförelse, men kan också ha en inverkan på hastigheten vid en databassökning.

Ett kompakt fingeravtryck kunde fås genom att addera vektorerna för alla enskilda ramar i ett musikstycke, men detta ger en ganska dålig noggrannhet vid jämförelse av ljudfiler. Musicbrainzs TRM system [1] använde denna metod med vektorer innehållande bl.a. taktslag per minut samt medelhastigheten på nollgenomgångarna. Detta system hade vissa problem med att liknande ljud kunde få olika fingeravtryck och att olika ljud kunde få lika fingeravtryck, vilket gjorde att Musicbrainz övergick till att använda andra metoder [7].

Shazam skapar hashvärden genom att kombinera punkter i stjärnbilden [5]. Detta görs genom att välja ankarpunkter som alla associeras med ett målområde, se Figur 4. Då ankarpunkterna kombineras med alla punkter i sitt målområde fås hashvärden som består av två frekvenser samt tidsskillnaden mellan punkterna, se Figur 5. Varje hashvärde associeras också med tidsskillnaden mellan ankarpunkten samt ljudfilens början.




*Figur 4 Kombination av ankarpunkter med målområden [5]*




*Figur 5 Beräkning av hashvärde [5]*

# 4 Databassökning

I detta kapitel behandlas sökningen efter fingeravtryck i en databas. Saker som tas upp är bl.a. hur fingeravtryck kan jämföras och hur databassökningen kan effektiveras. Slutligen ges ett exempel på hur databassökningen kan genomföras genom att Shazams system studeras mer ingående.

## 4.1 Sökmetoder

För att kunna bestämma hur mycket två ljudstycken liknar varandra måste man ha ett sätt att jämföra vektorerna i deras fingeravtryck. Ett sätt att bestämma likheten mellan vektorerna är att använda det euklidiska avståndet [1]. En annan metod som är vanlig speciellt i system där egenskapsvektorerna är kvantiserade är att använda sig av Manhattan avståndet som baserar sig på 1-normen. Haitsma m.fl. [8] som använder sig av en binär kvantisering av egenskapsvektorerna använder sig av Hammingavståndet mellan vektorerna.

Förutom en metod för att jämföra fingeravtrycksvektorerna är det också viktigt att ha en effektiv metod för att söka i databasen. Att beräkna avståndet från det fingeravtryck som sökningen görs på till vartenda fingeravtryck som finns i databasen kan vara en beräkningsmässigt väldigt dyr operation. Därför är det viktigt att skapa en sådan datastruktur som minskar antalet avstånd som ska beräknas vid en sökning.

Ett sätt att göra databassökningen effektivare är att först använda en enklare avståndsberäkning för att snabbt förkasta de flesta alternativen och sedan övergå till den beräkningsmässigt tyngre metoden för att hitta den bästa träffen.

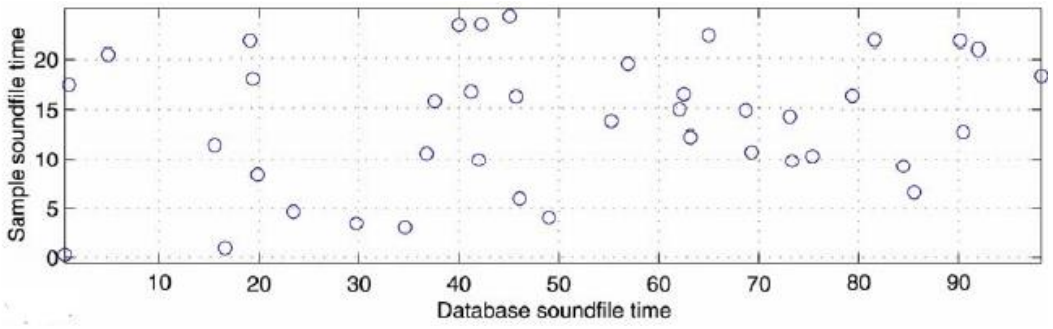
När databassökningen är gjord gäller det att avgöra om det sökta ljudstycket överhuvudtaget finns i databasen. Databassökningen ger som resultat en mängd poäng, baserade på avståndsberäkningen, som berättar hur lika de olika ljudstyckena är i förhållande till det eftersökta. Om poängsumman för en träff överstiger (eller understiger) en viss gräns anses träffen vara en korrekt identifiering av det sökta ljudstycket. Det kan vara ganska svårt att bestämma detta gränsvärde eftersom det finns många inverkande faktorer. Dessa faktorer kan vara t.ex. den använda fingeravtrycksmodellen, likheten mellan fingeravtrycken i databasen och

storleken på databasen [1]. Desto större databasen är, desto större blir också sannolikheten för så kallade falska positiva d.v.s. att databassökningen ger en felaktig träff.

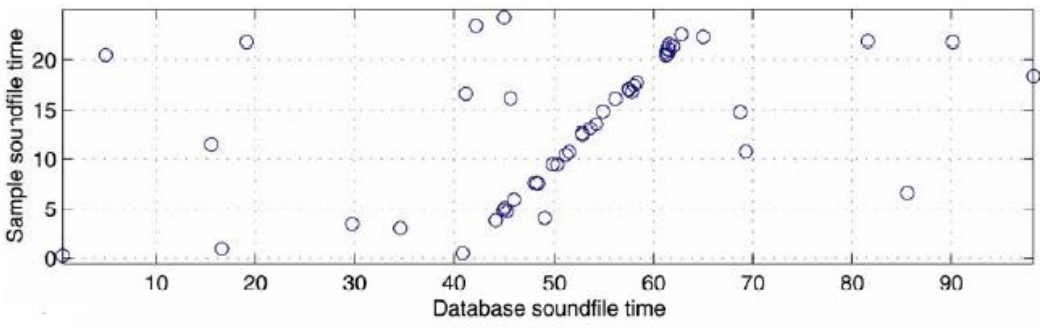
## 4.2 Shazam databassökning

Fingeravtrycken i Shazams algoritm består av mängd hashvärden samt deras tidsvärden. För att hitta en matchning görs en databassökning för vartenda hashvärde i fingeravtrycket. För varje träff som hittas i sökningen kombineras sökningens och träffens tidsvärden till par som indelas i grupper enligt det låt ID som är associerat med hashvärdet [5].

Paren av tidsvärden i varje grupp kan betraktas som punkter i ett spridningsdiagram, se Figur 6. Om två ljudfiler är lika borde punkterna i deras spridningsdiagram ligga i en diagonal linje, se Figur 7. Problemet med att avgöra om två ljudstycken är lika kan alltså reduceras till att hitta kluster av punkter som formar en diagonal linje. Shazam löser detta problem med effektiviteten $Nlog(N)$, där N är antalet punkter i spridningsdiagrammet [5].




*Figur 6 Spridningsdiagram över matchande hashvärden [5]*

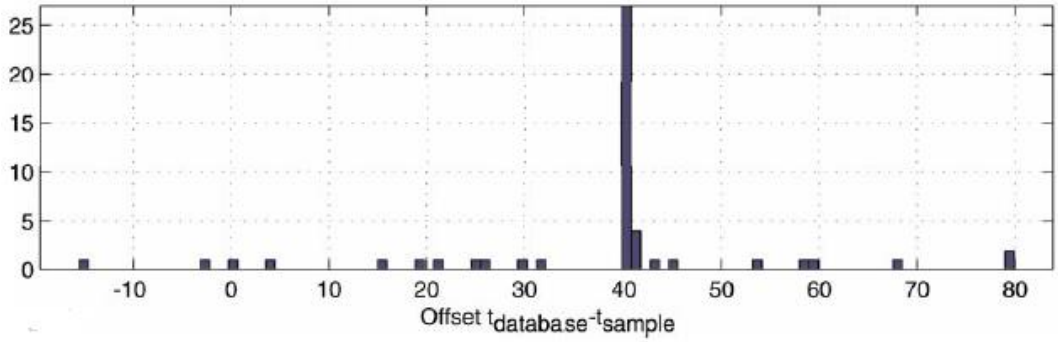



*Figur 7 Spridningsdiagram över matchande hashvärden, punkterna formar en linje [5]*

För varje punkt $(t', t)$ i spridningsdiagrammet görs beräkningen

$$\delta t = t' - t$$

där $t'$ är sampelns tidsvärde och $t$ är motsvarande tidsvärde i databasen. Ett histogram beräknas utifrån $\delta t$-värdena och toppar sökes. Figur 8 visar ett histogram över tidsskillnaderna i spridningsdiagrammet i Figur 7. Toppvärdet i histogrammet betraktas sedan som en poängsumma för hur mycket samplet liknar ljudet i databasen, och om poängsumman passerar ett bestämt gränsvärde anses resultatet vara en träff.




*Figur 8 Histogram över tidsskillnaderna i Figur 7 [5]*

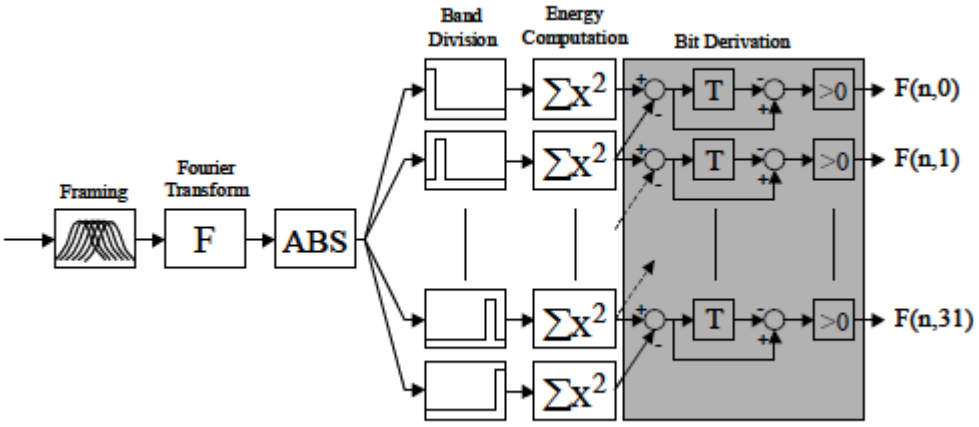
# 5 Exempel på system för ljudidentifiering

I detta kapitel presenteras två olika tillvägagångsätt för identifiering av musik. Det första exemplet av Haitsma och Kalker [2] följer i stora drag de steg som behandlades i kapitel 3 medan det andra exemplet använder sig av datorseende för att identifiera musiken.

## 5.1 Algoritm av Haitsma och Kalker

Jaap Haitsma och Ton Kalker presenterar i [2] ett system för akustiska fingeravtryck. En översikt av systemet visas i Figur 9. Ljudsignalen delas upp i överlappande ramar så som presenterades i kapitel 3.2. Delavtryck på 32 bitar extraheras med 11,6 millisekunders mellanrum, och dessa delavtryck motsvarar en ramstorlek på 370 millisekunder [2]. Ett delavtryck innehåller inte tillräckligt med information för att unikt identifiera ett ljudklipp och därför använder sig Haitsma

och Kalker [2] av fingeravtrycksblock. Ett fingeravtrycksblock är den basenhet som innehåller tillräckligt med data för en identifikation, Haitsma och Kalker använder sig i [2] av en grupp om 256 delavtryck. Detta motsvarar ungefär tre sekunder av ljud.




*Figur 9 Översikt över Haitsma och Kalkers system för beräkning av fingeravtryck [2]*

Ramarna i signalen transformeras med hjälp av fouriertransformen till frekvensplanet för lättare beräkning av relevanta egenskaper. Dessutom sparas endast det absoluta värdet av signalen [2].

Som nämndes tidigare beräknas ett 32 bitar långt delavtryck för varje ram. För att göra detta delas frekvensområdet mellan 300 Hz och 2000 Hz upp i 33 delområden. Haitsma och Kalker [2] påstår att det experimentellt har visats att tecknet på energidifferensen är en väldigt robust egenskap [2]. Detta motiverar deras val av beräkningsmetod för delavtrycken till

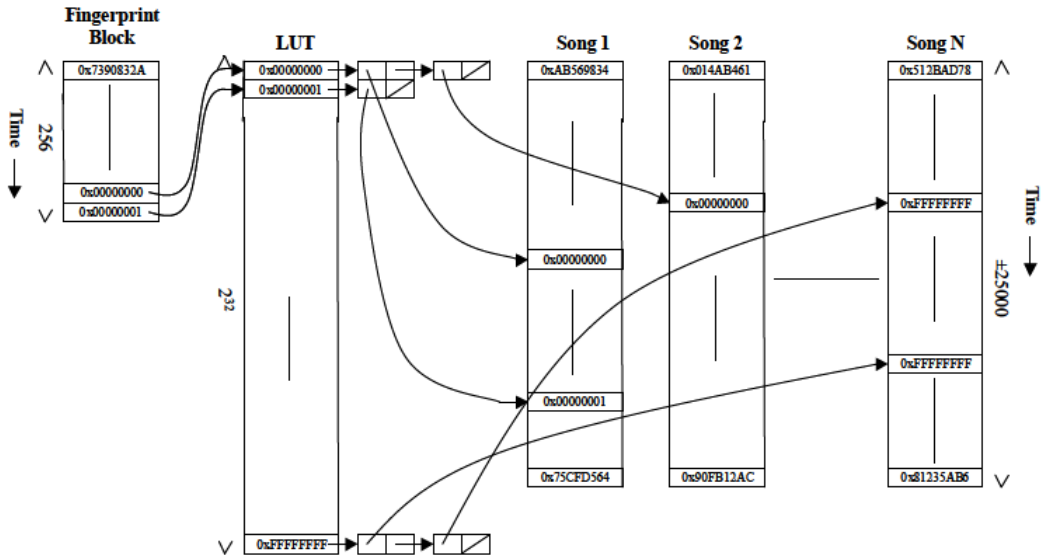
$$F(n,m) = \begin{cases} 1, om\ E(n,m) - E(n,m+1) - \big(E(n-1,m) - E(n-1,m+1)\big) > 0 \\ 0, om\ E(n,m) - E(n,m+1) - \big(E(n-1,m) - E(n-1,m+1)\big) \leq 0 \end{cases}$$

där *E(n,m)* är energin för område *m* i ram *n* och *F(n,m)* är bit nummer *m* i delavtrycket för ram *n*.

Två fingeravtrycksblock (3 sekunders ljudsignal) anses vara lika om Hammingavståndet mellan dem är mindre än ett bestämt gränsvärde *T*. Haitsma och Kalker [2] har med antagandet att extraheringsmetoden genererar slumpmässiga

bitar beräknat att ett gränsvärde $T = 2867$, vilket motsvaras av att 35 % av bitarna är olika, medför att andelen falska positiva är $3{,}6 \times 10^{-20}$.

En databas med 10000 låtar som har en medellängd på 5 minuter består av ca 250 miljoner delavtryck. Att med råstyrka söka igenom en sådan databas skulle enligt Haitsma och Kalker [2] ta ca 20 minuter på en modern dator (år 2002). Detta är därför ingen lämplig lösning för praktiska användningsändamål.




*Figur 10 Databas layout med referenstabell (LUT) [2]*

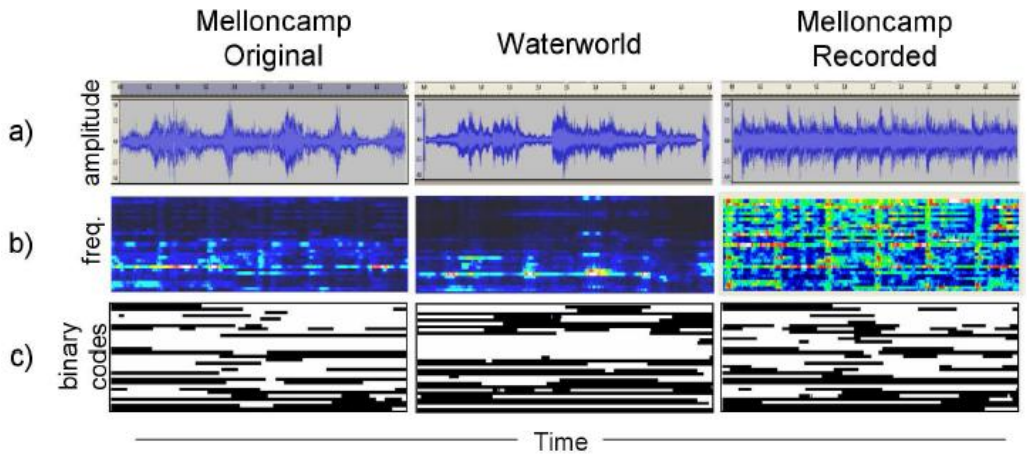
Om man antar att åtminstone ett delavtryck ger en exakt match till rätt låt i databasen räcker det att kontrollera de låtar som innehåller något av de 256 delavtrycken. Haitsma och Kalker [2] föreslår användning av en referenstabell (*lookup table*, LUT), se Figur 10. Tabellen innehåller för varje tänkbart delavtryck en lista med de låtar och positioner där delavtrycket finns. Denna tabell gör det möjligt att snabbt välja ut kandidater som sedan kan jämföras med hela fingeravtrycksblocket. I praktiska tillämpningar används en hashtabell eftersom begränsat minne ofta gör det orimligt att använda en referenstabell med $2^{32}$ poster [2]. Dessutom skulle en referenstabell vara glest fylld eftersom endast ett ändligt antal poster finns i databasen [2]. Användningen av en referenstabell försnabbar enligt Haitsma och Kalker [2] databassökningen med en faktor på ungefär 800000 vilket gör att sökningen nu genomförs på ca 1,5 millisekunder.

För en kraftigt försämrad signal stämmer inte alltid antagandet att det finns ett felfritt delavtryck. Haitsma och Kalker [2] använder sig därför av ett

rankningssystem där bitarna i delavtrycken rankas enligt hur pålitliga de är. Delavtrycken fås från tecknet på energidifferenser och om energidifferensen ligger väldigt nära tröskelvärdet är det därför mer troligt att biten i delavtrycket är felaktig. De bitar som anses minst pålitliga varieras sedan för att skapa en lista med sannolika delavtryck. I praktiken kan även bitar med hög pålitlighet visa sig vara felaktiga, men metoden ger ändå en klar förbättring i resultatet [2].

## 5.2 Musikidentifiering med hjälp av datorseende

En idé som behandlats av bl.a. Ke, Hoiem och Sukthankar [9] är att använda befintliga metoder för bildanalys som grund vid jämförelse av ljudsignaler. Genom att dela upp den endimensionella signalen (Figur 11a) i överlappande ramar samt applicera en fouriertransform skapas en tvådimensionell bild av signalen, ett spektrogram (Figur 11b). Ke m.fl. [9] delar upp signalen i 33 olika frekvensområden i vilka spektrogrammet representerar energin mätt över ramar med en längd på 0,372 s och en distans på 11,6 millisekunder mellan ramarna.



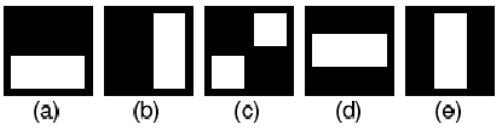

*Figur 11 Tre olika ljudsampel representerade i a) vågform b) spektrogram och c) binära delavtryck [9]*

Ke m.fl. [9] använder sig av ett antal filter som väljs med hjälp av maskininlärning. Kandidatfiltren representerar kännetecken som särskiljer olika låtar men ändå är motståndskraftiga mot störningar. Dessa kännetecken inkluderar (Figur 12):

a) Energiskillnader i närliggande frekvensområden vid en viss tidpunkt.
b) Skillnader i energin längs tidsaxeln för ett visst frekvensområde.
c) Ändring i dominant frekvens över tiden.

d) Energitoppar över frekvensområden vid en viss tidpunkt.
e) Energitoppar längs tidsaxeln för ett visst frekvensområde.

Bandbredden för filtren varieras mellan 1 och 33 frekvensområden och längden mellan 1 ram (11,6 ms) och 82 ramar (951 ms) [9]. Filtrens placering varieras också vilket gör att sammanlagt fås ungefär 25000 kandidatfilter. Av dessa kandidatfilter väljs med hjälp av maskininlärning M stycken filter och motsvarande gränsvärden, vilka används för att beräkna M-bitars delavtryck. Dessa delavtryck beräknas i algoritmen av Ke m.fl. [9] med 11,6 millisekunders mellanrum, se Figur 11c (M=32).




*Figur 12 Kandidatfilter. Filtren ger som resultat differensen mellan de olikfärgade områdena. [9]*

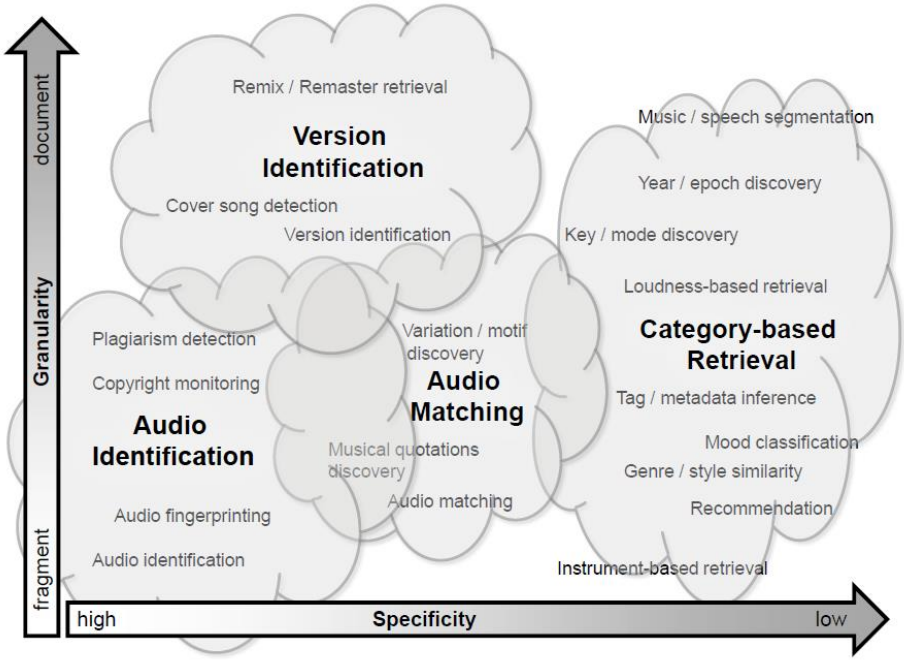
Bitarna i delavtrycken bestäms av tecknet på differensen mellan filtervärdet och gränsvärdet. Om två spektrogram ger filtervärden på samma sida av gränsvärdet har de alltså lika delavtryck.

Systemet av Haitsma och Kalker [2] i kapitel 5.1 använder sig av energiskillnaden för närliggande frekvens- och tidsområden. Detta kan betraktas som filter c i Figur 12 med en bandbredd på två frekvensområde och en längd på två ramar.

Databassökningen i systemet av Ke m.fl. [9] liknar till en viss del sökningen i Haitsma och Kalkers system [2]. Ke m.fl. [9] använder sig av en hashtabell i vilken en sökning görs för att hitta träffar med ett Hammingavstånd mindre eller lika med två från delavtrycken. De två systemen skiljer sig ändå åt i hur den bästa träffen väljs. Istället för att välja den post i databasen som ger flest träffar beaktar Ke m.fl. [9] huruvida de matchande delavtrycken är konsistenta över tiden. För detta används RANSAC [10] som itererar genom kandidaterna och resultatet från metoden används för att bedöma hur mycket kandidaten liknar samplet. Den kandidat som har högst poängsumma väljs och om poängen överstiger ett visst gränsvärde anses kandidaten vara en träff [9].

# 6 Användningsändamål

Det finns många olika användningsändamål för akustiska fingeravtryck. Ett sätt att gruppera dessa är att betrakta hur mycket samplet och databasfilen liknar varandra samt om samplet består av en hel ljudfil eller endast en del av den [11]. Denna indelning illustreras i Figur 13 där de olika användningsområdena är uppdelade i fyra överlappande grupper. De exempel som studerats i denna avhandling tillhör alla gruppen ljud identifiering.




*Figur 13 Gruppering av system för ljudsökning [11]*

De olika grupperna är:

- *Ljud identifiering*: Vid ljud identifiering är systemets uppgift att utgående från en del av ett ljudklipp korrekt identifiera källan. Vid ljud identifiering görs en skillnad inte bara mellan olika låtar, utan också mellan olika spelningar av samma låt.
- *Ljud matchning:* Vid ljud matchning är målet att hitta alla ljudstycken i databasen som motsvarar samplet. Här tillåts variationer som uppstår vid olika inspelningar, vilket gör att systemet kan hitta t.ex. både en inspelning från en konsert och samma låt inspelad i en studio.

- *Versions identifiering:* Vid versions identifiering är målet att i en databas identifiera olika versioner av samma låt. Här tillåts även mer extrema variationer, t.ex. förändringar i melodin eller tonart, som kan uppstå vid coverlåtar.
- *Kategori baserad sökning:* I den här gruppen görs sökningen efter musik som tillhör samma kategori. Kategorin kan vara t.ex. genre, rytm, tonart eller något specifikt instrument.

# 7 Avslutning

Akustiska fingeravtryck är mycket användbara vid jämförelse av ljudfiler. De algoritmer som presenterades i avhandlingen kan alla utgående från ett par sekunders rent ljudklipp korrekt identifiera vilken låt det är. Detta tar dessutom bara några sekunder och en förutsättning är förstås att låten finns i databasen. Det som skiljer algoritmerna åt är främst hur bra de klarar olika typer av degradering eller ändring av ljudfilen. Shazams algoritm klarar t.ex. inte av någon större skalning längs tidsaxeln eftersom linjen av punkter i spridningsdiagrammet (Figur 7) då skulle få en annan riktningskoefficient.

Nyare forskning inom området har gett förslag på lösningar som bättre klarar av olika typer förändring av ljudet. T.ex. Sonnleitner och Widmer [12] introducerar en algoritm som enligt dem är någorlunda robust även vid skalning längs tids- och frekvensaxeln.

# 8 Litteraturförteckning


[1] P. Cano, E. Batlle, T. Kalker och J. Haitsma, ”A Review of Algorithms for Audio Fingerprinting,” i *IEEE International workshop on MMSP*, 2002.

[2] J. Haitsma och T. Kalker, ”A Highly Robust Audio Fingerprinting System,” i *Proceedings of ISMIR,* 2002.

[3] F. Mapelli, R. Pezzano och R. Lancini, ”Robust audio fingerprinting for song identification,” i *European Signal Processing Conference*, 2004.

[4] R. Simha, B. Narahari, A. Youssef och S. Subramanya, ”Transform-based indexing of audio data for multimedia databases,” i *Proc. IEEE Int. Conf. Multimedia Comput. Syst.*, 1997.

[5] A. L.-C. Wang, ”An Industrial-Strength Audio Search Algorithm,” i *Proceedings of the 4 th International Conference on Music Information Retrieval,* 2003.

[6] P. Cano, E. Batlle, H. Mayer och H. Neuschmied, ”Robust song modeling for song detection in broadcast audio,” i *Proc. AES 112th Int. Conv.*, 2002.

[7] ”Musicbrainz,” [Online]. Available: https://musicbrainz.org/doc/Fingerprinting. [Använd 08 02 2015].

[8] J. Haitsma, T. Kalker och J. Oostveen, ”Robust audio hashing for content identification,” i *Proc. of the Content-Based Multimedia Indexing,* 2001.

[9] Y. Ke, D. Hoiem och R. Sukthankar, ”Computer Vision for Music Identification,” i *Proceedings of Computer Vision and Pattern Recognition*, 2005.

[10] M. Fischler och T. Bolles, ”Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” i *Communications of the ACM,* 1981.

[11] P. Grosche, M. Muller och J. Serrà, ”Audio Content-Based Music Retrieval,” i *Multimodal Music Processing,* 2012, pp. 157-174.

[12] R. Sonnleitner och G. Widmer, ”Quad-Based Audio Fingerprinting Robust to Time and Frequency Scaling,” i *Proc. of the 17th Int. Conf. of Digital Audio Effects,* 2014.