

### 13. naloga – Filtriranje šuma

Jože Zobec

## 1 Uvod

V tej nalogi se bomo lotili filtriranja šuma z uporabo različnih filtrov, prvi med njimi je Wienerjev filter. Izhodni signal je konvolucija vhodnega signala ‘ $u(t)$ ’ in odzivne funkcije našega „senzorja“ ‘ $v(t)$ ’, tj.

$$f(t) = \int_{-\infty}^{\infty} d\tau \, u(\tau) v(t - \tau) = (u * v)(t).$$

Vendar pa imamo poleg  $f(t)$  primešan še šum, tako da

$$c(t) = f(t) + n(t)$$

Naj bo  $\mathcal{F}(\bullet) = \hat{\bullet}(\omega)$  fourierova transformiranka, potem velja

$$\hat{c} = \hat{f} + \hat{n}, \quad \hat{f}(\omega) = \hat{u}(\omega) \cdot \hat{v}(\omega). \quad (1)$$

Osnovna predpostavka je

$$\Phi \hat{c} \approx \hat{f}, \quad (2)$$

oz. poiskati  $\Phi(\omega)$ , da bo  $n(t)$  minimalen. S  $\Phi$  množimo  $\hat{c}$  pred dekonvolucijo. Na koncu dobimo, da je

$$u(t) \approx -\frac{1}{2\pi} \mathcal{F} \left( \Phi(\omega) \frac{\hat{c}(\omega)}{\hat{v}(\omega)} \right), \quad (3)$$

kjer je multiplikativna konstanta  $-1/2\pi$  spredaj zato, ker dvakrat naredimo Fourierovo transformacijo „naprej“. Wienerjev filter  $\Phi(\omega)$  je definiran kot

$$\Phi(\omega) \equiv \frac{|\hat{f}(\omega)|^2}{|\hat{f}(\omega)|^2 + |\hat{n}(\omega)|^2} = \frac{1}{1 + \left| \frac{\hat{n}(\omega)}{\hat{f}(\omega)} \right|^2}, \quad (4)$$

torej bolj zašumljene podatke manj upoštevamo v rekonstrukciji. Tu naletimo še na en problem – ne vemo, kakšen je šum. Literatura priporoča oceno šuma  $n(t)$  z opazovanjem  $c(t)$  po dolgem času. Pričakujemo, da na koncu preživi le šum, ki ga potem ekstrapoliramo na kratke čase in določimo tudi  $f(t)$ .

Poleg filtra  $\Phi(\omega)$  lahko uporabimo tudi še kakšno okno  $H(\omega)$ , s katerim lahko sliko dodatno izostrimo, meglimo itd.

$$u(t) = \mathcal{F}^{-1} \left( H(\omega) \Phi(\omega) \frac{\hat{c}(\omega)}{\hat{v}(\omega)} \right). \quad (5)$$

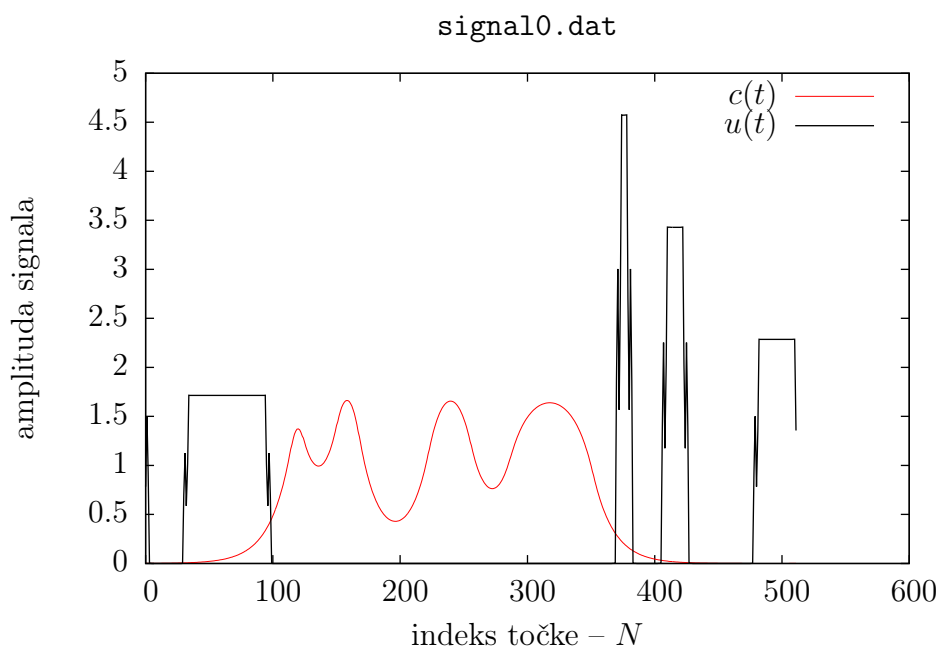
Kompresija slik s pomočjo razcepa na singularne vrednosti močno spominja na „trunkiranje“ iz TEBD DMRG in je precej intuitivno. Naša slika je matrika  $A = USV^\dagger$ ,  $A \in \text{Mat } n \times m$ . Matrika  $S$  je diagonalna matrika singularnih vrednosti. Majhne singularne vrednosti porežemo proč, popravimo število vrstic matrike  $V^\dagger$  in število stolpcev matrike  $U$ . Ko zmnožimo nazaj, dobimo spet matriko dimenzije  $n \times m$ , vendar smo za opis potrebovali manj informacije, zaradi česar potrebujemo manj računalniškega spomina.

## 2 Rezultati

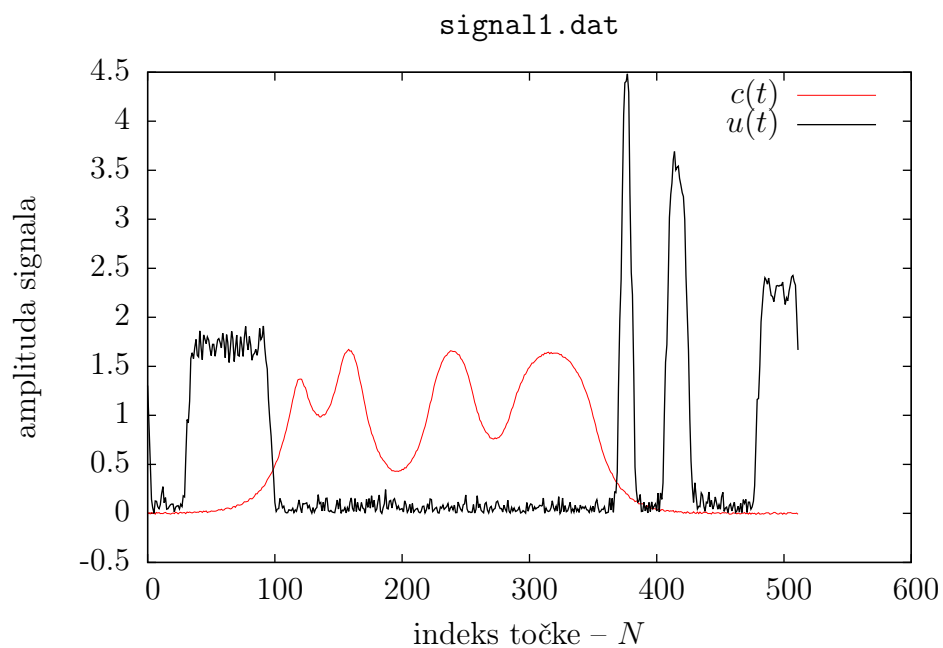
Za reševanje sem uporabljal programski jezik C, skriptni jezik sh in Octave.

### 2.1 Dekonvolucija signalov

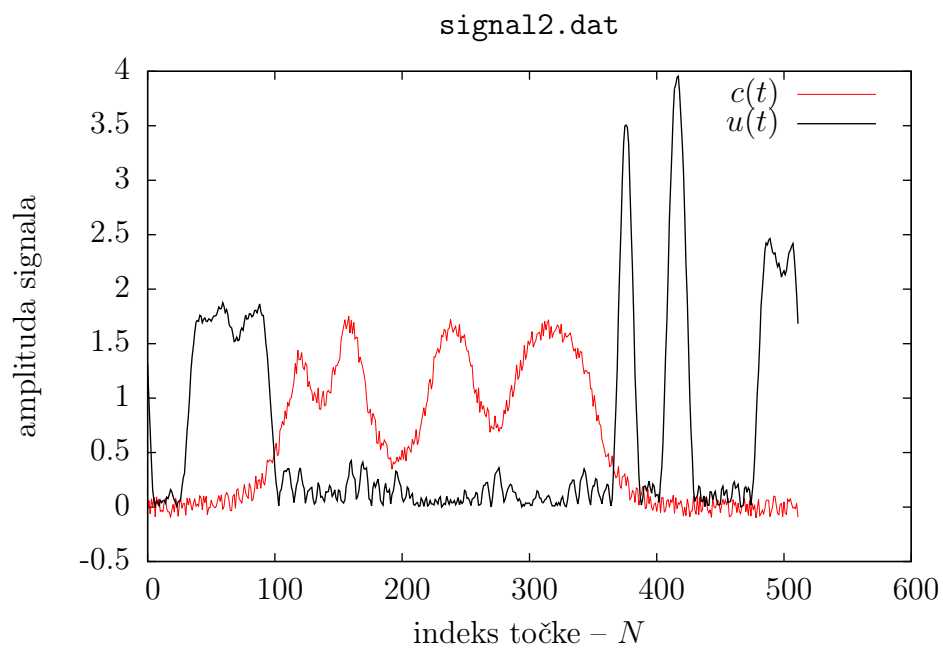
Za datoteke `signal{1,2,3}.dat` sem predpostavil, da je `signal0.dat` brez šuma in tako dobil  $n(t)$  eksplicitno ven, kot razliko. Zdelo se mi je tudi, da je čisti signal še vedno nekoliko zašumljen zaradi efektov diskretne Fourierjeve transformacije, zato sem poleg Wienerjevega filtra dodal še Hammingovo okno, ki se je izkazalo da najboljše sčisti oscilacije na robovih.



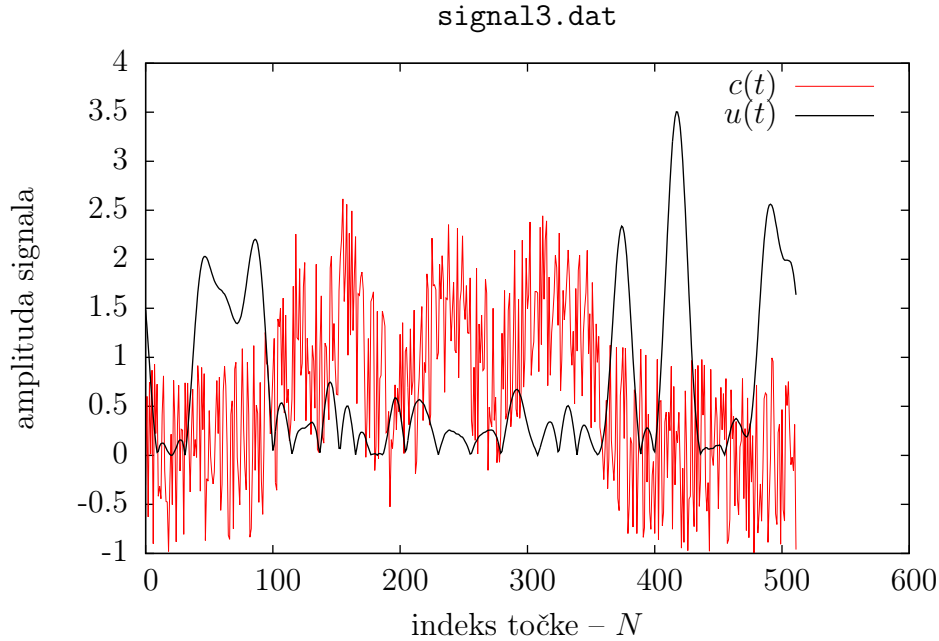
*Slika 1:* Signal je tukaj videti res lep, vidimo 4 impulze. Ugibam, da gre za simuliranje diskretne Diracove  $\delta$ -funkcije, in da imajo vsi štirje impulzi isto ploščino.



*Slika 2:* Signal je nekoliko zašumljen, vendar še vedno vidimo jasno sliko.



*Slika 3:* Anomalije med impulzi postajajo čedalje večje, vendar jih še vedno lahko ločimo od impulzov.



Slika 4: Ta slika je preveč zašumljena, vendar smo kljub temu izluščili vsaj bistveno informacijo, vendar bi težko rekli, koliko sunkov je bilo na začetku.

Šuma smo se znebili po goljufiji. Kaj pa, če ne bi imeli take sreče? Kako bi potem ocenilišum? Vidimo, da funkcija na začetku in na koncu prav prikladno pada proti nič in je skoraj konstantna. Od tam preberemo amplitudo šuma. Predpostavimo, da gre za beli šum, od koder lahko napovemo spekter  $\hat{n}(\omega) \approx 1/A$ , kjer je  $A$  amplituda, ki smo jo merili prej. Ker je napovedan  $n(\omega)$  zgolj konstanta, ta postopek ni najboljši in tudi grafi, ki sem jih dobil niso jasni, zato jih ne bom kazal tukaj.

## 2.2 Čiščenje Lincolнове brade

Slike bivšega predsednika severnoameriške federacije so bile zašumljene z belim šumom poleg tega, da so bile zamazane z eksponentno prenosno funkcijo  $v(t) = \exp(-t/\tau)/\tau$ ,  $\tau = 30$ .

Dobljenim slikam sem, tako kot prej, s Hammingovim oknom dodatno sčistil črte, ki so se poznale po dekonvoluciji z Wienerjevim filtrom. Rezultati so prikazani na slikah 5, 6, 7 in 8.



(a) Originalna slika.

(b) Dekonvolucija brez filtra.

(c) Dodan filter.

*Slika 5:* Na sliki imamo prikazan postopek čiščenja slike `lincoln_L30_N00.pgm`. Filter, dodan v sliki 5c, je Wienerjev filter, ki mu je dodano Hammingovo okno, zato je slika nekoliko temnejša.

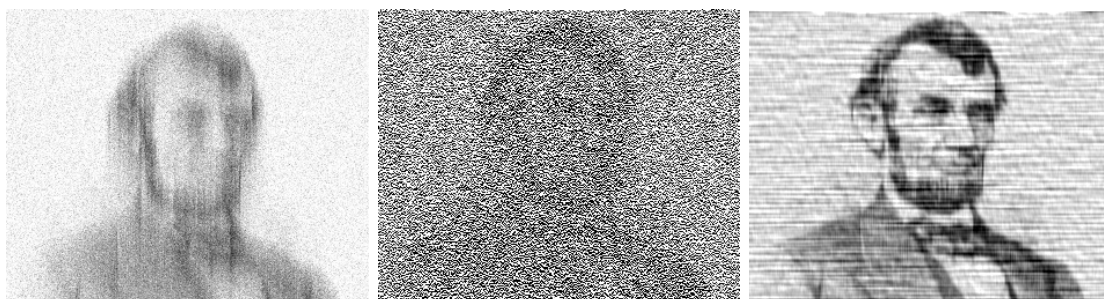


(a) Originalna slika.

(b) Dekonvolucija brez filtra.

(c) Dodan filter.

*Slika 6:* Na sliki imamo prikazan postopek čiščenja slike `lincoln_L30_N10.pgm`. Vidimo, da je kljub relativno majhnemu originalnemu šumu, dekonvoluirana slika močno zašumljena. Uporaba Wienerjevega filtra sliko močno izostri, defektov zaradi enodimenzionalnega branja slike skoraj ni več, ker smo dodali še Hammingovo okno.

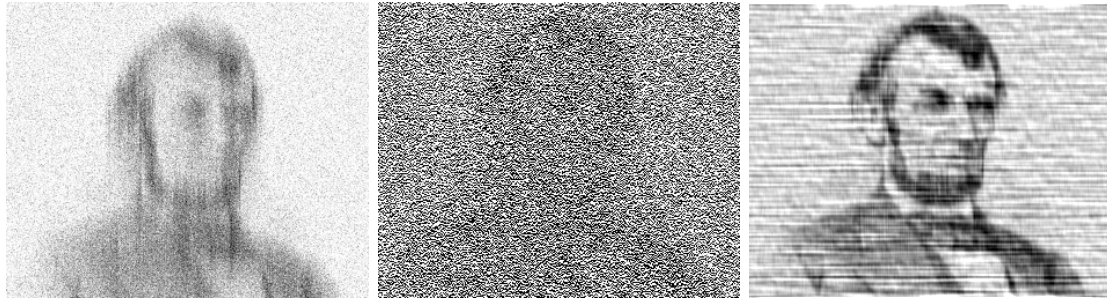


(a) Originalna slika.

(b) Dekonvolucija brez filtra.

(c) Dodan filter.

*Slika 7:* Na sliki imamo prikazan postopek čiščenja slike `lincoln_L30_N30.pgm`. Šum je močno prisoten, vendar je končna slika še vedno dovolj jasna. Defekti zaradi vrstičnega branja so bolj izraziti.



(a) Originalna slika.

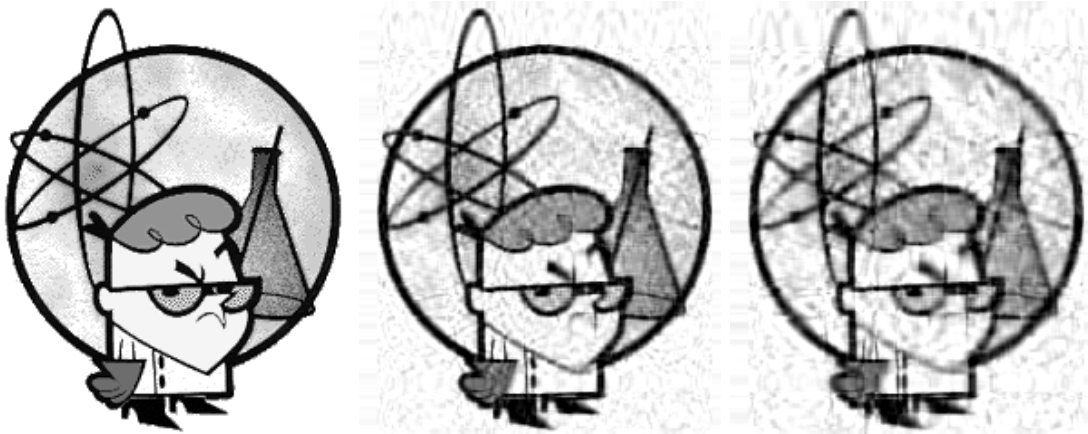
(b) Dekonvolucija brez filtra.

(c) Dodan filter.

*Slika 8:* Na sliki imamo prikazan postopek čiščenja slike `lincoln_L30_N40.pgm`. Slika 8b je praktično nerazpoznavna, po uporabi filtra, na sliki 8c, pa je tako jasna, da skorajda ni moč verjeti kako močno orodje so filtri.

## 2.3 Kompresija slik z uporabo SVD

Obdržimo  $M$  singularnih vrednosti in originalno matriko aproksimiramo s tako, ki ima isto dimenzijo, a ima nižji rang (rang  $M$ ). Rezultata za sliki `dexter.pgm` in `a.pgm` sta na slikah 9 in 10.



(a) `dexter.pgm`

(b)  $M = 30$ .

(c)  $M = 20$ .

*Slika 9:* Na sliki imamo prikazanih nekaj različnih izbir  $M$  za sliko `dexter.pgm`. Obe izbiri  $M = 30$  in  $M = 20$  sta še dobri, vendar na  $M = 20$  že težko ločimo Dexterjevo privzdignjeno obrv.



(a) `a.pgm`

(b)  $M = 30$ .

(c)  $M = 20$ .

(d)  $M = 10$ .

(e)  $M = 8$ .

*Slika 10:* Različne izbire kompresije prek SVD za sliko `a.pgm`.

S preprostim ukazom<sup>1</sup> lahko *ocenimo* učinkovitost naše kompresije s singularnim razcepom. Rezultati so v tabeli 1.

*Tabela 1:* Ocena velikosti datotek **dexter** po kompresiji z SVD. Vse datoteke so pretvorjene v format `.png`. Izvirnik v formatu `.png` potrebuje samo 14 kiB pomnilnika. Po spodnjih številih sodeč SVD kompresija verjetno potrebuje kak dodaten korak, saj naš rokodelski poskus kompresije klavrno propade in doseže ravno obratno – pomnilnik, potreben za te datoteke, se *poveča* za več kot 100%.

$M$	velikost [kiB]
2	32
8	37
10	37
20	36
30	36
39	36
50	36
60	36
80	36
100	35

Vidimo, da se taka kompresija ne izplača preveč. Tudi sicer obstajajo naprednejše metode.

### 3 Zaključek

Tekom te naloge smo se poglobljeje seznanili s pomembnostmi filtrov in preproste kompresije s singularnim razcepom, ki v praksi verjetno ni najboljši pristop.

---

<sup>1</sup>`$ du --apparent-size ime_datoteke`