Kompresia zvuku

Záverečná práca
Ukázky aplikací matematiky

Daniel Onduš

Univerzita Karlova v Praze

Matematicko-fyzikální fakulta

2017

Obsah

1 Úvod

1.1 Záznam zvuku

Prvé zariadenie na záznam zvuku bolo patentované v roku 1857. Volalo sa fonoautograf a na špeciálny papier dokázalo zaznamenať zvukové vibrácie. Neskôr sa tento spôsob záznamu zdokonalil a štandardizoval príchodom gramofónu. Zásadnú revolúciu priniesol vynález mikrofónu, ktorý dokázal premeniť zvuk na elektrický signál. Tento spôsob nahrávania spolu so záznamom na magnetickú pásku bol populárny niekoľko desaťročí a dodnes sa využíva v niektorých nahrávacích štúdiách. Všetky tieto spôsoby majú jednu vec spoločnú. Sú analógové.

Prechod na digitálny záznam spolu s formátom CD bol prelomový, keďže zásadne zjednodušil prácu so zvukom, najmä možnosti pri nahrávaní. Stále však nevyriešil pomerne veľký problém. Ak sa človek chcel dostať k nahrávke, musel si ju fyzicky kúpiť na nejakom nosiči, alebo si ju, v prípade, že mal internet, mohol stiahnuť. CD formát má dátový tok 1411 kbps, čo v minulosti znamenalo, že stiahnuť si celý album trvalo niekoľko hodín, ak nie dní. Riešením je tento súbor zmenšiť, čiže komprimovať.

1.2 Kompresia

Kompresia je proces, pri ktorom sa spracúva súbor, zvyčajne za účelom zmenšiť pamäťové nároky na jeho uloženie. Táto kompresia môže byť buď bezstratová, v takom prípade je možné dekompresiou obnoviť pôvodný súbor bez akejkoľvek zmeny, alebo stratová, kde sa časť informácie stratí a nie je možné ju obnoviť. Pri zvukových súboroch ide zvyčajne o nekomprimovaný formát wav, ktorý je možné bezstratovo komprimovať na formát flac, alebo stratovo napríklad na formát mp3.

Faktor kompresie je pomer medzi veľkosťou výstupného a vstupného súboru. Čím menšie je toto číslo, tým viac pamäte sa nám podarí kompresiou ušetriť.

1.3 CD záznam

Na vytvorenie digitálneho záznamu sa používa PCM (pulse code modulation). Táto metóda spočíva v použití A/D prevodníka na prevod spojitého analógového signálu na digitálny, ktorý je už diskrétny. V určitom intervale (pri CD je to 44100 krát za sekundu) sa zaznamená napätie elektrického signálu. Tomuto procesu sa hovorí aj samplovanie. Napätie sa zaznamenáva so 16 bitovou presnosťou, teda sa rozlišuje 65536 rôznych napätí. Jedným samplom rozumieme jednu nameranú hodnotu.

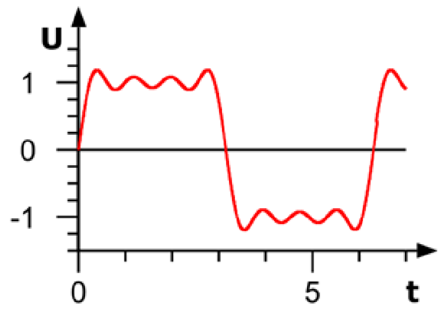
Tomuto záznamu zodpovedá nekomprimovaný formát wav. Nanešťastie, bežnými kompresnými metódami, ako napríklad zipovaním, sa takýto záznam dá bezstratovo komprimovať len o 10%. Formát flac pracuje s premenlivým dátovým tokom a preto sa u neho faktor kompresie pohybuje od 20% až takmer po 100%, teda nie je schopný vždy zabezpečiť dostatočnú kompresiu dát.

1.4 Stratová kompresia

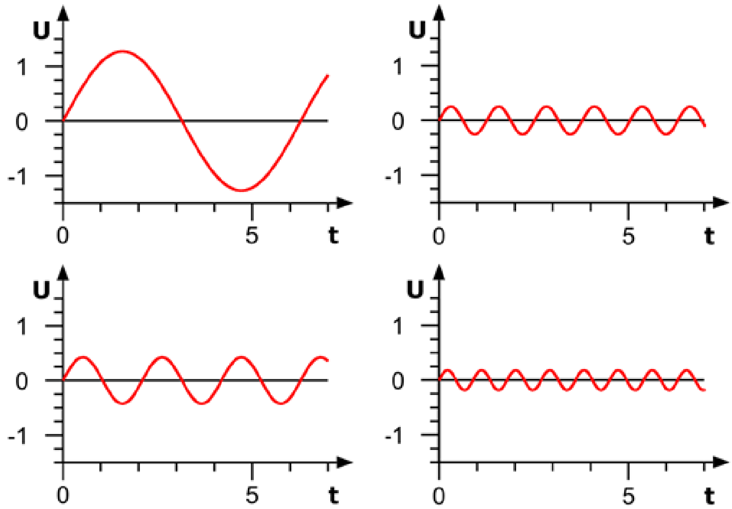
Ľudské ucho nie je schopné vnímať všetky zvuky, ktoré sme schopní nahrať. V skutočnosti je jeho rozsah približne 20 Hz až 20 kHz. Navyše frekvencie, ktoré sú pri sebe príliš blízko, často vnímame, ako keby splynuli do seba. Nepočujeme ani tiché zvuky, ak sú zamaskované oveľa hlasnejšími. To je pomerne veľa informácií, ktoré môžeme odstrániť bez toho, aby sme si to všimli. Najprv však musíme tieto informácie od seba oddeliť. V nasledujúcej časti si ukážeme, ako dokážeme vstupný signál rozdeliť na jednotlivé frekvencie.

2 Fourierova transformácia zvukovej vlny

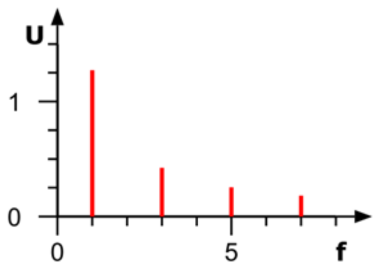
Po digitalizácii zvuku máme k dispozícii súbor, ktorý vyzerá ako vlna. Táto vlna má zaznamenanú hodnotu 44100 krát za sekundu. Čiže pre každú sekundu máme k dispozícií 44100 samplov. Následne si vyberieme nejaký úsek tejto vlny (istý počet samplov) a skúsime ho rozložiť na viacero zvukových frekvencií. Táto vec sa nám podarí vďaka Fourierovej transformácii. Fourierova transformácia dokáže rozložiť spojitú periodickú funkciu na sumu sínusoíd. [1]



Túto konkrétnu funkciu vieme rozložiť na súčet štyroch sínusoíd.



Navyše, po rozklade na tieto sínusoidy sa na našu vlnu nemusíme pozerať v závislosti na čase, ale v závislosti na frekvencii, čo sme chceli.



Teraz potrebujeme ukázať, ako presne našu periodickú funkciu rozložíme.

2.1 Fourierova rada

Spojitú periodickú funkciu s periódou T dokážeme vyjadriť ako sumu sínusoíd. Túto sumu nazývame Fourierovou radou a vyzerá nasledovne. [2]

$$g(t) = a_0 + \sum_{m=1}^{\infty} a_m cos(\frac{2\pi mt}{T}) + \sum_{n=1}^{\infty} b_n sin(\frac{2\pi nt}{T})$$

Zostáva vypočítať jednotlivé koeficienty a_m a b_n. Tieto koeficienty vyjadrujú závislosť našej pôvodnej funkcie od sínusoíd v sume. Túto závislosť vieme spočítať pomocou určitého integrálu na intervale 0 až T, čo je perióda našej funkcie.

$$a_0 = \frac{1}{T}\int_0^T f(t)dt$$

$$a_m = \frac{2}{T}\int_0^T f(t)cos(\frac{2\pi mt}{T})dt$$

$$b_n = \frac{2}{T}\int_0^T f(t)sin(\frac{2\pi nt}{T})dt$$

2.1.1 Náznak dôkazu pre pre $T = \pi$

Dôkaz. [3] Funkciu si zapíšeme v tvare

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty}[a_n \cos(nx) + b_n \sin(nx)]$$

Následne si ľubovoľne zvolíme prirodzené m a obe strany prenásobíme $sin(mx)$ a zintegrujeme od 0 po π. Keďže platí

$$\int_0^{\pi} \sin(nx)\sin(mx)\,dx = \begin{cases} 0, & \text{ak } n \neq m \\ \frac{\pi}{2}, & \text{ak } n = m \end{cases}$$

dostávame

$$b_m = \frac{2}{\pi}\int_0^{\pi} f(x)\sin(mx)\,dx$$

keďže m bolo volené ľubovoľne, tak dostávame požadovaný tvar pre $T = \pi$. Obdobne aj pre koeficienty a_n, keďže platí

$$\int_0^{\pi} \cos(nx)\cos(mx)\,dx = \begin{cases} 0, & \text{ak } n \neq m \\ \frac{\pi}{2}, & \text{ak } n = m \end{cases}$$

□

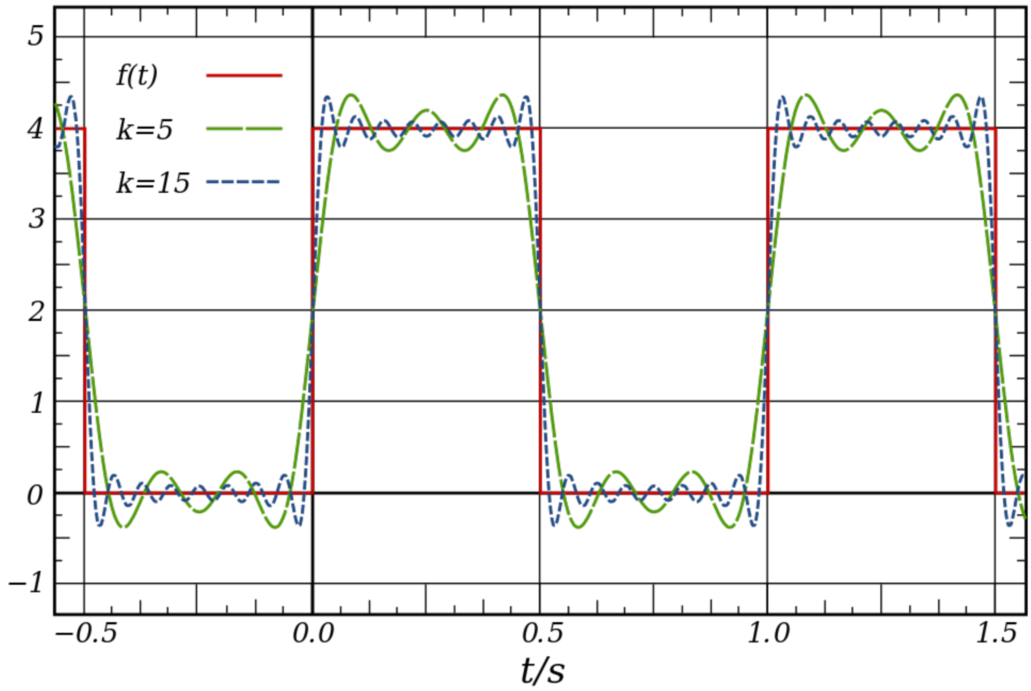
Podľa eulerovej identity platí $e^{it} = cost + isint$. Vďaka tomu môžeme vyjadriť našu funkciu novým spôsobom.

$$g(t) = \sum_{n=-\infty}^{\infty} c_n e^{i2\pi\frac{nt}{T}}$$

Koeficienty c_n môžeme vypočítať takto:

$$c_n = \frac{1}{T}\int_0^T f(t)e^{i2\pi\frac{nt}{T}}dt$$

V skutočnosti ale nemôžeme použiť nekonečne veľa sínusoíd. Preto presnosť vyjadrenia funkcie závisí od počtu sínusoíd, ktorými ju vyjadrujeme. Pri vyjadrení štvorcovej vlny vidno, že už 15 sínusoíd dokáže túto funkciu popísať pomerne presne. Pri mp3 kompresii sa týchto sínusoid používa niekoľko sto. [4]



2.2 Diskrétna Fourierova transformácia (DFT)

Keďže naše dáta po digitalizácii už nie sú spojité ale nasamplované v pravidelných intervaloch, môžeme na nich použiť diskrétnu Fourierovu transformáciu (DFT). Tá je definovaná nasledovne [5]:

DFT transformuje postupnosť N komplexných čísel $x_0, x_1, \ldots, x_{N-1}$ na druhú postupnosť N komplexných čísel $X_0, X_1, \ldots, X_{N-1}$ podľa vzťahu

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi\frac{kn}{N}}$$

U nás je N počet samplov (vzoriek), n je index momentálneho samplu, x_n je jeho hodnota, k je momentálna frekvencia medzi 0 a N-1 Hz a X_k je výsledný podiel frekvencie k v pôvodnom signáli. Inverznou Fourierovou transformáciou (IFT) dostaneme naspäť pôvodné hodnoty samplov podľa vzťahu

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{i2\pi\frac{kn}{N}}$$

čo sa značne podobá na vzťah, ktorý popisuje funkciu ako súčet sínusoíd pomocou Fourierovej rady. Navyše DFT môžeme zapísať aj pomocou matice tak, že platí $X = Wx$, kde x je vektor vstupného signálu, t. j. $x = (x_0, x_1, \ldots, x_{N-1})^T$, X je výstupný vektor, t. j. $X = (X_0, X_1, \ldots, X_{N-1})^T$ a W je matica, ktorá vyzerá nasledovne [6]:

$$\mathrm{W} = 1/\sqrt{N} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix}$$

kde $\omega = e^{\frac{2\pi i}{N}}$

2.2.1 Náznak dôkazu unitárnosti DFT matice

Dôkaz. Matica W má spĺňať $W^*W = WW^* = I$. Označme $W_{i,j}$ prvok na mieste i, j v matici WW^*. Platí, že

$$W_{i,j} = (\frac{1}{\sqrt{N}})^2 \sum_{k=0}^{N-1} \omega^{jk}\omega^{ik} = (\frac{1}{\sqrt{N}})^2 \sum_{k=0}^{N-1} \omega^{(j-i)k}$$

Pre prvky na diagonále, teda ak $i = j$, dostávame

$$W_{i,j} = (\frac{1}{\sqrt{N}})^2 \sum_{k=0}^{N-1} \omega^{(j-i)k} = \frac{1}{N} N = 1$$

Pre prvky mimo diagonály platí, že $i \neq j$, a teda $\omega^{(j-i)} \neq 1$. Označme $\omega^{(j-i)} = w_0$. Keďže platí $(e^{\frac{-2\pi i}{N}})^N = 1$, tak aj $w_0^N = 1$. Podľa súčtu geometrickej rady dostávame

$$W_{i,j} = \sum_{k=0}^{N-1} \omega_0^k = \frac{1 - \omega_0^N}{1 - \omega_0} = 0.$$

Teda matica $WW^* = I$. Obdobne pre W^*W. □

Tento fakt uľahčuje a zrýchľuje prácu s touto maticou.

2.3 Rýchla Fourierova transformácia (FFT)

Časová náročnosť výpočtu DFT podľa definície v závislosti na N je $O(N^2)$. Pri malých N je táto náročnosť postačujúca, nie však pri veľkých. Pri mp3 kompresii sa zvyčajne používa 1024 samplová DFT. Našťastie sa pomocou rôznych symetrií a periodicít podarilo túto náročnosť znížiť na $O(NlogN)$ s rovnakým výsledkom ako pri počítaní podľa definície. FFT označuje akýkoľvek algoritmus, ktorý dokáže spočítať DFT v tomto čase. Najznámejším a najpoužívanejším z nich je Cooley-Tukey algoritmus. Najprv dokážeme periodicitu DFT, ktorú používa aj tento algoritmus.

2.3.1 Periodicita DFT

Dôkaz. Chceme ukázať, že $X_k = X_{k+\frac{N}{2}}$. Podľa definície DFT

$$X_{k+\frac{N}{2}} = \sum_{n=0}^{N-1} x_n e^{-i2\pi\frac{(k+N/2)n}{N}} = \sum_{n=0}^{N-1} x_n e^{-i2\pi\frac{kn}{N}} e^{-i2\pi\frac{Nn}{2N}}$$

Keďže $e^{-\pi i} = 1$ tak môžeme náš tvar prepísať ako

$$\sum_{n=0}^{N-1} x_n e^{-i2\pi\frac{kn}{N}} e^{-\pi in} = \sum_{n=0}^{N-1} x_n e^{-i2\pi\frac{kn}{N}}$$

čo je z definície rovné X_k. □

2.4 Cooley-Tukey algoritmus

Tento algoritmus spočíva v rekurzívnom rozdeľovaní pôvodnej DFT na N prvkoch na N_1 menších DFT po N_2 prvkoch tak, že $N = N_1 N_2$. Ukážeme si najjednoduchšiu verziu, kde sa DFT delí na dve. [7]

V tomto prípade sa suma rozdelí na dve pre párne a nepárne indexy.

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N}(2m)k} + \sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N}(2m+1)k}$$

teraz môžeme využiť, že

$$\sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N}(2m+1)k} = e^{\frac{-2\pi i}{N}k} \sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N/2}mk}$$

čim dostávame

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N/2}mk} + e^{\frac{-2\pi i}{N}k} \sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N/2}mk}$$

Následne ak označíme E_k sumu cez párne indexy a O_k sumu cez nepárne indexy, tak z dokázanej periodicity vieme, že $E_k = E_{k+\frac{N}{2}}$ a $O_k = O_{k+\frac{N}{2}}$. Preto vieme X_k vyjadriť nasledujúcim spôsobom

$$X_k = \begin{cases} E_k + e^{\frac{-2\pi i}{N}k} O_k, & \text{pre } 0 \leq k < \frac{N}{2} \\ E_{k-\frac{N}{2}} + e^{\frac{-2\pi i}{N}k} O_{k-\frac{N}{2}}, & \text{pre } \frac{N}{2} \leq k < N \end{cases}$$

Takýmto spôsobom sme pôvodnú DFT na N prvkoch rozdelili na dve DFT s veľkosťami $N/2$. Rekurzívnym delením a použitím medzivýsledkov takto zrýchlime pôvodný algoritmus na $O(NlogN)$.

Táto verzia navyše predpokladá, že vstupný počet prvkov je mocnina čísla dva, pričom takýto počet prvkov väčšinou nie je problém zvoliť, existujú aj algoritmy, ktoré delia DFT na iný počet častí tak, aby mohlo byť N ľubovoľné prirodzené číslo.

3 Kódovanie

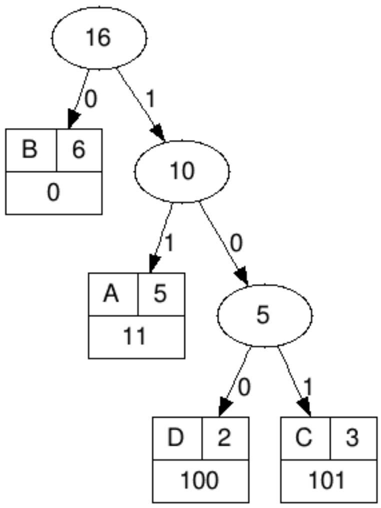
Veľkosť dát získaných Fourierovou transformáciou predstavuje zlepšenie oproti pôvodnej veľkosti, najmä keďže niektoré frekvencie, ktoré ucho nezachytí môžeme odstrániť. Napriek tomu sa toto množostvo dá ešte zmenšiť vhodným kódovaním. Pri mp3 súboroch sa štandardne používa Huffmanovo kódovanie. [8]

3.1 Huffmanovo kódovanie

Tento kód využíva fakt, že niektoré znaky sa v súbore vyskytujú častejšie ako iné. Preto je ideálne znakom, ktoré sa v súbore nachádzajú najčastejšie priradiť čo najkratší kód. Algoritmus vytvorí binárny strom, kde sa dané znaky vyskytujú v listoch tohoto stromu a ich kód je jednoznačne popísaný cestou z koreňa do tohoto listu.

Postup vytvárania stromu je pomerne jednoduchý a intuitívny. Na začiatku máme vstupnú abecedu, pričom každý znak má vypočítanú pravdepodobnosť výskytu a súčet týchto pravdepodností je jedna. Znaky zoradíme podľa tejto pravdepodobnosti a v každom kroku vyberieme dva znaky s najnižšou pravdepodobnosťou výskytu. Následne tieto znaky zlúčime do jedného znaku, ktorého pravdepodobnosť je súčtom pôvodných dvoch a pridáme tento znak do upravenej abecedy namiesto pôvodných znakov. V binárnom strome vytvoríme vrchol zodpovedajúci tomuto znaku a pôvodné dva znaky budú predstavovať jeho synov. Tento proces opakujeme, až kým nevytvoríme znak s pravdepodobnosťou výskytu 1, čo je koreň stromu.

Napr. pre text ABBBABABBABCCDDCA bude Huffmanov strom vyzerať takto [9]:



Vidíme, že najmenej často sa vyskytujú znaky C a D, teda sú zlúčené do jedného. Tento znak je následne zlúčený so znakom A a nakoniec aj so znakom B. Kódy pre jednotlivé znaky sú A:11, B:0, C:101 a D:100. Vidíme, že znak s najväčším počtom výskytov má kód najkratší a znak s najnižším počtom najdlhší, čo sme chceli dosiahnuť.

4 Zhrnutie

Teraz, keď máme popísané najdôležitejšie matematické metódy použité pri kompresii zvukového súboru, môžeme si zhrnúť celý vznik mp3 súboru.

Na začiatku máme analógový zvuk, ktorý sa šíri vzduchom ako mechanické vlny. Tieto vlny vieme zaznamenať a zosilniť pomocou mikrofónu na elektrický prúd. Hodnoty napätia tohoto elektrického prúdu zaznamenáme pomocou PCM metódy 44100-krát za sekundu so 16 bitovou presnosťou. Teraz máme súbor wav s dátovým tokom 1411 kbps.

Nasleduje najdóležitejšia časť. Vzniknutú digitálnu vlnu rozdelíme na krátke úseky, na ktorých prevedieme diskrétnu Fourierovu transformáciu pomocou Cooley-Tukey algoritmu. Tá nám rozloží vlnu na súčet frekvencií, pričom pre každú frekvenciu vieme aj silu, s akou sa v danom čase vyskytuje. Podľa psychoakustického modelu sa vyhodnotí, ktoré z daných frekvencií je schopné ľudské ucho rozpoznať a zvyšné sa vymažú.

Nakoniec sa získané dáta spracujú a zakódujú pomocou pomocou Huffmanovho kódovania. Po celom procese získame súbor, ktorý má dátový tok 128 kbps namiesto pôvodných 1411, čo znamená, že náš kompresný faktor je približne 9 precent a hudbu si vieme stiahnuť aj s oveľa pomalším internetom.

5 Zoznam použitých zdrojov

[1] https://upload.wikimedia.org/wikipedia/commons/thumb/f/fa/Fourier_synthesis.svg/748px-Fourier_synthesis.svg.png

[2] http://www.thefouriertransform.com/series/fourier.php

[3] https://math.stackexchange.com/questions/584113/fourier-series-coefficients-proof

[4] https://upload.wikimedia.org/wikipedia/commons/thumb/7/7e/Square_Wave_Fourier_Series.svg/
Square_Wave_Fourier_Series.svg.png

[5] https://en.wikipedia.org/wiki/Discrete_Fourier_transform

[6] https://math.stackexchange.com/questions/1708143/prove-that-the-dft-matrix-is-unitary

[7] https://en.wikipedia.org/wiki/Cooley-Tukey_FFT_algorithm

[8] http://pakuj.brek.sk/single/huffman.html

[9] http://huffman.ooz.ie/

https://www.theguardian.com/technology/2002/apr/04/internetnews.maths