

Støyreduksjon av hyperspektrale bilder

Sverre Fjerdingen

Master i elektronikk

Oppgaven levert: Juni 2010

Hovedveileder: Lise Lyngsnes Randeberg, IET

Oppgavetekst

Hyperspektral avbildning er en avbildningsteknikk som kombinerer høy spektral og romlig oppløsning. Hyperspektrale datakuber inneholder store mengder data og er krevende å prosessere. Hyperspektrale bilder kan prosesseres både ved hjelp av bildeanalyse og ved hjelp av tradisjonelle spektroskopiske teknikker. Støy skaper ofte problemer, spesielt for spektroskopisk analyse av data. Det er derfor interessant å se på støyreduksjon som kan gjøre det lettere for etterfølgende algoritmer og analyser. Oppgaven går ut på å teste forskjellige algoritmer til bruk for støyreduksjon av hyperspektrale bilder. Det er en føring for oppgaven at man ønsker spektral glatting med minst mulig reduksjon i romlig oppløsning.

Oppgaven forutsettes å inneholde følgende elementer

En teoretisk innføring i støy i hyperspektrale bilder og de metodene som er valgt for støyreduksjon i denne oppgaven. Valget av støyreduksjonsmetoder skal begrunnes.

Implementasjon, testing og sammenligning av ulike metoder for støyreduksjon i hyperspektrale bilder.

Implementasjon, testing og evaluering av ulike metoder for beregning av kovariansmatrisen til støyen i hyperspektrale bilder.

Oppgaven gitt: 22. januar 2010

Hovedveileder: Lise Lyngsnes Randeberg, IET

Sammendrag

Støyreduksjon har blitt utført på hyperspektrale bilder i både spektral retning og romlige retninger. Algoritmene som har blitt benyttet for å oppnå støyreduksjon er Principal Component Analysis (PCA), Maximum Noise Fraction (MNF) og wavelet-transform. MNF-algoritmen har blitt kjørt med mange forskjellige støyestimatorer for å bestemme hvilke av disse som gir høye signal-støyforhold. Å utføre støyestimatet i fourierrommet har også blitt undersøkt. Dette ga gode resultater når man benyttet fasedifferansen med nærliggende piksler som estimator for hyperspektrale bilder tatt under hvitt lys. Ble derimot kilden endret til en 355nm laserkilde fikk man langt dårligere resultater.

Det er bare Haar-transformen som har blitt brukt til wavelet-transformasjon. Haar-transformen ga dårlig støydempning i både spektral retning og romlige retninger.

Algoritmene PCA og MNF fungerer bra til støyreduksjon. I spektral retning er det liten forskjell mellom PCA og de ulike støyestimatene som er brukt under MNF. Ser man derimot på det romlige planet finner en større forskjeller mellom dem. Dette gjelder spesielt for spektralbånd med lav intensitet og mye støy. Her gir PCA bedre støydempning enn MNF.

Støyreduksjonen ved PCA og MNF kommer som en direkte følge av å begrense antallet prinsipalkomponenter under tilbaketransformasjonen. Hvor grensen bør settes for hvilke prinsipalkomponenter som skal bevares, ble også vurdert. Når lyskildeforholdene for de hyperspektrale bildene blir sammenlignet er grensen valgt slik at 99,25% av det opprinnelige signalet blir bevart.

Spekteret til hyperspektrale bilder tatt under hvitt lys har høy intensitet for lange bølgelengder, og lav intensitet ved korte bølgelengder. Endres derimot lyskilden til en 355nm laserkilde får man lav intensitet for lange bølgelengder og høy intensitet for korte.

Forord

Denne oppgaven er gjennomført på Institutt for elektronikk og telekommunikasjon ved Norges teknisk-naturvitenskapelige universitet (NTNU) våren 2010.

Veiledere har vært Lise Lyngsnes Randeberg og Martin Denstedt, begge ved NTNU.

Jeg vil takke begge veiledere for all hjelp og konstruktive tilbakemeldinger i løpet av prosjektet. Jeg vil også takke Åsmund Bakke Bø for gode råd.

Innholdsfortegnelse

1. Innledning	1
2. Teori	3
2.1. Støykarakterisering	3
2.2. Principal Component Analysis (PCA)	4
2.3. Maximum Noise Fraction (MNF)	7
2.3.1. Estimering av kovariansmatrisen til støyen	9
2.3.2. Støyestimering for homogent felt	11
2.3.3. Støyestimering i fourierrommet	12
2.3.4. Antall band under tilbaketransformering	14
2.4. Wavelets	15
3. Metode	17
3.1. Eksperimentelt oppsett	17
3.2. Valg av programmeringsspråk	19
3.3. Fjerning av horisontal støystripe	20
3.4. MNF	22
3.4.1. Estimering av kovariansmatrisen til støyen	24
3.4.2. Støyestimering for homogent felt	28
3.4.3. Antall band under tilbaketransformering	29
3.4.4. Støyestimering i fourierrommet	30
3.5. PCA	31
3.6. Wavelets	32
4. Resultater	35
4.1. Fjerning av horisontal støystripe	35
4.2. Sammenligning av MNF og PCA	37
4.2.1. Estimering av kovariansmatrisen til støyen	38
4.2.2. Støyestimering for homogent felt	39
4.2.3. Støyestimering i fourierrommet	40
4.2.4. Antall band under tilbaketransformering	42
4.3. Støyreduksjon ved ulik belysning	45
5. Diskusjon	51
5.1. Fjerning av horisontal støystripe	51
5.2. Sammenligning av MNF og PCA	52
5.2.1. Estimering av kovariansmatrisen til støyen	53
5.2.2. Støyestimering for homogent felt	55
5.2.3. Støyestimering i fourierrommet	57
5.2.4. Antall band under tilbaketransformering	59
5.3. Støyreduksjon ved ulik belysning	61
6. Konklusjon	67
7. Referanseliste	69

8. Appendiks	71
8.1. Utledning for kovariansen	71
8.2. Utledning for støyfraksjon	72
8.3. Utledning for venstrehånds-egenvektorer	73
8.4. Bildeutsnitt	74
8.5. SNR-verdier basert på totalt utsnitt	76
8.6. SNR-verdier basert på homogen del	81
8.7. SNR-verdier for støyestimering i fourierrommet	84
8.8. Spekter for enkel rad	86
8.9. 3D-utsnitt for spektralband 10 (444nm)	100
8.10. 3D-utsnitt for spektralband 70 (662nm)	114
8.11. Støyfordeling til spekter for enkel rad	128
8.12. Støyfordeling til 3D-utsnitt for spektralband 10 (444nm)	131
8.13. Støyfordeling til 3D-utsnitt for spektralband 70 (662nm)	134

Figurliste

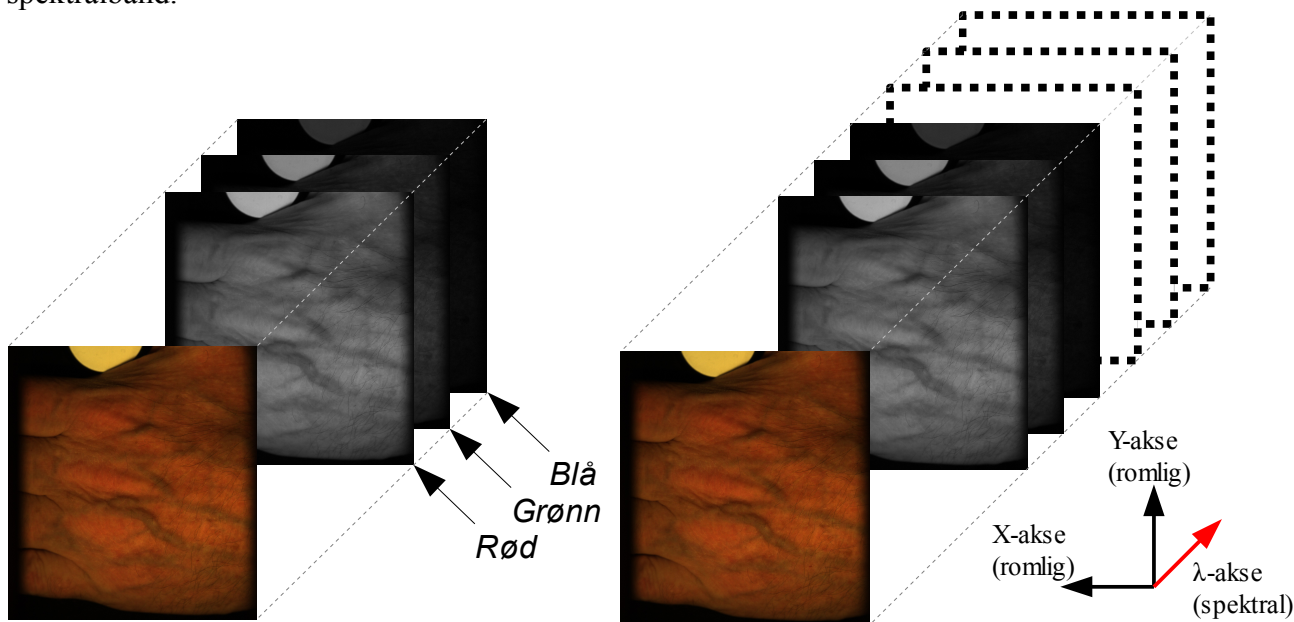
1.1: Vanlig fargebilde	1
1.2: Hyperspektralt bilde	1
2.1.1: Tetthets-sannsynlighets-fordelinger	3
2.2.1: Eksempelsett for PCA	4
2.3.1.1: Gråskala rampe	10
2.3.1.2: Illustrasjon av sadelpunkt	10
2.3.2.1: Plassering av refleksjonsstandard	11
2.3.3.1: Flytdiagram for MNF-algoritmen i fourierrommet	12
2.4.1: Dekomponering av signal med wavelet-transform	15
2.4.2: Flytdiagram for støyreduksjon under wavelet-transform	16
3.1.1: Oppsett for å ta hyperspektrale bilder	17
3.1.2: Testoppsett for hyperspektrale bilder	17
3.3.1: Hyperspektralt bilde med horisontal støystripe	20
3.4.1.1: Referanser til nabopiksler for 3x3 maske	24
3.4.1.2: 3x3 maske for horisontal nabopiksel	24
3.4.1.3: 3x3 maske for vertikal nabopiksel	24
3.4.1.4: 3x3 maske for snittet av vertikal og horisontal nabopiksler	25
3.4.1.5: 3x3 maske for lokalt gjennomsnitt	25
3.4.1.6: 3x3 maske for Gaussisk fordeling	25
3.4.1.7: 5x5 maske for Gaussisk fordeling	26
3.4.1.8: 7x7 maske for Gaussisk fordeling	26
3.4.1.9: 3x3 maske for median støyestimator	26
3.4.1.10: 3x3 maske for andrederiverte i x-retning (horisontal)	26
3.4.1.11: 3x3 maske for andrederiverte i y-retning (vertikal)	26
3.4.1.12: Summen av absoluttverdiene til de andrederiverte	27
3.4.2.1: Utsnitt av det hyperspektrale bildet som blir studert	28
3.4.3.1: Reduksjonsmatrise for å begrense antallet prinsipalkomponenter under MNF	29
3.5.1: Reduksjonsmatrise for å begrense antallet prinsipalkomponenter under PCA	31
3.6.1: Haar-transformen illustrert ved 3 iterasjoner	32
4.1.1: Akkumulert kvadrert differanse mellom horisontale linjer i hyperspektrale bilder	35
4.1.2: Første prinsipalkomponent til bilde 3 og 4	36
4.2.1: De 6 første prinsipalkomponentene til bilde 1 under MNF-algoritmen	37
4.2.1.1: Første prinsipalkomponent til bilde 3 med ulike estimatører	38
4.2.2.1: Prinsipalkomponenter til bilde 3 og 4 ved homogen støyestimering	39
4.2.3.1: Akkumulerte og normaliserte SNR-verdier	40
4.2.4.1: Spekter til enkelt piksel etter støyreduksjon	42
4.2.4.2: Grense for 2 og 40 prinsipalkomponenter	43
4.2.4.3: Utsnitt av spektralband 10 til bilde 2	43
4.3.1: Spektral støydempning for enkelt rad under hvitt lys	46
4.3.2: Spektral støydempning for enkelt rad belyste av 355nm laserkilde	47
4.3.3: Romlig støydempning for høyintensitets-spektralband	48
4.3.4: Romlig støydempning for lavintensitets-spektralband	49
5.2.1.1: Gauss-masker med størrelse 3x3, 5x5 og 7x7	53
5.3.1: Bruk av celle under Daubechie med 2 og 4 koeffisienter	62

Tabelliste

4.2.1.1: Akkumulerte og normaliserte SNR-verdier basert på det totale utsnittet	38
4.2.2.1: Akkumulerte og normaliserte SNR-verdier basert på homogent felt	39
4.2.3.1: Akkumulerte og normaliserte SNR-verdier for fourierrommet	40
4.2.4.1: Antall prinsipalkomponenter man må ta med for å beholde en gitt prosent	44

1 Innledning

Hyperspektrale bilder består av mange spektralbånd. Vanlige fargebilder består av 3 farger: rød, grønn og blå. En kan se på hyperspektrale bilder som fargebilder, men med mange flere farger eller spektralbånd.



Figur 1.1: Vanlig fargebilde

Figur 1.2: Hyperspektralt bilde

I figur 1.1 vises oppbygningen av et vanlig fargebilde med komponentene rød, grønn og blå. Disse tre fargene danner tilsammen fargebildet illustrert nederst. Figur 1.2 viser et hyperspektralt bilde bestående av 6 spektralbånd. De hyperspektrale bildene som blir brukt i denne oppgaven har 160 spektralbånd.

Hyperspektral avbildning har tradisjonelt vært brukt til fjernmåling og måldeteksjon for militære anvendelser. Bildene har vært overvåkningsbilder fra fly eller satellitt. Hyperspektral avbildning er ikke begrenset til disse anvendelsene, og i denne oppgaven blir hyperspektral avbildning brukt på medisinske applikasjoner. Nederst i figur 1.1 og 1.2 ser man oversiden av en hånd, som er et av objektene det blir sett på i dette prosjektet.

Denne oppgaven er en videreføring av et prosjekt som ble gjennomført høsten 2009. Da ble Principal Component Analysis (PCA) forsøkt brukt til reduksjon av dimensjonalitet for hyperspektrale bilder. Hovedresultatet fra dette prosjektet var at PCA ikke egnert seg til reduksjon av dimensjonaliteten for de medisinske bildene som ble analysert. Et alternativ til PCA er Maximum Noise Fraction (MNF). Denne algoritmen er også laget med tanke på å redusere dimensjonaliteten for hyperspektrale bilder. Som en videreføring av høstprosjektet vil derfor MNF bli sett på som en mulighet for å redusere dimensjonaliteten for de medisinske bildene.

Algoritmene PCA og MNF kan imidlertid benyttes til andre formål enn bare å redusere dimensjonalitet. For denne oppgaven vil hovedformålet med PCA og MNF være støydemping på hyperspektrale bilder. I tillegg til disse to algoritmene vil også wavelet-transform bli trukket frem som en mulighet for å oppnå støydemping for hyperspektrale bilder.

Hyperspektrale bilder består av både spektral og romlig informasjon. De romlige aksene ligger i 2D-planet. Dette er illustrert ved X-akse og Y-akse i figur 1.2. Spektral-aksen strekker seg innover langs med 3D-strukturen i figur 1.2, og retningen er vist ved en rød pil. For spektral retning ønsker

en å få til glatting av spekteret. Støyreduksjon i det romlige planet er også viktig. Her skal dette skje uten at man reduserer romlig oppløsning eller mister relevant informasjon.

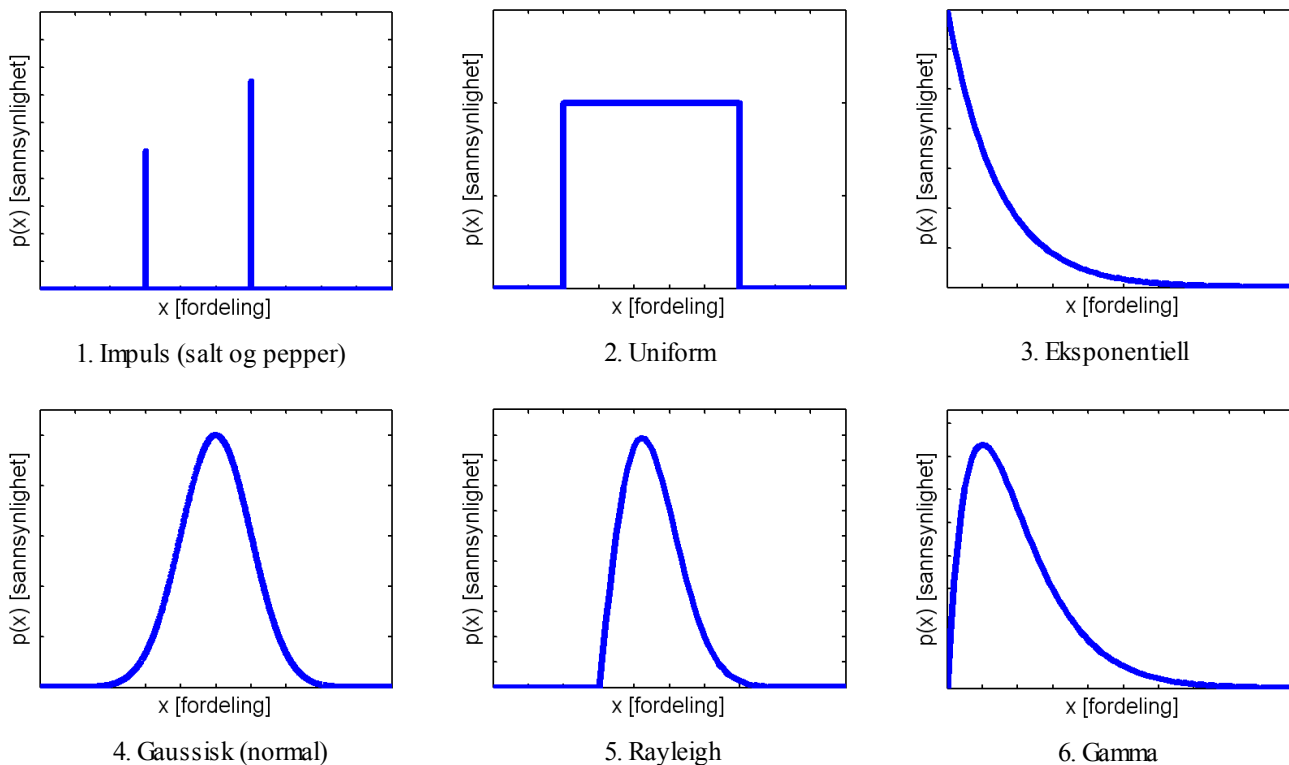
Slik støyfjerning gjør det langt lettere for etterfølgende analyser og algoritmer da mye av støyen har blitt filtrert bort. Dette vil typisk gjelde gjenkjenning av intensitetstopper for enkelte spektralbånd, men også romlige strukturer som kan ha blitt skjult av støy.

Algoritmene baseres på bruk av ulik informasjon. PCA benytter bare spektral informasjon. MNF er en videreutvikling av PCA. Den viktigste forskjellen mellom disse to algoritmene er at MNF tar med romlig informasjon i tillegg til spektral. Wavelet-transform vil i denne oppgaven bare bli basert på spektral informasjon.

2 Teori

2.1 Støykarakterisering

Støy er en uønsket komponent i bilder som forringer signalet. Her blir støy sett på som en additiv komponent. Det vil si at pikselverdiene Z i de hyperspektrale bildene kan skrives som $Z = S + N$, der S er signalet og N er støyen. Støykomponentene N er tilfeldige variable som kan karakteriseres av en sannsynlighets-tetthets-funksjon (PDF)^[1]. Støy kan opptre på mange forskjellige måter og figur 2.1.1 viser sannsynlighets-tetthets-funksjonene for noen støytyper.



Figur 2.1.1: Sannsynlighets-fordelings-funksjoner for støytyper^[1]

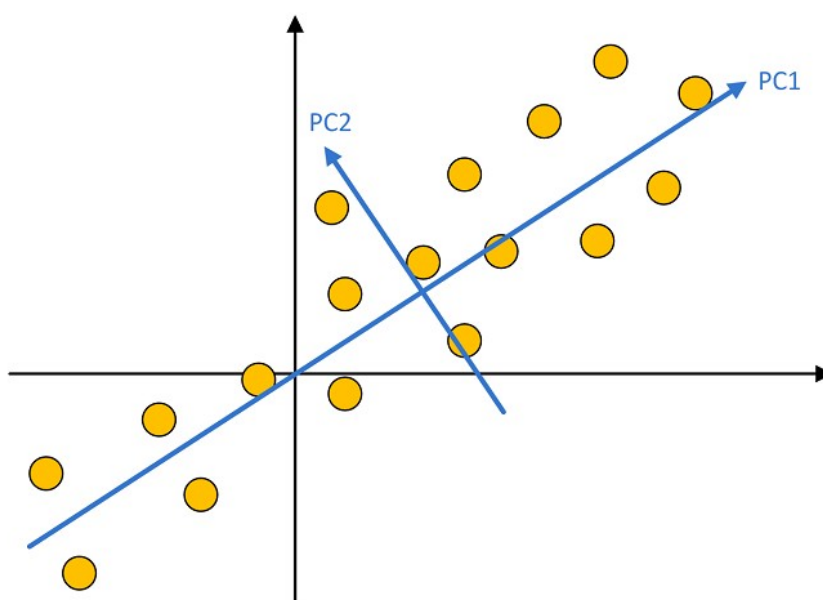
Matematiske funksjoner for distribusjonene vist i figur 2.1.1 er gitt i Gonzalez^[1]. Siden dette er sannsynlighetsfordelinger skal arealet under grafene være lik 1.

Disse støymodellene er i utgangspunktet benyttet for spatial støy. Hyperspektrale bilder har en spektrale akse i tillegg til to spatiale. Dette betyr at det vil være noe usikkerhet forbundet med bruk av de samme modellene for å karakterisere støy i spektral retning til de hyperspektrale bildene. Hvis man ser på det andre har gjort tidligere finner en at Ientilucci^[2] modellerer støyen som hvit gaussisk med et gjennomsnitt på null i spektral retning. Manolakis^[3] modellerer også additiv støy i spektral retning med gaussisk fordeling.

2.2 Principal Component Analysis (PCA)

PCA er en algoritme for å finne den underliggende statistiske dimensjonaliteten til et datasett. Velger her å kalle rommet PCA transformasjonen sender datasettet til for PCA-rommet. Algoritmen tar inn et datasett der komponentene kan være korrelert, og gir ut et nytt datasett med ukorrelerte komponenter. Med dette menes at komponentene man får i PCA-rommet står ortogonalt på hverandre. Dette PCA-rommet består av like mange dimensjoner som det er i datasettet PCA-algoritmen benyttes på. De hyperspektrale bildene som skal PCA-transformeres har i vårt tilfelle 160 bølgelengder. Hver av disse bølgelengdene tilsvarer en dimensjon. Dette betyr at i PCA-rommet vil det også være 160 prinsipalkomponenter som står ortogonalt på hverandre. Det transformerte datasettet i PCA-rommet er ordnet etter variansen. Langs med den første akse vil variansen være størst. Videre finner man den nest største variansen langs med den andre akse og minst varians finnes ved den siste prinsipalkomponenten.

En grafisk fremstilling av PCA, vil være at høyt korrelerte komponenter blir rotert til et ortogonalt rom. I det ortogonale rommet blir variansen bevart, og er ordnet etter synkende størrelse med stigende prinsipalkomponent.



Figur 2.2.1: Første prinsipalkomponent ligger langs med pilen PC1, andre prinsipalkomponent ligger langs med PC2^[9].

Figur 2.2.1 viser et eksempel på PCA for et todimensjonalt datasett. De gule prikkene skal illustrere dataene som finnes i datasettet. En kan se at datasettet vil ha størst varians langs med den blå pilen merket PC1. Dette vil være akse den første prinsipalkomponenten legger seg langs med i PCA-rommet. Ortogonalt på pilen PC1 står pilen PC2. Langs med denne pilen finner man den nest største variansen. Pil PC2 i figur 2.2.1 indikerer derfor akse den andre prinsipalkomponenten vil plasseres langs med. Siden dette datasettet bestod av bare 2 dimensjoner vil det også bare bli 2 prinsipalkomponenter i PCA-rommet.

Eksempel på PCA med hyperspektrale bilder er vist i rapporten til forprosjektet for denne oppgaven^[4]. Det er også vist et eksempel i Gonzales^[1].

PCA blir også kalt Hotelling transformasjon, Karhunen-Loève (K-L) transformasjon, "proper orthogonal decomposition" (POD) og egenvektor transformasjon^[5]. Dette er en algoritme for å

trekke ut informasjon, redusere dimensjonaliteten til og gi støyreduksjon av datasettet den blir benyttet på^[6, 7]. Hvis man igjen ser på figur 2.2.1 er det mulig å få et inntrykk av disse 3 tingene som PCA-algoritmen kan brukes til.

Det meste av informasjonen vil legge seg langs med første prinsipalkomponent vist i figur 2.2.1. Dette kommer av at det er størst varians langs med pilen PC1. For retningen til PC2 i figur 2.2.1 vil det være langt mindre informasjon, siden variansen i denne retningen er mye mindre.

Reduksjon av dimensjonalitet skjer ved at man unnlater de høyere prinsipalkomponentene. For figur 2.2.1 vil dette si at man beholder PC1 men dropper PC2. Rent grafisk fører det til at alle de gule datapunktene blir flyttet inntil den blå pilen PC1. Når man har flyttet alle prikkene som i utgangspunktet var plassert i det todimensjonale rommet inn på pilen PC1 ligger de langs med en endimensjonal strek. Dette vil si at man har redusert dimensjonaliteten fra to til en dimensjon. Merk at flyttingen av de gule prikkene skal skje i ortogonal retning mot pilen PC1, altså kortest avstand inn til PC1.

Støyreduksjon ser man først når en har tilbaketransformert datasettet fra PCA-rommet til rommet som var utgangspunktet. I PCA-rommet vil datasettet vist i figur 2.2.1 være dreid, slik at PC1 kommer langs med x-aksen, og PC2 kommer langs med y-aksen. Etter reduksjon av dimensjonalitet fra to- til en-dimensjonalt blir alle de gule sirklene flyttet ned til x-aksen eller PC1. Tilbaketransformasjonen vil da dreie datasettet slik at de gule prikkene blir plassert med svak stigning som PC1 har i figur 2.2.1.

Matematisk kan PCA beskrives som en algoritme på seks steg.

1. Først samles dataene i en 2D matrise. Hver rad representerer en dimensjon, og hver kolonne en samling av målinger for et spesifikt punkt.

$$\vec{X}_i = \begin{bmatrix} dim1 \\ dim2 \\ dim3 \\ \dots \\ dimP \end{bmatrix} \quad (2.2.1)$$

Over vises en P-dimensjonal kolonnevektor. For et hyperspektralt bilde bestående av MxN piksler, der hver piksel har P dimensjoner, gir dette en 2D-matrise med størrelse PxMN. Under vises feltet "Data" som lagrer det hyperspektrale bildet. Dette illustrerer også at PCA kan benyttes på alle typer datasett, og man kan fint unnlate enkeltpunkter uten at det har stor innvirkning på resultatet hvis MN er stor.

$$Data = [\vec{X}_1 \ \vec{X}_2 \ \vec{X}_3 \ \dots \ \vec{X}_{MN}] \quad (2.2.2)$$

2. Deretter trekkes gjennomsnittet for hver dimensjon fra datasettet. Gjennomsnittet er gitt ved

$$\overline{dimX} = \frac{1}{MN} (dimX_1 + dimX_2 + dimX_3 + \dots + dimX_{MN}) = \frac{\sum_{i=1}^{MN} dimX_i}{MN} \quad (2.2.3)$$

Dette gir et datasett med gjennomsnittsverdi lik 0.

3. I dette steget beregnes kovariansmatrisen til datasettet. Kovariansen mellom verdiene ved de ulike bølgelengdene til det hyperspektrale bildet viser hvor korrelert datasettet er. Kovariansen er gitt ved

$$C = cov(X, Y) = \frac{\sum_{i=1}^{MN} (dimX_i - \overline{dimX})(dimY_i - \overline{dimY})}{MN} \quad (2.2.4)$$

Her refererer X og Y i $\text{cov}(X, Y)$ til to tilfeldige bølgelengder eller spektralbånd. Merk at når $X=Y$ finner man variansen til spektralbåndet X .

4. Det neste er å beregne egenverdier og egenvektorer for kovariansmatrisen C . Dette vil i praksis være å løse ligningen $Cx = \lambda x$. For denne ligningen er λ egenverdier og x er egenvektorer^[8]. Egenvektorene står alltid ortogonalt på hverandre. Siden man finner egenvektorene til kovariansmatrisen, vil disse danne et grunnlag for koordinatsystemet man var på utkikk etter. Egenverdiene gir målet på andelen man trenger av hver egenvektor for å danne kovariansmatrisen. Dette betyr at hvis man plasserer aksene i retning gitt av egenvektorene i PCA-rommet etter synkende korresponderende egenverdi, vil også variansen langs med aksene i PCA-rommet være ordnet i synkende rekkefølge.

5. Videre sorteres egenvektorene etter korresponderende egenverdi. Dette gjøres i synkende rekkefølge, slik at egenvektoren med størst korresponderende egenverdi blir første kolonne i en egenvektormatrise.

$$\text{Egenvektormatrise} = [eig_1 \ eig_2 \ eig_3 \ \dots \ eig_n] \quad (2.2.5)$$

6. Til slutt multipliseres den transponerte til "Egenvektormatrise", med det opprinnelige datasettet der gjennomsnittsverdiene er trukket ifra. Dette transformerer det hyperspektrale bildet til PCA-rommet.

$$\text{SluttData} = \text{Egenvektormatrise}^T \cdot \text{DataSnitt} \quad (2.2.6)$$

"SluttData" er matrisen som holder alle prinsipalkomponentene. Prinsipalkomponentene er sortert etter synkende korresponderende egenverdi. Variansen vil derfor være synkende med stigende prinsipalkomponent. Reduksjon av dimensjonalitet skjer ved at man utelater de høyeste prinsipalkomponentene. Hvis egenverdiene til disse prinsipalkomponentene er små, vil man ikke miste mye informasjon ved å fjerne dem.

Ved å sortere prinsipalkomponentene etter synkende egenverdi ønsker en også at de blir sortert etter støy eller signal-støy-forhold. Med andre ord ønsker man mest støy blant de høyeste prinsipalkomponentene, og minst ved de laveste. Blir prinsipalkomponentene ordnet på denne måten vil det også være mulig å benytte PCA til støyreduksjon. Det utføres ved å begrense antallet prinsipalkomponenter som inngår i tilbaketransformeringen. Fjerner man de høye prinsipalkomponentene blir da det tilbaketransformerte datasettet støyreduert.

For å komme tilbake til det opprinnelige rommet multipliseres "SluttData" med den inverse til "Egenvektormatrise^T". Deretter blir gjennomsnittsverdiene som ble trukket ifra under punkt 2, igjen lagt til datasettet^[7].

$$\text{OpprinneligData} = (\text{Egenvektormatrise}^T)^{-1} \cdot \text{SluttData} + \text{Snitt} \quad (2.2.7)$$

For hele ligningen under ett blir det

$$\text{OpprinneligData} = (\text{Egenvektormatrise}^T)^{-1} \cdot \text{Egenvektormatrise}^T \cdot \text{DataSnitt} + \text{Snitt} \quad (2.2.8)$$

2.3 Maximum Noise Fraction (MNF)

MNF er en alternativ algoritme til PCA. Denne algoritmen blir også referert til som minimum noise fraction og støy-justert transformasjon (NAPC)^[5]. Ved å benytte PCA på et hyperspektralt bilde vil man ikke alltid få ut prinsipalkomponentene ordnet etter synkende bildeklarhet. Dette kom tydelig frem under forprosjektet som gikk på høsten 2009^[4]. Årsaken er at enkelte støykomponenter har høyere varians enn enkelte signalkomponenter^[10]. Signal-støy-forhold (SNR) er et mye brukt mål på bildeklarhet, og MNF garanterer at prinsipalkomponentene kommer ut med synkende SNR^[11]. Etter at man har transformert det hyperspektrale bildet til MNF-rommet kan støyreduksjon utføres på datasettet. Dette skjer ved å fjerne komponentene med mest støy. Når komponentene med mye støy er fjernet, transformerer man tilbake til det originale rommet. Da vil man ha oppnådd støyreduksjon uten å redusere romlig oppløsning, som kan skje ved bruk av for eksempel lavpassfilter.

Anta at et hyperspektralt bilde består av n spektralbånd, og det har romlige koordinater x og y . Datasettet til det hyperspektrale bildet kan da skrives^[11]

$$A_i(x, y), \quad i=1, \dots, n \quad (2.3.1)$$

Videre blir det antatt at man kan splitte $A_i(x, y)$ i en signalkomponent og en støykomponent. $S(x, y)$ er signalet og $N(x, y)$ er støykomponenten^[11].

$$A(x, y) = S(x, y) + N(x, y) \quad (2.3.2)$$

Denne forutsetningen krever at signalet og støyen er ortogonale. Hvis signalet og støyen er ortogonale er det ingen korrelasjon mellom dem^[12].

$$S(x, y)^T N(x, y) = N(x, y)^T S(x, y) = 0 \quad (2.3.3)$$

Dette gir at kovariansen til $A(x, y)$ blir (se appendiks 8.1 for utledning)

$$\text{Cov}\{A(x, y)\} = \Sigma = \Sigma_S + \Sigma_N \quad (2.3.4)$$

For spektralbånd nummer i er SNR gitt som forholdet mellom variansen til signalet, og variansen til støyen^[13]

$$\frac{\text{Var}\{S_i(x, y)\}}{\text{Var}\{N_i(x, y)\}} \quad (2.3.5)$$

MNF er definert slik at SNR for $Y_j(x, y)$ er maksimert blant alle lineære transformasjoner som står ortogonalt på $Y_j(x)$ der $j = 1, \dots, i-1$ ^[13]. Dette gjør at SNR synker for stigende prinsipalkomponent.

$$B_i(x, y) = a_i^T A(x, y), \quad i=1, \dots, n \quad (2.3.6)$$

SNR for $B_i(x, y)$ blir

$$\frac{\text{Var}\{a_i^T S(x, y)\}}{\text{Var}\{a_i^T N(x, y)\}} = \frac{a_i^T \Sigma a_i}{a_i^T \Sigma_N a_i} - 1 \quad (2.3.7)$$

Følger man utledningen for støyfraksjonen i appendiks 8.2 kommer man frem til

$$\frac{\text{Var}\{a_i^T N(x, y)\}}{\text{Var}\{a_i^T A(x, y)\}} = \frac{a_i^T \Sigma_N a_i}{a_i^T \Sigma a_i} \quad (2.3.8)$$

Dette resulterer i at vektorene a_i er gitt som egenvektorene til det generaliserte egenvektorproblemet^[13]

$$\det \{ \Sigma_N - \lambda \Sigma \} = 0 \quad (2.3.9)$$

SNR kan nå beregnes for $B_i(x, y)$ fra formelen

$$SNR = \frac{1}{\lambda_i} - 1 \quad (2.3.10)$$

der λ_i er egenverdiene^[13].

Matematisk kan MNF beskrives på flere måter. Prinsippkomponentene blir alltid ordnet etter SNR, men rekkefølgen kan både være i synkende og stigende SNR avhengig av definisjon. Green^[11] ordner prinsippkomponentene etter maksimal støyfraksjon. Altså er det mest støy i første prinsippkomponent, og minst i den siste. Dette vil si at SNR stiger med økende prinsippkomponent. Det er denne ordningen av prinsippkomponenter som henspiller til navnet "maximum noise fraction". Egenvektorene beregnes ved å finne venstrehånds-egenvektorer til $\Sigma_N \Sigma^{-1}$. Appendix 8.3 viser hvordan disse beregnes. En ulempe ved å finne egenvektorene på denne måten, er at beregningene kan være sensitive til aritmetisk presisjon^[14].

En tredje måte å utføre MNF på, er ved støy-justert transformasjon (NAPC). Her kan man se på MNF som to PCA transformasjoner kjørt etter hverandre^[5]. Først transformeres datasettet til et koordinatsystem der kovariansmatrisen til støyen er identitetsmatrisen^[14]. Dette kan også ses på som hvitning av originaldataene^[15], eller dekorrelering og reskalering av støyen i det hyperspektrale bildet. Støyen får enhetsvarians og den vil ikke ha noen korrelasjon mellom båndene^[16]. Deretter utføres PCA på det transformerte datasettet^[14]. Fordelen med å implementere MNF på denne måten, er at implementasjonen blir veldig enkel hvis man allerede har tilgang til PCA^[14]. Programvaren ENVI «the Environment for Visualizing Images» utfører MNF som PCA kjørt to ganger^[16].

Etter samme prinsipp som for PCA kan MNF brukes til støyfjerning. Det vil si at man beholder prinsippkomponentene med høyt signal-støy-forhold og utelater de med lavt under tilbaketransformeringen.

Tilbaketransformering fra MNF-rommet til det opprinnelige rommet er nesten lik som for PCA. Her skal det bare utføres en matrisemultiplikasjon med den inverse til a^T ^[11].

$$A(x, y) = (a^T)^{-1} B(x, y) \quad (2.3.11)$$

2.3.1 Estimering av kovariansmatrisen til støyen

Resultatet fra MNF vil avhenge av hvor godt kovariansmatrisen til støyen er estimert. Å se på forskjellige måter for estimering av støyen er derfor viktig. Den opprinnelige tanken for å estimere kovariansen til støyen, er at signalet endres tregere enn støyen. Altså tar man utgangspunkt i at signalet til et tilfeldig punkt er sterkt korrelert med signalet til nabopunktene^[11]. Denne antagelsen er gjort for hyperspektrale bilder som viser jordoverflaten fra fly eller satellitt. I dette prosjektet blir det sett på medisinske bilder. Det vil derfor være en mulighet at støyen og signalet kan opptre på en annen måte, enn for bildene tatt til fjernmåling og måldeteksjon. På prosjektet i høst ble resultatet at PCA ikke klarte å ordne prinsipalkomponentene etter synkende bildekvalitet^[4]. Ser man på resultatene til Chen^[5] for PCA av hyperspektrale bilder brukt til fjernmåling, er det en tydeligere sammenheng mellom stigende prinsipalkomponent og synkende bildekvalitet. Derfor er det nærliggende å tro at det man avbilder er med på å påvirke resultatene.

Det er skrevet mange artikler om estimering av støyen i hyperspektrale bilder. De aller fleste tar utgangspunkt i bilder for måldeteksjon og fjernmåling. Derfor velges det å gjøre denne analysen på nytt for medisinske bilder, som ser helt annerledes ut enn de tradisjonelle bildene tatt fra fly eller satellitt.

Utgangspunktet for å estimere kovariansmatrisen for støyen kalles minimum/maximum autocorrelation factors (MAF). MAF bygger på at signalet i hyperspektrale bilder tatt for fjernmåling er sterkt korrelert med nabopunkter. Dette gjelder ikke for støyen i disse bildene, som har svak korrelering med nabopiksler i det romlige domenet^[11]. Notasjonen $\Delta = (\Delta_x, \Delta_y)$ ^[17] benyttes for å beskrive spatial forskyvning. Et estimat for støykovariansmatrisen Σ_N kan beregnes ut i fra $\text{Cov}\{A(x, y) - A(x + \Delta_x, y + \Delta_y)\}$. Typiske valg for Δ er (1, 0) eller (0, 1). Det siste alternativet kan benyttes for å fjerne horisontale støystriper^[11].

Et problem med dette støyestimatet, er støy som ikke fremkommer ved å se på differansen med nærmeste nabo. Dette kan være periodisk støy med periode på flere piksler. Støyestimatet vil da ikke være tilfredsstillende^[14].

Istedenfor bare å se på differansen med en av de nærmeste naboene, kan man bruke gjennomsnittet av differansen mellom flere nabopiksler som støyestimat^[13, 18]. Denne måten å estimere kovariansmatrisen til støyen på, blir blant annet brukt av programmet ENVI (the Environment for Visualizing Images)^[16].

Andre metoder som også har vært brukt tidligere er differansen med lokalt gjennomsnitt, og differansen med lokalt median^[13, 18]. En ulempe ved bruk av filter som baserer seg på lokalt gjennomsnitt er utsmøring (bluring) av kanter og detaljer. Medianfilter unngår denne effekten.

Etter litteratursøk har man ikke klart å fastslå om det er blitt gjort forsøk på å estimere kovariansmatrisen ved bruk av Gaussisk kjerne eller den andrederiverte. Disse to måtene blir derfor også tatt med som estimat for støyen.

Den Gaussiske kjernen ligner på lokalt gjennomsnitt, men det er en ulik vektning av pikslene. Vektingen er gitt som en Gausskurve, der de nærmeste naboene vektet tyngst. Gaussiske filter har i likhet med lokale gjennomsnittsfilter en utsmørende effekt^[1]. Her skal ingen av de to filtrene direkte brukes som utsmøringsfilter (smoothing/glatting), men heller være med på å gi et estimat på støyen. Dette gjøres ved å se på differansen mellom pikselverdien $A_i(x, y)$ og den vektete gjennomsnittsverdien som Gaussfilteret gir for den aktuelle pikselen.

Valget om å ta med et filter som er Gaussfordelt gjøres fordi det vektet de nærmeste nabopunktene tyngre enn punkter som ligger litt lengre unna. Da det lokale gjennomsnittsfilteret vektlegger alle piksler likt i masken, vil Gaussfilteret være mer i tråd med Greens opprinnelige ide^[11]. Altså at signalet er tilnærmet likt mellom nabopunkter, mens støyen varierer. Dette bør

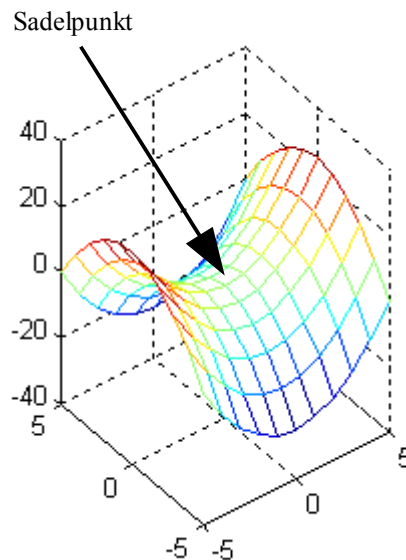
fremkomme tydeligere ved bruk av Gaussfilter over et vanlig gjennomsnittsfiler, da en endring mot en av de nærmeste nabopunktene blir vektet tyngre enn endringen mot punktene som ligger lengre unna.

MAF bruker den førstederiverte for å gi et støyestimant. Bruken av ligningen $A(x, y) - A(x + 1, y)$ langs med enhver rampe vil slå ut som støy, selv om rampen kan være en del av signalet. Den andrederiverte vil bare slå ut i overganger eller knekkpunkter. Andrederiverte har typisk blitt brukt til deteksjon av isolerte punkter^[1]. Salt-og-pepper støy kan ofte opptre som isolerte punkter^[19], derfor blir den andrederiverte tatt med som et mulig støyestimant.



Figur 2.3.1.1: Gråskala rampe. Førstederiverte $A(x, y) - A(x - 1, y)$ vil slå ut som støy langs med hele rampen, mens den andrederiverte bare vil gi utslag ved knekkpunkter.

Det er mulig å se på den andrederiverte i en retningen av gangen, for eksempel x- eller y-retning. En kan også slå sammen bidragene fra flere retninger. Hvis man slår sammen bidrag kan de utligne hverandre om de har motsatt fortegn, men er ellers like. Dette vil typisk gjelde sadelpunkt^[20]. Figur 2.3.1.2 illustrerer et sadelpunkt. For å unngå utslukking i sadelpunkt kan en summere sammen absoluttverdien av bidragene.



Figur 2.3.1.2: Illustrasjon av sadelpunkt. Gitt av ligningen $f(x, y) = x^2 - y^2$

Den diskrete andrederiverte kan defineres som følgende^[1]

$$\frac{\partial^2 f}{\partial x^2} = (f(x+1) - f(x)) - (f(x) - f(x-1)) = f(x+1) + f(x-1) - 2f(x) \quad (2.3.1.1)$$

Dette er for et endimensjonalt datasett. Bildene er imidlertid todimensjonale. Laplaceoperatoren kan da brukes, men som tidligere nevnt kan den få utslukking i sadelpunkt. Laplacian er^[1]

$$\Delta f = \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (2.3.1.2)$$

Ved bruk av absoluttverdien, blir summen av leddene følgende

$$\left| \frac{\partial^2 f}{\partial x^2} \right| + \left| \frac{\partial^2 f}{\partial y^2} \right| \quad (2.3.1.3)$$

2.3.2 Støyestimering for homogent felt

Når de hyperspektrale bildene for dette prosjektet ble tatt, var det med en refleksjonsstandard i bildene. Tanken med denne refleksjonsstandarden er å kunne konvertere bildene til refleksjonsbilder, og dermed fjerne bidraget fra lyskilden. Dette gjøres ved å først lese ut bildet som ligger innenfor refleksjonsstandarden. Deretter deler man det totale bildet på utsnittet fra refleksjonsstandarden. Dette gjør at påvirkningen fra lyskilden blir kraftig redusert^[4]. Her er ikke målet å fjerne påvirkningen fra lyskilden, men heller å få til et bedre støyestimat der støyestimeringen er begrenset til en homogen del av bildet. Fordelen med å benytte en homogen eller jevn del av et bildet, er at man vet hvordan det egentlige signalet skal være for denne biten. Signalet skal være helt jevnt uten noen form for ukorrelert støy. Når man bruker felter som ikke er homogene, eventuelt hele bildeutsnittet for å gjøre støyestimeringen kan man ikke garantere at støyestimatoren bare slår ut på støy. Det vil alltid være en mulighet for at deler av signalet registreres som støy. Dette vil ikke skje for jevne områder der alle endringer må stamme fra støy og kan derfor ikke være en del av signalet. Refleksjonsstandarden er et stabilt homogent område i samtlige av de hyperspektrale bildene som ble tatt for prosjektet, og egner seg derfor til dette formålet.



Figur 2.3.2.1: Den sorte pilen i den grønne firkanten peker på refleksjonsstandarden.

Refleksjonsstandarden er den hvite disken som er delvis synlig innenfor den grønne firkanten i figur 2.3.2.1. Denne egenskapen med at de hyperspektrale bildene har et homogent område er ikke like lett å få til for overvåkningsbilder tatt med fly eller satellitt. Årsaken er naturlig nok at man ikke kan plassere inn et homogent objekt på samme måten. Det har imidlertid vært utført forsøk på å finne homogene objekter i de hyperspektrale bildene, da støyestimering fra slike områder kan gi bedre støyestimat^[21, 22].

En ulempe med å basere støyestimatet på et jevnt utsnitt fra de hyperspektrale bildene er at dette begrenser støyestimeringen til et lite område i det romlige domenet. Støy som avhenger av romlig plassering i de hyperspektrale bildene kan derfor risikere å ikke bli tatt hensyn til.

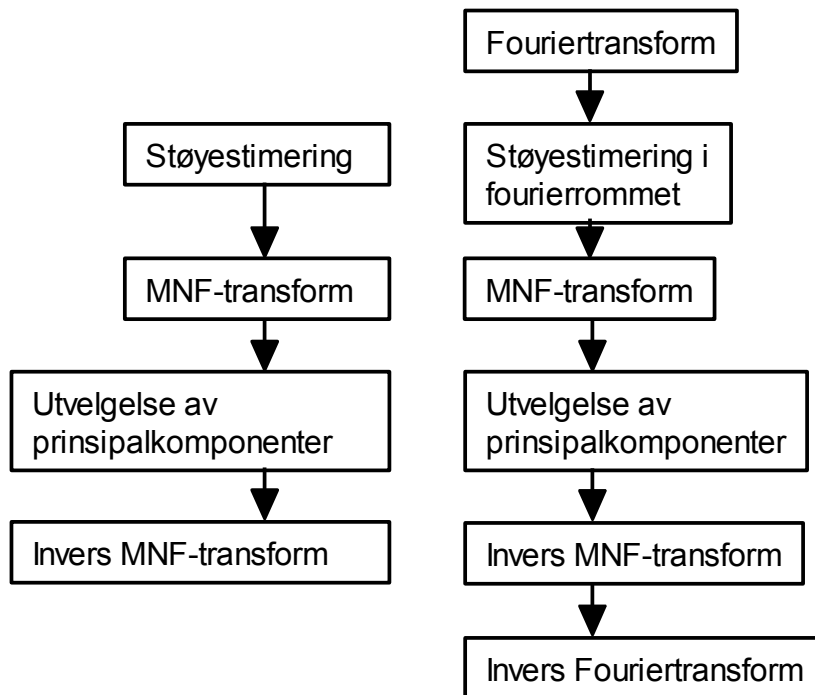
I programmet ENVI kan man også velge ut et homogent område for det hyperspektrale bildet til støyestimeringen^[16]. Her utføres det også for å oppnå et best mulig støyestimat.

2.3.3 Støyestimering i fourierrommet

Etter undersøkelser har man ikke klart å finne tidligere forsøk på støyestimering i fourierrommet. Dette trekkes derfor her frem som en alternativ form for støyestimering. I det vanlige spatiale rommet foreslo Green^[11] å benytte differansen med nærmeste nabo i horisontal eller vertikal retning for å utføre støyestimeringen. Andre forslag for tilsvarende rom er bruk av differansen med median eller lokalt gjennomsnitt^[18] til en maske. Her blir det presentert en ny variant som gjør støyestimering i fourierrommet.

I frekvensrommet har man ikke tilgang til differansen med nærmeste spatiale naboer som er benyttet i mange varianter nevnt over. Det som derimot kan beregnes er differansen med nære frekvenser. Green^[11] benytter at signalet endres relativt tregt i det spatiale rommet sammenlignet med støy. Hurtige endringer i det spatiale rommet er i praksis firkantfunksjoner. Dette vil gi sincfunksjoner når man ser på absoluttverdien i fourierrommet.

Videre er det fouriertransformerte bildet representert ved komplekse tall, og ikke bare reelle. Dette betyr at de inneholder både amplitude og fase for de forskjellige frekvenskomponentene. Her er det differansen til amplitude og fase for nærliggende frekvenskomponenter som vil bli brukt under støyestimeringen.



Figur 2.3.3.1: MNF-algoritmen med støyestimering i fourierrommet er vist til høyre, og den tradisjonelle til venstre.

Figur 2.3.3.1 viser hvordan det opprinnelige oppsettet er bygget opp med støyestimering i det spatiale rommet, og hvordan det vil se ut med støyestimering i fourierrommet. Den vanlige MNF-algoritmen er illustrert til venstre i figuren og den nye til høyre. De sammenlignbare trinnene er vist ved siden av hverandre. Den nye algoritmen begynner først med en fouriertransformasjon av det opprinnelige hyperspektrale bildet før resten av MNF-algoritmen kan bli kjørt. Understreker at ideen er å gjøre en 2D fouriertransform av hvert spektralbånd i det hyperspektrale bildet og ikke en eventuell "3D" av hele bildet. Tanken er altså å gjøre støyestimeringen for hvert enkelt spektralbånd, og en skal ikke trekke inn spektral informasjon som blant annet blir gjort av Liu^[22]. Etter at det hyperspektrale bildet har blitt fouriertransformert utføres trinnet støyestimering. Her ligger den store forskjellen mellom den tradisjonelle støyestimeringen i det spatiale rommet, da den nå vil bli utført i fourierrommet. Dette betyr at en får tilgang på andre data enn det man vanligvis har. Støyestimatene vil bli gjort med to grunnleggende forskjellige varianter. Den ene varianten tar

utgangspunkt i absoluttverdiene for frekvenskomponentene. Dette betyr at sammenligningen som gjøres mellom nærliggende frekvenskomponenter er basert på den lokale differansen i mengde eller amplitude det er mellom dem. Den andre varianten baseres på differansen til fasen mellom frekvenskomponentene. Fasen vil alltid ligge mellom -180° og 180° . I motsetning til amplituden reflekterer fasen ikke på noen måte mengden eller amplituden til frekvenskomponentene. Fasen anslår hvor den aktuelle frekvensen skal ha topp og bunn. Etter at støyestimeringen har blitt utført i fourierrommet fortsettes algoritmen på vanlig måte med MNF-transform, utvelgelse av prinsipalkomponenter og invers MNF-transform. På slutten blir det også lagt til ett ekstra trinn. Dette skal ta datasettet tilbake til det spatiale rommet fra fourierrommet, og utføres med en invers fouriertransform.

Tar en et overblikk over de to variantene av MNF-algoritmen er kjernen tilnærmet identisk. Starten og slutten er endret med en fouriertransform for å komme til fourierrommet, og en invers for å komme tilbake til der en startet. Til høyre i figur 2.3.3.1 foregår hele MNF-algoritmen i fourierrommet. Den viktigste endringen dette medfører er at støyestimatet blir nå gjort med differanser for nærliggende frekvenser istedenfor spatiale naboer.

2.3.4 Antall band under tilbaketransformering

Det finnes mange teorier for hvordan man skal velge grensen for hvilke prinsipalkomponenter som skal være med under tilbaketransformasjonen og hvilke som skal utelates^[23]. En mulighet er å benytte seg av visuell inspeksjon av prinsipalkomponentene i MNF-rommet for å se hvilke som inneholder informasjon og hvilke som består hovedsakelig av støy. Den største ulempen med denne teknikken er at den ikke kan automatiseres. Det kreves et menneske som kan se på prinsipalkomponentene for å bestemme grensen mellom dem som inneholder hovedsakelig støy eller signal. Dette vil heller ikke være et objektivt resultat da grensen som velges kan avhenge av øyet som ser. Egenverdiene til prinsipalkomponentene er et mål på støyen for komponentene. Når egenverdien er lav er det lite støy og vice versa. Sammenligning av egenverdiene mot en grenseverdi for å bestemme hvilke komponenter som skal med fra MNF-rommet under reverstransformasjon har også vært gjort tidligere^[23].

Et ytterpunkt ved utvelgelse av antall prinsipalkomponenter under tilbaketransformeringen vil være å kun ta med første prinsipalkomponent. Da vil alle spektralbandene man får bli identisk lik prinsipalkomponenten. Dette er opplagt et uakseptabelt resultat, da det er langt fra hvordan de forskjellige spektralbandene faktisk ser ut. Det andre ytterpunktet er å ta med alle prinsipalkomponentene. Da vil spektralbandene etter transformeringen bli identiske med det datasettet man ga inn i algoritmen. Dette resultatet er også for dårlig siden en ikke har oppnådd noen form for støyreduksjon.

Det man skal velge er en mellomting mellom disse to ytterpunktene^[12]. Tar man med for få prinsipalkomponenter under tilbaketransformeringen blir støyreduksjonen bra, men reduksjonen av signalet kan også fort bli for høy. Velger man å ta med for mange komponenter vil det fremdeles være igjen mye støy etter tilbaketransformasjonen.

Her vil det bli sett på en måte som tar utgangspunkt i at man vil beholde en gitt prosent av det opprinnelige signalet. Dette betyr at det kan variere fra bilde til bilde hvor mange prinsipalkomponenter som skal være med under tilbaketransformasjonen for å beholde den bestemte prosenten. Signal-støyforholdet er gitt som $1/\lambda_i - 1$ der λ_i er egenverdiene^[13]. Mengden av det opprinnelige signalet som blir med i tilbaketransformasjon kan bestemmes ved å se på de akkumulerte SNR-verdiene. Når de akkumulerte SNR-verdiene ligger ved den prosenten man vil beholde av det opprinnelige signalet, har en også antallet prinsipalkomponenter som må beholdes under tilbaketransformasjonen.

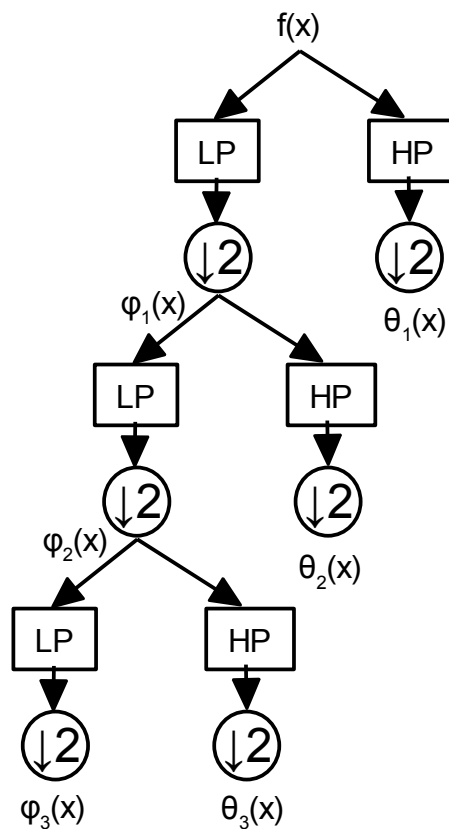
Det finnes også andre teknikker for å oppnå støyreduksjon enn å fjerne prinsipalkomponentene som ikke skal med under tilbaketransformeringen. Fjerning av prinsipalkomponenter er den mest drastiske formen for støyreduksjon. Ulempen med denne teknikken er at den vil også ta bort signalet som finnes i prinsipalkomponentene som blir utelatt under tilbaketransformeringen^[11]. For å få til støyreduksjon uten å fjerne noen av prinsipalkomponentene er det vanlig å benytte spatial støyfiltrering av de høyere prinsipalkomponentene som inneholder for det meste støy. En variant er å benytte et adaptivt filter med økende maskestørrelse når signal-støynivået synker for de høyere prinsipalkomponentene^[24]. Når signal-støyforholdet synker med stigende prinsipalkomponent kan man bruke et filter med større utsmørende effekt enn filter med kjerne på 3x3. Dette skjer ved å øke maskestørrelsen. Store filter gir mye støyfiltrering samtidig som et eventuelt lavfrekvent signal blir bevart.

2.4 Wavelets

Wavelet-transform kan sammenlignes med fouriertransform. Wavelet-transform benytter wavelets som basisfunksjoner i motsetning til fouriertransform som har sinus og cosinus funksjoner^[1]. Wavelets er bølger som består av flere frekvenser og har begrenset varighet eller "compact support"^[25]. Wavelet-transform gir både informasjon om frekvenser til et signal og hvor de befinner seg i signalet.

Wavelet transform har tidligere blitt brukt til reduksjon av dimensjonalitet for hyperspektrale bilder^[26]. I tillegg til å redusere dimensjonalitet kan wavelet-transform benyttes til støyreduksjon^[27]. Her vil det bli sett på støyreduksjon for hyperspektrale bilder ved bruk av wavelet-transform.

Wavelet-transform deler et signal opp i en detalj-del og en gjennomsnitt-del. Detalj-delen blir ofte kalt for wavelet-del. Inndelingen i detalj- og gjennomsnitt-del kan også bli sett på som en høypass- og lavpass-del.

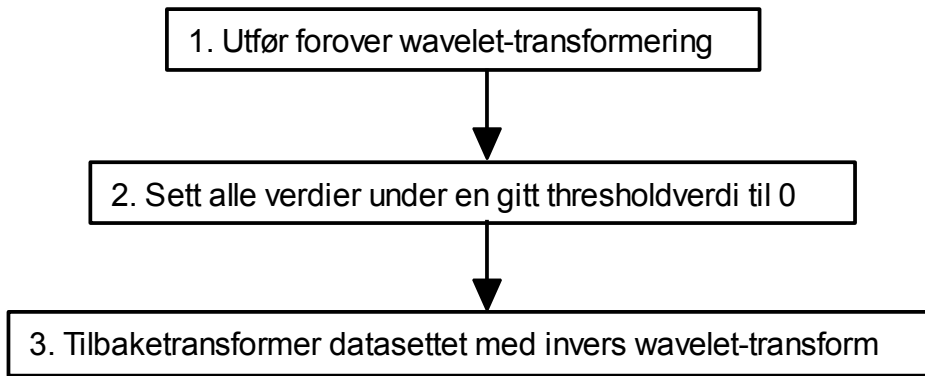


Figur 2.4.1: Dekomponering av signal med wavelet-transform^[28]

Figur 2.4.1 viser de 3 første iterasjonene av en wavelet-transform. Wavelet-transformen kan forlenges med flere iterasjoner enn de som er vist i figur 2.4.1. Funksjonen $f(x)$ i toppen av figur 2.4.1 er signalet som skal wavelet-transformeres. $\varphi_i(x)$ og $\theta_i(x)$ henpeiler til henholdsvis lavpassfaktorene og høypassfaktorene etter den i 'te iterasjonen. Pil pekende nedover etterfulgt av et 2-tall symboliserer nedsampling av signalet.

Wavelet-transformen deler signalet i to. En detaljert del og en del som beskriver gjennomsnittet. Ved å sløyfe svake signaler etter inndelingen kan man oppnå støyreduksjon uten at dette påvirker hovedtrekkene i signalet. Hvis et signal er under en gitt thresholdverdi blir det tolket som svakt og satt til null. Dette kalles thresholding^[27]. De endrede koeffisientene blir tatt med under tilbaketransformasjonen.

Støyreduksjonen ved hjelp av wavelet-transformasjon kan beskrives som en prosedyre i 3 trinn. Dette blir vist i figur 2.4.2.



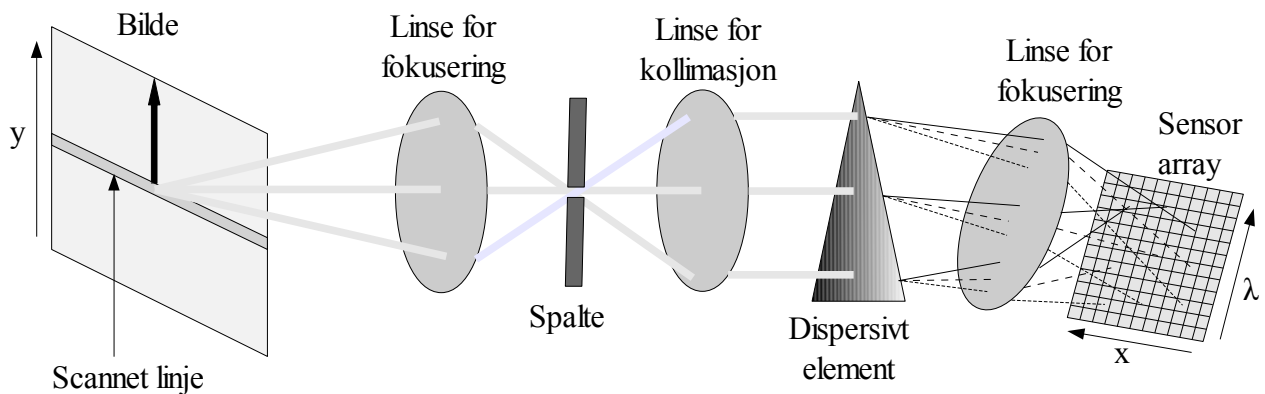
Figur 2.4.2: Støyreduksjon ved hjelp av wavelet-transform

Det finnes mange forskjellige wavelet-transformer. Her blir Haar-transform benyttet (se kapittel 3.6 for mer informasjon om Haar-transformen). Haar-transformen blir valgt fordi den er enkel å implementere^[25], og den gir eksakt rekonstruksjon av signalet ved invers transformering.

3 Metode

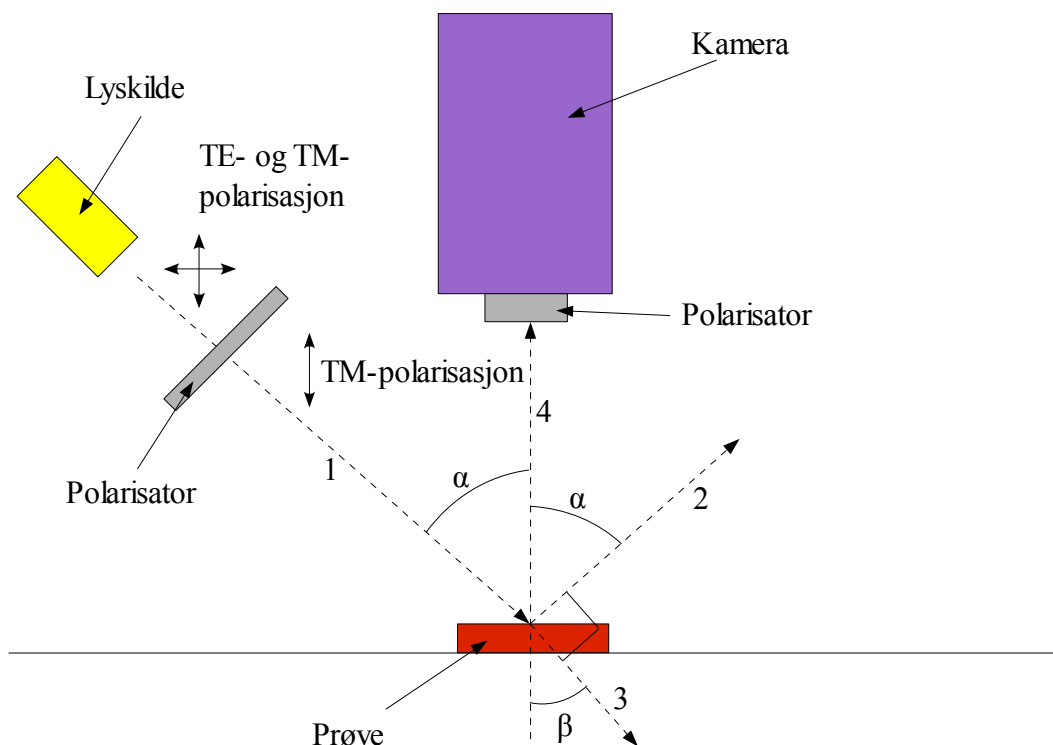
3.1 Eksperimentelt oppsett

Hyperspektrale bilder blir fotografert i en skanneprosess som er illustrert med figur 3.1.1. Motivet eller scenen som skal avbildes blir skannet linjevis i y-retning. Skanneretningen er vist ved den tykke sorte pilen. Hyperspektrale bilder inneholder en spektral dimensjon som de forskjellige bølglengdene ligger langs med. For å skille bølglengdene fra hverandre under skanningen benyttes et dispersivt element. Til høyre i figur 3.1.1 vises sensorarrayet som registrerer alle bølglengdene for den aktuelle linjen som blir skannet.



Figur 3.1.1: Oppsett for å ta hyperspektrale bilder^[4]

Testoppsettet for å ta hyperspektrale bilder av en vevsprøve er vist i figur 3.1.2.



Figur 3.1.2: Testoppsett for hyperspektrale bilder^[4]

Lys fra lyskilden treffer prøven langs med pilen merket 1. Når hvitt lys benyttes vil lyset være

upolarisert ut av lyskilden, og inneholder da både TE- og TM-polarisasjonskomponenter. Polarisatoren filtrer bort TE-polarisasjonen, og bare TM-polarisasjonen vil fortsette å propagere. TM-polarisasjonen treffer prøven med en vinkel α . Dette lyset blir depolarisert på overflaten til prøven fordi overflaten er ru. Depolarisasjonen gir opphav til en TE-polarisert komponent som i utgangspunktet kan bli registrert av kameraet. En lineærpolarisator blir derfor plassert foran kameraet for å fjerne en eventuell TE-polarisert komponent. Det meste av det depolariserte lyset fra prøven vil imidlertid følge retningen til pil 2, som også er hovedretningen til spekulær refleksjon.

Når α er lik Brewstervinkelen, blir TM-polarisert lys brutt ned i prøven langs med retningen til pil 3^[4]. Brytningsvinkelen β er gitt av Snells brytningslov. Lys som blir reflektert fra dypere lag i prøven kan bli registrert av kameraet. For at dette skal skje må det reflekterte lyset ut av prøven ha retningen til pil 4.

Testoppsettet er bygget opp av følgende komponenter:

1. Kamera: HySpex VNIR 1600, Norsk Elektro Optikk AS, Lørenskog, Norway
Kameraet har bølgelengdeoppløsning på 3.7 nm i området fra 400 – 1000 nm. Radiometrisk kalibrering blir gjort automatisk i kamerasoftwaren.
2. Lyskilde hvitt lys: DCR II, Fostec, Auburn, NY, USA
3. Lyskilde laser: 355nm triplet nd:YAG-laser
4. Polarisator ved kamera: NT52-547, Edmund Optics, York, UK
5. Polarisator ved lyskilde: International Polarizer Inc, Marlboro, MA, USA

3.2 Valg av programmeringsspråk

Det finnes mange forskjellige programmeringsspråk. Hva man skal velge avhenger av flere faktorer. Rask implementasjon, kjørehastighet for ferdig kode og portabilitet mellom systemer er noen kriterier for valg av programmeringsspråk.

For prosjektet i høst var kjørehastighet et viktig kriterium. Da ble valget C++ som gir stor grad av kontroll på hvordan algoritmene skal implementeres. Ulempen med C++ er at lite er ferdig implementert i utgangspunktet, og man må basere seg på å skrive en god del av basisfunksjonaliteten selv.

I dette prosjektet skal flere forskjellige algoritmer implementeres og sammenlignes. Rask implementasjon gjør at man får testet koden på et tidlig stadium for å se om den virker. Hvor hurtig en algoritme lar seg implementere vil avhenge av rammeverket til programmeringsspråket. Siden algoritmene for dette prosjektet benytter matriser vil derfor ferdigimplementerte matriseoperasjoner være en stor fordel. Matlab inneholder et ferdig rammeverk for bruk av matriser. Dette programmeringsspråket har også implementert funksjonalitet for å finne egenverdier og egenvektorer til matriser. Slike beregninger trengs både under PCA- og MNF-algoritmen. På grunn av disse tingene velges Matlab som programmeringsspråk.

Det er imidlertid flere ulemper med Matlab. Kjørehastigheten for koden kan bli treg. Matlab koster også penger i innkjøp. Det siste problemet kan omgås ved å velge Octave, som er et gratis alternativ til Matlab.

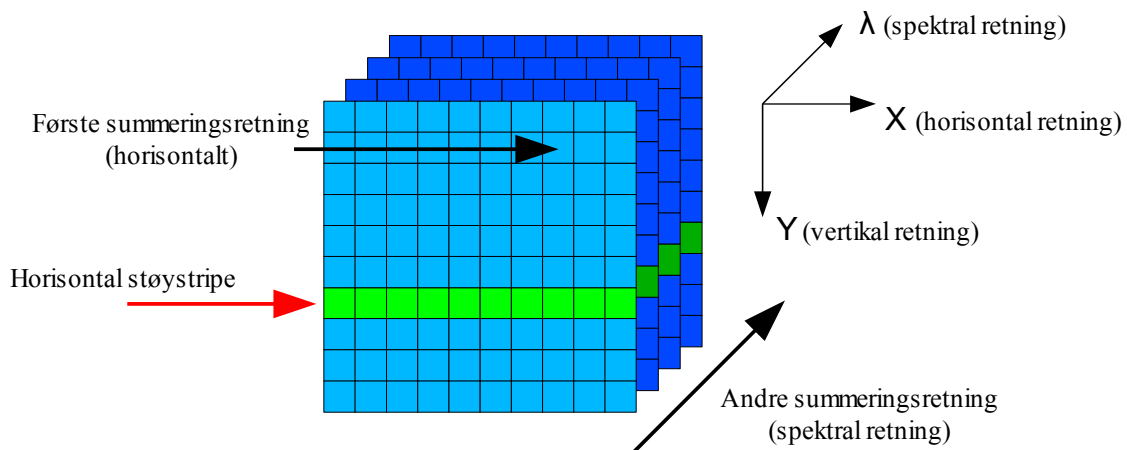
3.3 Fjerning av horisontal støystripe

Blant de hyperspektrale bildene inneholder noen en veldig kraftig horisontal strek som kan ødelegge for MNF algoritmen. Her vil det bli vist en algoritmen som detekterer og fjerner en eventuell skarp horisontal støystripe i de hyperspektrale bildene.

Den horisontale støystripen kjennetegnes ved en abrupt endring i vertikal retning fra resten av bildet på bare en piksel. Hvis man ser på kvadratet av den deriverte vil dette fortone som én topp og ikke to grunnet bredden på bare en piksel. Kvadratet av den deriverte benyttes for at både en spiss topp og et dypt søkk skal kunne detekteres på samme måte. Dette vil si at begge disse konturene vil fremkomme som en topp siden kvadrering gjør at fortegnet forsvinner.

Generering av den deriverte

Den deriverte mellom to diskrete verdier er gitt som $f(x) - f(x + 1)$ ^[1]. For vertikal retning i et todimensjonalt bilde vil dette være $f(x, y) - f(x, y + 1)$. Denne verdien kan i utgangspunktet være både negativ og positiv. Det er abrupt endring man ser etter, så det spiller liten rolle om den er positiv eller negativ. For å gjøre deteksjonen lettere er det en fordel om den bare er positiv. Da slipper man å se på tilfellet for et negativt søkk. Dette oppnår man ved å kvadrere differansen mellom to nabopiksler i vertikal retning. Det er endring mellom to linjer som er av interesse, og ikke mellom to enkeltstående piksler. En finner endringen mellom to linjer ved å summer sammen alle de kvadrerte differansene mellom to og to piksler. Da blir den kvadrerte differansen fra hele linjen tatt med i beregningen. Videre har det hyperspektrale bildet tre dimensjoner. Til nå har man kun sett på to av de, altså man har holt seg innenfor et bestemt spektralbånd. For å ta høyde for at denne horisontale støykomponenten ikke fremkommer like tydelig i alle spektralbånd, summeres derfor opp summene av kvadrerte differanser mellom spektralbåndene.



Figur 3.3.1: Hyperspektralt bilde med horisontal støystripe

Figur 3.3.1 illustrerer en horisontal støystripe i det hyperspektrale bildet. Denne er vist med en rød pil. Etter at differansen har blitt kvadrert opp mellom to piksler i vertikal retning, summeres de kvadrerte tallene sammen langs med første summeringsretning. Når endringen mellom hver horisontale linje er funnet i et spektralbånd, summeres dette opp med endringen for tilsvarende linjer til de andre spektralbåndene. Denne summeringen skjer langs med andre summeringsretning som vist i figuren. Nå sitter man på et datasett som viser den kvadrerte differansen mellom hver linje i det hyperspektrale bildet.

Deteksjon av topp for de deriverte linjene

Deteksjon av en eventuell topp skjer ved at en søker etter en verdi som er langt høyere enn gjennomsnittet for datasettet som ble generert i punktet over. Det benyttes en vanlig thresholdverdi for å avgjøre om det er en topp eller ikke. Hvis den deriverte for linjen man søker i er større enn thresholdverdien, blir linjen merket som en horisontal støylinje. Er den mindre enn thresholdverdien, går man bare videre til neste linje.

Interpolasjon for fjerning av støylinje

Etter at støylinjen har blitt detektert skal den erstattes med en interpolasjon av nabolinjene^[17]. Dette utføres pikselvis for hele linjen. Verdiene man erstatter pikslene i støylinjen med er gitt av følgende formel

$$f(x, y) = \frac{f(x, y-1) + f(x, y+1)}{2} \quad (3.3.1)$$

Når den horisontale støylinjen har blitt erstattet med interpolasjonen av nabolinjene, kan MNF-algoritmen kjøres som vanlig på det behandlede bildet.

3.4 MNF

MNF algoritmen har blitt implementert som et skript i Matlab. Koden ble delt inn i blokker, avhengig av hva som skulle beregnes. I tillegg til operasjonene beskrevet under kapittel 2.3 er det noen overganger mellom 2- og 3-dimensjonalitet. Dette kommer av at enkelte steg i MNF-algoritmen må utføres der det hyperspektrale bildet er lagret som en 2-dimensjonal matrise, mens de fleste romlige beregninger krever at de utføres når det hyperspektrale bildet er lagret som et 3-dimensjonalt datasett.

Fra harddisk blir det hyperspektrale bildet lest inn i Matlab som en 3-dimensjonal datastruktur. Dette utføres ved hjelp av kommandoen "multibandread" som finnes i Matlab. Støyen for det innleste hyperspektrale bildet blir deretter estimert ved en av flere måter beskrevet i kapittel 3.4.1. Felles for støyestimaterne er at de gjør sammenligninger blant piksler som ligger nært i rommet, og dette må derfor utføres mens man har den romlige datastrukturen.

Etter at støyen er estimert skal kovariansmatrisen for både støyen og selve bildet beregnes. Dette kan implementeres uavhengig av romlig informasjon, og kan derfor gjennomføres både når dataene er lagret som 2- og 3-dimensjonal struktur. Matlab har en innebygget funksjon "cov" som kan beregne kovariansen til en matrise. Denne funksjonen forutsetter en 2-dimensjonal struktur, der hver rad er en observasjon og hver kolonne en variabel^[29]. En velger her å konvertere datasettet til 2-dimensjonal struktur slik at funksjonen "cov" kan benyttes. Det er kovariansen mellom de spektrale båndene som skal beregnes. For N bånd blir kovariansmatrisen en matrise med størrelsen $N \times N$. Konverteringen fra 3- til 2-dimensjonal datastruktur utføres med følgende skript

```
% matrise3D(rad, kolonne, spektralband)
% matrise2D(spektralband, radOgKolonne)
for z = 1:antallSpektralband
    for y = 1:antallRader
        matrise2D(z, ((y -1) * antallKolonner) + 1:y * antallKolonner)...
            = matrise3D(y, :, z);
    end
end
end
```

Merk at størrelsen på feltet `radOgKolonne` blir lik størrelsen på feltene `rad` og `kolonne` multiplisert med hverandre.

Videre følger beregningen av det generaliserte egenvektor-problemet for kovariansmatrisen til støyestimatet med hensyn på kovariansmatrisen til det hyperspektrale bildet^[13]. Dette utføres med kommandoen "eig" i Matlab. Ved bruk av følgende skript får man ut egenvektormatrisen til det generaliserte egenvektor-problemet

```
% egenvektorMatrise = matrisen som holder egenvektorene, egenvektorene er ordnet
% etter korresponderende egenverdi i egenverdiMatrise
% egenverdiMatrise = matrise med egenverdiene langs med diagonalen, sortert
% etter stigende egenverdier
% kovStoy = kovariansmatrisen til støyestimatet
% kovHyper = kovariansmatrisen til det hyperspektrale bildet
[egenvektorMatrise, egenverdiMatrise] = eig(kovStoy, kovHyper);
```

Prinsipalkomponentene for MNF får man nå ved å matrisemultiplisere egenvektormatrisen med datasettet av det hyperspektrale bildet gitt som en 2-dimensjonal datastruktur. Matlab skriptet er som følger

```
% prinsipalkomponenter = matrisen som holder prinsipalkomponentene fra MNF
prinsipalkomponenter = egenvektorMatrise * matrise2D;
```

Videre beregninger i MNF-rommet krever en konvertering fra 2- til 3-dimensjonal ordning av

prinsipalkkomponentene. Da vil man få tilbake den romlige ordningen av datasettet. Merk at det er like mange spektralbånd som prinsipalkkomponenter.

```
% matrise3dMNF(rad, kolonne, prinsipalkkomponent)
% prinsipalkkomponenter(prinsipalkkomponent, radOgKolonne)
% antallRader = antallet rader i det opprinnelige hyperspektrale bildet
% antallKolonne = antall kolonner i det opprinnelige hyperspektrale bildet
for z = 1:antallSpektralband
    for y = 1:antallRader
        matrise3dMNF(y, :, z) = prinsipalkkomponenter(z, (antallKolonne * ...
            (y - 1)) + 1:antallKolonne * y);
    end
end
```

Nå som prinsipalkkomponentene er gitt ved en 3-dimensjonal struktur kan andre algoritmer som krever romlig struktur kjøres på dem. Her vil typisk algoritmer for støyfjerning bli utført, før en tilbaketransformering til det opprinnelige rommet. Det er også her en får ut bildene av hvordan de forskjellige prinsipalkkomponentene ser ut.

Tilbaketransformeringen krever at datasettet er gitt ved 2-dimensjonal datastruktur. Derfor må enda en konvertering til før man igjen kommer til det opprinnelige rommet. Denne konverteringen vil være nesten lik den første 3- til 2-dimensjonale transformen. Forskjellen ligger i datasettet som skal konverteres. Her er det matrisen som holder prinsipalkkomponentene og ikke det innleste hyperspektrale bildet fra harddisken.

Når datasettet igjen er gitt ved en 2-dimensjonal struktur kan tilbaketransformering skje ved en matrisemultiplikasjon mellom den inverse egenvektormatrisen og prinsipalkkomponentmatrisen. Den inverse matrisen blir beregnet med Matlab funksjonen "inv". Skriptet viser tilbaketransformeringen

```
% tilbaketransformert = det tilbaketransformerte datasettet i det opprinnelige
% rommet
tilbaketransformert = inv(eigenvektorMatrise) * prinsipalkkomponenter
```

For å kunne sammenligne resultatet etter MNF med det originale hyperspektrale bildet må enda en transformering fra 2- til 3-dimensjonalt sett utføres. Dette gjøres da på matrisen som holder de tilbaketransformerte dataene.

3.4.1 Estimering av kovariansmatrisen til støyen

Referering til nabopikslene for en gitt piksel blir gjort som følgende^[1]. Hvis pikselen man ser på i det spatiale plan er $A(x, y)$, kan man bruke figur 3.4.1.1 for å se adresseringene til nabopikslene rundt piksel $A(x, y)$. Figur 3.4.1.1 viser en 3x3 maske. Det blir også brukt masker på 5x5 og 7x7 størrelse. De følger samme referansemønster. Valget om å ta med flere maskestørrelser er gjort for at en skal kunne sammenligne påvirkningen maskestørrelsen har på filteret. Det er også viktig at man kan sammenligne virkningen mellom filtertypene for en gitt maskestørrelse.

$A(X-1, Y-1)$	$A(X, Y-1)$	$A(X+1, Y-1)$
$A(X-1, Y)$	$A(X, Y)$	$A(X+1, Y)$
$A(X-1, Y+1)$	$A(X, Y+1)$	$A(X+1, Y+1)$

Figur 3.4.1.1: Referanser til nabopiksler for 3x3 maske

Minimum/maximum autocorrelations factors (MAF)

Fra kapittel 2.3.1 har vi at MAF kan beskrives matematisk som funksjonsuttrykket $A(x, y) - A(x + \Delta_x, y + \Delta_y)$, der Δ_x og $\Delta_y \in \{-1, 0, 1\}$. For å lette sammenligning og det implementasjonsmessige blir MAF her fremstilt som en maske. Støyestimatet for en gitt piksel beregnes følgende. Multipliser hver piksel med tallet som står inni korresponderende celle i masken. Deretter summerer man opp alle leddene gitt av masken. For en maske på 3x3 blir det 9 ledd, og en på 5x5 gir 25 ledd. Nummerene inni masken henspeiler til vekten av hver piksel.

Støyestimat basert på horisontal nabopiksel

Masken for støyestimatet er vist i figur 3.4.1.2. Figuren fremkommer ved å sette $\Delta_x = 1$, og $\Delta_y = 0$. Denne masken kan ikke anvendes på kolonnen som står helt ytterst til høyre i bildet. Dette kommer av at man ikke kan hente ut verdien på en piksel som ligger utenfor bildet. På grunn av dette er denne kolonnen utelatt fra støyestimatet. Å utelate denne ene kolonnen skal ha lite å si for det totale støyestimatet^[4].

0	0	0
0	1	-1
0	0	0

Figur 3.4.1.2: 3x3 maske basert på horisontal nabopiksel

Støyestimat basert på vertikal nabopiksel

Figur 3.4.1.3 viser masken som blir brukt for å danne støyestimatet. Masken kommer som en følge av å sette $\Delta_x = 0$, og $\Delta_y = 1$. Her kan ikke øverste rad bli tatt med. Begrunnelsen blir tilsvarende som for horisontal nabopiksel.

0	-1	0
0	1	0
0	0	0

Figur 3.4.1.3: 3x3 maske basert på vertikal nabopiksel

Gjennomsnitt av flere korrelasjonsfaktorer

I figur 3.4.1.4 vises masken som tar utgangspunkt i gjennomsnittet basert på vertikal og horisontal nabo. Det er denne implementasjonen som blir brukt av programmet ENVI^[16]. En merknad er at hvis man summerer sammen alle cellene i masken, skal summen alltid bli 0. Hvis summen av cellene hadde vært ulik 0, ville en helt flat eller homogen overflate gi utslag i støyestimatet.

Faktoren $\frac{1}{2}$ kommer av at en må ta verdien i punktet (x, y) , altså $A(x, y)$ å subtrahere fra snittet mellom differansene i vertikalt nabopunkt og horisontalt nabopunkt.

$$\begin{array}{|c|c|c|} \hline 0 & -1 & 0 \\ \hline 0 & 2 & -1 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \cdot \frac{1}{2}$$

Figur 3.4.1.4: 3x3 maske for snittet av vertikal og horisontal nabo.

Differanse med lokalt gjennomsnitt (Mean)

3x3 maskestørrelse

Her skal en også bevare tanken om å se på differansen mellom et punkt og et filter for å estimere støyen. En illustrativ formel for bruk av lokalt gjennomsnitt er vist i figur 3.4.1.5.

$$A(x, y) - \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \cdot \frac{1}{9} = A(x, y) + \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & -1 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array} \cdot \frac{1}{9} = \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array} \cdot \frac{1}{9}$$

Figur 3.4.1.5: 3x3 maske for lokalt gjennomsnitt

Til venstre for det første likhetstegnet vises ideen med støyestimatet. Det vil si å trekke fra det lokale gjennomsnittet til punkt (x, y) fra verdien som eksisterer i punktet vist ved $A(x, y)$. Til høyre for det siste likhetstegnet vises den faktiske masken som blir implementert.

5x5 og 7x7 maskestørrelse

Tilsvarende filter med maskestørrelser på 5x5 og 7x7 blir også brukt. Senterverdien i masken for 5x5 blir 24, og man må gange hele masken med en faktor på $1/25$ istedenfor $1/9$. For 7x7 masken blir senterverdien 48, og masken må ganges med en faktor $1/49$.

Differanse med lokalt snitt gitt av Gaussisk fordeling

Disse filtrene ligner på lokale gjennomsnittsfiltre. Forskjellen ligger i ulik vektning av nabopikslar. Fordelingen er gitt av Gausskurven, der de nærmeste naboene vektlegges mer enn pikslene som ligger lengre unna. For lokale gjennomsnittsfiltre blir alle pikslene som ligger innenfor masken vektet likt.

3x3 maskestørrelse

Figur 3.4.1.6 viser masken for Gaussfilteret når størrelsen er på 3x3. Verdiene for alle Gaussfiltrene har blitt beregnet med standardavvik på 1 i både x-retning og y-retning.

$$A(x, y) - \begin{array}{|c|c|c|} \hline 0,68 & 1,1 & 0,68 \\ \hline 1,1 & 1,8 & 1,1 \\ \hline 0,68 & 1,1 & 0,68 \\ \hline \end{array} \cdot \frac{1}{9} = \begin{array}{|c|c|c|} \hline -0,68 & -1,1 & -0,68 \\ \hline -1,1 & 7,2 & -1,1 \\ \hline -0,68 & -1,1 & -0,68 \\ \hline \end{array} \cdot \frac{1}{9}$$

Figur 3.4.1.6: 3x3 maske for Gaussisk fordeling

5x5 maskestørrelse

Figur 3.4.1.7 viser masken til Gaussfilter med størrelse på 5x5.

$$A(x, y) - \begin{array}{|c|c|c|c|c|} \hline 0,07 & 0,33 & 0,55 & 0,33 & 0,07 \\ \hline 0,33 & 1,5 & 2,5 & 1,5 & 0,33 \\ \hline 0,55 & 2,5 & 4,1 & 2,5 & 0,55 \\ \hline 0,33 & 1,5 & 2,5 & 1,5 & 0,33 \\ \hline 0,07 & 0,33 & 0,55 & 0,33 & 0,07 \\ \hline \end{array} \cdot \frac{1}{25} = \begin{array}{|c|c|c|c|c|} \hline -0,07 & -0,33 & -0,55 & -0,33 & -0,07 \\ \hline -0,33 & -1,5 & -2,5 & -1,5 & -0,33 \\ \hline -0,55 & -2,5 & 21 & -2,5 & -0,55 \\ \hline -0,33 & -1,5 & -2,5 & -1,5 & -0,33 \\ \hline -0,07 & -0,33 & -0,55 & -0,33 & -0,07 \\ \hline \end{array} \cdot \frac{1}{25}$$

Figur 3.4.1.7: 5x5 maske for Gaussisk fordeling

7x7 maskestørrelse

Figur 3.4.1.8 viser sluttmasken for estimering av støyen med utgangspunkt i Gaussfilter av størrelse 7x7.

$$\begin{array}{|c|c|c|c|c|c|c|} \hline 0 & -0,01 & -0,05 & -0,09 & -0,05 & -0,01 & 0 \\ \hline -0,01 & -0,14 & -0,64 & -1,1 & -0,64 & -0,14 & -0,01 \\ \hline -0,05 & -0,64 & -2,9 & -4,7 & -2,9 & -0,64 & -0,05 \\ \hline -0,09 & -1,1 & -4,7 & 41 & -4,7 & -1,1 & -0,09 \\ \hline -0,05 & -0,64 & -2,9 & -4,7 & -2,9 & -0,64 & -0,05 \\ \hline -0,01 & -0,14 & -0,64 & -1,1 & -0,64 & -0,14 & -0,01 \\ \hline 0 & -0,01 & -0,05 & -0,09 & -0,05 & -0,01 & 0 \\ \hline \end{array} \cdot \frac{1}{49}$$

Figur 3.4.1.8: 7x7 maske for Gaussisk fordeling

Differanse med median

I utgangspunktet ligner differanse med median på differanse med lokalt gjennomsnitt. Forskjellen er at støyestimatet blir gjort med differansen til median av masken som benyttes, istedenfor gjennomsnittsverdien til masken. For de respektive maskestørrelsene 3x3, 5x5 og 7x7 finner man medianverdien innenfor masken. Deretter blir $A(x, y)$ verdien trukket fra medianverdien, og dette blir støyestimatet. Figur 3.4.1.9 viser oppsettet for maskestørrelsen 3x3.

$$A(x, y) - \text{median} \left\{ \begin{array}{|c|c|c|} \hline A(X-1, Y-1) & A(X, Y-1) & A(X+1, Y-1) \\ \hline A(X-1, Y) & A(X, Y) & A(X+1, Y) \\ \hline A(X-1, Y+1) & A(X, Y+1) & A(X+1, Y+1) \\ \hline \end{array} \right\}$$

Figur 3.4.1.9: 3x3 median støyestimator

Andrederiverte

Ligningen for den andrederiverte er gitt i kapittel 2.3.1, og blir på matrisform vist i figur 3.4.1.10 og 3.4.1.11. Figur 3.4.1.10 viser den andrederiverte i x-retning, og figur 3.4.1.11 i y-retning.

0	0	0
1	-2	1
0	0	0

Figur 3.4.1.10: 3x3 maske for andrederiverte i x-retning

0	1	0
0	-2	0
0	1	0

Figur 3.4.1.11: 3x3 maske for andrederiverte i y-retning

Merk at her blir ikke støyestimatet beregnet som differansen mellom en celleverdi $A(x, y)$ og en annen celle eller et filter. Støyestimatet er summen av celleverdiene multiplisert med de korresponderende faktorene i masken som benyttes.

For å få med begge retningen i samme støyestimat har det blitt valgt å summer sammen absoluttverdien av de andrederivert i horisontal og vertikal retning. Dette vil bli på formen

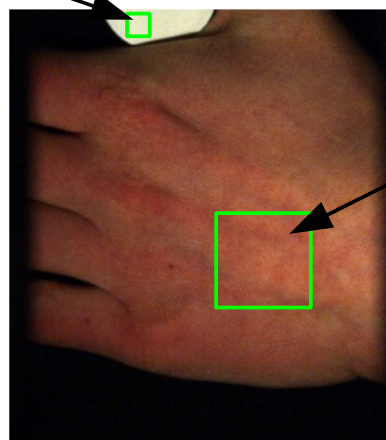
$$\text{abs} \left\{ \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 1 & -2 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \right\} + \text{abs} \left\{ \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 0 & -2 & 0 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \right\}$$

Figur 3.4.1.12: Summen av absoluttverdiene til de andrederiverte

3.4.2 Støyestimering for homogent felt

Støyestimering basert på homogent felt ligner mye på støyestimeringen som benytter hele bildet. Ved støyestimering basert på hele utsnittet, blir både støyestimatet og resten av MNF-transformen beregnet i en operasjon. Tenker da på beregning av støyestimat beskrevet i kapitlet 3.4.1, og utregning av egenverdier samt egenvektorer for det generaliserte egenvektorproblemet vist i kapittel 2.3. Når støyestimeringen blir utført på et homogent felt som ikke ligger innenfor utsnittet man studerer i de hyperspektrale bildene, blir MNF-transformen delt i to deler. En del estimerer støyen fra den homogene biten, og den andre gjennomfører beregningen av kovariansen til utsnittet som studeres samt egenvektorer og egenverdier.

Homogent område innenfor refleksjonsstandarder som benyttes under støyestimeringen. Størrelse 50x50 piksler.



Utsnitt av hyperspektralt bilde som blir studert. Størrelse 400x400 piksler.

Figur 3.4.2.1: Utsnitt av det hyperspektrale bildet som blir benyttet til støyestimering, og utsnittet som blir studert.

Årsaken til at MNF-transformen blir gjort todelt når støyestimatet er basert på en homogen del kommer av minnebegrensninger for datamaskinen som ble benyttet. Denne oppdelingen gjør at man kun trenger å lese inn de delene av det hyperspektrale bildet som til en hver tid benyttes. I prinsippet kan man lese inn det totale hyperspektrale bildet, for deretter å dele det opp i de ønskede områdene som skal studeres. Dette krever imidlertid mye minne, og da resultatene skal bli identiske uavhengig av denne oppdelingen skal det derfor være helt greit å utføre operasjonen i to steg. En ulempe med å gjøre det på denne todelte måten er at det krever lengre tid å lese inn dataene to ganger fra de hyperspektrale bildene, istedenfor bare en.

3.4.3 Antall band under tilbaketransformering

Støyreduksjon utført på et hyperspektralt bilde skjer ved å utelate enkelte prinsipalkomponenter under reverstransformasjonen.

Først transformeres det hyperspektrale bildet til MNF-rommet. Ligning 2.3.6 i teorien for MNF gir transformasjonen av det hyperspektrale bildet til MNF-rommet. For å komme tilbake til det opprinnelige bildet, må man inverstransformere fra MNF-rommet. Inverstransformasjonen er beskrevet i ligning 2.3.11. I tillegg til dette skal en del av prinsipalkomponentene utelates under inverstransformasjonen.

For å bestemme hvilke prinsipalkomponenter i MNF-rommet som skal være med under tilbaketransformasjonen benyttes en reduksjonsmatris $R^{[1]}$. Denne matrisen består av 1-ere langs med diagonalen som korresponderer med prinsipalkomponentene som skal være med under tilbaketransformasjonen. Resten av matrisen er 0-ere. Det gjelder også verdiene langs med diagonalen i matrisen for de prinsipalkomponentene man ikke skal ha med tilbake under tilbaketransformasjonen.

1	0	0	0	0
0	1	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Figur 3.4.3.1: Reduksjonsmatrise som beholder de 2 første prinsipalkomponentene og ignorerer de 3 siste i et sett på totalt 5 komponenter.

Figur 3.4.3.1 viser et eksempel på en reduksjonsmatrise som trekker ut de to første prinsipalkomponentene i ett sett på fem prinsipalkomponenter. Merk at siden man kan ta med maksimalt fem prinsipalkomponenter vil antallet spektralband til det korresponderende hyperspektrale bildet også være fem.

Reduksjonsmatrisen skal matrisemultipliseres med ligning 2.3.6. Dette gir ligningen

$$R(x, y) B(x, y) = R(x, y) a^T A(x, y) \quad (3.4.3.1)$$

Hvis man kombinerer ligning 2.3.6 og ligning 2.3.11 uten å ta med noen reduksjonsmatrise får man tilbake det nøyaktig samme hyperspektrale bildet man puttet inn uten noen form for støyreduksjon. De kombinerte ligningene vil ta formen vist under

$$A(x, y) = (a^T)^{-1} B(x, y) = (a^T)^{-1} a^T A(x, y) \quad (3.4.3.2)$$

Merk at $(a^T)^{-1} a^T$ gir identiteten, så dette kan forkortes bort. Tar man med reduksjonsmatrisen blir ligningen for det reduserte datasettet

$$\hat{A}(x, y) = (a^T)^{-1} R(x, y) B(x, y) = (a^T)^{-1} R(x, y) a^T A(x, y) \quad (3.4.3.3)$$

Her er \hat{A} det støyreduerte hyperspektrale bildet.

3.4.4 Støyestimering i fourierrommet

Støyestimeringen blir utført på prinsipielt to forskjellige måter. En med utgangspunkt i differansen mellom andelen det er av de forskjellige frekvenskomponentene i fourierplanet uttrykt ved absoluttverdien til de komplekse tallene. Den andre bruker forskjellen mellom fasene til frekvenskomponentene. Ideen om å bruke differansen med nærliggende verdier er den samme som Green^[11] benyttet, men her ser man på nærliggende frekvenser istedenfor nabopiksler. Dette vil gjelde begge tilfellene om en bruker absoluttverdien eller fasen til frekvenskomponentene. Støy fremstår som hurtige endringer i det spatiale rommet. Hurtige endringer kan sees på som skarpe firkantpulser. Dette gir sincfunksjoner i fourierplanet.

Frekvensplanet er ordnet som et polart koordinatsystem med frekvensaksen liggende radielt ut i fra sentrum. Grensesnittet til dataene er imidlertid ordnet som et kartesisk koordinatsystem. På grunn av dette er det valgt å bruke et "mean" filter med maskestørrelser på 3x3, 5x5 og 7x7 for å gjøre støyestimatene basert på absoluttverdi og fase. Dette vil gjøre det enkelt å få med differansen til nærliggende frekvenser. "Mean" filter er vist i figur 3.4.1.5.

Ved utregningene av kovariansmatrisene til støyestimatet og det hyperspektrale bildet blir den transponerte til matrisene benyttet når man er i det vanlige spatiale rommet. For beregninger av kovariansmatriser til matriser som består av komplekse tall må den transponerte byttes ut med den konjugert transponerte eller Hermitian transponerte^[30]. Siden man får komplekse verdier etter en fouriertransformasjon, vil dette gjelde for matrisene man arbeider med i fourierrommet.

Når man har funnet kovariansmatrisen til støyestimatet og løst det generaliserte egenvektorproblemet følger MNF-transformasjonen. Den blir litt annerledes i fourierrommet enn i det vanlige spatiale rommet. Dette kommer også av at matrisene man jobber med består av komplekse tall og ikke rent reelle. Fra ligning 2.3.6 under teorien for MNF får man overgangen til MNF-rommet. Her må også den transponerte byttes ut med den konjugert transponerte. Dette kommer av ligningen $a^T \Sigma a = 1$ fra Green^[11]. For matriser med komplekse tall må dette endres til $a^H \Sigma a = 1$ for at det fremdeles skal gjelde.

Figur 2.3.3.1 fra teorien viser at det neste steget i algoritmen er utvelgelsen av prinsipalkomponenter. Dette vil skje på samme måte som for det spatiale rommet. Reduksjonsmatrisen man matrisemultipliserer med blir den samme som er vist i figur 3.4.3.1.

3.5 PCA

Matlab er valgt som programmeringsspråk. I Matlab benyttes funksjonen "eig" for å finne egenvektorene med korresponderende egenverdier. Denne funksjonen ordner imidlertid egenverdiene i stigende rekkefølge. Fra kapittel 2.2. har man at de laveste prinsipalkomponentene inneholder mest varians, gitt at egenverdiene er sortert i synkende rekkefølge med den største egenverdien først. Dette betyr at rekkefølgen på prinsipalkomponentene er snudd i forhold til det de var for MNF-algoritmen. For å beholde de laveste prinsipalkomponentene og utelate de høyeste under tilbaketransformasjonen må man derfor endre på reduksjonsmatrisen i forhold til MNF-algoritmen. Dette betyr at reduksjonsmatrisens 1-tall som indikerer hvilke prinsipalkomponenter som skal bli bevart under tilbaketransformasjonen vil være plassert langs med diagonalen nederst i høyre hjørne. For MNF-algoritmen ble de plassert øverst i venstre hjørne. En alternativ løsning til å endre på reduksjonsmatrisen vil være å sortere egenverdiene slik at de kommer i synkende rekkefølge for å få det man opprinnelig ville ha. Dette blir ikke gjort da det første alternativet er enklere å implementere.

Reduksjonsmatrisen blir som vist i figur 3.5.1 for PCA. En kan også sammenligne denne med reduksjonsmatrisen for MNF-algoritmen som er vist ved figur 3.4.3.1.

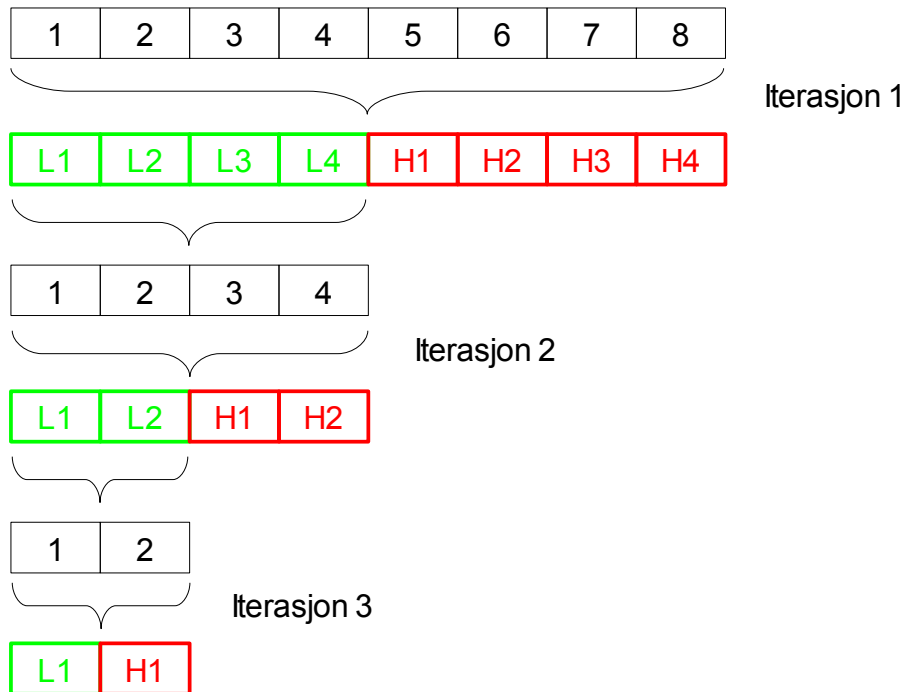
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	1	0
0	0	0	0	1

Figur 3.5.1: Reduksjonsmatrise som beholder de 2 siste prinsipalkomponentene og ignorerer de 3 første i et sett på totalt 5 komponenter.

Merk at for MNF trengte man ikke å snu om denne rekkefølgen. Det kommer av valget som ble tatt da ligningene ble satt opp for å gi ut minst støy for laveste prinsipalkomponent. Minst støy vil også si at korresponderende egenverdi er den laveste, og at signal-støyforholdet er det høyeste blant alle prinsipalkomponentene. Her passer det bra et Matlab sorterer egenverdiene i stigende rekkefølge med at første prinsipalkomponent skal ha størst signal-støyforholdet.

3.6 Wavelets

Det finnes mange forskjellige wavelet-transformer, men her er det Haar-transformen som vil bli brukt. Haar-transformen velges fordi den er enkel å implementere. De hyperspektrale bildene består av tre dimensjoner. To av dimensjonene er romlige og spenner ut et vanlig 2D-bilde. Den tredje dimensjonen holder den spektrale delen til de hyperspektrale bildene. Det er langs med denne dimensjonen wavelet-transformen skal benyttes. Dette betyr at man får en én-dimensjonal wavelet-transform i spektral retning.



Figur 3.6.1: Tre iterasjoner av Haar-transformen

Figur 3.6.1 illustrerer tre iterasjoner av Haar-transformen. Hvis man ser først på iterasjon 1 ligger spektralaksen langs med cellene merket med tallene fra 1 til 8. Dette betyr at figur 3.6.1 viser et spektralt utsnitt for én enkelt piksel. Haar-transformen leser inn to og to etterfølgende celleverdier av gangen og benytter dette til å beregne en L- og en H-celle. Det er cellene med nummer 1 og 2 som brukes for å bestemme L1 og H1, cellene 3 og 4 gir L2 og H2 osv. Lavpassdelen man finner i L1 blir beregnet på følgende måte av celle 1 og 2:

$$L1 = \frac{\text{celle1} + \text{celle2}}{2} \quad (3.6.1)$$

Høypassdelen som kommer i H1 er gitt av ligningen:

$$H1 = \frac{\text{celle1} - \text{celle2}}{2} \quad (3.6.2)$$

L2 og H2 blir beregnet på tilsvarende måte, men ut i fra det neste parret med verdier som er celle 3 og 4. Med utgangspunkt i lavpassdelen og høypassdelen kan man rekonstruere det opprinnelige signalet. Ligningen for å finne verdien til celle 1 fra L1 og H1 er gitt av ligningen:

$$\text{celle1} = L1 + H1 \quad (3.6.3)$$

For å få tilbake verdien til celle 2 benyttes ligningen:

$$celle1 = L1 - H1 \quad (3.6.4)$$

Figur 3.6.1 viser tre iterasjoner av Haar-transformen. Her er utgangspunktet en rekke med 8 celler. Siden det er de lavpassfiltrerte verdiene man fortsetter å iterere videre på, vil det si at det er 4 celler som blir benyttet i andre iterasjon. I tredje iterasjon fortsetter en med halvparten av det som var i andre iterasjon, altså to celler. Etter tredje iterasjon kan man ikke fortsette transformasjonen med ytterligere iterasjoner. Dette kommer av at det er et odde antall celler igjen. Antallet iterasjoner man kan gjennomføre vil være lik antallet 2-potenser i primtallfaktoreringen av den opprinnelig lengden.

For PCA og MNF er det benyttet de 81 første spektralbåndene av totalt 160 til de hyperspektrale bildene. Siden 81 ikke har noen 2-potenser blir det istedenfor valgt ut de 80 første spektralbåndene for Haar-transformen. Dette gir maksimalt 4 iterasjoner. Ved heller å velge 64 eller 128 spektralbånd vil man henholdsvis få muligheten for 6 og 7 iterasjoner. Valget faller likevel på 80 da dette ligger nærmere 81, som er antallet benyttet for PCA og MNF.

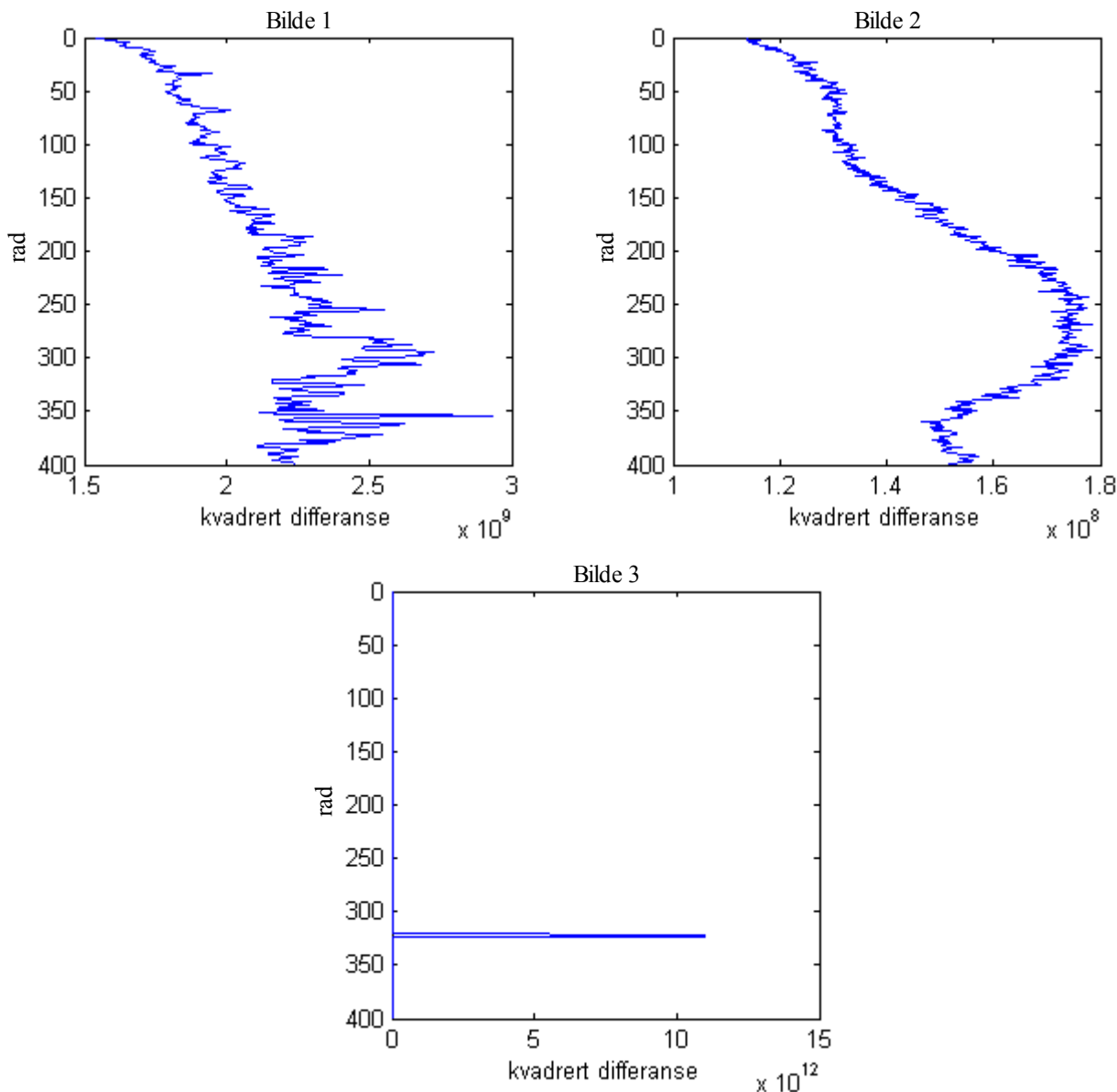
Etter forover wavelet-transform skal alle verdier under en gitt thresholdverdi settes til 0. Thresholdverdien blir satt slik at man beholder en gitt prosent av dataene. For å få til dette opprettes det først en kopi av de wavelet-transformerte dataene som deretter blir sortert. Datasettet sorteres i synkende rekkefølge. Videre blir thresholdverdien plukket ut blant de sorterte dataene. Hvis en skal beholde 25% av dataene velger man ut verdien som ligger 25/100-deler langs med det sorterte settet. Dette vil da være thresholdverdien man skal benytte på de usorterte dataene som kommer ut av forover wavelet-transform.

Merk at wavelet-transformen gir ut både positive og negative tall. Dette betyr at thresholdverdien må baseres på absoluttverdien av dataene, da det er verdiene med lav intensitet man vil fjerne. Ved sammenligningen om verdiene er høyere eller lavere enn thresholdverdien må også absoluttverdien benyttes.

Til slutt tilbaketransformeres alt med invers Haar-transform.

4 Resultater

4.1 Fjerning av horisontal støystripe

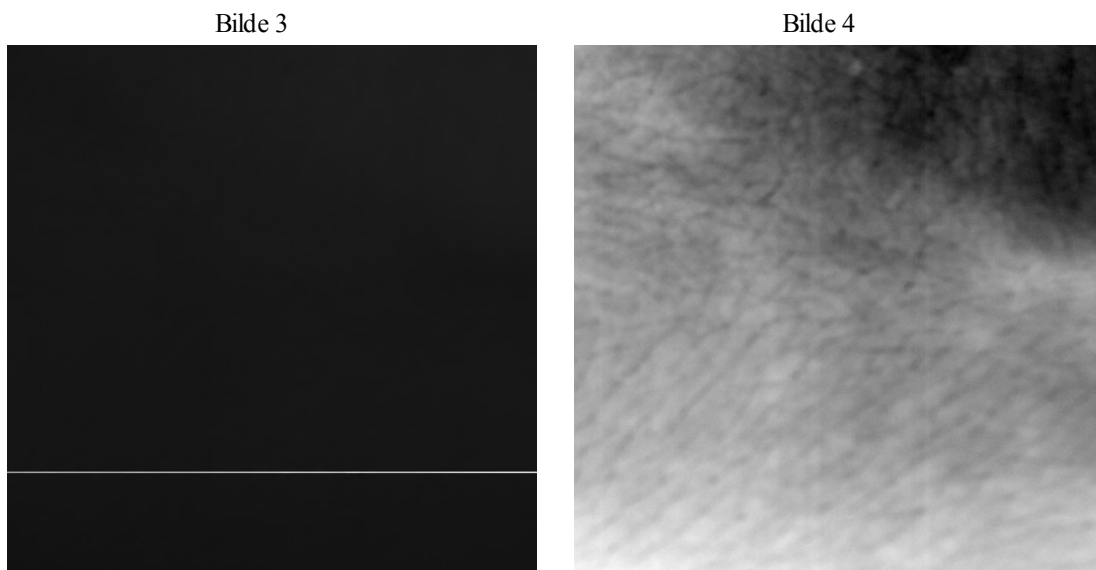


Figur 4.1.1: Akkumulert kvadrert differanse mellom horisontale linjer i hyperspektrale bilder.

I figur 4.1.1 vises den kvadrerte differansen mellom linjene i de hyperspektrale bildene. Siden det er horisontale streker man ser etter har grafene blitt dreid 90 grader for lettere å kunne sammenligne med de faktiske bildene. Dette betyr at kvadrert differanse ligger langs med x-aksen, mens radnummer ligger på y-aksen. Det er bilde 3 som har en horisontal støystripe, grafene for bilde 1 og 2 er tatt med for sammenligningsgrunnlag.

Figur 4.1.2 sammenligner tilfellene av første prinsipalkomponent for MNF-algoritmen med og uten erstatning av den horisontale støylinjen. Bilde 3 har en horisontal støystripe. Bilde 4 viser det nøyaktige samme utsnitt som bilde 3, men støystripen har blitt erstattet av snittet mellom pikslene som ligger rett over og under stripen. Denne utbytingen av linjen med interpolasjon mellom

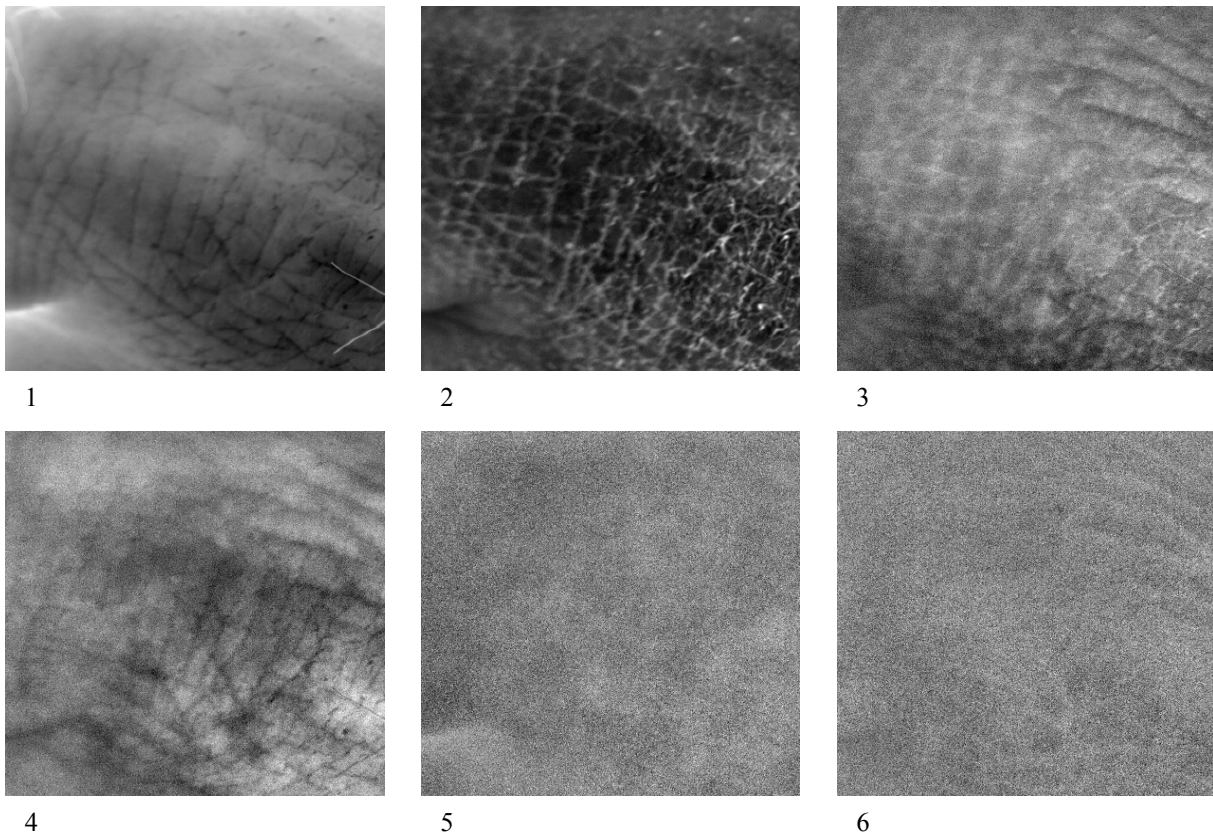
nabolinjene har blitt utført før MNF-algoritmen. Til venstre vises første prinsipalkomponent uten interpolasjon mellom nabolinjene og til høyre vises samme prinsipalkomponent når den er interpolert vekk. Støyestimeringen under MNF-algoritmen ble utført ved hjelp av differansen med horisontal nabopiksel. Dette skjedde både når den horisontale støylinjen var til stede i det hyperspektrale bildet, og når den var blitt fjernet med interpolasjon mellom nabolinjene.



Figur 4.1.2: MNF-komponent 1 for bilde 3 og 4 når støyestimateret er utført ved å benytte differansen til nærmeste horisontale nabopiksel.

4.2 Sammenligning av MNF og PCA

I figur 4.2.1 ser man de 6 første prinsipalkomponentene som kommer ut av MNF-algoritmen til bilde 1. Disse utsnittene av det opprinnelige bildet er på 400x400 piksler. Støyestimatoren som ble benyttet under MNF er illustrert i figur 3.4.1.4. Dette er forøvrig den samme støyestimatoren som programmet ENVI bruker^[16]. I likhet med PCA gir MNF ut like mange prinsipalkomponenter som det er spektralbånd i det hyperspektrale bildet man gir som input til algoritmen. Spektralbåndene som ble benyttet fra det hyperspektrale bildet er de 81 første av totalt 160 bånd. Etter spektralbånd 81 (702 nm) begynner spekteret å svinge. Dette kommer mest sannsynlig av at bruksområdet til polarisatorene ikke strekker seg over 700 nm^[4].



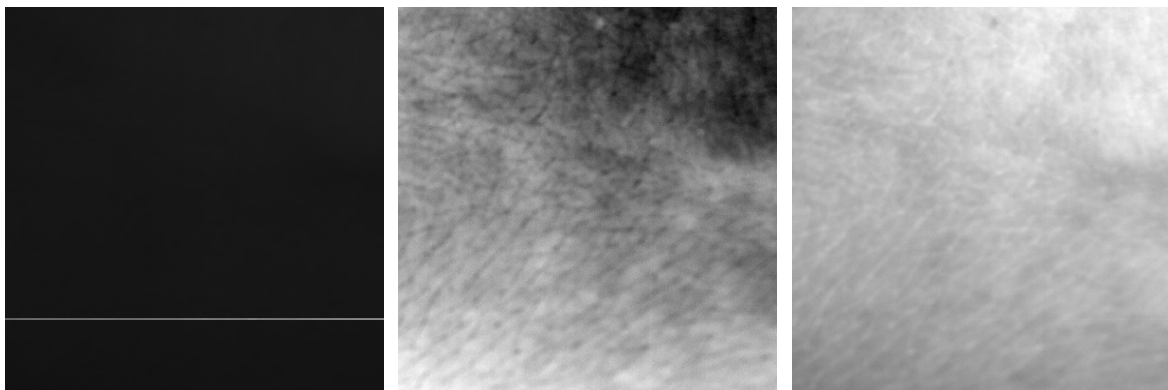
Figur 4.2.1: Bilder av de 6 første prinsipalkomponentene til bilde 1 utført med MNF-algoritmen

4.2.1 Estimering av kovariansmatrisen til støyen

Tabell 4.2.1.1: Akkumulerte og normaliserte SNR-verdier for de 5 første prinsipalkomponentene til bilde 1 og 2 basert på det totale utsnittet.

Bilde 1						Bilde 2					
Filter	MNF 1	MNF 2	MNF 3	MNF 4	MNF 5	Filter	MNF 1	MNF 2	MNF 3	MNF 4	MNF 5
Horisontalt	0,92	0,97	0,98	0,99	0,99	Horisontalt	0,96	0,99	0,99	0,997	0,997
Vertikalt	0,91	0,97	0,98	0,99	0,99	Vertikalt	0,96	0,99	0,99	0,997	0,997
Horisontalt + Vertikalt	0,92	0,97	0,98	0,99	0,99	Horisontalt + Vertikalt	0,96	0,99	0,99	0,997	0,998
Mean 3x3	0,83	0,89	0,90	0,91	0,91	Mean 3x3	0,94	0,96	0,97	0,97	0,97
Mean 5x5	0,79	0,83	0,84	0,85	0,86	Mean 5x5	0,92	0,95	0,95	0,96	0,96
Mean 7x7	0,74	0,78	0,79	0,81	0,81	Mean 7x7	0,89	0,93	0,94	0,94	0,94
Median 3x3	0,85	0,91	0,92	0,92	0,93	Median 3x3	0,94	0,97	0,97	0,97	0,97
Median 5x5	0,79	0,84	0,85	0,86	0,86	Median 5x5	0,92	0,95	0,96	0,96	0,96
Median 7x7	0,73	0,77	0,79	0,80	0,81	Median 7x7	0,90	0,93	0,94	0,94	0,94
Gauss 3x3	0,82	0,89	0,90	0,90	0,90	Gauss 3x3	0,93	0,96	0,96	0,97	0,97
Gauss 5x5	0,81	0,86	0,87	0,88	0,88	Gauss 5x5	0,93	0,96	0,96	0,96	0,96
Gauss 7x7	0,80	0,86	0,87	0,88	0,88	Gauss 7x7	0,93	0,96	0,96	0,96	0,96
Andrederivert horisontal	0,92	0,98	0,99	0,99	0,99	Andrederivert horisontal	0,97	0,99	0,996	0,998	0,998
Andrederivert vertikal	0,91	0,98	0,99	0,99	0,99	Andrederivert vertikal	0,97	0,99	0,996	0,998	0,998
Andrederivert absolutt	0,88	0,97	0,98	0,98	0,98	Andrederivert absolutt	0,95	0,99	0,99	0,99	0,99

Beregningen av SNR-verdiene er gjort som vist i ligning 2.3.10. Merk at det er de akkumulerte SNR-verdiene som blir vist i tabell 1. For kolonnen MNF 2 betyr dette at man ser summen av SNR-verdien for første og andre prinsipalkomponent. Ved kjøring av MNF-algoritmen har de 81 første spektralbandene for de hyperspektrale bildene blitt benyttet. Normaliseringen gjør at summen av SNR-verdien for de totalt 81 prinsipalkomponentene man får i MNF-rommet blir 1. MNF ordner prinsipalkomponentene etter støyandel. Prinsipalkomponenten med minst støy kommer ut som den første, og prinsipalkomponenten med mest støy blir den siste. Rekkefølgen for signalet blir da motsatt. Altså vil systemet ordne prinsipalkomponentene etter synkende signal-støy-forhold. Tabell 4.2.1.1 viser akkumulerte og normaliserte SNR-verdier for prinsipalkomponentene man får når forskjellige støyestimatorer blir benyttet. Dette gjøres for å kunne sammenligne virkningen av de forskjellige estimatorene med hverandre når man tar med de N første prinsipalkomponentene. I siste del av appendiks 8.5 kan man se grafene for akkumulerte og normaliserte SNR-verdier til bilde 1.



Figur 4.2.1.1: Første prinsipalkomponent ut av MNF for bilde 3 ved bruk av støyestimator "horisontalt", "vertikalt" og "horisontalt+vertikalt".

Figur 4.2.1.1 viser første prinsipalkomponent til 3 utvalgte støyestimat. Bildene for første prinsipalkomponent til de restrerende estimatene ligner på disse bildene.

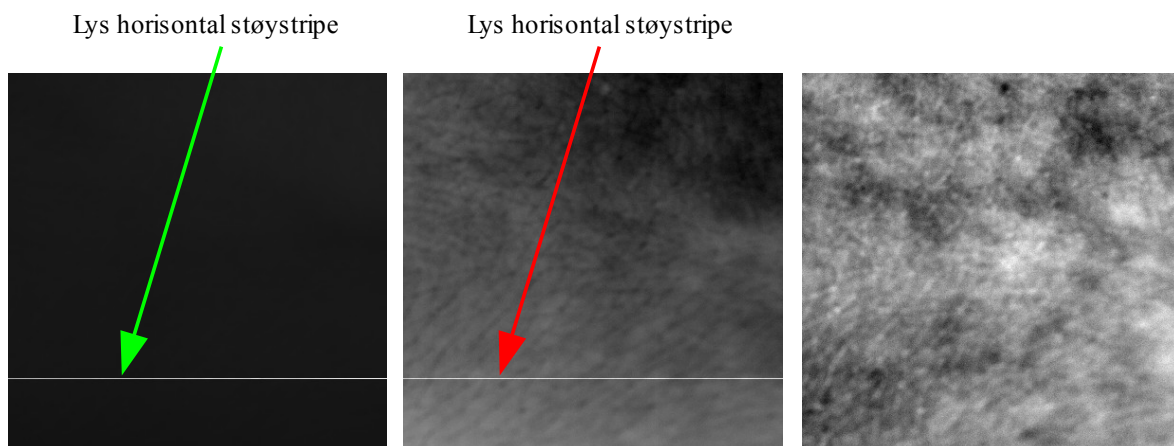
4.2.2 Støyestimering for homogent felt

Tabell 4.2.2.1: Akkumulerte og normaliserte SNR-verdier for de 5 første MNF-komponentene til bilde 1 og 2 basert på homogent felt.

Bilde 1						Bilde 2					
Filter	MNF 1	MNF 2	MNF 3	MNF 4	MNF 5	Filter	MNF 1	MNF 2	MNF 3	MNF 4	MNF 5
Horisontalt	0,91	0,96	0,96	0,96	0,96	Horisontalt	0,95	0,98	0,98	0,98	0,98
Vertikalt	0,91	0,96	0,96	0,97	0,97	Vertikalt	0,96	0,98	0,98	0,98	0,98
Horisontalt + Vertikalt	0,91	0,96	0,96	0,97	0,97	Horisontalt + Vertikalt	0,96	0,98	0,98	0,98	0,98
Mean 3x3	0,91	0,96	0,96	0,97	0,97	Mean 3x3	0,96	0,98	0,98	0,98	0,98
Mean 5x5	0,91	0,96	0,96	0,97	0,97	Mean 5x5	0,96	0,98	0,98	0,98	0,98
Mean 7x7	0,91	0,96	0,96	0,97	0,97	Mean 7x7	0,96	0,97	0,98	0,98	0,98
Median 3x3	0,91	0,96	0,96	0,97	0,97	Median 3x3	0,96	0,98	0,98	0,98	0,98
Median 5x5	0,91	0,96	0,96	0,97	0,97	Median 5x5	0,96	0,98	0,98	0,98	0,98
Median 7x7	0,91	0,96	0,96	0,97	0,97	Median 7x7	0,95	0,97	0,98	0,98	0,98
Gauss 3x3	0,92	0,96	0,96	0,97	0,97	Gauss 3x3	0,96	0,98	0,98	0,98	0,98
Gauss 5x5	0,92	0,96	0,96	0,97	0,97	Gauss 5x5	0,96	0,98	0,98	0,98	0,98
Gauss 7x7	0,91	0,96	0,96	0,97	0,97	Gauss 7x7	0,96	0,98	0,98	0,98	0,98
Andrederivert horisontal	0,91	0,96	0,96	0,96	0,96	Andrederivert horisontal	0,95	0,97	0,98	0,98	0,98
Andrederivert vertikal	0,91	0,96	0,96	0,96	0,97	Andrederivert vertikal	0,96	0,98	0,98	0,98	0,98
Andrederivert absolutt	0,92	0,96	0,96	0,97	0,97	Andrederivert absolutt	0,96	0,98	0,98	0,98	0,98

Tabell 4.2.2.1 viser akkumulerte og normaliserte SNR-verdier for bilde 1 og 2 til de 5 første MNF-komponentene. Dette vil være et mål på hvor mye av det opprinnelige signalet som er bevart til og med den prinsipalkomponenten man ser på. Ved kjøring av MNF-algoritmen har de 81 første spektralbandene for de hyperspektrale bildene blitt benyttet. Tallene i tabellen er beregnet på tilsvarende måte som i tabell 4.2.1.1 for hele bildeutsnittet. Forskjellen ligger i at det her er benyttet et støyestimat som er dannet på bakgrunn av et homogent eller jevnt område i de hyperspektrale bildene.

Figur 4.2.2.1 viser bildene til tre av prinsipalkomponentene som kom ut av MNF-algoritmen når støyestimatet ble gjort for et homogent område i det hyperspektrale bildet. Til venstre i figur 4.2.2.1 er første prinsipalkomponent for bilde 3 med støyestimering utført som differansen med vertikal nabopiksel. De to andre bildene er andre prinsipalkomponent med støyestimator "mean" og maskestørrelse på 3x3 for henholdsvis bilde 3 i midten og bilde 4 til høyre.



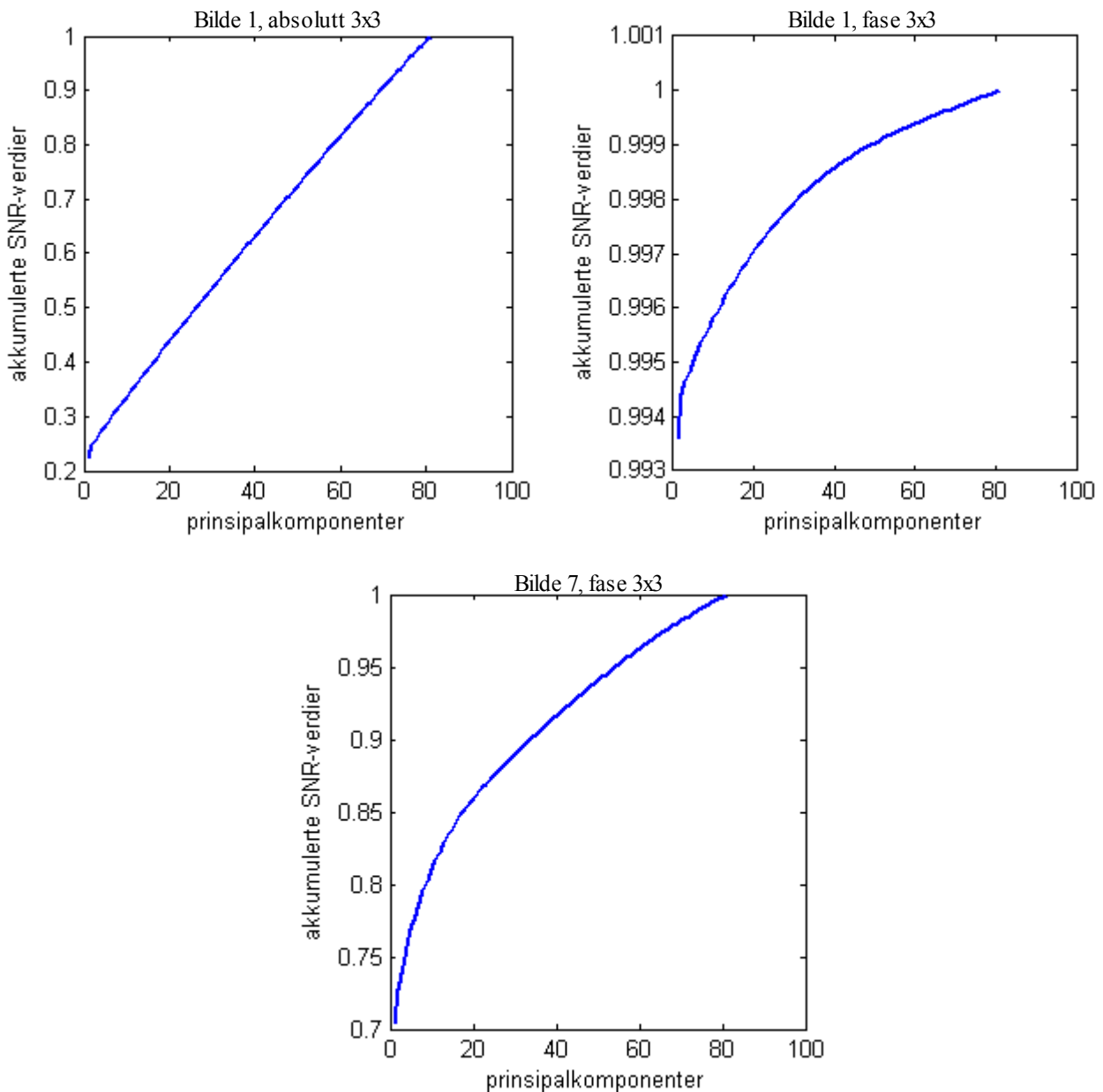
Figur 4.2.2.1: Til venstre er første prinsipalkomponent med støyestimering "vertikal" fra MNF for bilde 3, i midten er andre prinsipalkomponent med støyestimering "mean" til bilde 3, og til høyre er andre prinsipalkomponent med støyestimering "mean" for bilde 4.

4.2.3 Støyestimering i fourierrommet

Tabell 4.2.3.1: Akkumulerte og normaliserte SNR-verdier for de 5 første MNF-komponentene til bilde 1 og 2.

Bilde 1						Bilde 2					
Filter	MNF 1	MNF 2	MNF 3	MNF 4	MNF 5	Filter	MNF 1	MNF 2	MNF 3	MNF 4	MNF 5
Absolutt 3x3	0,22	0,25	0,26	0,27	0,28	Absolutt 3x3	0,21	0,24	0,25	0,27	0,28
Absolutt 5x5	0,53	0,55	0,56	0,57	0,58	Absolutt 5x5	0,51	0,53	0,55	0,56	0,57
Absolutt 7x7	0,74	0,76	0,76	0,77	0,77	Absolutt 7x7	0,76	0,77	0,78	0,78	0,79
Fase 3x3	0,99	0,99	0,99	0,99	0,995	Fase 3x3	0,996	0,997	0,997	0,997	0,997
Fase 5x5	0,99	0,99	0,99	0,99	0,99	Fase 5x5	0,996	0,997	0,997	0,997	0,997
Fase 7x7	0,99	0,99	0,99	0,99	0,99	Fase 7x7	0,996	0,997	0,997	0,997	0,997

Akkumulerte og normaliserte SNR-verdier for bilde 1 og 2 til de 5 første MNF-komponentene er vist i tabell 4.2.3.1. Tallene i tabellen er beregnet på tilsvarende måte som i tabell 4.2.1.1 for hele bildeutsnittet. Forskjellen ligger i støyestimeringen som her er gjort i fourierrommet.

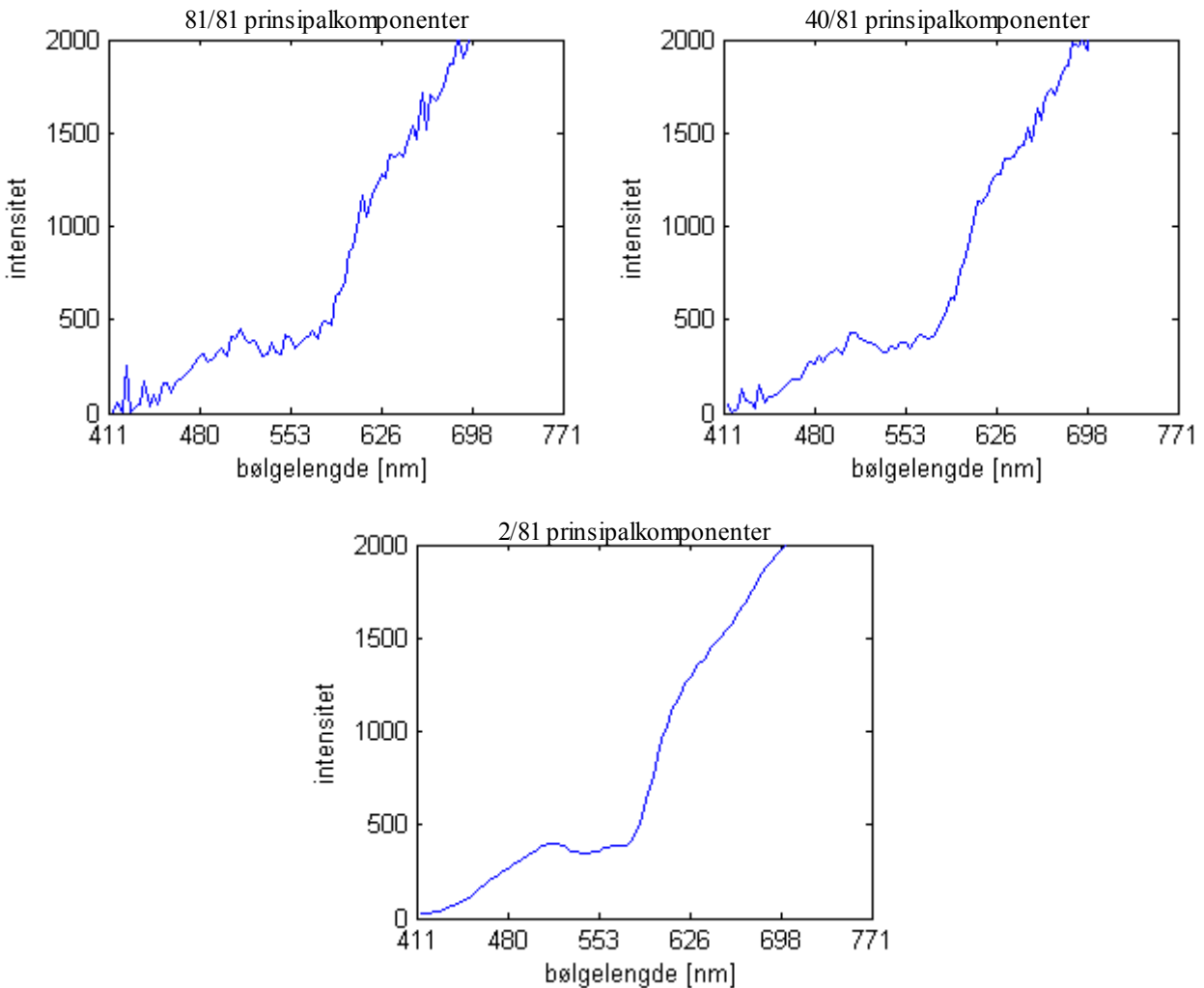


Figur 4.2.3.1: Akkumulerte og normaliserte SNR-verdier

Figur 4.2.3.1 viser grafene for SNR-verdiene når man tar med samtlige av prinsipalkomponentene. I appendiks 8.7 vises alle støyestimaterne gjort i fourierrommet for bilde 1. I figur 4.2.3.1 er det i tillegg tatt med grafen for bilde 7 når støyestimering er basert på fase med maskestørrelse 3x3. Merk at aksene som viser akkumulerte SNR-verdier for grafene i figur 4.2.3.1 er forskjellige. Dette er gjort for å synliggjøre endringene til kurvene.

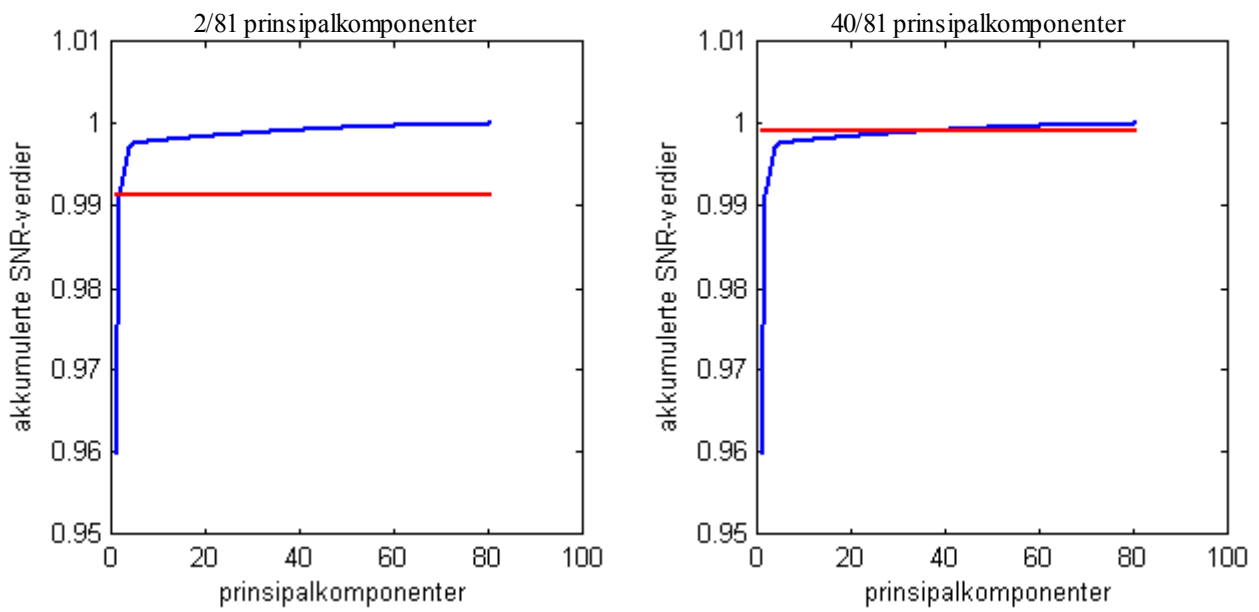
4.2.4 Antall band under tilbaketransformering

Masken og filteret som ble benyttet til støyestimeringen er vist i figur 3.4.1.4. Figur 4.2.4.1 viser spekteret til pikselen med koordinater $x = 128$ og $y = 111$ i bilde 2. Grafen øverst til venstre er det originale spekteret fra det hyperspektrale bildet uten å benytte MNF-algoritmen til støyreduksjon. En oppnår identisk spekter hvis man tar med alle prinsipalkomponentene under reverstransformasjonen fra MNF-rommet. Grafen til høyre viser resultatet der det er beholdt de 40 første av de totalt 81 prinsipalkomponentene. For grafen nederst er det bare tatt med de 2 første komponentene ved reverstransformasjonen.

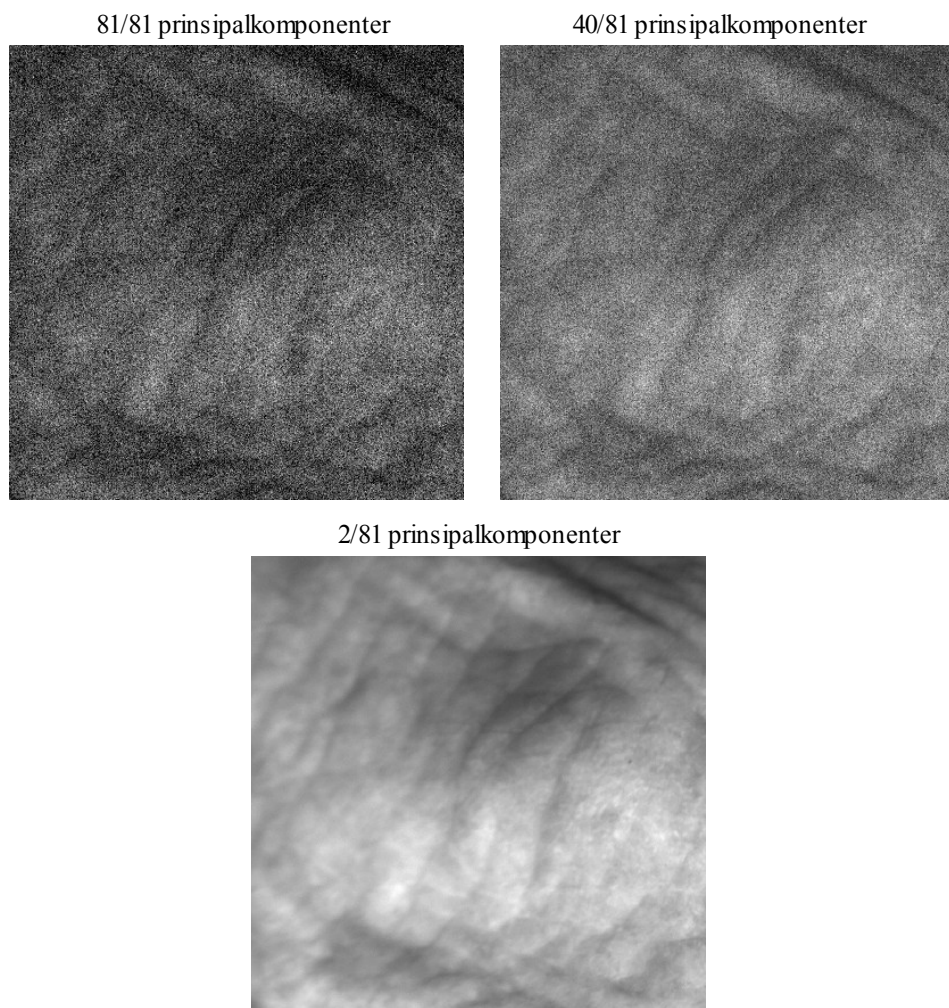


Figur 4.2.4.1: Spekter til enkelt piksel etter støyreduksjon

Figur 4.2.4.2 viser akkumulerte SNR-verdier til prinsipalkomponentene for bilde 2. De røde strekene er plassert på grenseverdien for hvilke prinsipalkomponenter som skal bli tatt med under reverstransformasjonen. For grafen til høyre er den røde streken plassert på summen av SNR-verdiene opp til prinsipalkomponent nummer 40. Altså blir alle prinsipalkomponenter fra 1 til og med komponent 40 tatt med i reverstransformasjonen. I grafen til venstre er den røde streken plassert på summen av SNR-verdien til de to første prinsipalkomponentene. På grunn av normaliseringen vil akkumulert SNR-verdi for den høyeste prinsipalkomponenten alltid være lik 1.



Figur 4.2.4.2: De blå grafene viser normaliserte og akkumulerte SNR-verdier for bilde 2. De røde linjene viser grensen for 2 og 40 prinsipalkomponenter.



Figur 4.2.4.3: Utsnitt av spektralbånd 10 til bilde 2. Bildet øverst til venstre er det opprinnelige utsnittet for det hyperspektrale bildet. Øverst til høyre er det utført støyreduksjon ved å beholde de 40 første prinsipalkomponentene under reverstransformasjonen. I midten nederst vises utsnittet når bare de 2 første prinsipalkomponentene ble bevart.

I figur 4.2.4.3 ser man de faktiske resultatene av støyreduksjonen på det hyperspektrale bildet som en følge av begrensningen av prinsipalkomponenter. Øverst til venstre vises et utsnitt på 400x400 piksler fra spektralband 10 i det hyperspektrale bildet uten støyreduksjon. Til høyre ser man resultatet fra samme utsnitt når det er bevart de 40 første prinsipalkomponentene under reverstransformasjonen. Nederst i midten er resultatet når kun de 2 første prinsipalkomponentene blir tatt med.

Tabell 4.2.4.1: Antall prinsipalkomponenter man må ta med for å beholde en gitt prosent.

	Bilde 1	Bilde 2	Bilde 3	Bilde 4	Bilde 5	Bilde 6	Bilde 7	Bilde 8	Bilde 9
97,00%	2	2	4	2	2	5	2	2	2
98,00%	3	2	6	2	2	5	3	3	3
99,00%	4	2	27	3	2	6	24	5	5
99,25%	5	3	36	3	2	7	33	14	14
99,50%	18	4	46	5	3	9	43	26	27
99,75%	38	5	58	26	4	19	55	43	45
99,90%	55	32	68	48	25	43	66	57	59

Tabell 4.2.4.1 viser hvor mange prinsipalkomponenter man må beholde under reverstransformasjonen for å bevare en gitt prosent av de opprinnelige hyperspektrale bildene. Antallet prinsipalkomponenter er gitt av grensen til de akkumulerte SNR-verdiene vist i første del til appendiks 8.5.

4.3 Støyreduksjon ved ulik belysning

Antallet prinsipalkomponenter for de forskjellige bildene som skal bli bevart under PCA- og MNF-algoritmen er valgt slik at en hele tiden beholder 99,25% av det opprinnelige signalet. Av tabell 4.2.4.1 gir dette 7 prinsipalkomponenter for bilde 6, 33 for bilde 7, samt 14 for bilde 8 og 9. For å muliggjøre sammenligning mellom algoritmene er dette tallet likt uavhengig om det er benyttet PCA eller forskjellige støyestimat for MNF. Skal man imidlertid sammenligne støyreduserte resultat for tilsvarende algoritme, men på tvers av bildene må man imidlertid ta høyde for denne forskjellen.

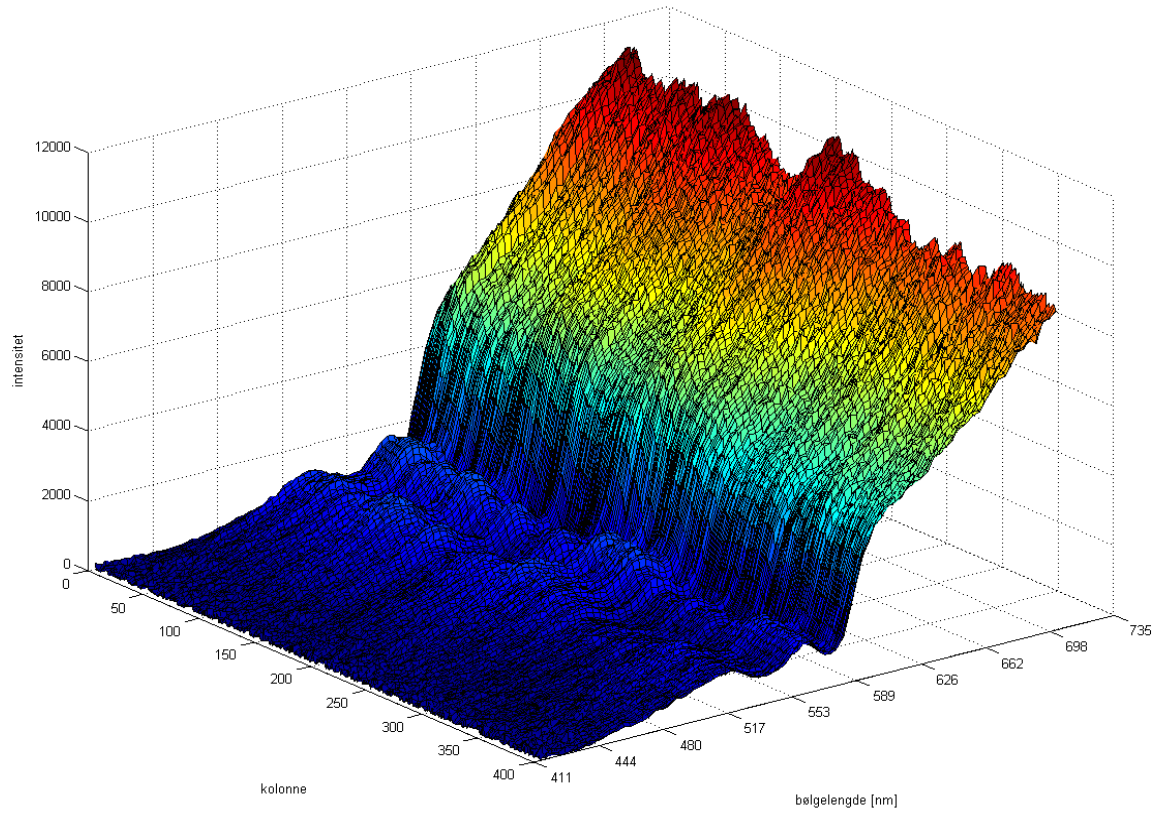
Den øverste grafen i figur 4.3.1 viser det opprinnelige spekteret for en enkelt rad uten støydemping til bilde 6. Nederst vises hvordan den samme raden blir seende ut etter støydempning. Det er rad 121 man ser i figur 4.3.1 fra utsnittet av det hyperspektrale bildet med størrelse 400x400 piksler. Bilde 6 er tatt under belysning av hvitt lys. MNF-algoritmen basert på støyestimator "Horisontal + Vertikal" er benyttet for å utføre støydempningen. Under tilbaketransformering i MNF-algoritmen ble 7 av totalt 81 prinsipalkomponenter bevart. I figur 4.3.1 er kolonnene plassert langs med x-aksen. Kolonnene går fra 1 til 400. På y-aksen ligger bølgelengdene til de forskjellige spektralbåndene. Det er totalt 81 spektralbånd som vises i grafene, og dette tilsvarer bølgelengdene fra 411nm til 702nm. Z-aksen viser intensiteten til de forskjellige punktene. Grafene illustrerer også intensitetsnivåene med farger der blått tilsvarer lav intensitet og rødt høy intensitet.

Figur 4.3.2 viser det opprinnelige spekteret for en enkelt rad med og uten støydempning til bilde 8. Her er det også rad 121 man ser spekteret til. Bilde 8 ble tatt under belysning av 355nm laserlys og ikke hvitt lys. Øverst i figur 4.3.2 vises utgangspunktet uten støydempning, og nederst ser man resultatet etter støydempningen. Det er MNF-algoritmen basert på støyestimatorene "Andrederiverte vertikal" som ble benyttet for å utføre støyreduksjonen. Under tilbaketransformeringen ble 14 av totalt 81 spektralbånd tatt med. Aksene i figur 4.3.2 gir samme informasjon som de man finner i figur 4.3.1.

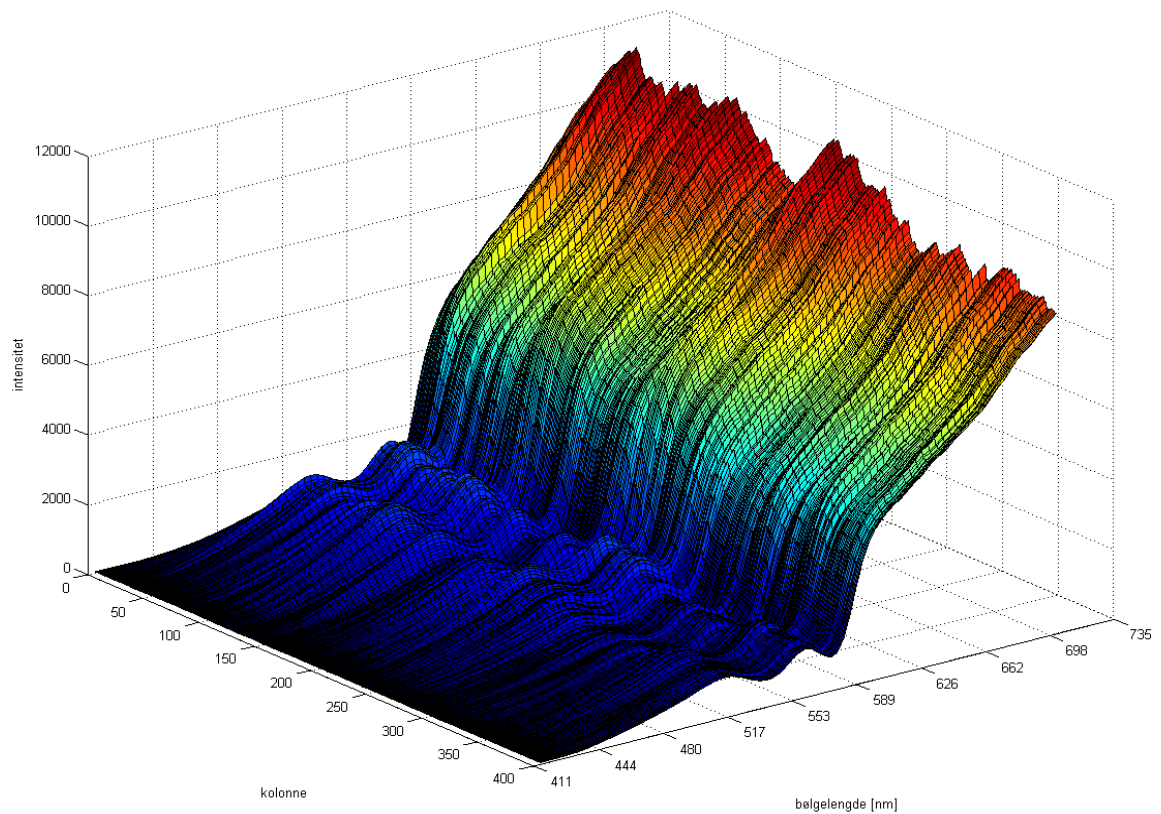
I figur 4.3.3 ser man et romlig utsnitt fra bilde 6 før og etter støydempning. Utsnittet er fra spektralbånd nummer 70 som har bølgelengde 662nm. Den øverste grafen viser utgangspunktet uten støyfjerning, og nederst ser man resultatet etter støyfjerning. Det er MNF-algoritmen med støyestimator "Mean 3x3" som ble benyttet til støyfjerningen. Her ble 7 av 81 spektralbånd beholdt under reverstransformasjonen. X-aksen som ligger nederst i figurene og y-aksen som er plassert til høyre henspiller til hvilke piksler man ser fra utsnittet på 400x400. På z-aksen til venstre vises intensiteten til de forskjellige pikslene. Her blir også intensitetsnivåene gjengitt med farger. Blått er fremdeles lav intensitet og rødt er høy.

Figur 4.3.4 viser et romlig utsnitt fra bilde 8. Øverst vises hvordan det ser ut uten støydempning, og nederst er resultatet etter støydempning. Her er det PCA-algoritmen som har blitt brukt for å oppnå støyreduksjon. Under tilbaketransformasjonen ble 14 av 81 spektralbånd bevart. Spektralbåndet man ser er nummer 70 med bølgelengde 662nm. Utsnittet er også likt som i figur 4.3.3. Dette kan man se ved å sammenligne verdiene på x- og y-aksen mellom figur 4.3.3 og 4.3.4.

Originalt utsnitt for alle spektralbånd til enkelt rad for bilde 6

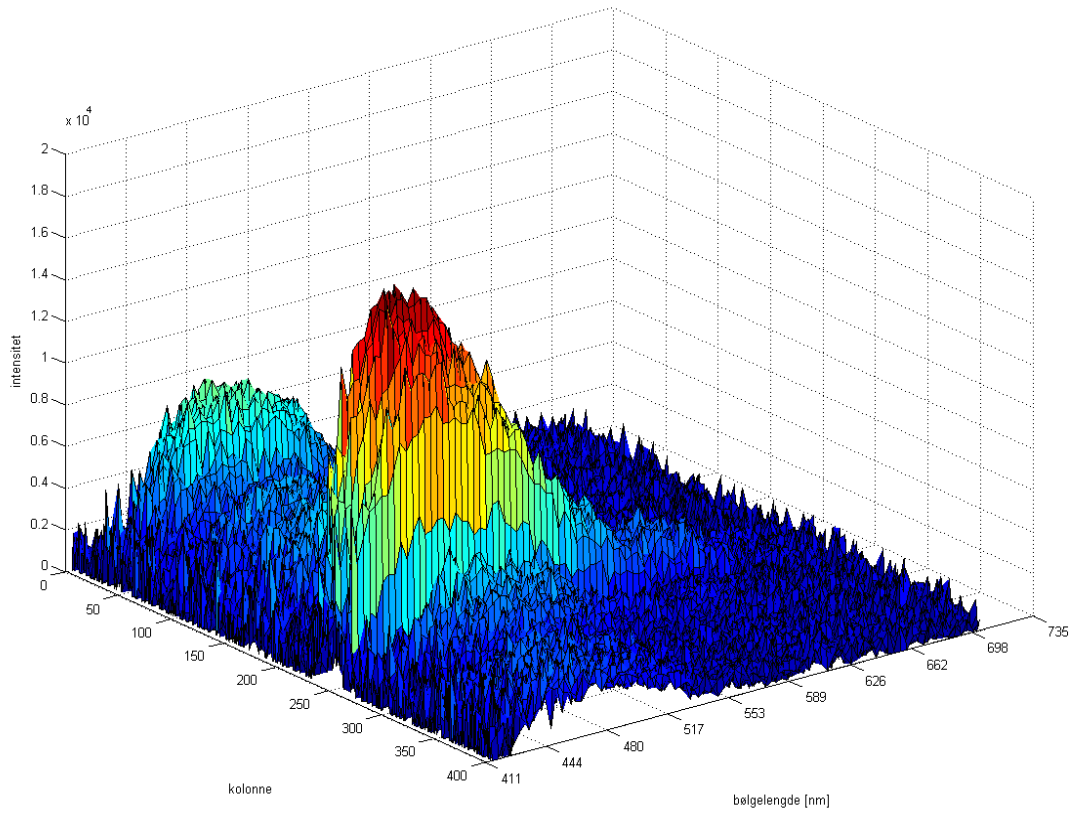


Støyreduisert resultat fra MNF-algoritmen basert på støyestimatoren "Horisontalt + Vertikalt" til enkelt rad for bilde 6

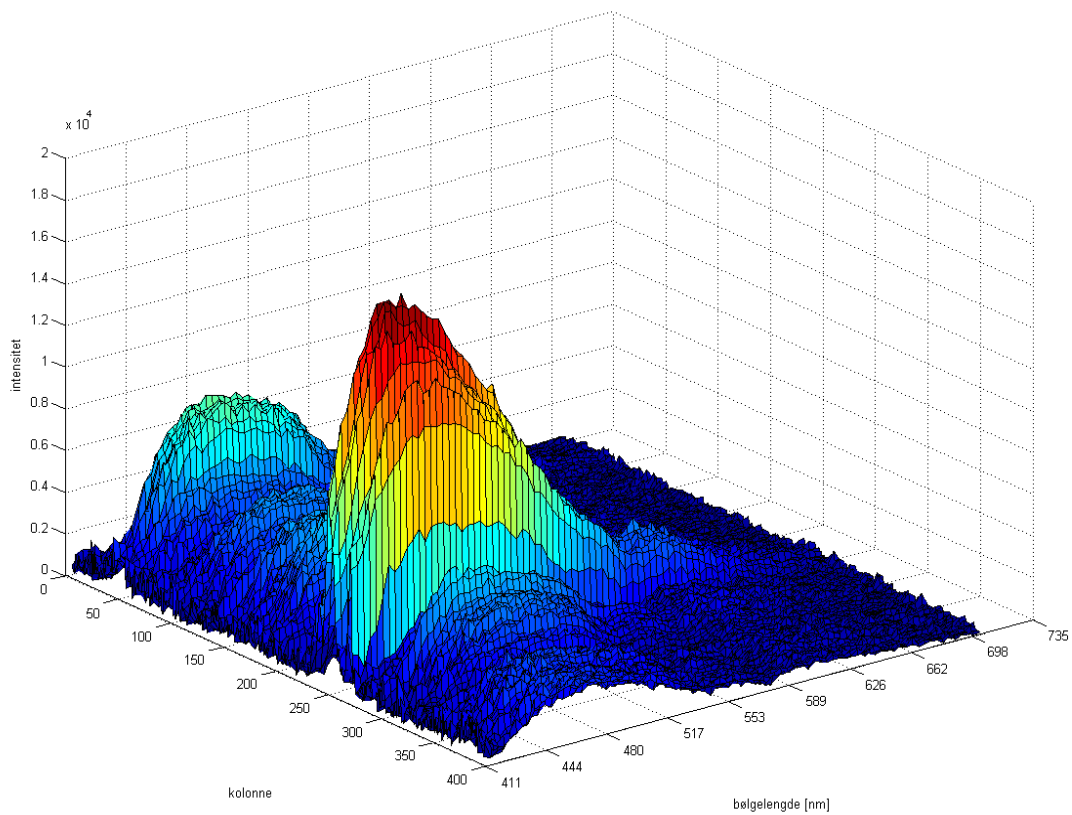


Figur 4.3.1: Spektral støydem্পning for en enkelt rad til hyperspektralt bilde som er tatt under hvitt lys (bilde 6).

Originalt utsnitt for alle spektralbånd til enkelt rad for bilde 8

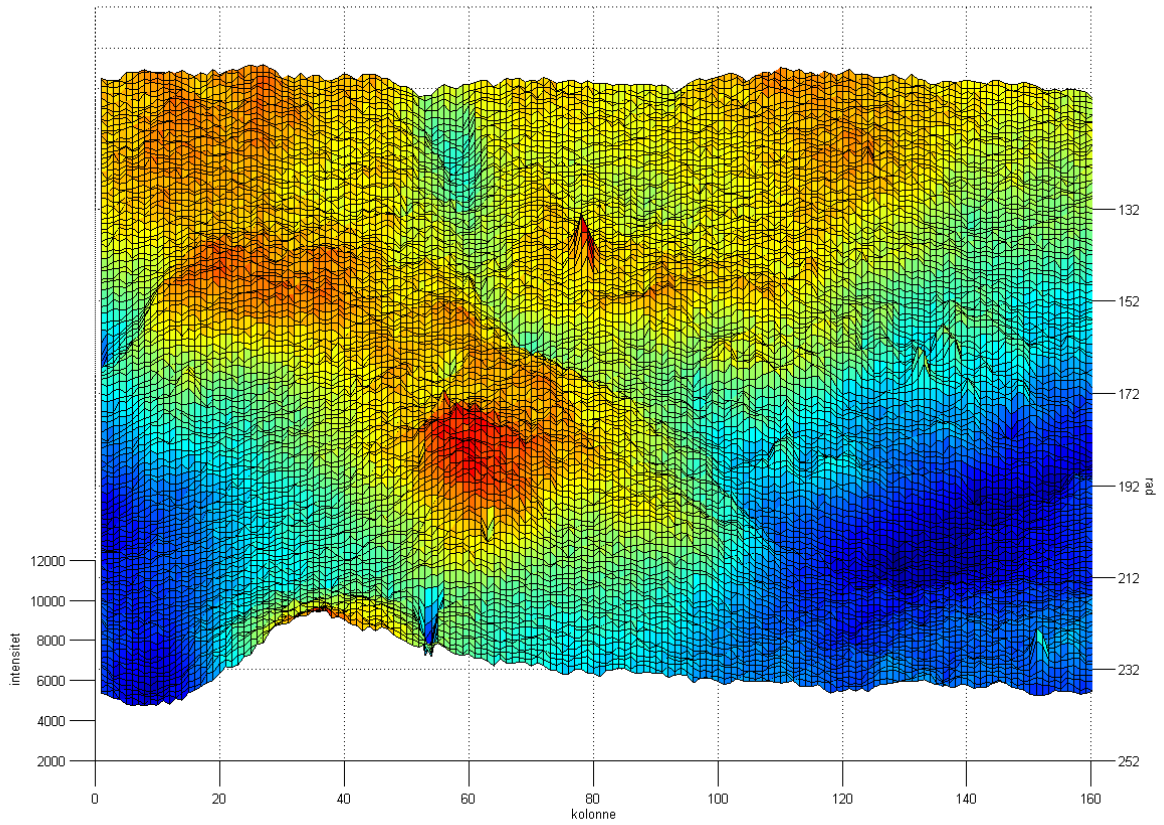


Støyreduisert resultat fra MNF-algoritmen basert på støyestimatoren "Andrederivert vertikalt" til enkelt rad for bilde 8

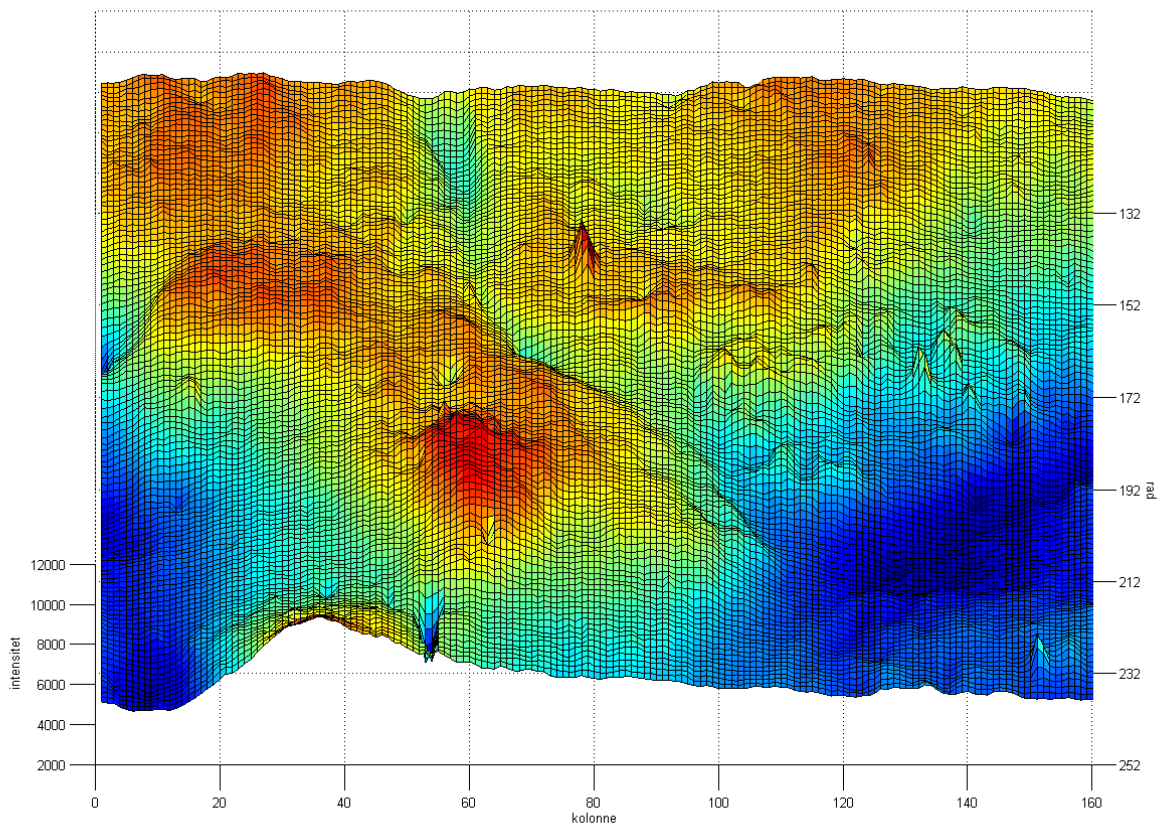


Figur 4.3.2: Spektral støydempering for en enkelt rad til hyperspektralt bilde ved belysning av 355nm laser (bilde 8).

Originalt utsnitt for høyintensitets spektralbånd 70 (662nm) til bilde 6

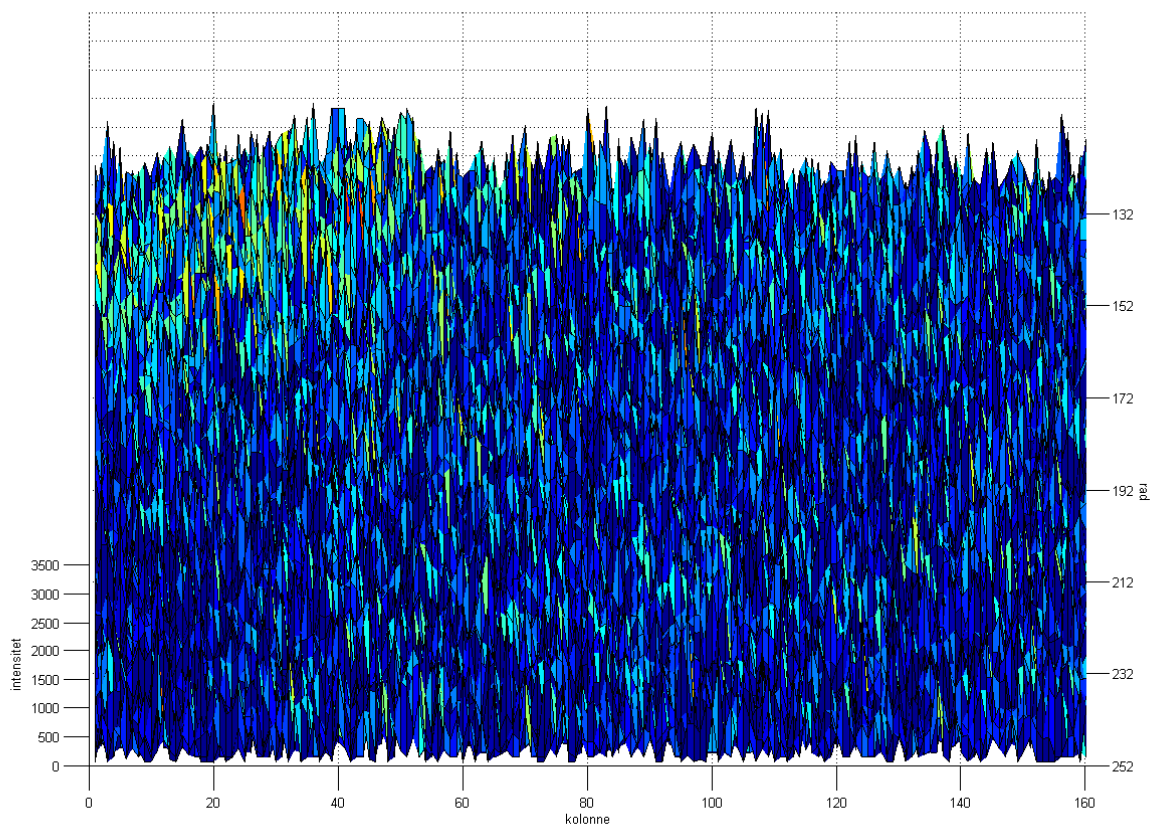


Støyreduert resultat fra MNF-algoritmen basert på støyestimatoren "Mean 3x3" for spektralbånd 70 til bilde 6

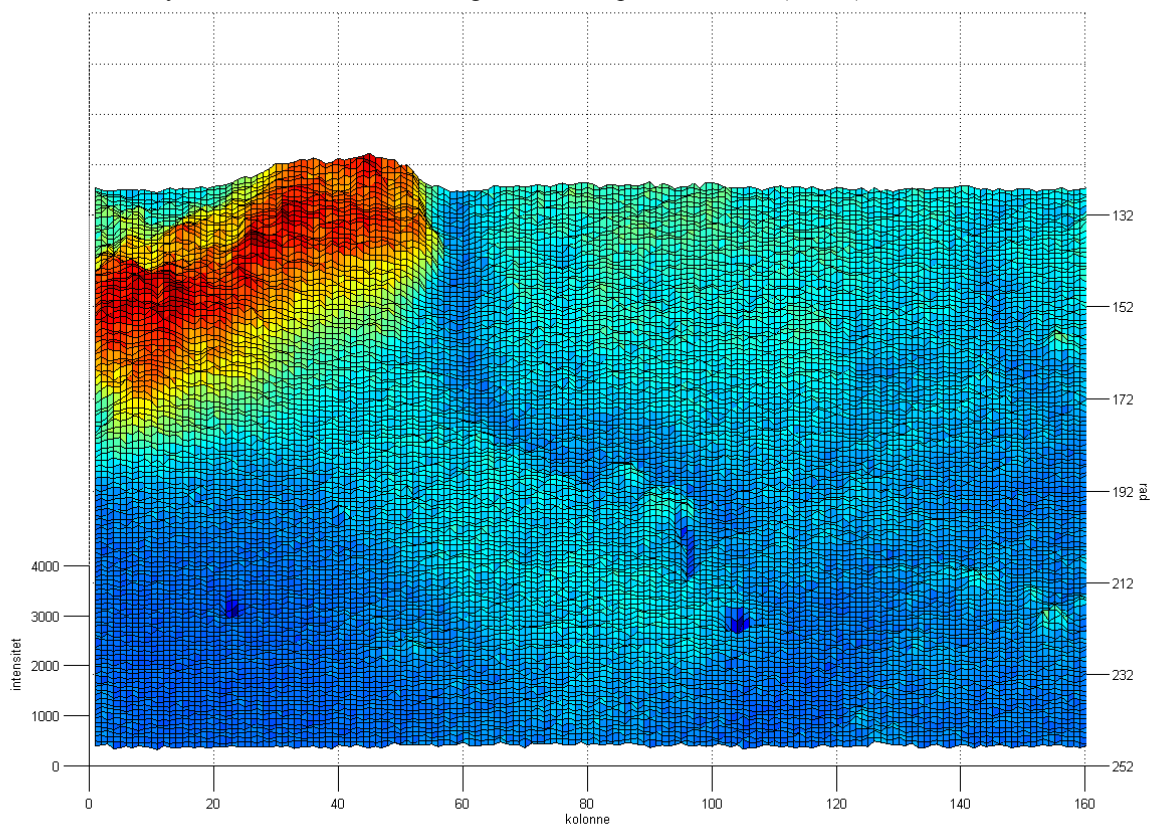


Figur 4.3.3: Romlig støydemping for høyintensitets spektralbånd 70 (662nm) til bilde 6

Originalt utsnitt for lavintensitets spektralbånd 70 (662nm) til bilde 8



Støyredusert resultat fra PCA-algoritmen for spektralbånd 70 (662nm) til bilde 8



Figur 4.3.4: Romlig støydempling for lavintensitets spektralbånd 70 (662nm) til bilde 8

5 Diskusjon

5.1 Fjerning av horisontal støystripe

De to grafene øverst i figur 4.1.1 er for bilder uten noen horisontal støystripe. Grafen til høyre er ganske jevn, og en kan derfor forvente at det ikke registreres noen egen topp. Grafen til venstre svinger litt mer opp og ned, men det er ingen dominant topp. Den nederste grafen er for bilde 3 som inneholder en dominant horisontal støystripe. Dette kommer tydelig frem som en kraftig topp i grafen. Årsaken til at denne toppen blir registrert som en horisontal støystripe og ikke de mindre variasjonene for grafen øverst til venstre er valget av thresholdnivå. Her er det en variabel grense som toppen blir sammenlignet med. Hvis thresholdnivået settes for lavt vil typisk mindre toppe som finnes i grafen øverst til venstre i figur 4.1.1 bli tatt for å være egne støylinjer. Settes thresholdnivået for høyt vil faktiske støytopper som vist i grafen for bilde 3 ikke bli registrert. Begge disse tilfellene fører til et resultat som ikke er tilfredsstillende. For analysene av bildene er thresholdnivået valgt slik at en topp blir detektert som tegn på støylinje hvis verdien på toppen er minst 10 ganger høyere enn gjennomsnittsverdien for hele grafen. Ser man på grafen øverst til venstre er toppene mindre enn 2 ganger gjennomsnittet. Derfor er det ingen mulighet for at disse skal detekteres som mulig støylinje. Hva thresholdverdien bør være kan avhenge av det som er avbildet i de hyperspektrale bildene. Dette er noe man bør undersøke før man bestemmer thresholdnivå.

Uten noen form for støyfjerning viser det venstre bildet i figur 4.1.2, den karakteristiske horisontale støystripen i første prinsipalkomponent fra MNF. Her har støyestimeringen under MNF-algoritmen blitt beregnet ved hjelp av differansen med nærmeste horisontale nabopiksel. Dette er et tydelig tegn på en svakhet ved støyestimatoren når det kun er denne støystreken som kommer frem i første prinsipalkomponent. Bildet til høyre viser også første prinsipalkomponent man får ut av MNF-algoritmen. Det er dette en forventer å se for denne prinsipalkomponenten. Her er det ingen tegn til den horisontale støystripen som dominerer det venstre bildet. Videre er det fremdeles benyttet samme støyestimering. Denne drastiske endringen kommer av en støyfjerning før MNF-algoritmen ble utført på bildet til høyre. Under støyfjerningen har den dominante horisontale støystripen blitt erstattet med en interpolasjon av nabolinjene. Denne sammenligningen viser at operasjonen der støystripen erstattes med interpolasjonen av nabolinjene virker tilfredsstillende.

Sammenligner man grafen for bilde 3 i figur 4.1.1 med bilde til venstre i figur 4.1.2 vil man se en tydelig sammenheng. Toppen fra grafen i figur 4.1.1 sammenfaller med den horisontale støylinjen. Dette er ikke tilfeldig da det er denne toppen som blir brukt til å detektere støylinjen for bilde 1.

5.2 Sammenligning av MNF med PCA

Prinsipalkomponentene til PCA finnes i rapporten fra høstprosjektet^[4]. Disse prinsipalkomponentene har størrelse på 600x600 piksler, mens de fra MNF er på 400x400. Utsnittet er også tatt fra et litt annet sted i bildet, men ellers så er alt likt. Her er det hovedtrekkene med PCA og MNF som er viktig, så disse små avvikene mellom utsnittene bør derfor ha lite å si for resultatene.

Den viktigste forskjellen mellom MNF og PCA, er at MNF klarer å sortere prinsipalkomponentene etter bildekvalitet. PCA klarer ikke dette, da den andre prinsipalkomponenten til PCA var helt dominert av støy. Videre inneholder prinsipalkomponentene 4, 5 og 6 til PCA et signal. Dette gjør at PCA ikke er egnet til bruk for å redusere dimensjonaliteten, da PCA ikke klarer å samle prinsipalkomponentene med signal. Figur 4.2.1 som viser de 6 første prinsipalkomponentene ut av MNF har en monotont synkende bildekvalitet. Her er støyen blitt skilt fra signalet i større grad enn for PCA, og den har blitt samlet blant de høyere prinsipalkomponentene til MNF. Dette gjør at MNF kan egne seg til reduksjon av dimensjonalitet ved at man forkaster de høyere prinsipalkomponentene og beholder bare de med et tilfredsstillende signal-støy-forhold.

Et viktig poeng som bør bemerkes er at resultatet avhenger av hva som avbildes. For disse hyperspektrale bildene som er tatt av oversiden til hender, blir allerede andre prinsipalkomponent til PCA totalt dominert av støy. Dette gjelder ikke for de mer tradisjonelle fly- og satellitt-fotoene av jordoverflaten. Chen^[5] gjør tilsvarende sammenligning for et slikt tradisjonelt datasett.

Årsaken til at PCA klarer å sortere prinsipalkomponentene etter bildekvalitet for flyfotoene, men ikke klarer det for disse medisinske bildene kommer trolig av forskjellen mellom støyen i de to bildetyperne. Fra kapittel 2.3 får man at MNF er det samme som "noise adjusted principal component analysis" (NAPC). NAPC er en prosedyre i to steg. Det første steget dekorrelerer og reskalerer støyen i de hyperspektrale bildene. Det andre steget er et standard PCA trinn. Hvis støyen hadde vært dekorrelert i de opprinnelige hyperspektrale bildene, ville MNF og PCA vært tilsvarende prosedyrer og skulle derfor ha gitt ut det samme resultatet. Siden MNF lyktes og PCA feilet med å ordne prinsipalkomponentene etter bildekvaliteten tyder dette på at støyen i de hyperspektrale bildene var korrelert.

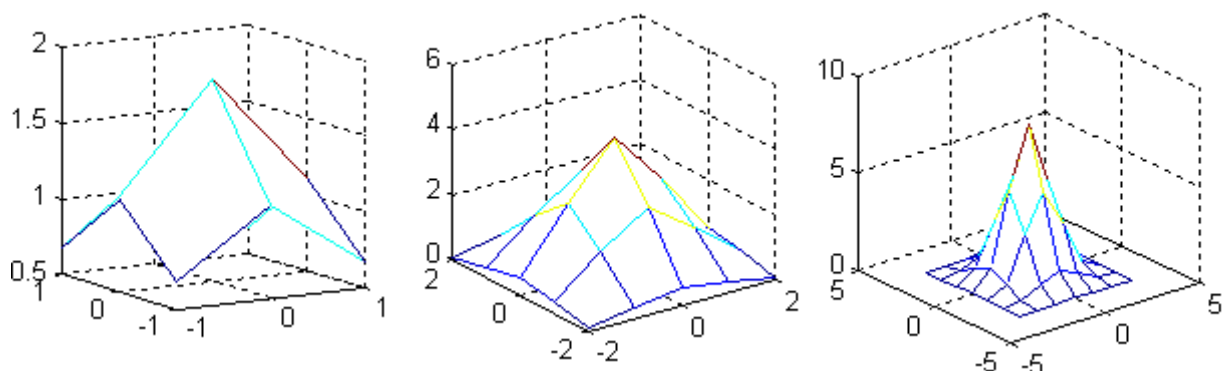
PCA og MNF har en vesentlig forskjell mellom seg når det kommer til hvilken strukturell informasjon som blir tatt hensyn til under algoritmene. PCA er en punktvis operasjon som ikke tar høyde for den romlige strukturen^[13]. En fordel med dette er at man kan plukke ut hvilke punkter man vil fra de hyperspektrale bildene. Med dette så menes det at pikslene i de hyperspektrale bildene blir sett på som individuelle målinger, og de utvalgte punktene trenger ikke å ligge ved siden av hverandre. Dette kan være en god idé hvis man vil unngå partier med mye støy men vil beholde resten av bildene. Et slikt parti kan for eksempel være støystripen som finnes i bilde 3.

I motsetning til PCA benytter MNF den romlige strukturen i de hyperspektrale bildene. Dette blir gjort under støyestimering, når man for eksempel bruker differansen med nærmeste nabopiksel som estimat. Her er fordelen at en klarer å nyttegjøre seg av mer informasjon. Dette fører også til at settet med prinsipalkomponenter som kommer ut av MNF-algoritmen blir ordnet etter signal-støyforholdet.

5.2.1 Estimering av kovariansmatrisen til støyen

Tabell 4.2.1.1 viser akkumulerte og normaliserte SNR-verdier for prinsipalkomponentene til bilde 1 og 2. Hvis man sammenligner støyestimatene for bilde 2 av typen "mean", "median" og "Gauss" med maskestørrelse 3×3 , ser en at de har cirka like akkumulerte SNR-verdier for prinsipalkomponentene. En kan gjøre tilsvarende sammenligning for bilde 1. Forskjellen mellom estimatene er større for bilde 1 enn for bilde 2. Til bilde 1 er det tydelig at "median" får den høyeste SNR-verdien for første prinsipalkomponent når man sammenligner filtrene med maskestørrelse på 3×3 . At "median" har større SNR-verdi enn "mean" og "Gauss" kan komme av at "median" ikke har en utsmørende effekt. Lavpassfilter som tilsvarende estimatene "mean" og "Gauss" har denne effekten på kanter i bilder^[13, 18]. Ved valg av støyestimator må man også ta hensyn til kompleksiteten og kjøretiden for støyestimatene. Dette kommer ikke frem i tabell 4.2.1.1, der bare resultatene foreligger. Gjennomsnittsestimatet "mean" er veldig likt "Gauss", men "Gauss" har en individuell vektning av celleverdiene innenfor masken. Det gjør at hver celleverdi må multipliseres med en egen konstant for "Gauss". Støyestimatet "mean" slipper dette da vektningen er lik 1 for alle cellene innenfor masken. På grunn av dette blir estimatet "Gauss" litt mer komplekst enn "mean". Støyestimatet som kommer best ut blant de tre estimatene som har vært nevnt over i tabell 4.2.1.1 er "median". Det tar imidlertid også lengst tid å beregne dette støyestimatet. For å finne medianverdien innenfor masken kreves det at verdiene blir sortert etter størrelse, som har en kjøretid på $O(n \cdot \log n)$ ^[31] der n er antallet celler i masken. De to andre støyestimatene "mean" og "Gauss" må summere opp cellene i masken etter at de har blitt multiplisert med forhåndsbestemte konstanter. Dette vil ha en kjøretid på $O(n)$ som er raskere enn kjøretiden til "median"-estimatet.

Ser man på virkningen maskestørrelsen har innenfor en type støyestimat, kommer det frem at små maskestørrelser har høyere SNR-verdi for en prinsipalkomponent enn tilsvarende estimat med større maskestørrelser. Dette er tydelig både for "mean" og "median" estimatene. Estimering av støyen med maskestørrelse på 3×3 fungerer bedre enn de større 5×5 og 7×7 maskene. For "Gauss" er ikke dette like tydelig. Her er forskjellen for SNR-verdiene til prinsipalkomponentene noe mindre når en sammenligner 3×3 maskestørrelsen med 5×5 og 7×7 . Forskjellen mellom "Gauss" og de to andre kommer mest sannsynlig fra at "Gauss" har en tyngre vektningen av de midterste cellen i masken, og de mer perifere cellene har en vektning ikke langt fra 0. Sammenligner man Gausskurvene i figur 5.2.1.1 som er benyttet til masker, ser en det tydelig for 7×7 masken at ytre ramme har verdier ikke langt fra 0. Dette fører til at "Gauss"-estimatene er mindre avhengig av maskestørrelsen.



Figur 5.2.1.1: Gauss-masker med størrelse på 3×3 , 5×5 og 7×7

Resultatene er klare på at støyestimatene som benytter små maskestørrelser gir et bedre støyestimat enn de med større maskestørrelse. Det går i tillegg raskere å beregne støyestimatet for små masker enn for store, så det er liten tvil om at maskevalg bør være lite.

Støyestimaterne basert på den andrederiverte har høye SNR-verdier i forhold til de andre estimatorene som er tatt med i denne testen. Fra og med andre MNF-komponent er de høyest både for bilde 1 og 2. De andrederiverte estimaterne som kun baserer seg på en retning, ligner mye på de to første estimatene som bare benytter differansen med en nabopiksel; horisontal nabo til høyre og vertikal nabo rett opp. Bildet av første prinsipalkomponent estimert med den andrederiverte i horisontal retning for bilde 3, som inneholder en horisontal støystripe, ligner veldig på det venstre bildet i figur 4.2.1.1. Den samme karakteristiske streken tvers igjennom bildet i horisontal retning er det eneste man ser. Årsaken til at denne streken ikke blir behandlet som støy, er den samme som for støyestimatet som bare baseres på høyre nabopiksel. Det er ingen registrering av svært hurtig endring i vertikalretningen når estimatet kun baseres på den andrederiverte i horisontal retning. For estimeringen med den andrederiverte i vertikalretning blir bildet av første prinsipalkomponent lignende det midterste bildet i figur 4.2.1.1. Støyestimatet "andrederivert absolutt" summerer absoluttverdien av den andrederiverte i horisontal og vertikal retning. Fra tabell 4.2.1.1 får man at SNR-verdien for dette støyestimatet er litt lavere enn de to andre som bare benytter seg av en retning. Siden "andrederivert absolutt" krever at man beregner absoluttverdien, vil dette gjøre at det har høyere kompleksitet enn de to andre. Signal-støy-forholdet er også dårligere enn de to som bare baseres på en retning. Disse to tingene gjør at man bør velge et av de enkleste. Hvilket av de to en bør velge avhenger av om det er horisontale støystriper i de hyperspektrale bildene man ser på. Hvis det er det bør ikke "andrederivert horisontal" velges, da denne støyestimatoren ikke registrerer disse stripene som støy.

I figur 4.2.1.1 blir bilde 3 trukket frem. Dette hyperspektrale bildet har en dominant støystripe. Valget av støyestimator vil ha stor påvirkning på om denne stripen blir registrert som støy, eller tatt for å være en del av signalet. Støyestimatorene "horisontalt" og "vertikalt" er støyestimat basert på én nabopiksel. Estimatoren "horisontalt" benytter differansen med horisontal nabopiksel som estimat, og "vertikalt" bruker differansen med en nabopiksel som ligger i vertikalretningen. For bilde 3 er SNR-verdien til første prinsipalkomponent 0,95 ved bruk av "horisontal" støyestimator, og 0,92 med "vertikal" støyestimator. Dette skal i utgangspunktet antyde at støyestimat utført med horisontal nabo er bedre enn med vertikal nabo. Hvis man ser på det faktiske bildet av første prinsipalkomponent til de to støyestimaterne er sannheten noe annerledes. De er vist i figur 4.2.1.1. For "horisontalt" støyestimat ser man kun en hvit strek mot en svart bakgrunn, mens bildet man får når "vertikalt" estimat benyttes er mer som forventet. Dette kommer av den dominante støystreken som finnes i bilde 3. Støystreken går horisontalt igjennom bildet, og et støyestimat som ikke tar med noen form for endring i vertikal retning vil derfor ikke registrere denne tydelige støykomponenten. MNF garanterer at signal-støy-forholdet skal synke med stigende prinsipalkomponent. Det er også tilfellet ved bruk av horisontalt filter for bilde 3. Problemet ligger ved støyestimatet som skal estimere støyen. Hvis støyestimatoren ikke merker en slik dominant støystripe, vil MNF-algoritmen registrere den som en del av signalet og den blir da trukket frem i de første prinsipalkomponentene. Dette illustrerer også poenget med bruk av differansen til vertikal nabopiksel er langt bedre enn å benytte differansen til horisontal nabopiksel, der det er sterke støykomponenter som går tvers igjennom bildet i horisontal retning^[11].

Hvis man sammenligner bildene av første prinsipalkomponent for MNF-algoritmen til bilde 3 vist i midten og til høyre i figur 4.2.1.1 er det en tydelig forskjell mellom dem. Det som er lyst i det midterste er mørkt i det til høyre og vice versa. Det er MNF-algoritmen som styrer hva som blir lyst og mørkt. Dette valget avhenger både av hvilket støyestimat som benyttes, samt hva som blir avbildet.

Tidsmålinger for å se på kjøretiden til de ulike estimatene er ikke blitt utført. I mange tidskritiske sammenhenger er det viktig å vite hvor lang tid en analyse tar. En bør derfor undersøke dette for de ulike estimatorene som har blitt benyttet under MNF-transformen.

5.2.2 Støyestimering for homogent felt

Sammenligner man akkumulerte SNR-verdier basert på homogent område med totalt utsnitt, ser en at resultatene blir en del bedre for mange av støyestimatene basert på homogent område. Dette gjelder spesielt estimatene "mean", "median" og "Gauss" både for bilde 1 og bilde 2. For de andre estimatorene er forskjellen mellom å basere støyestimatet på et homogent område eller hele utsnittet liten. Noen av disse blir bedre om en bruker homogent område, men andre blir også litt dårligere. Hvis man undersøker resultatene der hele utsnittet ble brukt under estimeringen ser en at estimatene "mean", "median" og "Gauss" er dårligere enn de seks andre. Det er også de dårligste estimatorene som hovedsakelig blir forbedret ved bruk av et homogent felt til støyestimering. Av dette kan man slutte at å benytte et homogent felt til støyestimeringen er ikke alltid nødvendig da resultatet blir omtrent det samme ved å velge en bra estimator i utgangspunktet.

Trekker her spesielt frem estimeringen med differansen til vertikal nabopiksel. Første prinsipalkomponent med denne støyestimeringen til bilde 3 er vist til venstre i figur 4.2.2.1. Sammenligner man dette bildet med tilsvarende man fikk når støyestimeringen ble utført med hele utsnittet er det et interessant resultat. Dette bildet er vist i figur 4.2.1.1, og viser at første prinsipalkomponent estimert med vertikal nabopiksel inneholder ikke den horisontale støystrifen. Den opprinnelige årsaken til at man valgte akkurat dette støyestimatet var at det skulle registrere den horisontale støystrifen som støy. Det er også tilfellet når hele utsnittet blir brukt som grunnlag for støyestimatet. Når man istedenfor benytter et homogent felt som ikke inneholder en slik støystripe blir ikke støystrifen registrert som støy, og den fremkommer i første prinsipalkomponent uavhengig av støyestimator. Dette er et tegn på at hvis man skal benytte seg av et homogent område for å utføre et bedre støyestimat, må det hyperspektrale bildet være uten sterke horisontale støykomponenter. Her skyldes altså problemet med horisontal støystripe det hyperspektrale bildet i seg selv, og ikke valget av støyestimator som det gjorde når man tok utgangspunkt i hele utsnittet for å gjøre støyestimeringen.

Resultatene i tabell 4.2.2.1 mellom de forskjellige støyestimatene er mye likere, enn resultatene for tabell 4.2.1.1. Siden tabellen viser akkumulerte verdier reduseres naturlig nok differansen mellom estimatene med økende prinsipalkomponent. Den normaliserte og akkumulerte verdien når man tar med alle prinsipalkomponentene vil alltid være lik 1 for alle estimatene. Sammenlignes SNR-verdiene for første prinsipalkomponenter mellom de forskjellige estimatene gitt i tabell 4.2.2.1, ser en at variasjonen mellom dem er små. Dette tyder på at valget av støyestimator har langt mindre å si når man beregner støyestimatet ut i fra et homogent område, istedenfor det totale utsnittet.

At resultatene er langt mer stabile for støyestimatet basert på det homogene området gjelder både på godt og vondt. Fordelen er at man kan velge en enkel støyestimator, å få et tilnærmet like bra støyestimat som enkelte av de mer avanserte estimatorene. Ulempen kommer hvis utsnittet man skal studere inneholder en horisontal støykomponent. Da vil støystreken dominere i de første prinsipalkomponentene uavhengig av hvilken støyestimator som benyttes. Slike resultater som inneholder en dominant støykomponent i de første prinsipalkomponentene vil ofte ikke være akseptabelt. Dette kommer av at støyfjerningen skjer ved utelatelse av de høyere prinsipalkomponentene, og hvis støystreken ligger blant de lavere vil den ikke bli fjernet under reverstransformasjonen.

I figur 4.2.2.1 vises andre prinsipalkomponenten basert på støyestimator "mean" med 3x3 maskestørrelse for bilde 3 og 4. Forskjellen mellom bilde 3 og 4 er kun den horisontale støystrifen i utsnittet som blir benyttet og ikke det homogene støyestimatet. Bilde 3 inneholder en horisontal støystripe, mens for bilde 4 har denne blitt interpolert vekk ved å erstatte den med snittet av naboradene. Det homogene feltet som er valgt for å gjøre støyestimatet er det samme for begge bildene. Her kan man se at utsnittet som studeres avgjør hvor vidt det dukker opp en horisontal

støykomponent eller ikke blant de første prinsipalkomponentene. Hvis en kan garantere at utsnittene man ser på ikke inneholder horisontale støystriper fungerer støyestimat basert på homogene felter bra. Er det derimot støystreker i bildene vil disse fremkomme i de første prinsipalkomponentene, og derfor heller ikke bli fjernet under støyreduksjon ved å utelate høyere prinsipalkomponenter.

Alternativt til å ta utgangspunkt i refleksjonsstandarden kunne man ha brukt et område fra det som er helt sort i bildet. Dette er også homogene områder. Ved å benytte seg av det sorte området er det imidlertid et problem som kan oppstå, og det er hvis den generelle belysningen i rommet der fotograferingen utføres øker. Da vil underlaget som hånden ligger på være med å påvirke resultatene. Det skjer i mindre grad så lenge man benytter seg av refleksjonsstandarden.

5.2.3 Støyestimering i fourierrommet

Tabell 4.2.3.1 viser akkumulerte og normaliserte SNR-verdier for de 5 første MNF-komponentene til støyestimatene utført i fourierrommet. SNR-verdiene øker en del med økende maskestørrelse for støyestimatene basert på absoluttverdi. Estimaten basert på fase er så godt som uendret ved stigende maskestørrelse. Dette gjelder både for bilde 1 og 2. Ser man på differanser mellom de ulike maskestørrelsene er den mye mindre enn forskjellen mellom valget å benytte absoluttverdi eller fase til støyestimeringen. Den dominerende forskjellen i tabell 4.2.3.1 er mellom støyestimatene utført med absoluttverdi og fase. Dette kan komme av at å bruke differansen til absoluttverdien mellom nærliggende frekvenskomponenter vil avhenge av intensiteten eller styrken til støyen, men differansen til fasen for tilsvarende frekvenskomponenter ikke har den samme avhengigheten. Absoluttverdien til et komplekst tall er det direkte målet på intensiteten, og differansen mellom absoluttverdiene for to nærliggende frekvenser vil si om den ene er sterkere enn den andre. Bruk av fasen på samme måte gir en annen betydning. Fasen vil alltid ligge mellom -180° og 180° uavhengig av intensiteten til frekvenskomponenten. Dette betyr at sterke og svake komponenter blir behandlet helt likt. Bruk av differansen mellom absoluttverdiene kan ligne på de tradisjonelle støyestimatene når man ser på differansen mellom intensiteten for nabopiksler i det spatiale planet. Begge disse benytter et mål på lokal differanse i intensitet, selv om det ene gjelder pikselverdier og det andre frekvenskomponenter. Differansen mellom nærliggende faser for frekvenskomponentene er noe helt annet, siden alle intensitetsvariasjoner blir behandlet likt. Hvis fasen mellom to frekvenskomponenter er liten vil de stykkevis svinge tilnærmet i takt. I det motsatte tilfellet er den stor og de vil svinge i utakt, og dette vil da regnes som støy. For det tradisjonelle tilfellet er det en god idé å benytte differansen mellom nærliggende piksler til støyestimering. Bruk av differansen mellom absoluttverdier for nære frekvenskomponenter gir imidlertid ikke like godt resultat. Dette kan komme av at intensiteten til frekvensene ikke sier noe om hvor de vil inntreffe. Denne informasjonen har man i det spatiale rommet da dette er nærliggende naboer. Bruk av differansen mellom fasene i fourierrommet gir et svært bra resultat, og her vet man at frekvenskomponentene tidvis vil svinge i takt. Dette kan forklare hvorfor bruk av fasen fungerer bra til støyestimering, mens intensiteten til frekvenskomponentene ikke gjør det.

For bilde 1 til 6 er støyestimering basert på absoluttverdien jevnt over dårlig. Resultatene for bilde 3 til 6 ligner stort sett på det man ser for de to første bildene, og støyestimatene basert på fase er fremdeles veldig gode. For de tre siste bildene 7, 8 og 9 endres resultatene mye for støyestimatet basert på fase. Den nederste grafen i figur 4.2.3.1 viser akkumulerte SNR-verdier når støyestimatet fase er benyttet med 3×3 maskestørrelse for bilde 7. Hvis man sammenligner denne grafen med den øverst til høyre i figur 4.2.3.1 som viser tilsvarende støyestimat for bilde 1, ser en at de akkumulerte SNR-verdiene er betydelig lavere i dette tilfellet. Her har første prinsipalkomponent en SNR-verdi på 0,70. For bilde 8 og 9 er den på henholdsvis 0,79 og 0,80. Disse to resultatene er riktignok noe bedre enn det var for bilde 7, men det er allikevel langt unna hva det var på de 6 første bildene. Da lå den stabilt rundt 0,99. Den viktigste forskjellen mellom de 6 første og de 3 siste bildene er vist i appendiks 8.4. Fra appendikset har man at de 6 første hyperspektrale bildene er tatt under hvitt lys, men for de tre siste er lyskilden endret til en 355nm YAG-laser. Når en ser på SNR-verdiene for første prinsipalkomponent til de tre siste bildene har også bilde 8 og 9 litt høyere verdi enn bilde 7. Dette kommer også mest sannsynlig av forskjellen i lysforholdene mellom dem. Bilde 8 er tatt med $3 \times$ midling. Det betyr at hver rad blir skannet inn 3 ganger, og middelverdien av disse 3 gangene blir så benyttet. Dette vil naturlig nok gi en form for støyreduksjon i forhold til å bare skanne inn radene én gang, siden endringer blir glattet ut når man benytter gjennomsnittet. Bilde 9 skanner i likhet med bilde 7 inn radene bare én gang, men for bilde 9 har integrasjonstiden blitt doblet i forhold til bilde 7 og 8. Dette betyr at sensorarrayet registrer lys dobbelt så lenge. Økning av integrasjonstiden vil gi en utsømnende effekt på raske fluktuasjoner, som igjen betyr at det gir en form for støyreduksjon. Bilde 7 benytter ingen av disse to teknikken, og en må derfor også forvente at dette bildet har mest støy blant de som er tatt ved belysning av YAG-laser.

Støyestimering av hyperspektrale bilder i fourierrommet er en ny variant, og man bør derfor gjennomføre flere tester for å se om dette er et godt supplement til eksisterende metoder. En bør også legge merke til at for bildene med annet lys en hvitt lys så fungerer denne metoden langt dårligere. Hvorvidt dette skyldes lyskilden bildene er tatt med, eller mengden støy som finnes i bildene er uvisst. Dette bør derfor også undersøkes nærmere.

Det skal understrekes at det er vanskelig å se de fulle konsekvensene av å velge fourierrommet til støyestimering. Dette kommer av at bare ved å endre en piksel i det spatiale rommet kan en potensielt endre alle verdier i fourierrommet. Med andre ord er det både vanskelig å lokalisere endringene i fourierrommet når det påtrykkes støy i det spatiale rommet, samt se konsekvensene av det som har blitt forandret i fourierrommet. Dette er imidlertid lettere når man benytter wavelet-transform. Da beholder man til en viss grad den spatiale lokaliseringen etter transformasjonen, og det blir også derfor lettere å se konsekvensene av støy i det spatiale rommet.

Støyestimering i fourierrommet er derfor høyst eksperimentelt og det bør undersøkes videre før eventuell bruk i endelige systemer.

5.2.4 Antall band under tilbaketransformering

I figur 4.2.4.1 vises spekteret til en piksel uten støyreduksjon øverst til venstre, og med støyreduksjon som en følge av at det er blitt bevart henholdsvis 40 og 2 prinsipalkomponenter under tilbaketransformasjonen. Spekteret man oppnår hvis man tar med alle prinsipalkomponentene under tilbaketransformasjonen blir identisk med spekteret uten støyreduksjon. Dette følger direkte av ligning 3.4.3.2. En kan se at grafen øverst til høyre, der det er tatt med de 40 første prinsipalkomponentene, er mer glatt enn det opprinnelige spekteret. Enda jevnere er grafen nederst der det bare ble bevart de to første prinsipalkomponentene. Dette er et viktig resultat som følger av å bruke MNF til støyreduksjon. Ved å begrense antallet prinsipalkomponenter når man tilbaketransformerer datasettet fra MNF-rommet vil spekteret for pikslene bli glattet ut.

Figur 4.2.4.2 viser akkumulerte SNR-verdier for prinsipalkomponentene. En kan se at de blå grafene stiger brått for de første prinsipalkomponentene, og langt svakere for høyere prinsipalkomponenter. Dette viser at signal-støy-forholdet er langt bedre for de første prinsipalkomponentene, enn det er for de etter den kraftige knekken. Den røde streken viser hvor grenseverdien for akkumulerte SNR-verdier er plassert, med tanke på hvilke prinsipalkomponenter som blir tatt med under tilbaketransformasjonen. Dette betyr at krysspunktet der den blå kurven skjærer den røde streken angir grensen for antallet prinsipalkomponenter som skal med under reverstransformasjonen. Til venstre er den plassert slik at de 2 første prinsipalkomponentene blir bevart og resten utelatt. I grafen til høyre viser den grensen når de 40 første komponentene blir tatt med. Dette ser man hvis det trekkes en vertikal strek fra krysningspunktet mellom den blå kurven og den røde streke ned til x-aksen. Verdien på x-aksen viser da antallet prinsipalkomponenter som blir tatt med. De blå grafene er normalisert til 1. Dette betyr at hvis man legger grensen for hvilke prinsipalkomponenter som skal med under reverstransformasjonen til 1 vil alle komponentene bli bevart. Hvis dette utføres og man trekker en vertikal strek ned til x-aksen vil den krysse aksen ved 81 prinsipalkomponenter.

Utsnittet av det hyperspektrale bildet som viser den faktiske virkningen av å begrense antallet prinsipalkomponenter under tilbaketransformasjonen fra MNF-rommet er vist i figur 4.2.4.3. Øverst til venstre er utsnittet av spektralband nummer 10 uten støyreduksjon. Til høyre vises resultatet der det er beholdt de 40 første prinsipalkomponentene. Nederst er resultatet man får når kun de 2 første prinsipalkomponentene blir tatt med under tilbaketransformasjonen. Bildet som har begrenset antallet prinsipalkomponenter til 40 under tilbaketransformasjonen inneholder klart mindre støy enn det opprinnelige utsnittet. Selv om dette utsnittet har redusert mengden støy betraktelig fra utgangspunktet, er det allikevel mye støy igjen. For resultatet vist nederst der bare de 2 første prinsipalkomponentene ble bevart har støynivået blitt redusert ytterligere. Dette er for såvidt bra, men her kan man se at prinsippet med å fjerne støy ved å redusere antallet prinsipalkomponenter har sine ulemper. Når man utradrer prinsipalkomponenter for å fjerne støy, vil man også miste et eventuelt signal som finnes blant støyen. Bildet der bare 2 prinsipalkomponenter er bevart virker langt lysere enn det originale utsnittet. Det kan også se ut som om det har blitt lagt til informasjon som ikke var der i utgangspunktet.

Det overordnede målet med å bestemme hvilke prinsipalkomponenter som skal bli bevart under inverstransformasjonen, er å bevare mest mulig av signalet samtidig som støyen blir redusert i størst mulig grad. Dette er ikke et enkelt mål å nå siden både støyen og signalet blir redusert for hver prinsipalkomponent man tar bort. Det ene ytterpunktet for en slik utvelgelse vil være å bevare samtlige av prinsipalkomponentene, og det andre er å kun ta med én komponent. Hvis alle prinsipalkomponentene blir bevart mister man ingen ting av det opprinnelige signalet. Problemet er at støyen blir heller ikke redusert. Tar man bare vare på én prinsipalkomponent under inverstransformasjonen vil i praksis alle spektralbandene fremstå som like. Dette kommer av at man

sitter igjen med én variabel å spille på, og da vil bare totalintensiteten mellom båndene kunne varieres.

Målet blir da å finne en akseptabel grense for hvor mye av signalet en er villig til å ofre, eventuelt mengden støy man vil akseptere. Dette skal gjøres uten bruk av visuell inspeksjon. Disse kriteriene er satt med tanken på to ting. For det første vil man prøve å automatisere prosessen mest mulig. Det andre er at utvelgelsen av grensen mellom signal og støy skal være objektiv. Begge disse kriteriene krever at en unngår det menneskelige element når grensesettingen skjer.

Her velges det å bestemme denne overgangen ved at man beholder en gitt mengde av det opprinnelige signalet. Tabell 4.2.4.1 viser hvor mange prinsipalkomponenter man må ta med for at en gitt prosent av de hyperspektrale bildene skal bli bevart. Hvor en plasserer den prosentvise grensen for antallet prinsipalkomponenter som skal bli bevart kan variere mellom ulike bildetyper. Appendiks 8.4 gir en kort beskrivelse av de forskjellige bildene. Bilde 1, 2, 4 og 5 viser oversiden av hender. For disse kan man tenke seg at det settes en grense. I overgangen mellom 99,25% og 99,50% stiger antallet prinsipalkomponenter bilde 1 trenger under reverstransformasjonen betraktelig. Ved å legge grensen i dette intervallet begrenses støyen som blir værende etter støyreduksjon. Bilde 3 er nesten identisk til bilde 4 og viser også oversiden av en hånd. Forskjellen er at bilde 3 inneholder en sterk horisontal støystripe. Ved å kreve at 99,25% til 99,50% prosent av signalet skal bli bevart vil dette føre til at man må ta med 36 til 46 prinsipalkomponenter under reverstransformasjonen for bilde 3. Dette kravet vil bevare mye av støyen, og for bilder som inneholder en slik dominant støystripe må man muligens sette grensen noe lavere for å slippe en del støy. Bildene 6, 7, 8 og 9 viser det nøyaktig samme utsnittet av en blodåre. Forskjellen mellom dem er lysforholdene de er tatt ved. Bilde 6 er tatt under belysning av hvitt lys, mens for de tre siste er det benyttet en ultrafiolett lyskilde på 355nm. For å få med mest mulig av signalet og minst mulig støy bør grensen legges mellom 99,00% og 99,25%.

Det vil være mange strategier for å bestemme hvor grensen mellom prinsipalkomponentene som inneholder hovedsakelig signal eller støy skal plasseres. En annen mulighet enn den som er valgt er å legge grensen tettest mulig opp til knekken i grafene vist i appendiks 8.5. Dette vil føre til at man begrenser mengden støy som blir med, siden det meste av støyen kommer etter knekken. Ulempen er at man ikke kan garantere for at en gitt prosent av det opprinnelige signalet blir bevart.

I figur 4.2.4.3 er det støyestimator "Horisontal + Vertikal" som er blitt benyttet på bilde 2 når man har beholdt henholdsvis 2 og 40 prinsipalkomponenter. Tabell 4.2.1.1 gir at denne estimatoren beholder 96% av signalet med én prinsipalkomponent og 99% med to. Hvis en heller hadde utført støyestimeringen i fourierrommet med "Fase 3x3" ville man ha beholdt litt mer av signalet. Fra tabell 4.2.3.1 får man at dette tilsvarer 99,6% av signalet med én prinsipalkomponent og 99,7% med to. Av disse resultatene tyder det på at støyestimering i fourierrommet med "Fase 3x3" vil gi best resultat. Dette støyestimatet ga imidlertid langt dårligere resultater enn "Horisontal + Vertikal" for de mer støyfulle bildene 7, 8 og 9. Det virker som støyestimering i fourierrommet med "Fase 3x3" fungerer bra på bilder med begrenset støy. Hvis man utfører to runder med støydemping ved bruk av MNF i kaskade, kan en kanskje oppnå et stabilt godt resultat. Under den første runden utføres støyestimering ved "Horisontal + Vertikal" i det spatiale rommet. Da skal bare et lavere antall prinsipalkomponenter fjernes for å få bort en god del støy, samtidig som man beholder det meste av signalet. Deretter kan en utføre runde nummer to med estimatoren "Fase 3x3" til støyestimering i fourierrommet. Her er tanken å fjerne flere prinsipalkomponenter enn ved første runde, slik at ytterligere støy blir utelatt under reverstransformasjonen. Forhåpentligvis blir også en større andel av signalet bevart. Dette er også noe man bør undersøke videre.

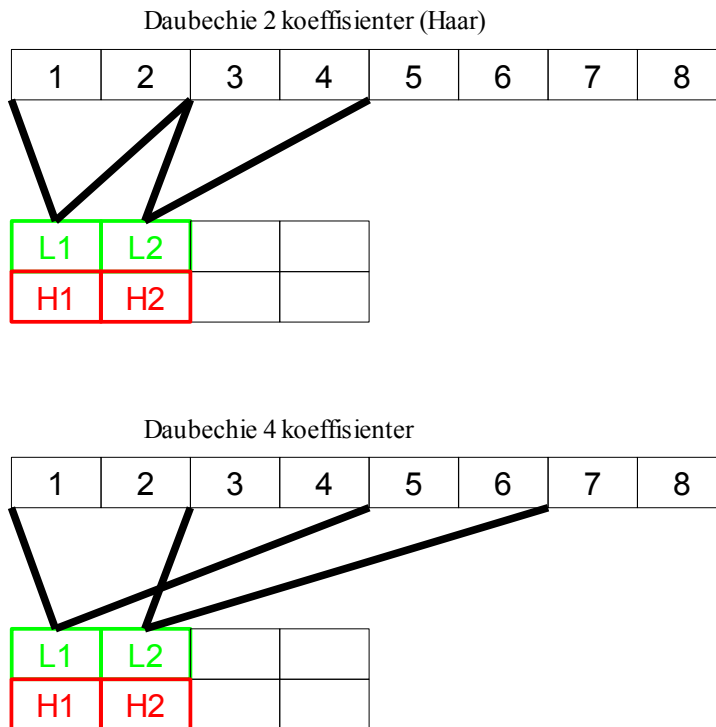
5.3 Støyreduksjon ved ulik belysning

Her vil det bli diskutert påvirkningen valg av lyskilde har for resultatene. Bildene 6, 7, 8 og 9 er av det samme utsnittet. Det eneste som varierer mellom disse fire bildene er belysningen. Bilde 6 ble tatt med hvitt lyskilde, mens de tre siste ble tatt under belysning av 355nm laserlys. Forskjellen mellom 7, 8 og 9 er midling og integrasjonstid. Dette er vist i appendiks 8.4. Midling vil si at en skanner samme linje flere ganger og deler på antallet for å benytte gjennomsnittet. Økning av integrasjonstiden er at bildebrikken blir belyst i et tilsvarende lengre tidsintervall.

Kapittel 4.3 inneholder bare et utdrag av alle resultatene. De samlede resultatene er vist i appendiks. Figur 4.3.1 og 4.3.2 viser hvordan spekteret til en enkelt rad for henholdsvis bilde 6 og 8 ser ut. Det er den samme raden som er avbildet i begge figurene. De øverste grafene i figur 4.3.1 og 4.3.2 viser originale utsnitt uten noen form for støydemping. Bilde 6 har lav intensitet for korte bølgelengder, og høy intensitet for lange. For bilde 8 er det motsatt. Hvis man sammenligner den udempede grafen av bilde 8 med tilsvarende for bilde 7 og 9 som er vist i appendiks 8.8 ser man at de ligner veldig. Dette betyr at valget av lyskilde eller lysforhold har mye å si for resultatene. Hvitt lys gir høy intensitet for lange bølgelengder, og 355nm laserlys har høy intensitet for korte bølgelengder. Merk at belysning fra bare en bølgelengde på 355nm gir opphav til mange andre bølgelengder. Dette kommer mest sannsynlig som en følge av fluorescens.

Den nederste grafen i figur 4.3.1 viser hvordan spekteret til den samme raden blir seende ut etter støydemping når MNF-algoritmen er basert på støyestimatoren "Horisontal + Vertikal". Her kan en tydelig se en glatting av grafen langs med retningen til bølgelengden. I appendiks 8.8 finner man tilsvarende graf for støyreduksjon basert på PCA og wavelet, samt andre støyestimeringer for MNF. Med unntak av wavelet er resultatene for de forskjellige variantene av MNF og PCA ganske like.

I appendikset vises raden etter at henholdsvis 75% og 25% av verdiene har blitt satt til 0 under wavelet-transformen. Her er det Haar-transformen som er valgt. Det finnes mange forskjellige wavelet-transformer, og forhåpentligvis vil noen av de andre gi bedre resultater enn Haar-transformen. Haar-transformen er lik den enkleste av Daubechies-transformene og benytter bare 2 koeffisienter. Daubechies-transformene øker med to koeffisienter mellom hver, så den neste har 4 og så følger en med 6. En viktig forskjell mellom Haar-transformen og de andre mer avanserte Daubechies-transformene er mangelen på overlapping for Haar. Dette illustreres i figur 5.3.1. For Haar eller D2 ser en to og to celler brukes for å danne et lavpass- og høypass-par. For eksempel brukes cellene 1 og 2 for å danne L1 og H1. Den mer avanserte D4 vist nederst i figur 5.3.1 benytter cellene 1, 2, 3 og 4 for å danne L1 og H1. Her blir cellene 3 og 4 benyttet nok en gang når L2 og H2 skal beregnes. Dette vil si at det er en overlapping ved bruk av celleverdier når neste iterasjon med verdier blir beregnet. Siden Haar ikke har denne overlappingen vil det kunne bli en veldig hard overgang mellom verdiene L1 og L2. Dette skjer hvis de 2 første cellene er i snitt høyere enn de 2 neste. Lar man det gå flere iterasjonene blir dette synligere da flere og flere celler blir med og trekker gjennomsnittet i en retning. Det er mest sannsynlig dette som skaper den veldige blokk-effekten en kan se i grafen for bilde 6 etter wavelet-transformering med 75% reduksjon i appendiks 8.8. Merk også at de høyere intensitetsverdiene ikke har det samme problemet. Intensitetsverdiene for de kortere bølgelengdene er langt flatere enn de er for de lange bølgelengdene. Dette vil si at endringen mellom korte bølgelengder er liten i forhold til endringen mellom lange bølgelengder. Siden det er de 75% minste verdiene etter forover wavelet-transformering som blir satt til 0, vil stort sett endringene mellom korte bølgelengder bli utvisket og ikke mellom lange. Grafen av bilde 6 med 25% reduksjon etter forover wavelet-transform er ikke like preget av blokk-effekten. Her er ikke reduksjonen av verdier like drastisk, slik at gjennomsnittsnivåene blir bevart og det er de små lite synlige forandringene som blir glattet ut.



Figur 5.3.1: Bruk av celler for å danne neste nivå med lavpass- og høypass-verdier

Siden man starter med 80 spektralbånd når wavelet-transformen begynner kan man maksimalt utføre 4 iterasjoner. Hvis utgangspunktet hadde vært 64 eller 128 spektralbånd kunne man ha fått til henholdsvis 6 og 7 iterasjoner. Hadde man fremdeles presset på med mye reduksjon ville det gitt større blokker til blokk-effekten. Etter 4 iterasjoner er det 16 celler som gir det totale opphavet for en lavpass-verdi. Dette betyr at 16 er maksimal lengde på blokkene man ser i grafen til bilde 6 med 75% reduksjon under wavelet-transform i appendiks 8.8. De nest lengste blokkene vil ha en lengde på 8 som oppstår etter 3 iterasjoner.

Den støyfjilte grafen vist nederst i figur 4.3.2 er mer ruglete enn grafen man ser etter støyfjerning i figur 4.3.1. Det første en skal merke seg er at to forskjellige algoritmer for støyfjerning har blitt benyttet, og en må derfor forvente litt ulike resultat. I appendiks 8.8 finner man imidlertid et komplett sett av grafer, så en kan sammenligne grafene mellom de ulike bildene når tilsvarende støyfjerningsalgoritme har blitt benyttet. Sammenligner en tilsvarende støyfjerningsalgoritme for PCA og MNF blant de forskjellige bildene med hverandre er det en tydelig tendens. De støydempede grafene av bilde 6 er langt glattere enn tilsvarende grafer for de andre bildene. Dette kan komme som en følge av lyskilden man har brukt. Bilde 6 er tatt under hvitt lys, mens de tre andre er belyst av en 355nm laserkilde.

Det er også en annen vesentlig forskjellen mellom bildene under støyfjerningsprosessene. Antallet prinsipalkomponenter som blir tatt med under reverstransformasjonen endres fra bilde til bilde. Den faktiske støydempningen skjer ved at man begrenser antallet prinsipalkomponenter under tilbaketransformasjonen. Dette gjelder både for PCA- og MNF-algoritmen. Hvor mange prinsipalkomponenter som blir bevart for de forskjellige bildene er opplyst i kapittel 4.3. Merk at like mange prinsipalkomponenter blir bevart for ulike algoritmer når man sammenligner innenfor et bilde. Dette gjør det mulig å se på effekten av de forskjellige algoritmene i forhold til hverandre. Hvis man derimot sammenligner resultatene fra samme algoritme mellom ulike bilder må man imidlertid ta høyde for at antallet prinsipalkomponenter som har blitt bevart vil kunne variere.

For bilde 6 blir 7 prinsipalkomponenter bevart under reverstransformasjonen. Bilde 7 får beholde hele 33 prinsipalkomponenter. Siden bilde 7 beholder langt flere prinsipalkomponenter vil også en større andel av støyen bli bevart, istedenfor å bli utelatt under reverstransformasjonen.

Mengden signal som blir bevart er imidlertid den samme, siden antallet prinsipalkomponenter er valgt med tanke på å beholde en gitt prosent av signalet for de forskjellige bildene. De to siste bildene 8 og 9 tar begge med 14 prinsipalkomponenter under reverstransformasjonen. Dette er også langt flere enn bilde 6. Som en følge av dette kan en også forvente at de støydempede resultatene for bilde 6 er glattere enn for de tre siste bildene

Grensesettingen for antallet prinsipalkomponenter som skal bevares under reverstransformasjonen til PCA og MNF kan bestemmes på forskjellige måter. For å lette sammenligningen mellom bildene etter de støyreducerende algoritmene ville det ha vært en fordel om man alltid hadde bevart like mange prinsipalkomponenter. Valget er imidlertid gjort slik at en bestandig skal sitte igjen med en bestemt andel av det opprinnelige signalet etter støyreduksjonen. Dette gjøres riktignok på bekostning av ønsket om å redusere støy. Utfordringen ligger i å bevare signalet samtidig som støyen fjernes. Hvis man øker antallet prinsipalkomponenter under reverstransformasjonen for at mer av signalet skal bli bevart, vil man også beholde en større andel av støyen. Den motsatte tilnærmingen til problemet vil være at en alltid skal redusere støyen med en gitt prosent. Da ville man oppnå en like stor prosentvis støydempende effekt mellom bildene. Her blir ulempen at man ikke kan garantere for at en bestemt del av det opprinnelige signalet skal være bevart.

Etter at man fikk resultatene for signal-støy-forholdet til prinsipalkomponentene er det en tredje mulighet som dukker opp. Første del av appendiks 8.5 viser akkumulert signal-støy-forhold for alle bildene når samme algoritme har blitt brukt. Siste del av appendiks 8.5 viser hvordan akkumulert SNR blir seende ut når man benytter ulike algoritmer på det samme bildet. Alle grafene i disse to appendiksene inneholder en kraftig knekk. Før knekken stiger SNR kraftig som betyr at hoveddelen av signalet er samlet for de første prinsipalkomponentene. Etter knekken er kurven til SNR mye slakkere, og en langt mindre del av signalet vil befinne seg blant prinsipalkomponentene bak dette punktet. I motsetning til signal vil det derimot være mye støy etter knekken. Å bestemme antallet prinsipalkomponenter man skal bevare under reverstransformasjonen ut i fra plasseringen til denne knekken virker derfor som en mulighet. Da vil en beholde mye av signalet samtidig som man får en god støydempende effekt. Dette kan virke som en mellomting mellom å bevare en gitt prosent av signalet, eventuelt å fjerne en bestemt andel av støyen. Dette er derfor noe en bør se på videre.

Utgangspunktet man starter med før eventuelle støydempende algoritmer kjøres kan også ha en del å si for resultatene. Her ser en også at spekteret for bilde 6 virker glattere enn de andre udempede spektrene for bildene 7, 8 og 9. Spektrene for de tre siste bildene ligner veldig, men også her er det noen viktige forskjeller. Hvis man sammenligner de udempede grafen av bildene 7 til 9, ser en at grafen for bilde 7 er mer ruglete eller inneholder mer støy enn de to andre. Dette kan komme av at bildene 8 og 9 har blitt tatt ved bruk av midling eller økt integrasjonstid, mens bilde 7 ble tatt uten slike virkemidler.

Til nå har bare virkningen av algoritmene blitt diskutert i spektral retning for de hyperspektrale bildene. Figur 4.3.3 og 4.3.4 viser hva som skjer i det romlige planet når man ser på et enkelt spektralbånd. I disse to figurene er det spektralbånd 70 med bølgelengde på 662nm som blir trukket frem. Hvis en kaster et blikk tilbake på figur 4.3.1 og 4.3.2 ser en at denne bølgelengden har høy intensitet for bilde 6, og lav intensitet for de 3 andre bildene. Appendiks 8.10 inneholder en større samling grafer for de ulike bildene og algoritmene i det romlige planet med bølgelengde 662nm. Spektralbånd 10 med bølgelengde 444nm har motsatt intensitetsmønster mellom bildene i forhold til spektralbånd 70. For spektralbånd 10 har bilde 6 lav intensitet mens de tre andre bildene har høy intensitet. Grafene for spektralbånd 10 er vist i appendiks 8.9.

Den nederste grafen i figur 4.3.3 viser hvordan et høyintensitetsbånd blir seende ut etter støyreduksjon for bilde 6. Her ser man en klar utglatting av små variasjoner samtidig som tydeligere topper blir bevart. Dette tyder på at man får både en støydempende effekt og man klarer å beholde viktige detaljer i signalet. Både PCA og MNF-algoritmene får til denne støyreducerende effekten.

I figur 4.3.4 er utgangspunktet før støyreduksjon mye mer støyrikt for bilde 8. Dette skyldes at spektralbånd 70 er et lavintensitetsbånd for bilde 8, mens det er et høyintensitetsbånd for bilde 6. Det eneste signalet en kan antyde ligger øverst til venstre i grafen. Her ser det ut som om det er en liten forhøyning. Ser man på det støydempede resultatet illustrert i den nederste grafen kommer det langt tydeligere frem hva som er avbildet. Forhøyningen øverst til venstre blir også klart tydeligere i dette støyreduerte utsnittet. Det er PCA-algoritmen som er benyttet for å frembringe den støydempede grafen nederst i figur 4.3.4. MNF-algoritmen basert på forskjellige støyestimat gir også bra støyreduksjon, men PCA-algoritmen er klart best. Dette er lett å se ved å sammenligne de støyreduerte resultatene vist i appendiks 8.10. I dette appendikset ser en også at forskjellen mellom PCA og MNF-algoritmene er langt mindre for utsnittet med høy intensitet vist i figur 4.3.3. Der spiller det mindre rolle om man velger PCA eller MNF-algoritmen. For utsnitt med lav intensitet som vist i figur 4.3.4, blir det derimot klart at PCA er å foretrekke til støyreduksjon.

I høstprosjektet kom det frem at PCA ikke egnet seg til reduksjon av dimensjonaliteten for et hyperspektralt bilde, da algoritmen ikke klarer å ordne prinsipalkomponentene etter synkende signal-støy-forhold. MNF egner seg derimot til dette som man ser i kapittel 4.2. Her er det helt tydelig at prinsipalkomponentene blir ordnet etter signal-støy-forholdet. Når det nå kommer til støyreduksjon viser det seg at PCA virker bra til støyfjerning for spektralbånd med mye støy. Dette kan komme av mange forskjellige grunner. Typen støy som skal bli filtrert bort vil trolig ha en innvirkning på resultatene. Enkelte typer støy kan være lettere å filtrere ut for en algoritme i forhold til en annen. Det er også en mulighet at PCA gir bedre resultater enn det grunnlaget skulle tilsi. Med dette så menes det å stille spørsmål ved hvor godt resultat kan man egentlig forvente. Her siktes det spesielt til å kopiere signal fra spektralbånd med sterkt signal til spektralbånd med svakt signal. Dette skjer under reverstransformasjonen når man begrenser antallet prinsipalkomponenter som blir bevart. Spektralbånd som i utgangspunktet er konstruert hovedsakelig av fjernede prinsipalkomponentene, vil nå heller bli basert på de gjenværende prinsipalkomponentene. Støyen forsvinner ved å ta bort de høye prinsipalkomponentene, men det vil også et eventuelt svakt signal som eksisterer blant dem. Ut av dette kan man få et klart og tydelig bilde basert på prinsipalkomponenter med sterkt signal, men trenger nødvendigvis ikke å gjengi virkeligheten. Dette illustreres hvis man bare tar med én prinsipalkomponent under reverstransformasjonen. Da vil alle spektralbåndene se like ut, og man har egentlig byttet ut både støyen og det egentlige signalet med det som vises i første prinsipalkomponent. Denne effekten skjer både for PCA og MNF.

Hvis man sammenligner det støyreduerte utsnittet av bilde 8 nederst i figur 4.3.4 med det støyreduerte av bilde 6 i figur 4.3.3 kan en kjenne igjen et tydelig element. Det ser ut som en forsinking eller kanal igjennom utsnittene. Kanalen slynger seg vertikalt igjennom bildene. Årsaken til at en kan gjenkjenne enkelte elementer samtidig som mye er endret mellom disse to bildene har en sammenheng med belysningen bildene ble tatt under. Bilde 6 ble tatt under hvitt lys, mens bilde 7, 8 og 9 ble belyst med en 355nm laserkilde.

Appendiks 8.9 viser det samme utsnittet man ser i figur 4.3.3 og 4.3.4, men her vises det for spektralbånd 10 med bølgelengde 444nm. For dette spektralbåndet har bilde 6 lav intensitet og de tre siste bildene høy intensitet. En kan gjenkjenne at det er likt utsnitt som tidligere på den samme vertikale kanalen som strekker seg igjennom grafene. Resultatene av støyfjerning ligner for det meste det man fant for spektralbånd 70.

En gjenganger for alle tilfellene man har sett på er at Haar-wavelet-transformen fungerer jevnt over dårlig til støyreduksjon. Dette er veldig tydelig i grafen med 75% reduksjon under wavelet-transformen til bilde 6 i appendiks 8.9. Her ser en at signalet har blitt redusert helt til 0 for store deler av utsnittet. Selv om Haar-transformen ikke egner seg til støydempning i dette tilfellet, kan den kanskje brukes til objektgjenkjenning. Det skal også påpekes at det alltid er muligheter for feil i koden som kan være med å ødelegge resultatene. Haar-transformen er en relativt enkel wavelet-transform så dette virker dermed ikke fullt så sannsynlig.

Figurene 4.3.3 og 4.3.4 er 3D fremstillinger av 2D bilder. Dette er gjort for å lette sammenligningen mellom de originale utsnittene og de støyreduerte resultatene. Alternativt kunne man bare ha vist før og etter resultatene som vanlige 2D bilder. Da vil det imidlertid være vanskelig å se små variasjoner i intensitet mellom pikslene. Når bildene illustreres som 3D med intensiteten langs med z-aksen vil selv små variasjoner bli synlige.

Appendiksene 8.11, 8.12 og 8.13 viser fordelingen til støyen som har blitt filtrert bort av PCA, MNF og Haar-wavelet-algoritmen. Figur 4.3.1 og 4.3.2 viser effekten av støydemping i spektral retning for de hyperspektrale bildene. Støyfordelingen i spektral retning er vist i appendiks 8.11. Det man ser er fordelingene til differansene mellom de støydempede resultatene og originale utsnitt. Merk at fordelingene bare gjelder for den ene utplukkede raden vist i figur 4.3.1 og 4.3.2. En ser med andre ord ikke fordelingsfunksjonene for de totale hyperspektrale bildene. Sammenligner en MNF-algortimene for et og et bilde ligner de mye på hverandre. I kapittel 2.1 er det vist 6 forskjellige fordelingsfunksjoner. En ser at den gaussiske fordelingen ligner mest på fordelingene til MNF-algortimene. Histogrammene for PCA ligner også mest på gaussisk fordeling. Her er det imidlertid en kraftigere topp midt i kurvene for bilde 7, 8 og 9. Siden bilde 6 ikke har det og de tre siste bildene er tilnærmet like hverandre kan dette være tilfeldig for raden som ble valgt ut. En mulighet er også at lysforholdene gjør denne forskjellen. Bilde 6 ble tatt under hvitt lys, mens de 3 siste bildene ble belyst av en 355nm laser. Toppen i bilde 7 er kraftigere enn for bilde 8 og 9. Dette kan komme av mengden støy i bildene da de ble tatt. Bilde 7 ble tatt uten teknikker for å dempe støy, mens for bilde 8 ble det brukt midling og for bilde 9 hadde man økt integrasjonstid.

Tidligere har man sett at Haar-transformen fungerte dårlig til støyreduksjon. Nå ser en itillegg at fordelingene til det som har blitt fjernet under Haar-transformen ikke ligner på resultatene fra PCA og MNF. Dette er også med på å bekrefte Haar-transformens uegnethet til støyreduksjon, siden fordelingen avviker så mye fra andre algoritmer som gir bra støydemping.

Hvor vidt støyen i spektral retning for hyperspektrale bilder skal være gaussisk fordelt er vanskelig å bestemme. Fordelingene man får fra PCA og MNF ser gaussiske ut, og dette er med på å støtte opp om denne antagelsen. Det skal understrekes at fordelingene er basert på det som har blitt fjernet av PCA og MNF. Dette trenger nødvendigvis ikke bare være støy, men kan også være en del av signalet. For modellering av støy i spektral retning til hyperspektrale bilder har både Ientilucci^[2] og Manolakis^[3] benyttet gaussisk fordeling. Dette viser at flere har kommet til slutningen om at støyen i spektral retning kan være gaussisk. Flere undersøkelser kan derfor være lurt for å avgjøre dette. En test kan være å kjøre PCA og MNF på et konstruert hyperspektralt bilde med kjent gaussisk støy. Hvis disse algortimene klarer å skille ut støyen fra et konstruert bilde vil de også mest sannsynlig klare det for et ekte bilde.

Fra figur 4.3.3 og 4.3.4 så man støyreduksjon i det romlige planet for de hyperspektrale bildene. Fordelingene til det som har blitt fjernet under disse prosessene blir vist i appendiks 8.12 og 8.13. Fremdeles ser fordelingene gaussiske ut både for PCA og MNF. Variasjonen mellom formen til histogrammene i spektralbånd 10 og 70 er også liten. Spredningen er derimot langt større for båndene med lav intensitet enn de med høy intensitet. Bilde 6 har lav intensitet for spektralbånd 10 og høy intensitet for spektralbånd 70. Utstrekningen til histogrammene for dette bilde er typisk på -0,2 til 0,2 for spektralbånd 10 og -0,02 til 0,02 for spektralbånd 70. De tre siste bildene har motsatt intensitetsmønster. Spektralbånd 10 har høy intensitet og spektralbånd 70 har lav intensitet. Utstrekningen er nå typisk -0,05 til 0,05 for spektralbånd 10 og -0,5 til 0,5 for spektralbånd 70. Dette kommer trolig som en følge av at signalet drukner langt mer i støy ved svak intensitet enn ved sterk.

6 Konklusjon

Deteksjon og fjerning av horisontal støystripe før MNF-algoritmen utføres, åpner opp for flere muligheter. Et viktig resultat er at man står mye friere ved valg av type støyestimator. Dette vil være en fordel hvis man har en estimator som gir generelt gode resultater, men som ikke klarer å skille ut en slik støystripe. En annen mulighet er bruk av et homogent felt til støyestimeringen. Hvis dette benyttes og det er støystriper i bildet, vil ikke disse bli skilt ut som støy. Er derimot støystripen fjernet ved interpolasjon på forhånd blir ikke dette noe problem.

Estimering av kovariansmatrisen til støyen i MNF-algoritmen kan utføres på mange forskjellige måter. Ved bruk av estimatoren "vertikal" listet i tabell 4.2.1.1, er det mulig å skille ut horisontale støystriper som støy under MNF-transformen. Andre estimatorene som "horisontal" klarer derimot ikke dette. Ved estimering av kovariansmatrisen til støyen er det de enklere estimatorene som gir best signal-støy-forhold. Det lønner seg derfor å velge estimatorene som for eksempel "vertikal" eller "horisontal + vertikal", siden de både gir bedre signal-støy-forhold og er lettere å implementere.

Å utføre støyestimering i fourierrommet er en lite prøvd teknikk. Bruk av fouriertransform gir tilgang til nærliggende frekvenser istedenfor nabopiksler. Hvis estimatet baseres på fasen kan dette gi gode resultater. Signal-støy-forholdet blir bedre enn ved bruk av den tradisjonelle formen for støyestimering som skjer i det spatiale rommet. Dette gjelder imidlertid ikke bildene som ble tatt under belysning av en 355nm laserkilde. Siden dette er en ny teknikk bør man undersøke videre om det er mulig å oppnå gode resultater for andre bilder.

Plassering av grensen for hvilke prinsipalkomponenter som skal med under reverstransformasjonen til MNF og PCA kan bestemmes på mange måter. Her er det valgt å beholde en gitt prosent av det opprinnelige signalet. Alternativt kunne man satt grensen etter et krav om å fjerne en bestemt andel av støyen. En tredje mulighet vil være å plassere grensen etter den karakteristiske knekken man ser for grafene til de akkumulerte SNR-verdiene i appendiks 8.5. Dette kan bli et bra kompromiss mellom å beholde mye av signalet samtidig som man fjerner en god del av støyen. Denne metoden er imidlertid noe en bør undersøke videre.

Valg av lyskilde har mye å si for hvordan spekteret blir sendt ut for de hyperspektrale bildene. Hvitt lys gir lav intensitet for korte bølgelengder og høy intensitet for lange. Benyttes derimot 355nm laserlys blir dette endret til høy intensitet for korte og lav ved lange bølgelengder. Både PCA- og MNF-algoritmen gir bra støyreduksjon i spektral retning.

I motsetning til PCA og MNF virker Haar-transformen jevnt over dårlig. Haar-transformen er imidlertid bare en av mange wavelet-transformer. Å benytte andre wavelet-transformer istedenfor Haar er derfor noe som bør undersøkes for å se om dette kan gi bedre resultat. Spesielt høyere ordens Daubechies-transformene vil være interessant å prøve. Haar-transformen tilsvarer den enkleste av Daubechies-transformene.

Når man ser på støyreduksjonen for et enkelt spektralbånd er også denne god for både PCA og MNF. Forskjellen mellom PCA og MNF er også nevneverdige for spektralbåndene med lav intensitet som inneholder mye støy. Her virker det som PCA gir bedre resultater sammenlignet med MNF. Fra høstprosjektet kom det frem at PCA ikke egnet seg til reduksjon av dimensjonalitet. I figur 4.2.1 ser man derimot at MNF er godt egnet til dette, siden prinsipalkomponentene blir ordnet etter signal-støy-forholdet. Når det kommer til støyfjerning virker PCA bra, og for spektralbånd med lav intensitet blir resultatene til PCA bedre enn ved bruk av MNF.

7 Referanseliste

- [1] R. C. Gonzalez og R. E. Woods, «Digital Image Processing Third Edition», Pearson Education, 2008, s. 68, 152-153, 158-161, 313-319, 461, 696-697, 846-849
- [2] E. Ientilucci, «Hyperspectral Image Classification Using Orthogonal Subspace Projections: Image Simulation and Noise Analysis», Center for Imaging Science, 2001, s. 8
- [3] D. Manolakis, D. Marden, G. A. Shaw, «Hyperspectral Image Processing for Automatic Target Detection Applications», Lincoln Laboratory Journal, 2003
- [4] S. Fjerdings, «Utvikling og testing av algoritmer for analyse av hyperspektrale bilder», 2009, s. 2, 9-10, 19, 29-32, 42
- [5] C. Chen, «Comparison of Principal Components Analysis and Minimum Noise Fraction Transformation for Reducing the Dimensionality of Hyperspectral Imagery», Geographical Research, 2000
- [6] J. Shlens, «A Tutorial on Principal Component Analysis», 2009
- [7] L. I. Smith, «A tutorial on Principal Components Analysis», 2002
- [8] E. Kreyszig, «Advanced Engineering Mathematics 8th Edition», Wiley, 1999, s. 569-570, s. 574 s. 917-918, s. 929-937
- [9] H. M. Gundersen, B. F. Rasmussen, «An Application of Image Processing Techniques for Enhancement and Segmentation of Bruises in Hyperspectral Images», 2007, s. 11
- [10] R. Larsen, «Decomposition using maximum autocorrelation factors», 2002
- [11] A. A. Green, M. Berman, P. Switzer, M. D. Craig, «A Transformation for Ordering Multispectral Data in Terms of Image Quality with Implications for Noise Removal», Transactions on geoscience and remote sensing, 1988
- [12] L. Sun, J. Rieger, H. Hinrichs, «Maximum noise fraction (MNF) transformation to remove ballistocardiographic artifacts in EEG signals recorded during fMRI scanning», NeuroImage, 2009
- [13] A. A. Nielsen, R. Larsen, «Restoration of GERIS data using the maximum noise fractions transform», 1994
- [14] J. B. Lee, A. S. Woodyatt, M. Berman, «Enhancement of High Spectral Resolution Remote-Sensing Data by a Noise-Adjusted Principal Components Transform», 1990, IEEE Transactions on geoscience and remote sensing
- [15] D. R. Hundley, M. J. Kirby, M. G. Anderle, «A solution procedure for blind signal separation using the maximum noise fraction approach: algorithms and examples», 2001
- [16] ENVI User's Guide, September 2001 Edition, s. 563-564
- [17] R. Larsen, A. A. Nielsen, K. Conradsen, «Restoration of Hyperspectral Push-Broom Scanner Data»

- [18] A. A. Nielsen, «Analysis of Regularly and Irregularly Sampled Spatial, Multivariate, and Multi-temporal Data», 1994
- [19] G. Li, B. Song, «Image Salt-Pepper Noise Elimination by Detecting Edges and Isolated Noise Points», 2004
- [20] C. H. Edwards, D. E. Penney, «Calculus 6e», Prentice-Hall, 2002, s. 836
- [21] L. Gao, B. Zhang, X. Zhang, W. Zhang, Q. Tong, «A New Operational Method for Estimating Noise in Hyperspektrale Images», IEEE Geoscience and Remote Sensing Letters, 2008
- [22] X. Liu, B. Zhang, L. Gao, D. Chen, «A maximum noise fraction transform with improved noise estimation for hyperspektral images», Science in China Series F Information Sciences, 2009
- [23] U. Amato, R. M. Cavalli, A. Palombo, S. Pignatti, F. Santini, «Experimental approach to the selection of the components in the Minimum Noise Fraction», 2008
- [24] R. D. Phillips, C. E. Blinn, J. Zhang, L. T. Watson, R. H. Wynne, «An adaptive noise filtering algorithm based on the maximum noise fraction», 2008
- [25] L. M. Bruce, C. H. Koger, J. Li, «Dimensionality Reduction of Hyperspectral Data Using Discrete Wavelet Transform Feature Extraction», IEEE Transactions on geoscience and remote sensing, 2002
- [26] A. Agarwal, T. El-Ghazawi, J. L. Moigne, J. Joiner, «An Application of Wavelet Based Dimension Reduction to AIRS Data», 2005
- [27] A. Graps, «An Introduction to Wavelets», IEEE Computational Science and Engineering, 1995
- [28] C. M. Leavey, M. N. James, J. Summerscales, R. Sutton, «An introduction to wavelet transforms: a tutorial approach», 2005
- [29] The MathWorks, Inc. «MATLAB 7, Data Analysis», Rev. Mars 2010, seksjon 3-2 til 3-4
- [30] R. G. Gallager, «Circularly-Symmetric Gaussian random vectors», 2008
- [31] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, «Introduction to Algorithms second edition», The MIT Press, 2001, s.165

8 Appendiks

8.1 Utledning for kovariansen

Gitt at

$$A(x, y) = S(x, y) + N(x, y)$$

der $S(x, y)$ og $N(x, y)$ er ortogonale. Dette gir

$$\begin{aligned} A(x, y)^T A(x, y) &= (S(x, y) + N(x, y))^T (S(x, y) + N(x, y)) \\ &\quad \downarrow \\ &= S(x, y)^T S(x, y) + N(x, y)^T S(x, y) + S(x, y)^T N(x, y) + N(x, y)^T N(x, y) \\ &\quad \downarrow \\ &= S(x, y)^T S(x, y) + N(x, y)^T N(x, y) \end{aligned}$$

Av dette følger det at kovariansen til $A(x, y)$ blir

$$\text{Cov}\{A(x, y)\} = \Sigma = \Sigma_S + \Sigma_N$$

8.2 Utledning for støyfraksjon

SNR for $B_i(x, y)$ blir

$$\frac{\text{Var}\{a_i^T S(x, y)\}}{\text{Var}\{a_i^T N(x, y)\}} = \frac{a_i^T \Sigma a_i}{a_i^T \Sigma_N a_i} - 1$$

Dette gir følgende

$$\begin{aligned} 1 + \frac{\text{Var}\{a_i^T S(x, y)\}}{\text{Var}\{a_i^T N(x, y)\}} &= \frac{a_i^T \Sigma a_i}{a_i^T \Sigma_N a_i} \\ &\downarrow \\ \frac{\text{Var}\{a_i^T N(x, y)\}}{\text{Var}\{a_i^T N(x, y)\}} + \frac{\text{Var}\{a_i^T S(x, y)\}}{\text{Var}\{a_i^T N(x, y)\}} &= \frac{a_i^T \Sigma a_i}{a_i^T \Sigma_N a_i} \\ &\downarrow \\ \frac{\text{Var}\{a_i^T A(x, y)\}}{\text{Var}\{a_i^T N(x, y)\}} &= \frac{a_i^T \Sigma a_i}{a_i^T \Sigma_N a_i} \\ &\downarrow \\ \frac{\text{Var}\{a_i^T N(x, y)\}}{\text{Var}\{a_i^T A(x, y)\}} &= \frac{a_i^T \Sigma_N a_i}{a_i^T \Sigma a_i} \end{aligned}$$

8.3 Utledning for venstrehånds-egenvektorer

Green^[11] finner venstrehånds-egenvektorer til $\Sigma_N \Sigma^{-1}$ for å bestemme vektorene a_i i ligningen

$$B_i(x, y) = a_i^T A(x, y), \quad i = 1, \dots, n$$

Venstrehånds-egenvektorer er gitt ved

$$a \Sigma_N \Sigma^{-1} = \lambda a$$

Dette gir videre

$$a \Sigma_N \Sigma^{-1} = \lambda a$$

↓

$$(a \Sigma_N \Sigma^{-1})^T = (\lambda a)^T$$

↓

$$(\Sigma_N \Sigma^{-1})^T a^T = \lambda a^T$$

Ser nå at problemet har blitt et standard egenvektorproblem, som er tilsvarende punkt 4 i gjennomføringen av PCA.

8.4 Bildeutsnitt

De fem første bildene er av oversiden på hender. Disse hyperspektrale bildene har en oppløsning på 1600x1800 piksler og utsnittet som er benyttet er vist i tabellene. De fire siste bildene er av en blodåre, og har en oppløsning på 1600x1600 piksler. Forskjellen mellom de fire siste er lysforholdene bildene ble tatt under.

- Bilde 1 (Perfusion_P6_after_s1_20000_us_2x_2008-12-12T133712_corr_rad):

Bilde utsnitt				Homogent utsnitt			
Fra kolonne:	140	Til kolonne:	539	Fra kolonne:	520	Til kolonne:	570
Fra rad:	250	Til rad:	649	Fra rad:	30	Til rad:	80
Hudtype:	IV	Alder:	45	Kjønn:	M		
Lystype:	Hvit						

- Bilde 2 (Perfusion_P7_after_s1_20000_us_2x_2008-12-12T134331_corr_rad):

Bilde utsnitt				Homogent utsnitt			
Fra kolonne:	220	Til kolonne:	619	Fra kolonne:	540	Til kolonne:	590
Fra rad:	300	Til rad:	699	Fra rad:	50	Til rad:	100
Hudtype:	II-III	Alder:	71	Kjønn:	M		
Lystype:	Hvit						

- Bilde 3 (Perfusion_p13_s1_20000_us_2x_2008-12-12T142949_corr_rad):

Bilde utsnitt				Homogent utsnitt			
Fra kolonne:	820	Til kolonne:	1219	Fra kolonne:	510	Til kolonne:	560
Fra rad:	1050	Til rad:	1449	Fra rad:	70	Til rad:	120
Hudtype:	II	Alder:	34	Kjønn:	K		
Lystype:	Hvit						

Merknad: Bilde 3 inneholder en dominant horisontal støystripe.

- Bilde 4 (Perfusion_p13_s1_20000_us_2x_2008-12-12T142949_corr_rad):

Bilde utsnitt				Homogent utsnitt			
Fra kolonne:	820	Til kolonne:	1219	Fra kolonne:	510	Til kolonne:	560
Fra rad:	1050	Til rad:	1449	Fra rad:	70	Til rad:	120
Hudtype:	II	Alder:	34	Kjønn:	K		
Lystype:	Hvit						

Merknad: Bilde 4 er det nøyaktig samme utsnittet som bilde 3, men har er den horisontale støykomponenten blitt interpolert vekk. Dette er utført ved å erstatte den med snittet av naboradene som ligger rett over og under støyraden.

- Bilde 5 (Perfusion_p16_s1_20000_us_2x_2008-12-12T144844_corr_rad):

Bilde utsnitt				Homogent utsnitt			
Fra kolonne:	150	Til kolonne:	549	Fra kolonne:	540	Til kolonne:	590
Fra rad:	230	Til rad:	629	Fra rad:	70	Til rad:	120
Hudtype:	III	Alder:	69	Kjønn:	M		
Lystype:	Hvit						

- Bilde 6 (AA11a_hyp_ll_p01_d01_20000_us_2x_HSNR03_2009-01-21T160101_corr_rad):

Bilde utsnitt					
Fra kolonne:	403	Til kolonne:	802	Integrasjonstid:	Normal
Fra rad:	1010	Til rad:	1409	Midling:	3x
Lystype:	Hvit				

- Bilde 7 (AA11a_hypfluor_1W_p05_d01_20000_us_2x_2009-01-21T161057_corr_rad):

Bilde utsnitt					
Fra kolonne:	400	Til kolonne:	799	Integrasjonstid:	Normal
Fra rad:	1000	Til rad:	1399	Midling:	Ingen
Lystype:	UV, 355nm triplet nd:YAG-laser				

- Bilde 8 (AA11a_hypfluor_1W_p05_d01_20000_us_2x_HSNR03_2009-01-21T161136_corr_rad):

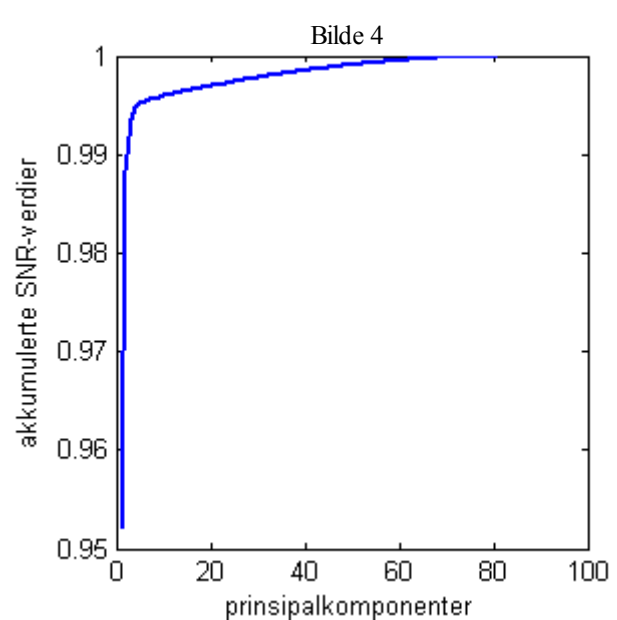
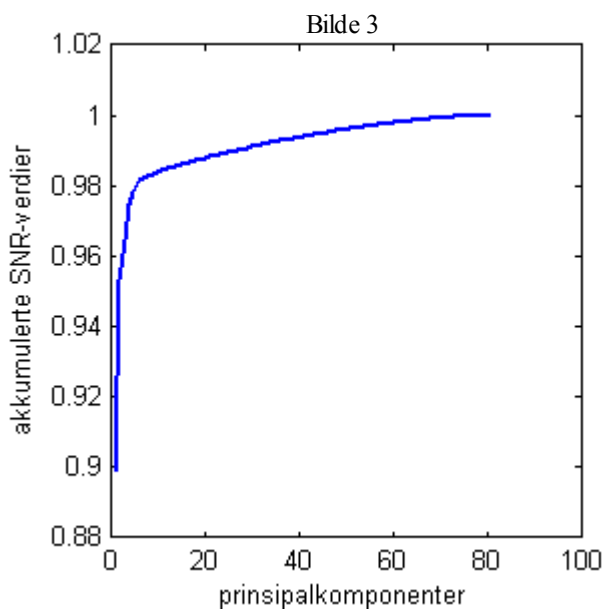
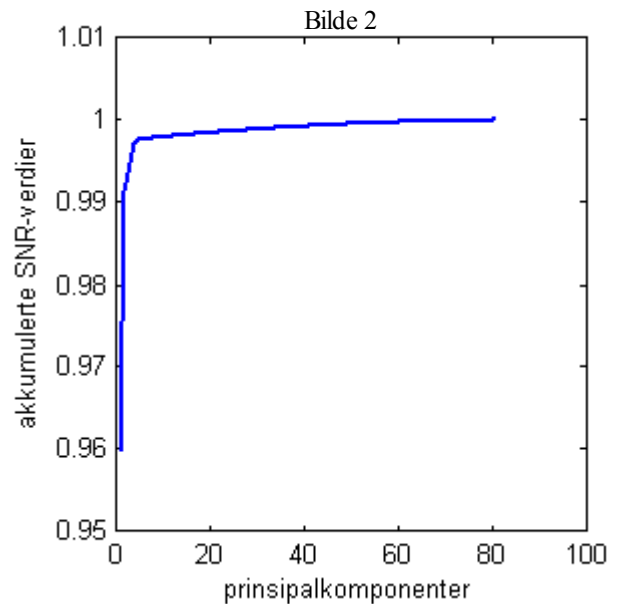
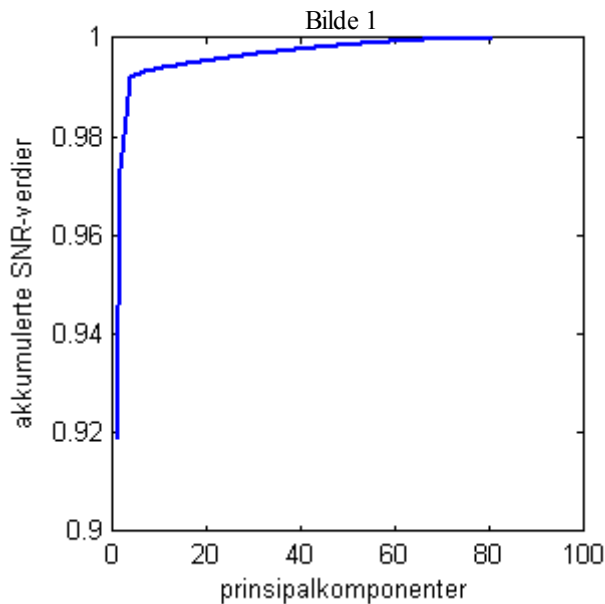
Bilde utsnitt					
Fra kolonne:	400	Til kolonne:	799	Integrasjonstid:	Normal
Fra rad:	1001	Til rad:	1400	Midling:	3x
Lystype:	UV, 355nm triplet nd:YAG-laser				

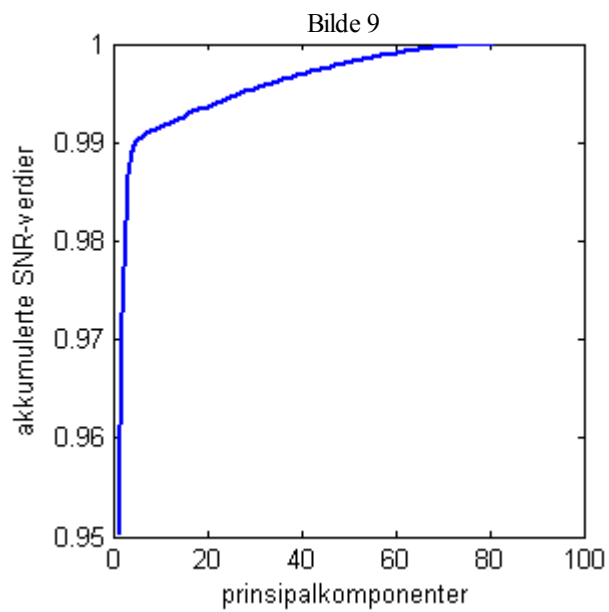
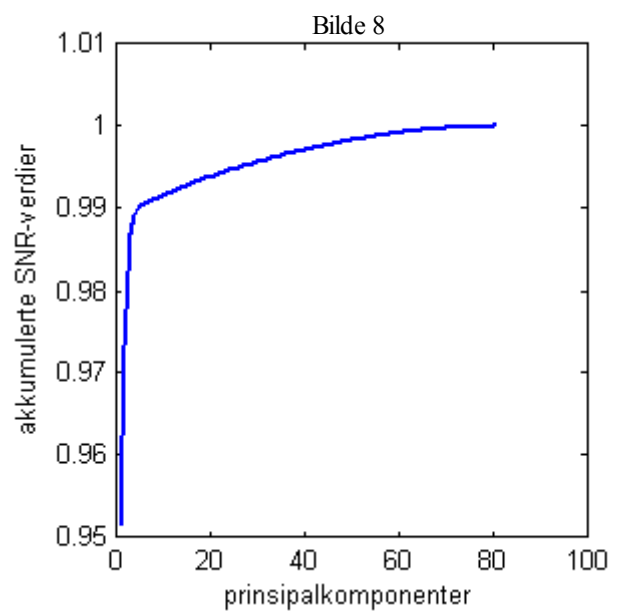
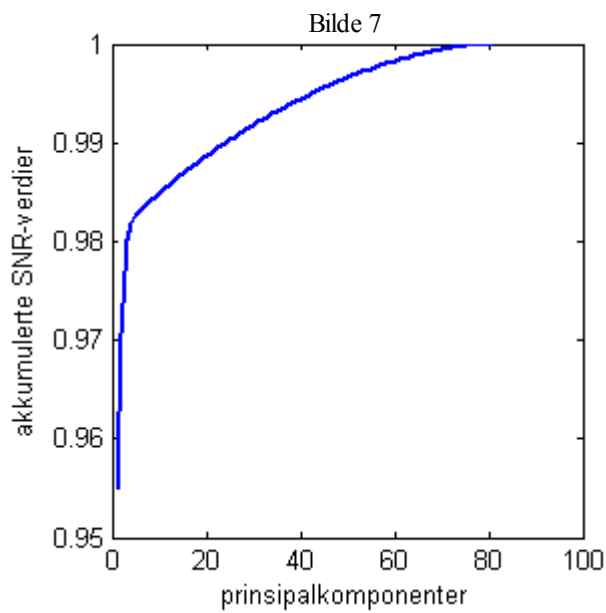
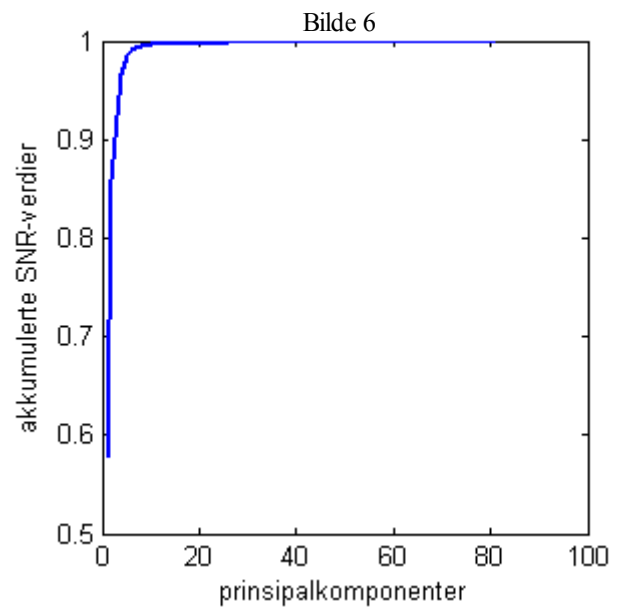
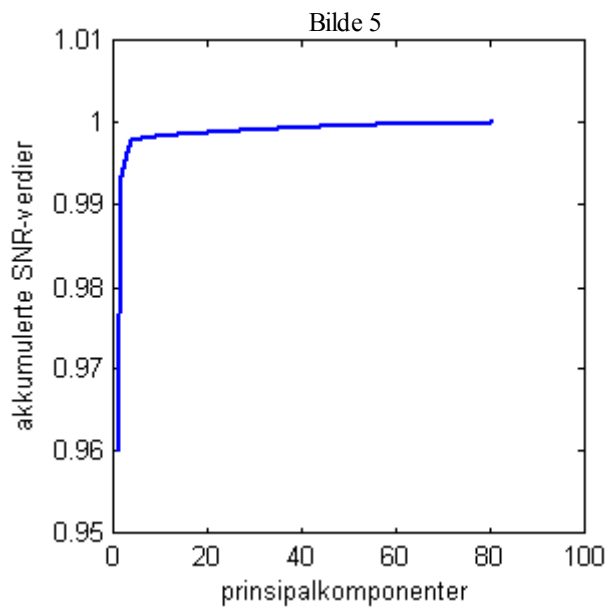
- Bilde 9 (AA11a_hypfluor_1W_p05_d01_40000_us_2x_2009-01-21T161339_corr_rad):

Bilde utsnitt					
Fra kolonne:	400	Til kolonne:	799	Integrasjonstid:	2x
Fra rad:	1000	Til rad:	1399	Midling:	Ingen
Lystype:	UV, 355nm triplet nd:YAG-laser				

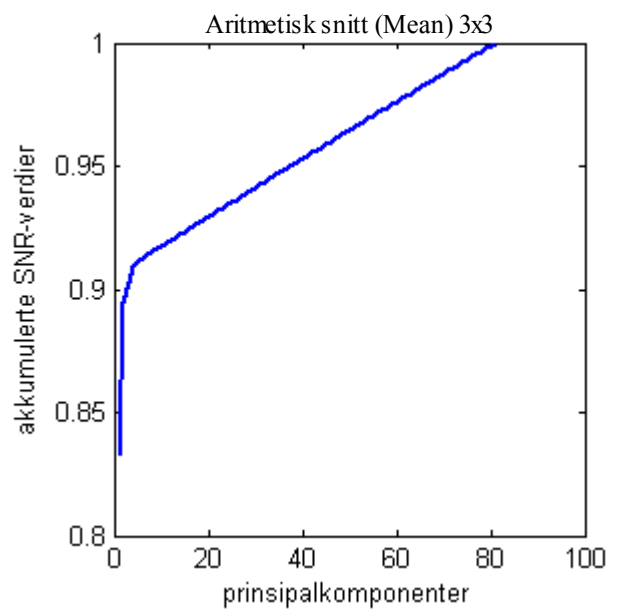
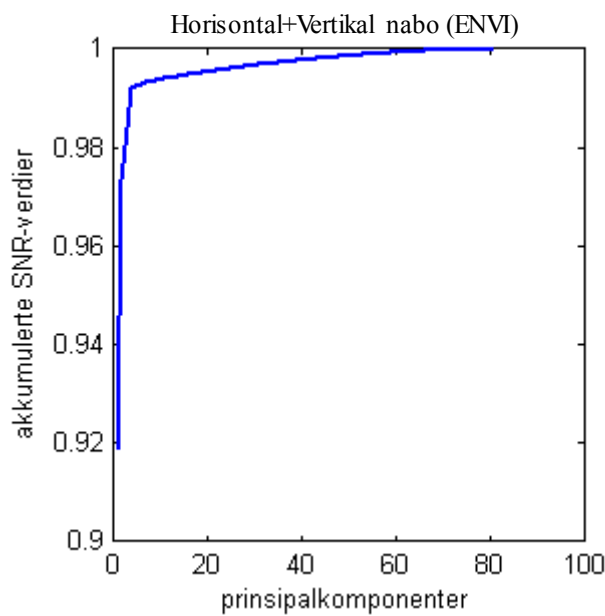
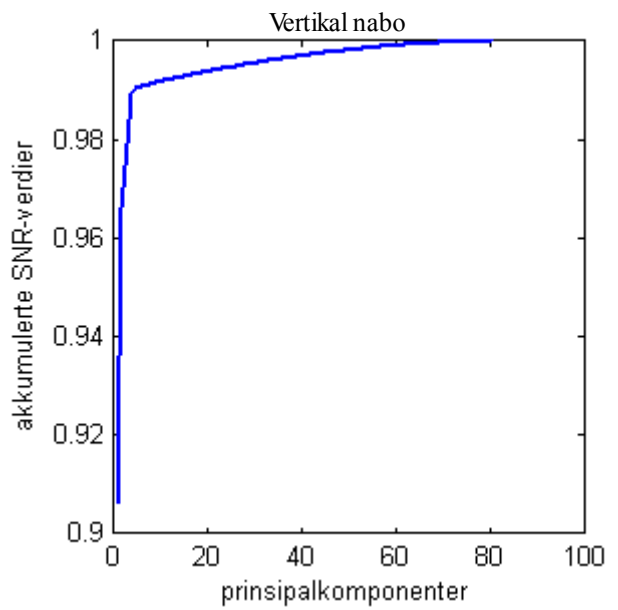
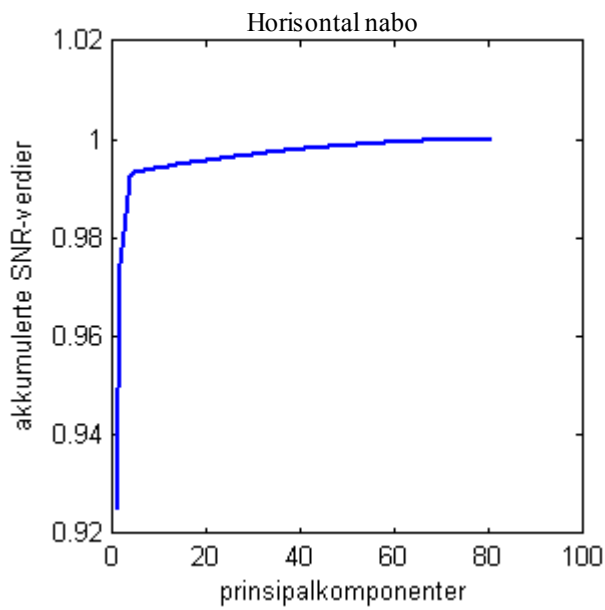
8.5 SNR-verdier basert på totalt utsnitt

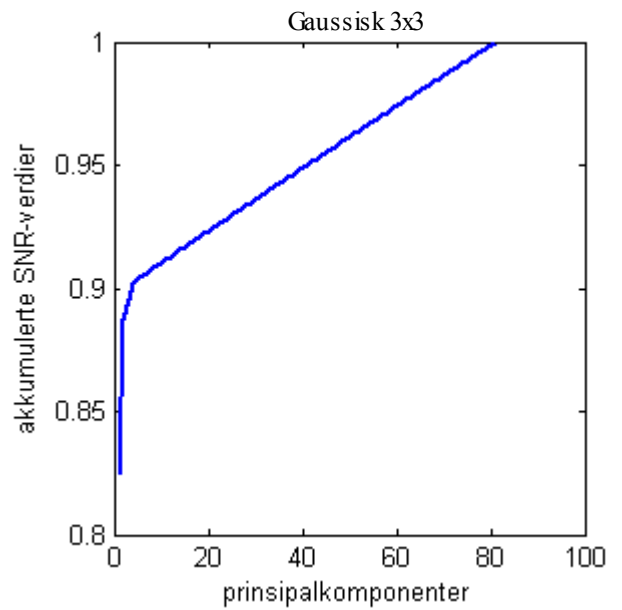
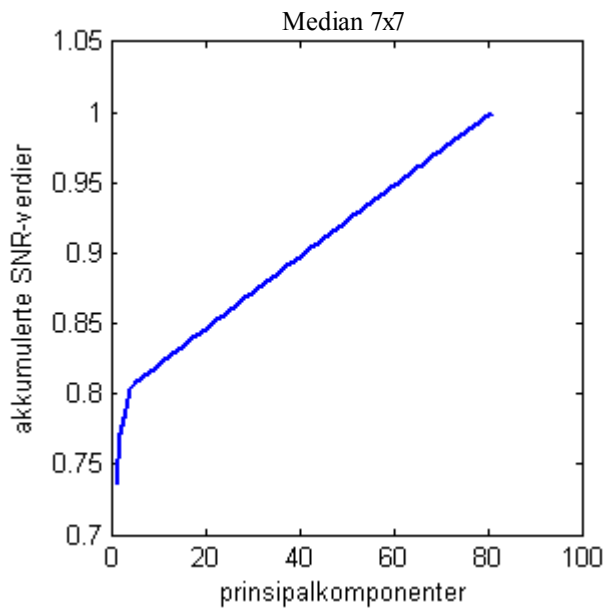
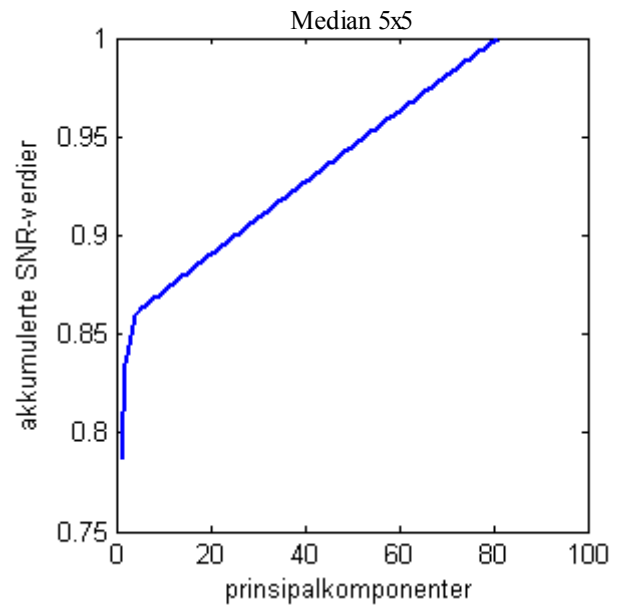
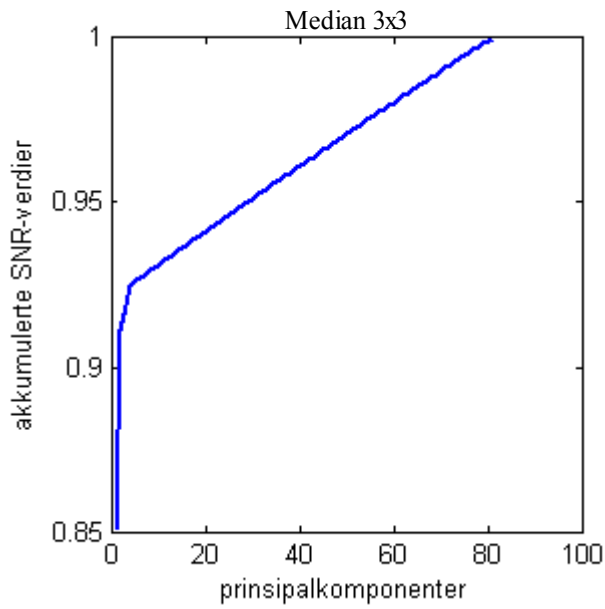
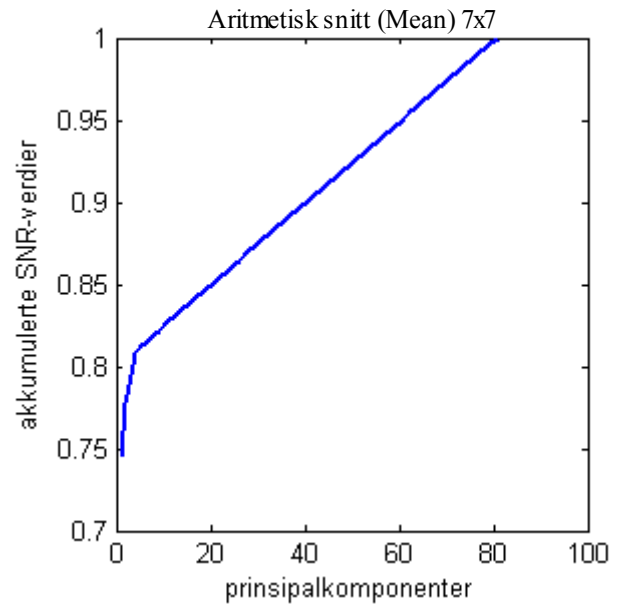
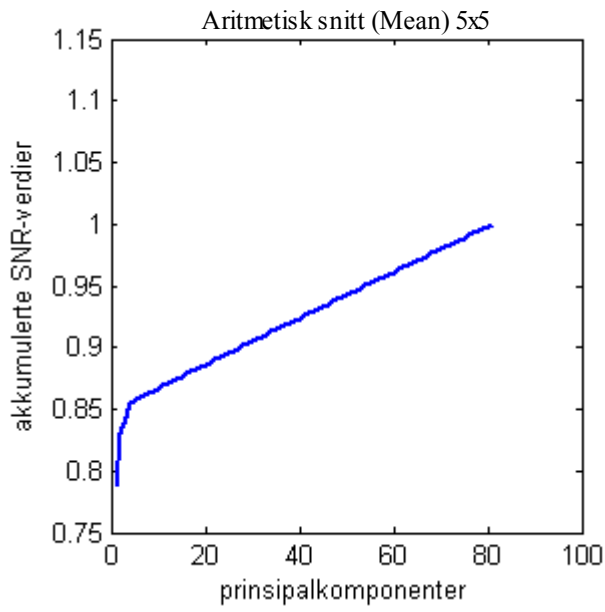
Akkumulerte og normaliserte SNR-verdier der støyestimeringen baseres på det totale utsnittet. I MNF-algoritmen har støyestimatene blitt utført med differansen til nærmeste vertikale og horisontale nabopiksel. Dette er den samme måten som programmet ENVI bruker. Merk at aksene som viser akkumulerte SNR-verdier er forskjellige. Dette er gjort for å synliggjøre endringene til kurvene.

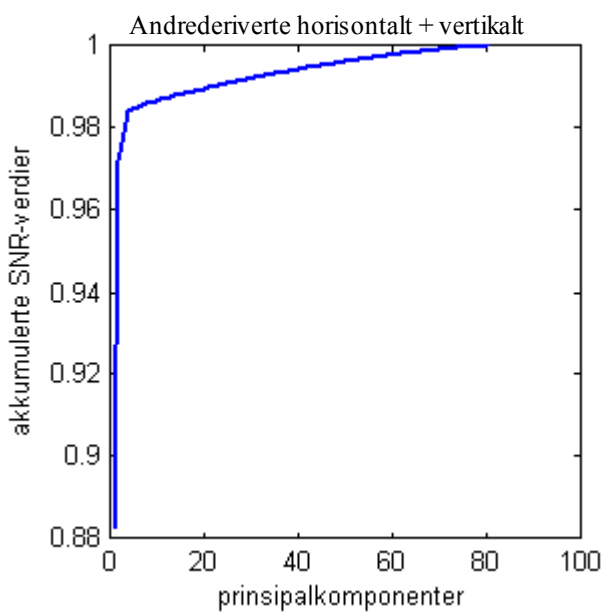
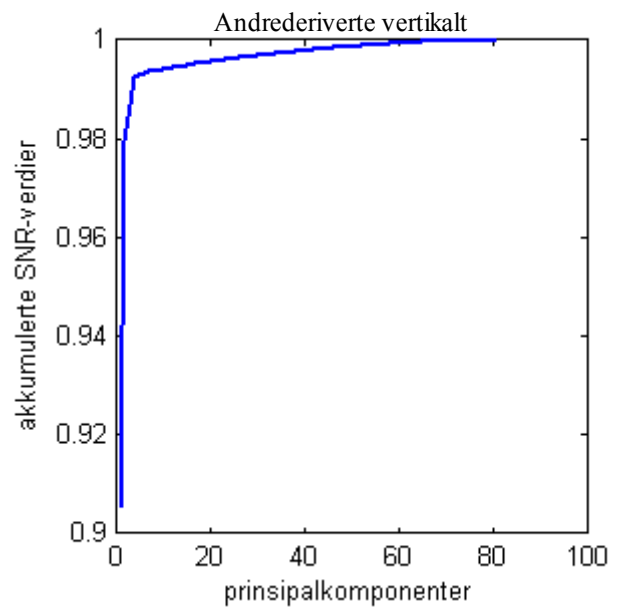
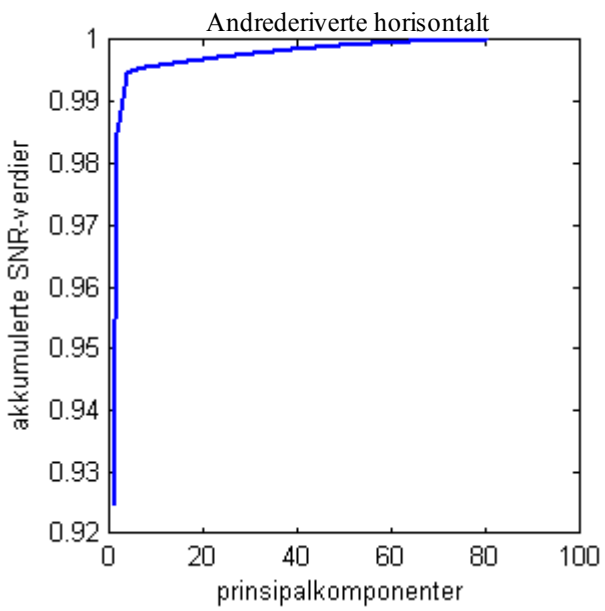
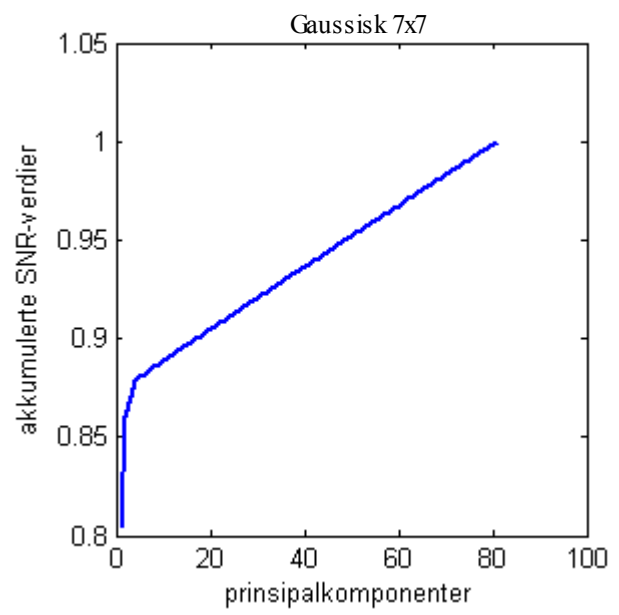
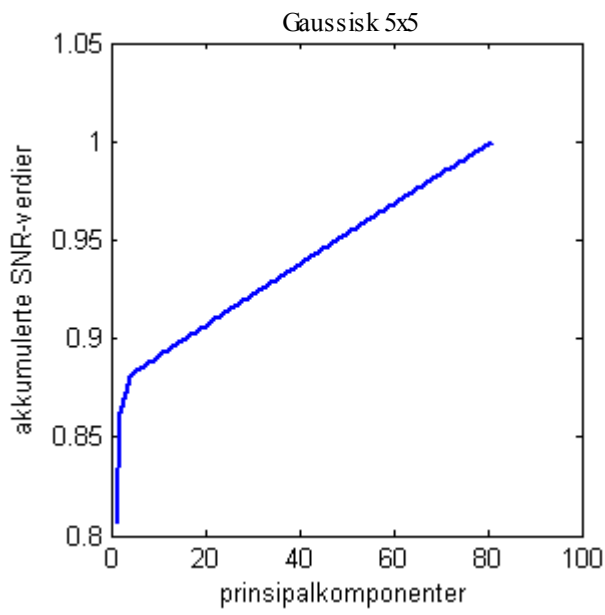




Resultatene for ulike støyestimat når man tar utgangspunkt i et bestemt bilde. Her er det bilde 1 som blir benyttet.

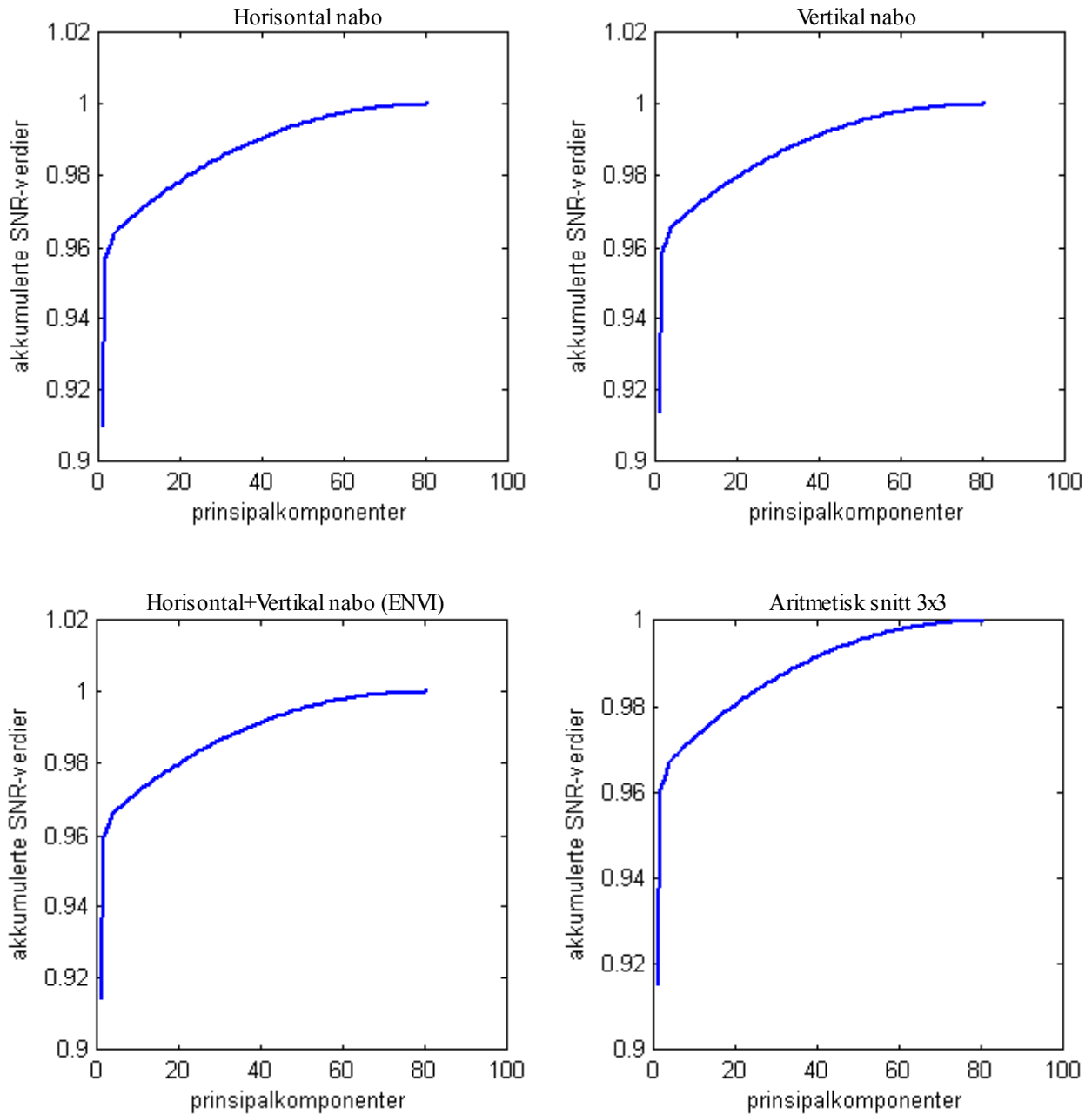


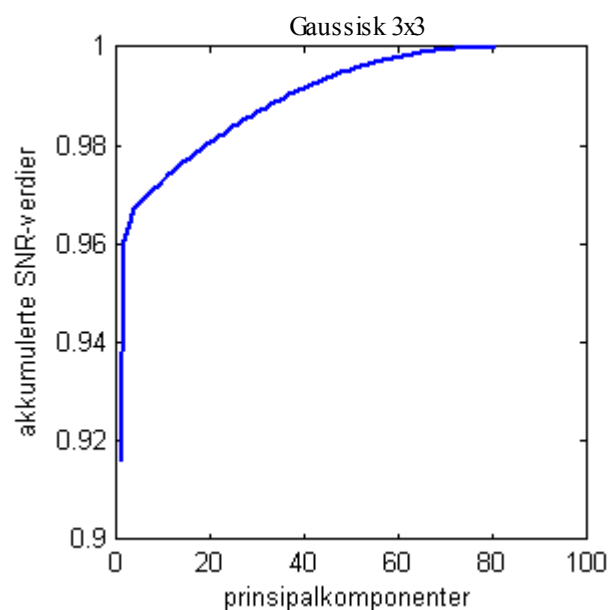
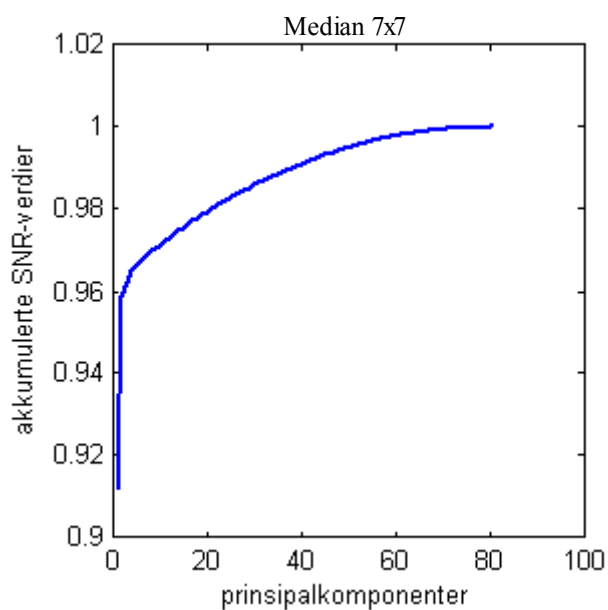
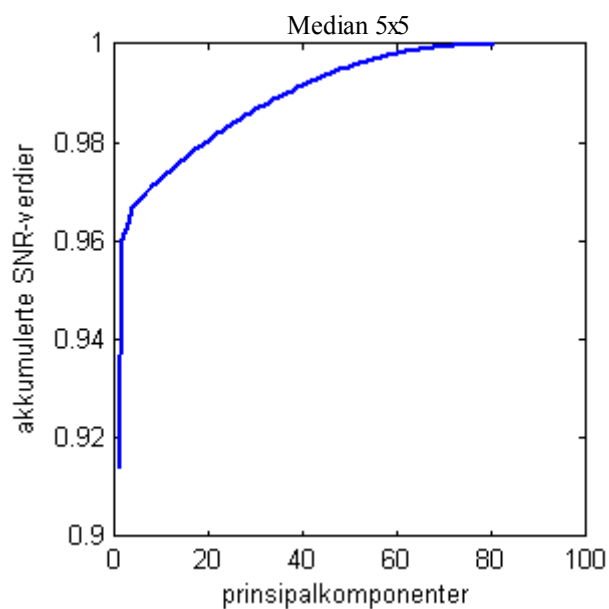
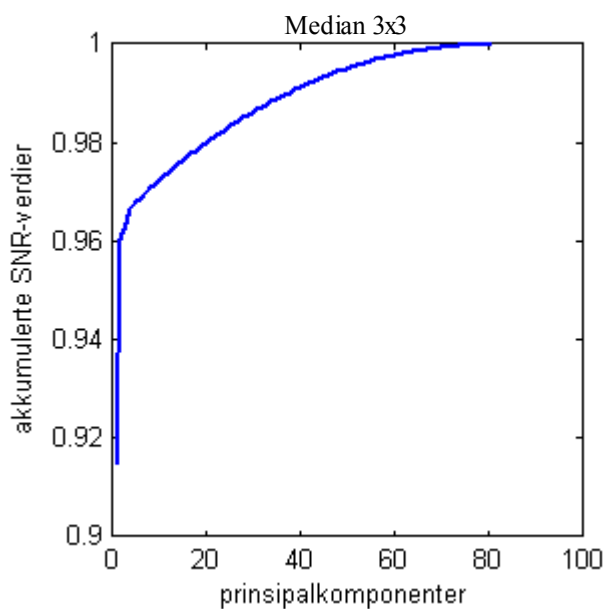
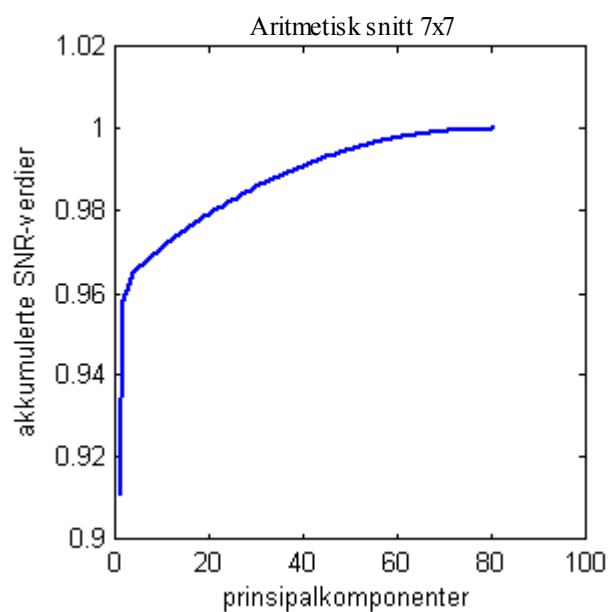
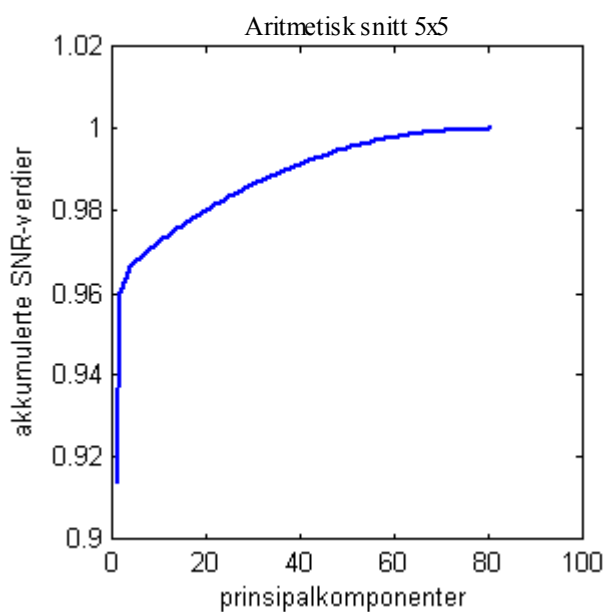


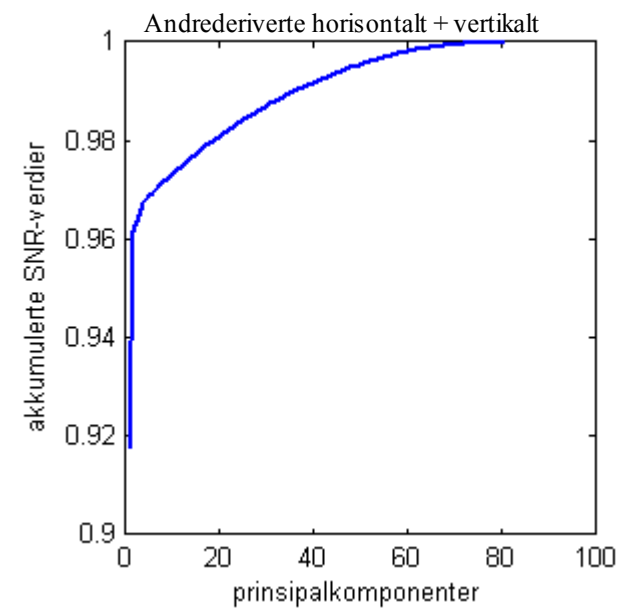
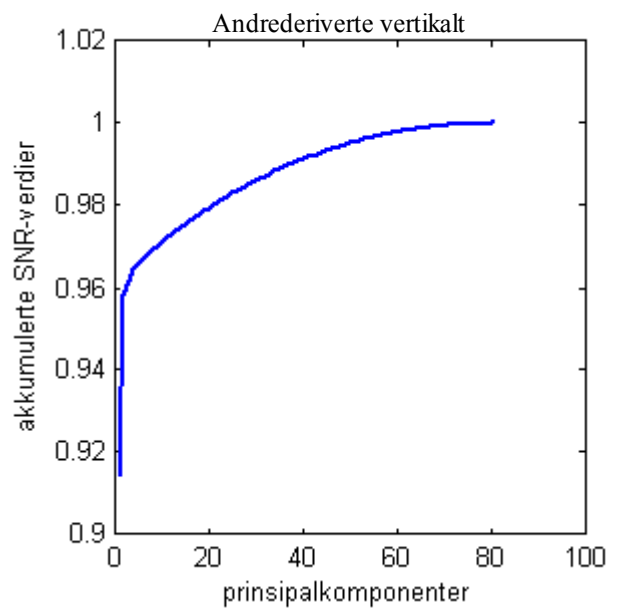
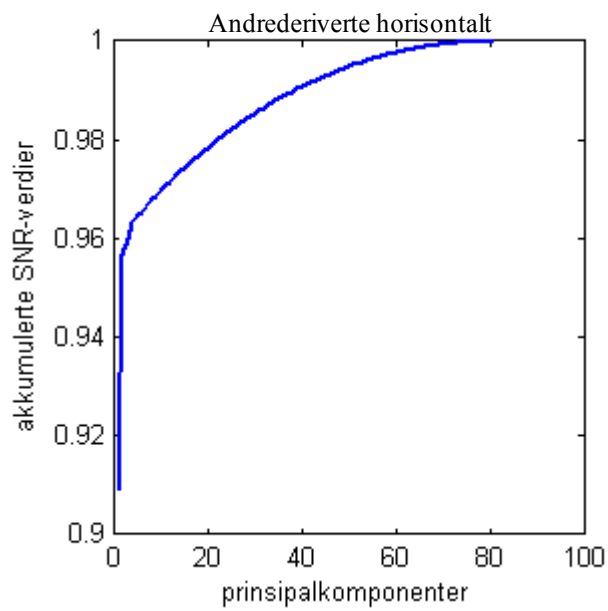
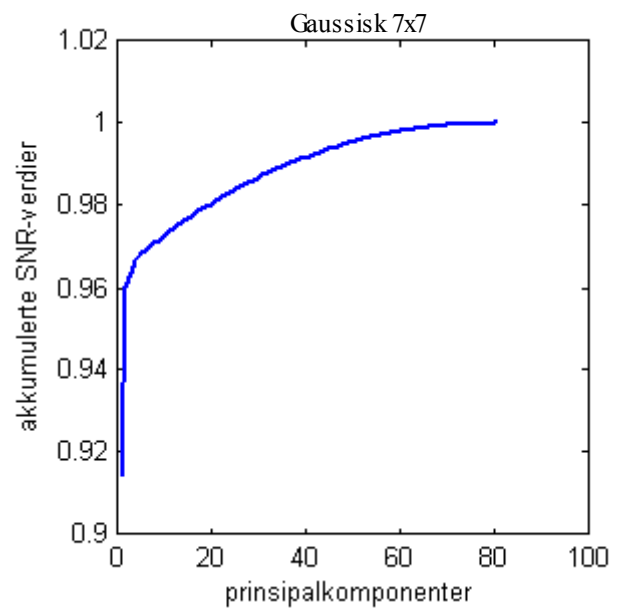
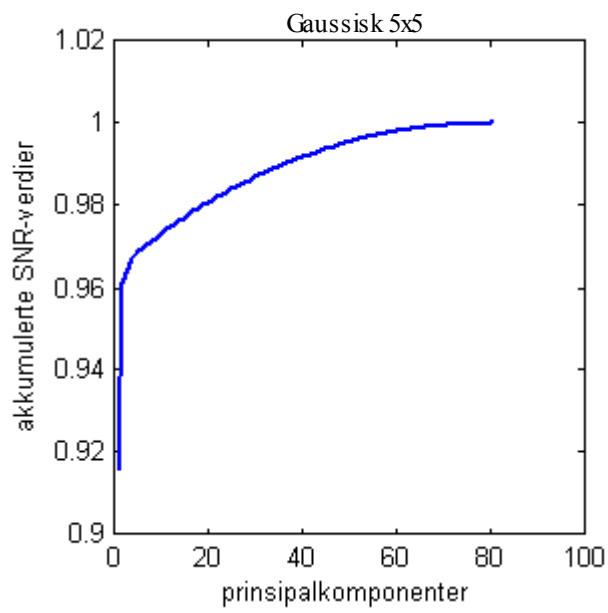


8.6 SNR-verdier basert på homogen del

Akkumulerte og normaliserte SNR-verdier der støyestimeringen baseres på en homogen del av utsnittet. Resultatene for ulike støyestimat når man tar utgangspunkt i en homogen del av bildet. Her er det bilde 1 som blir benyttet. Merk at aksene som viser akkumulerte SNR-verdier er forskjellige. Dette er gjort for å synliggjøre endringene til kurvene.

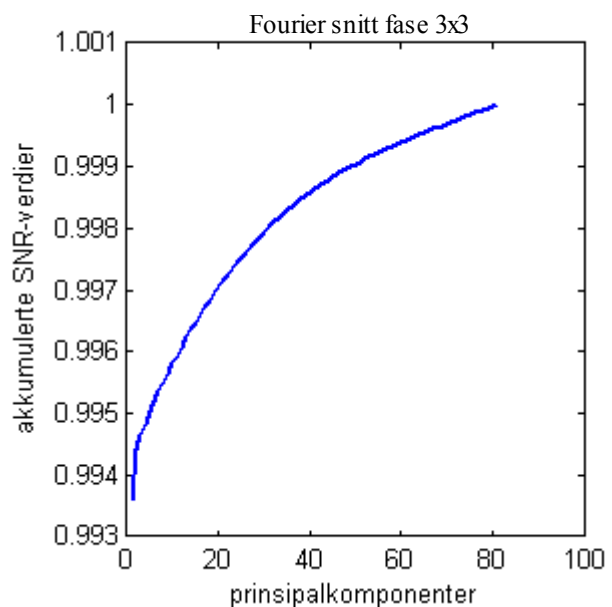
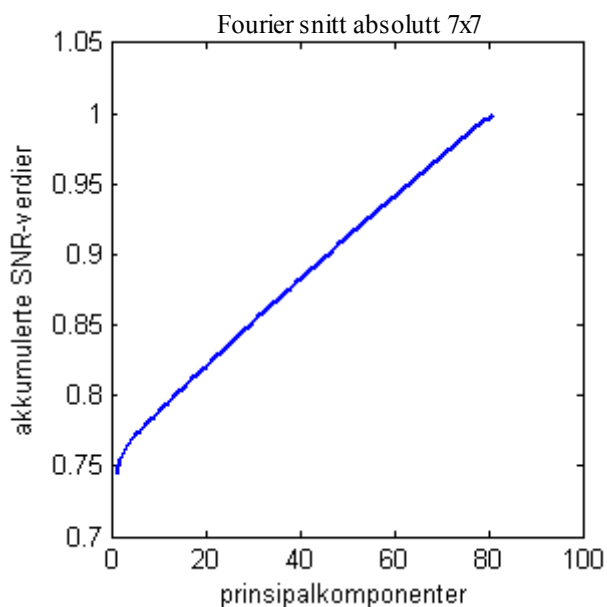
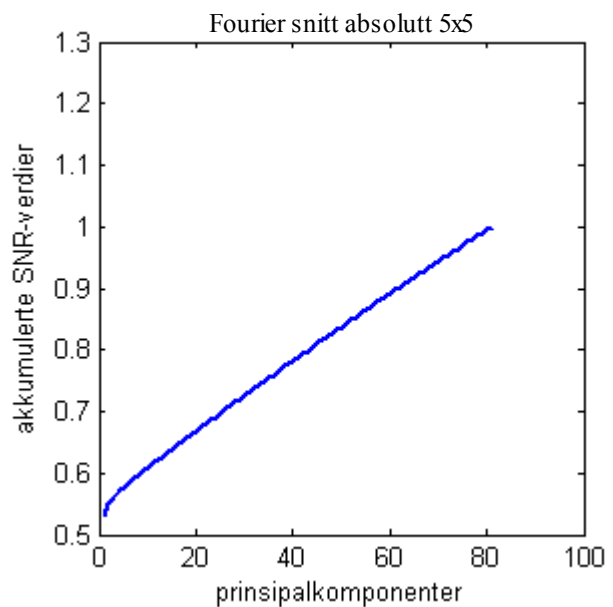
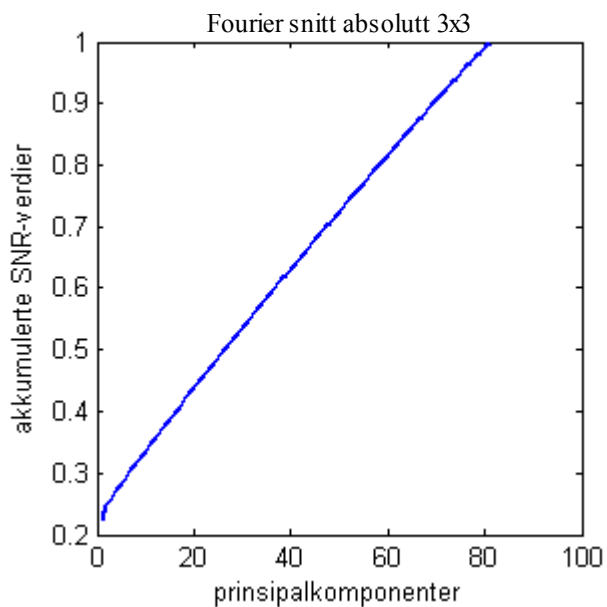


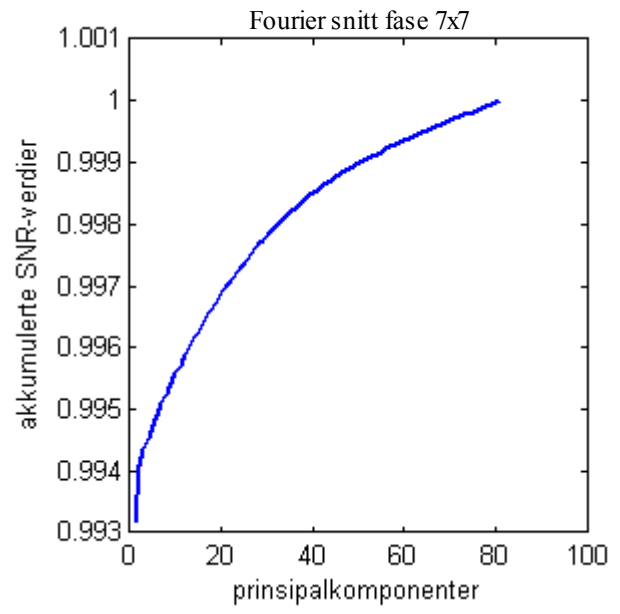
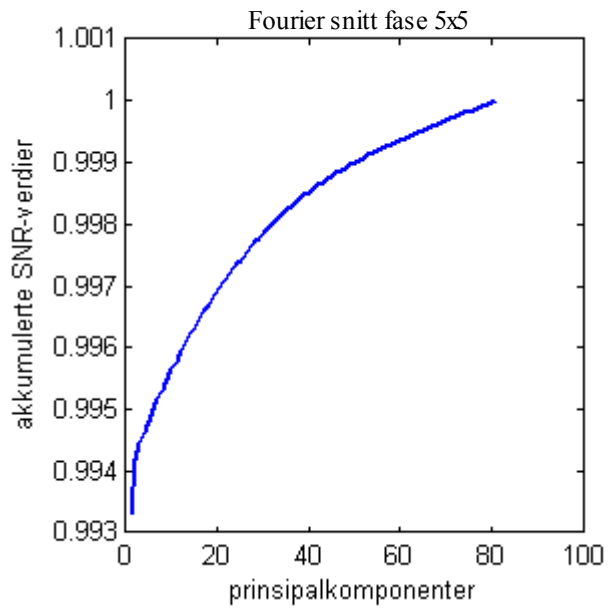




8.7 SNR-verdier for støyestimering i fourierrommet

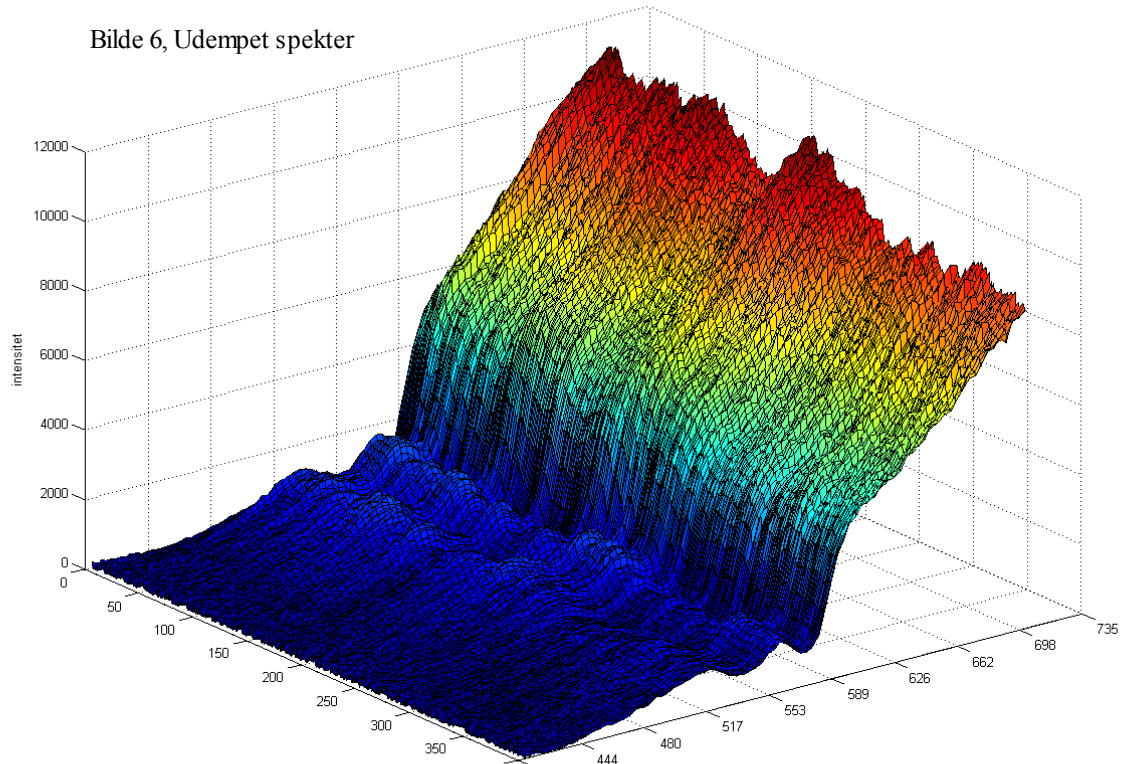
Akkumulerte og normaliserte SNR-verdier der støyestimeringen er utført i fourierrommet. Resultatene for ulike støyestimat når støyestimeringen skjer i fourierrommet. Her er det bilde 1 som blir benyttet. Merk at aksene som viser akkumulerte SNR-verdier er forskjellige. Dette er gjort for å synliggjøre endringene til kurvene.



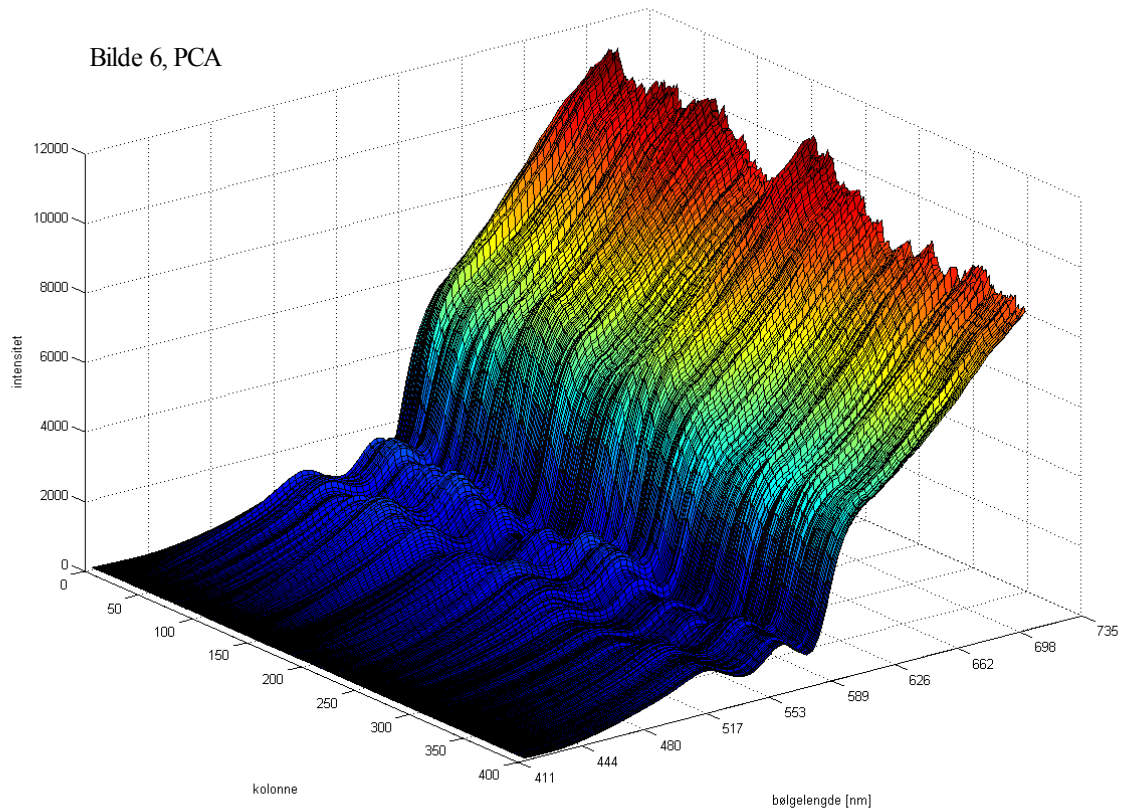


8.8 Spekter for enkel rad

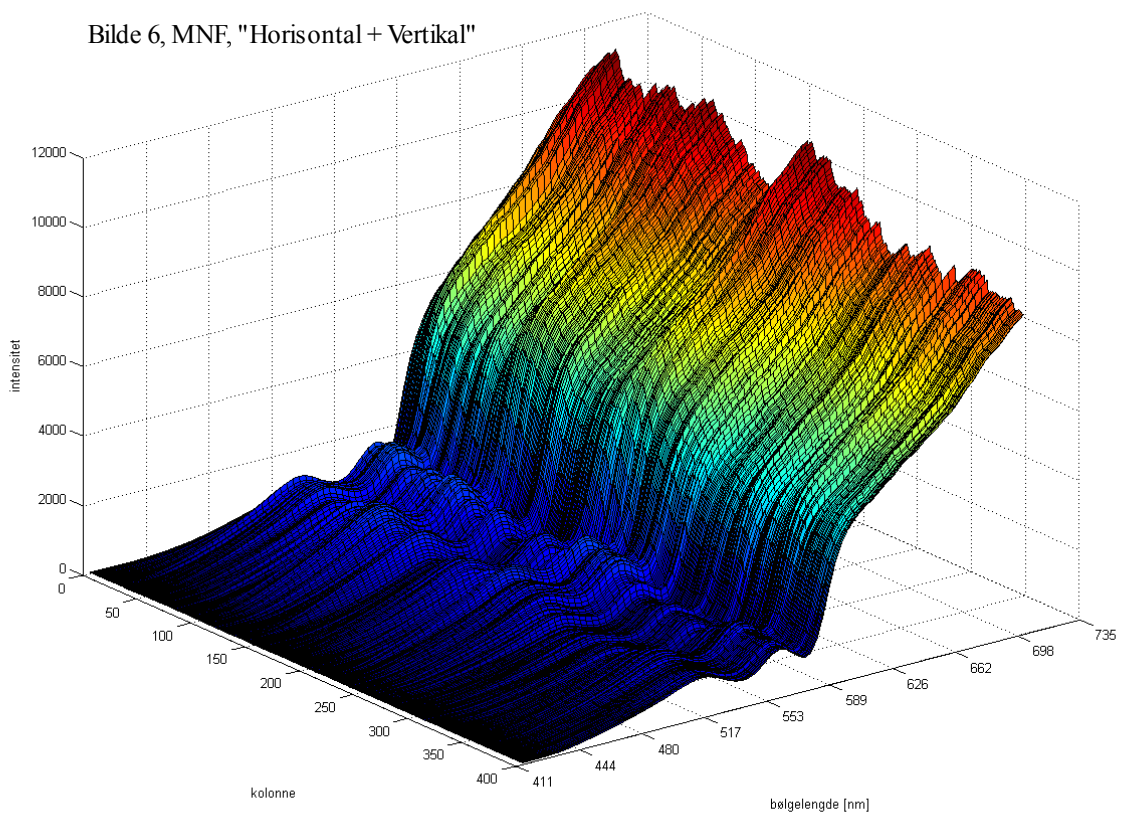
Bilde 6, Udempet spekter



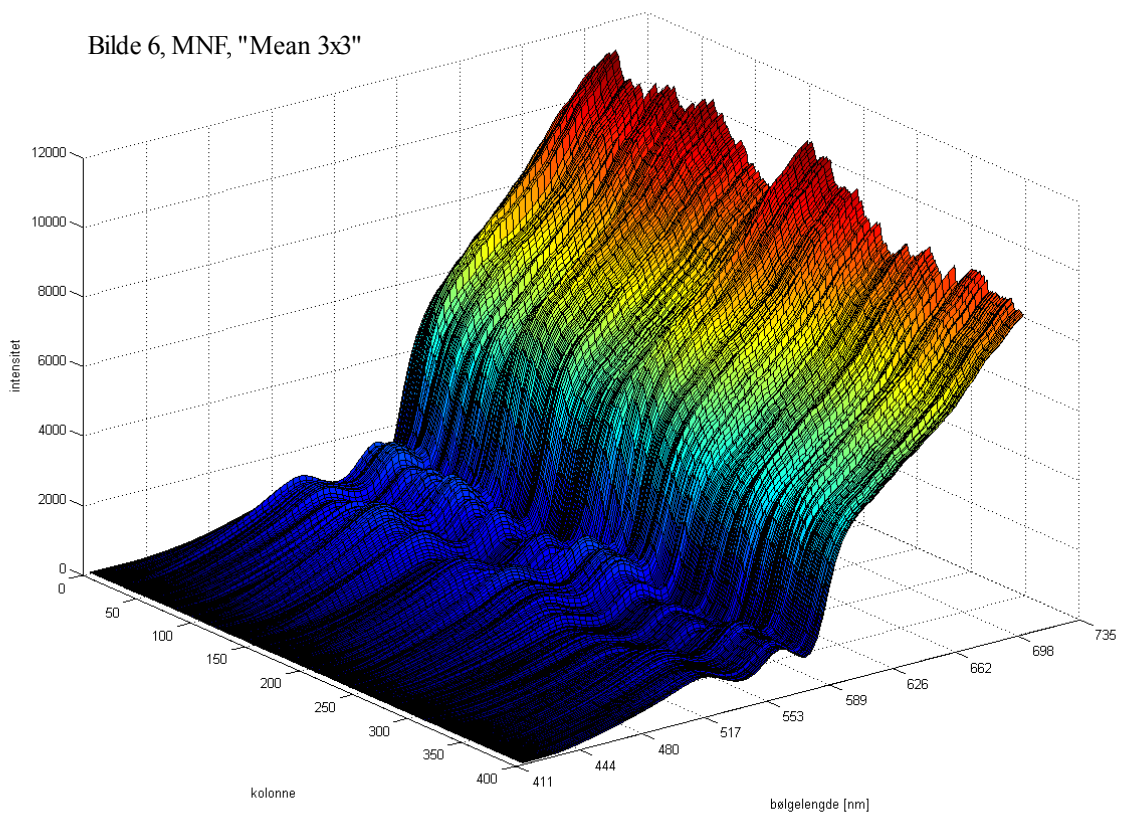
Bilde 6, PCA



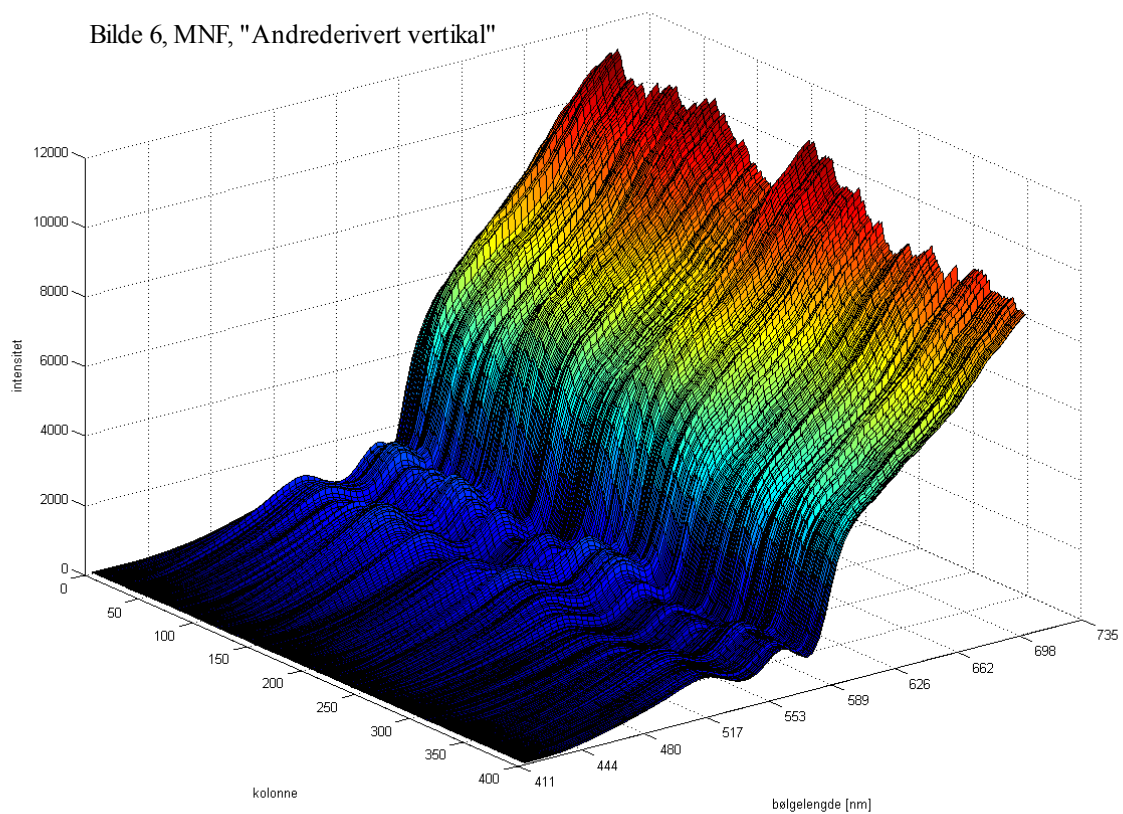
Bilde 6, MNF, "Horizontal + Vertikal"



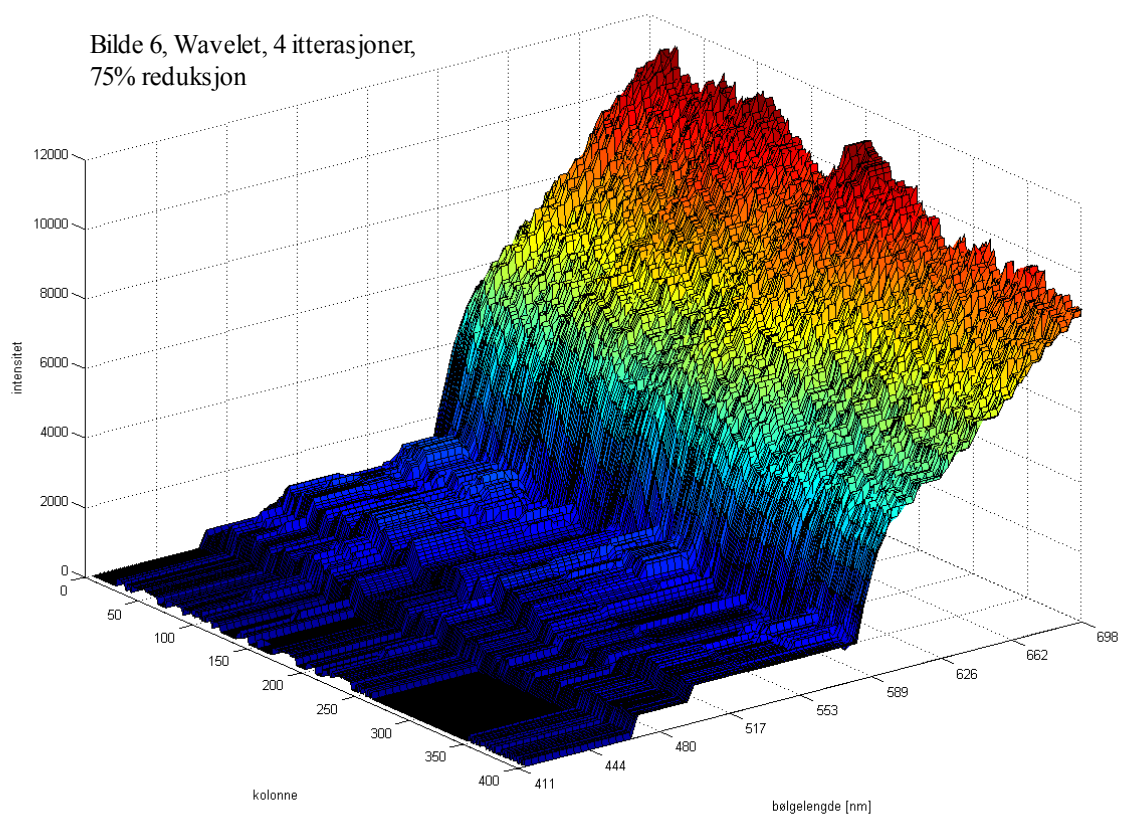
Bilde 6, MNF, "Mean 3x3"



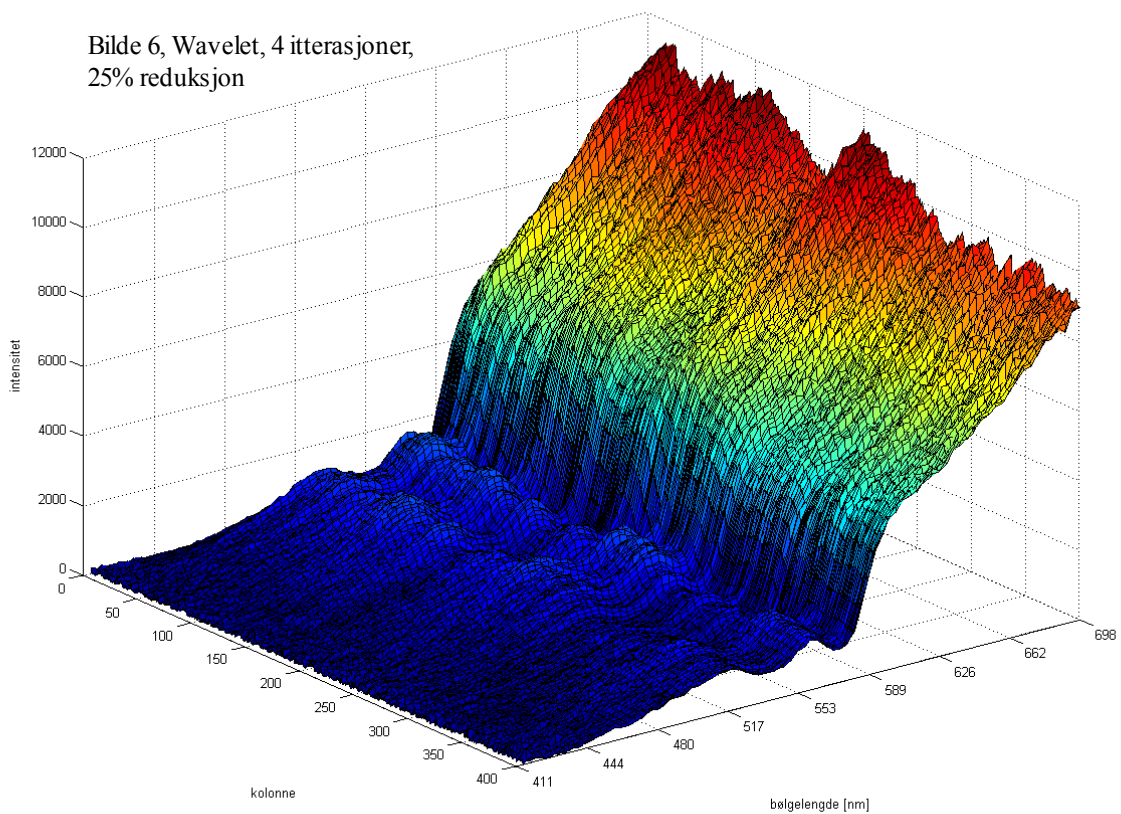
Bilde 6, MNF, "Andrederivert vertikalt"



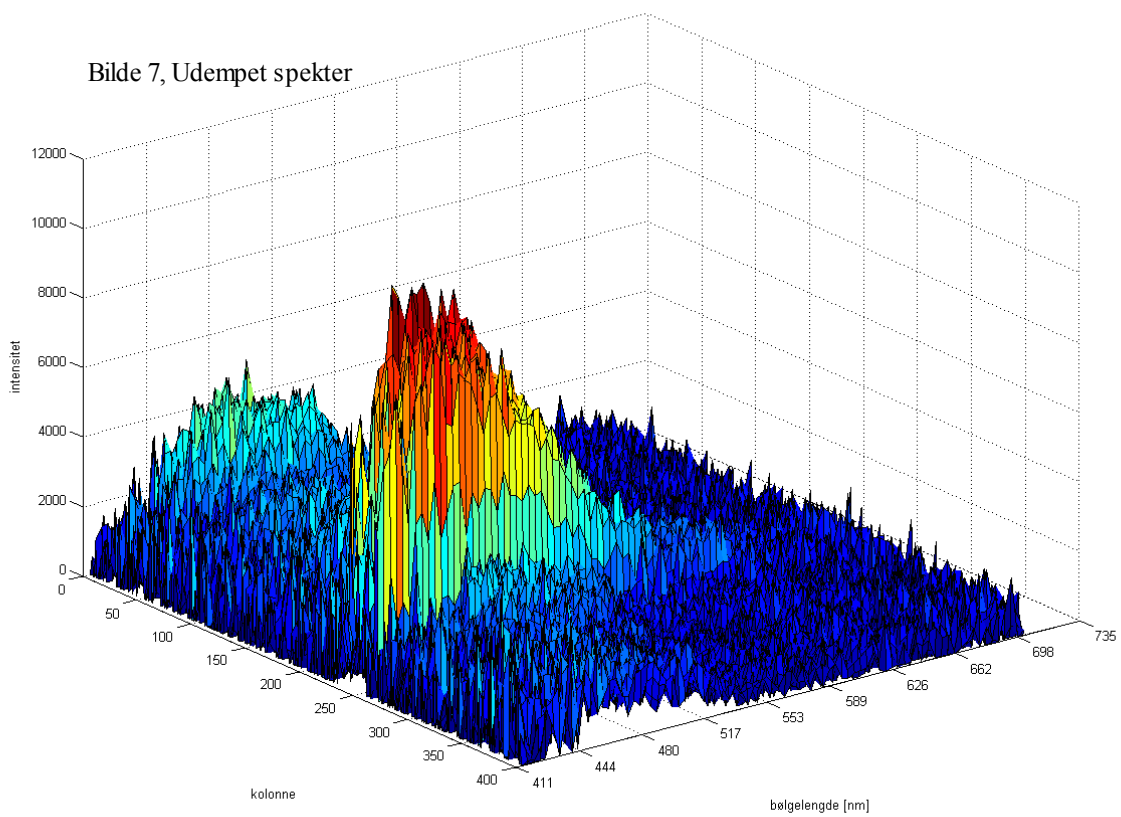
Bilde 6, Wavelet, 4 itterasjoner, 75% reduksjon



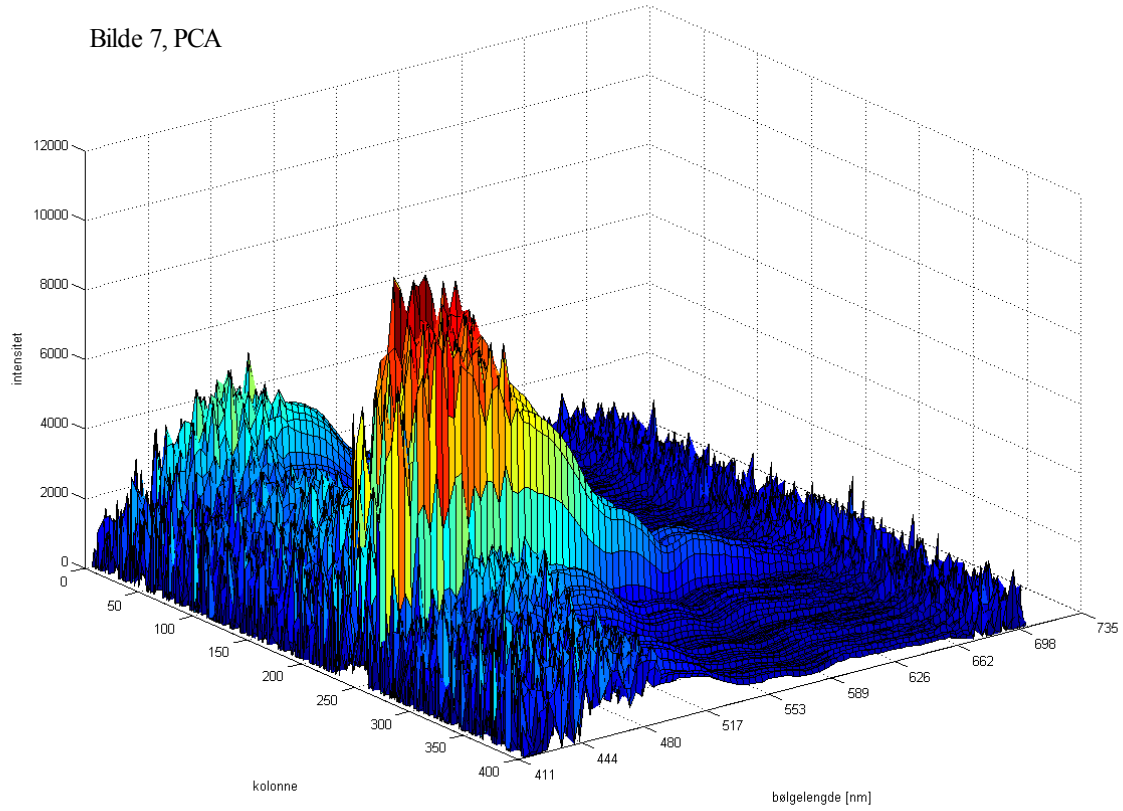
Bilde 6, Wavelet, 4 itterasjoner,
25% reduksjon



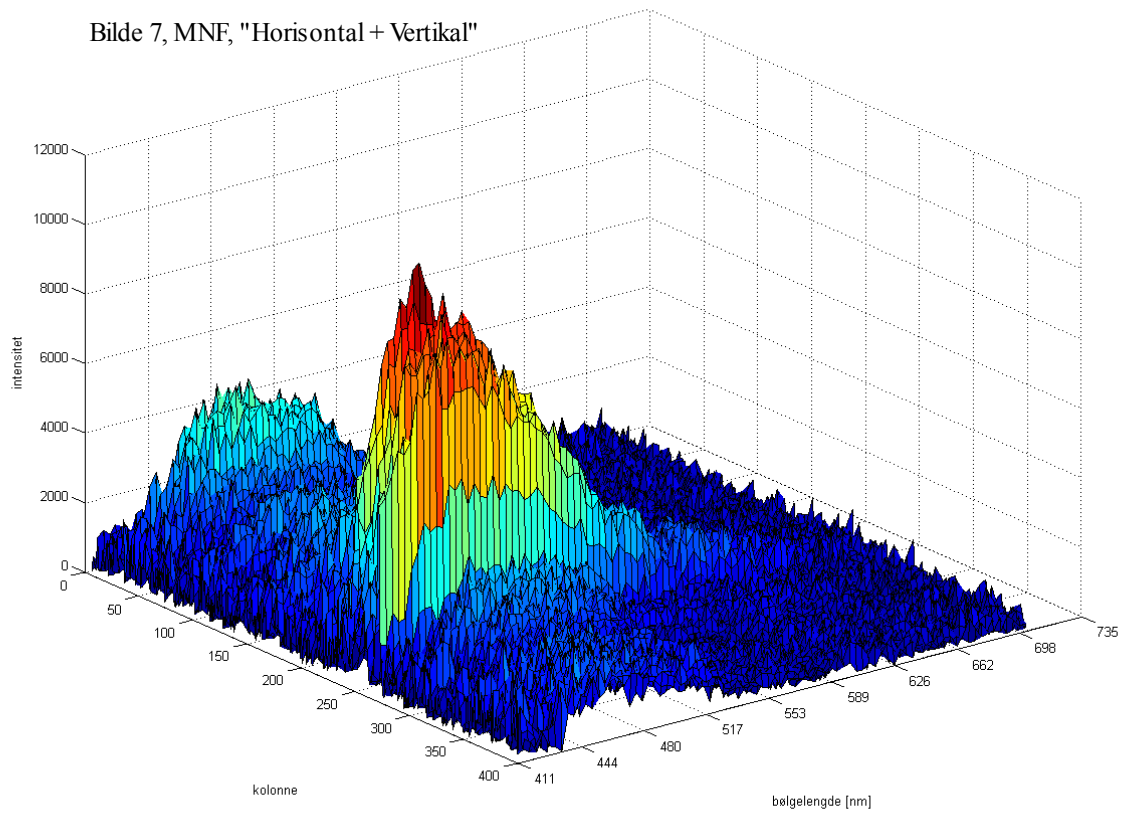
Bilde 7, Udempet spekter



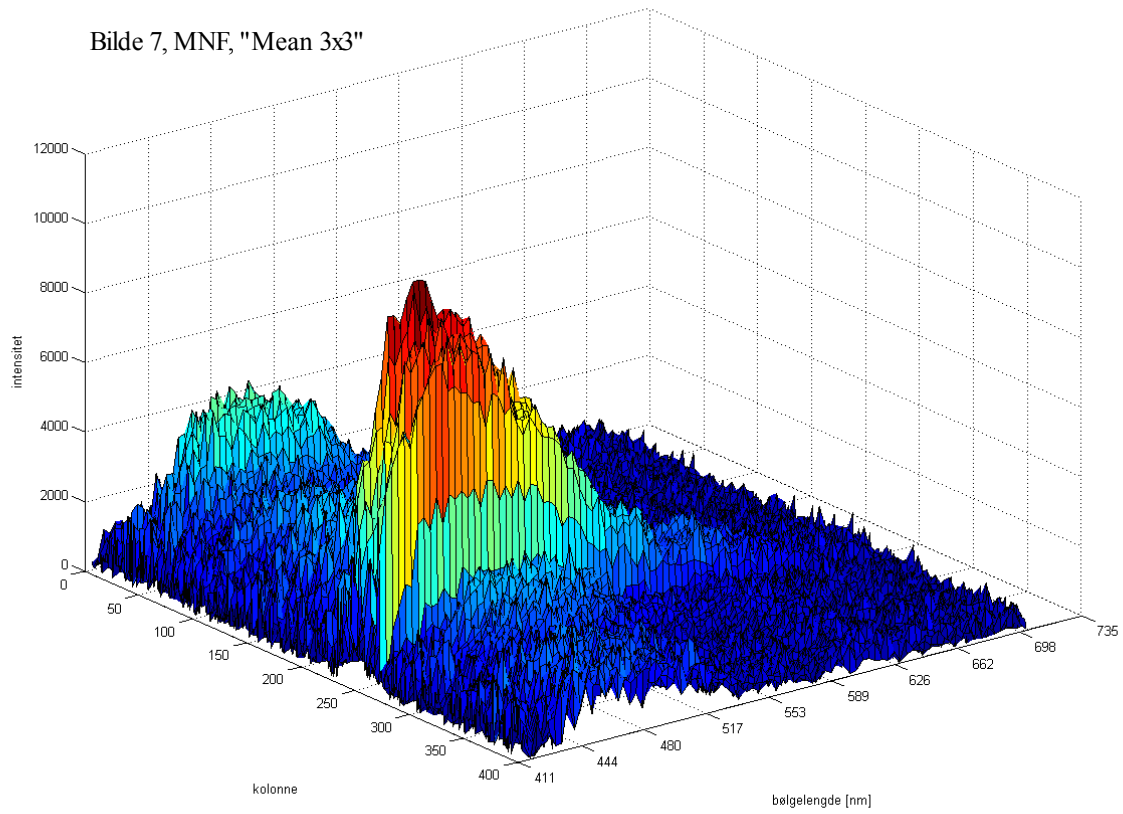
Bilde 7, PCA



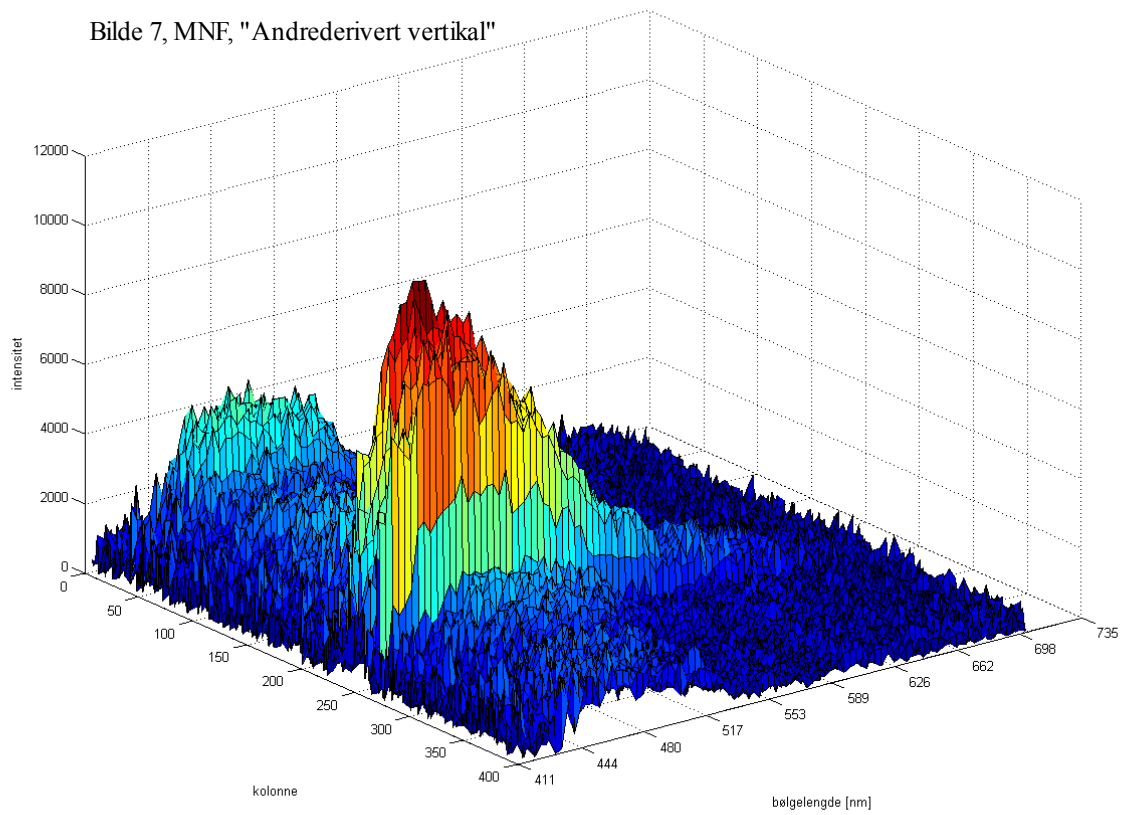
Bilde 7, MNF, "Horizontal + Vertikal"



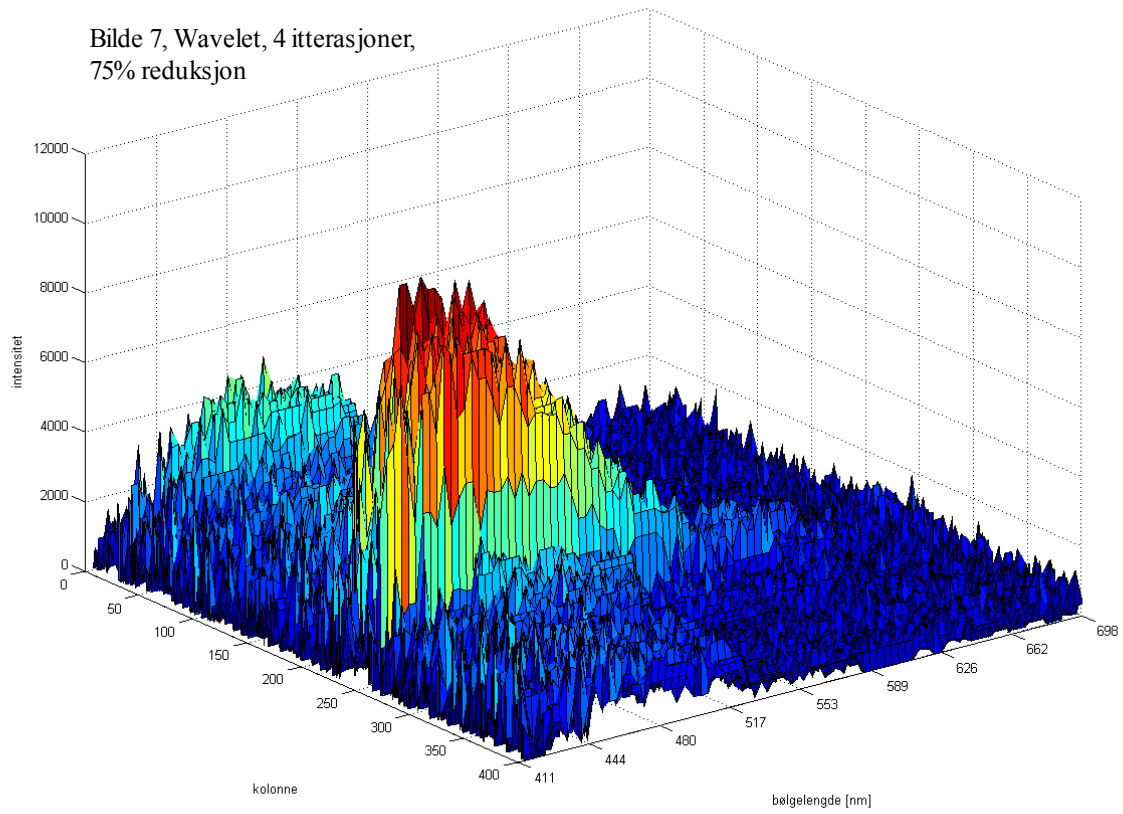
Bilde 7, MNF, "Mean 3x3"



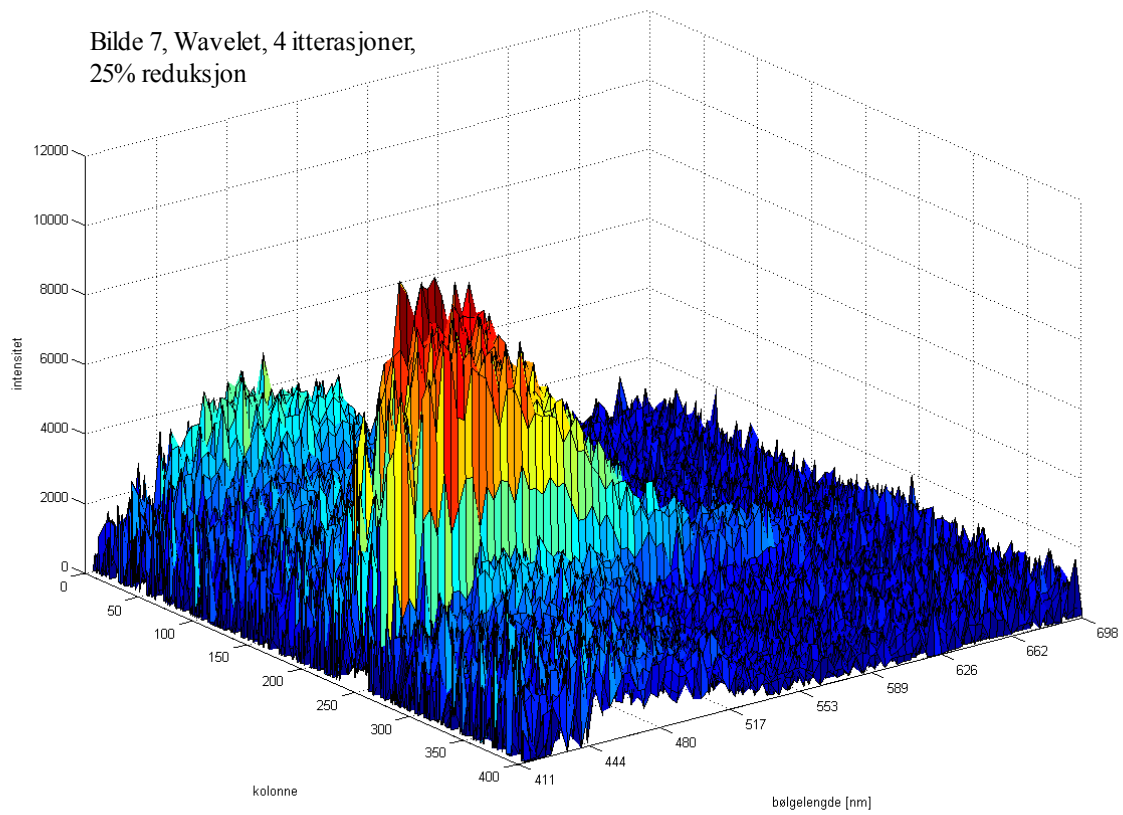
Bilde 7, MNF, "Andrederivert vertikalt"



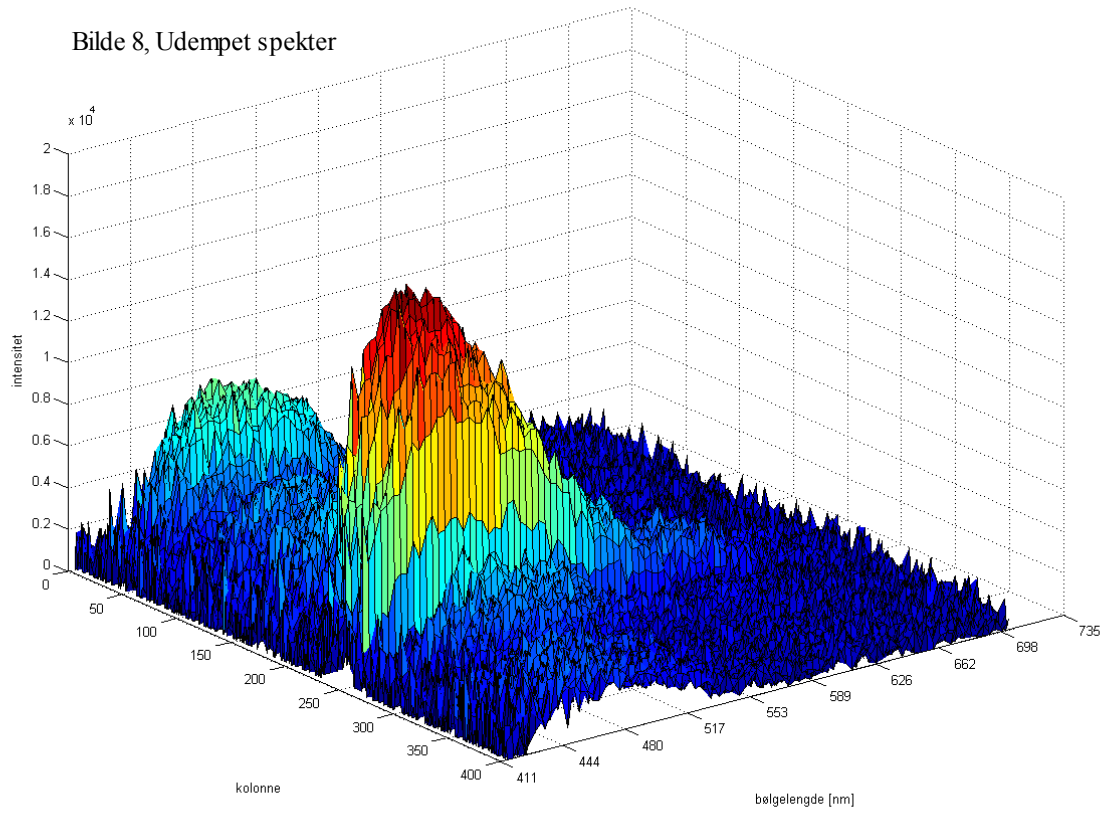
Bilde 7, Wavelet, 4 itterasjoner,
75% reduksjon



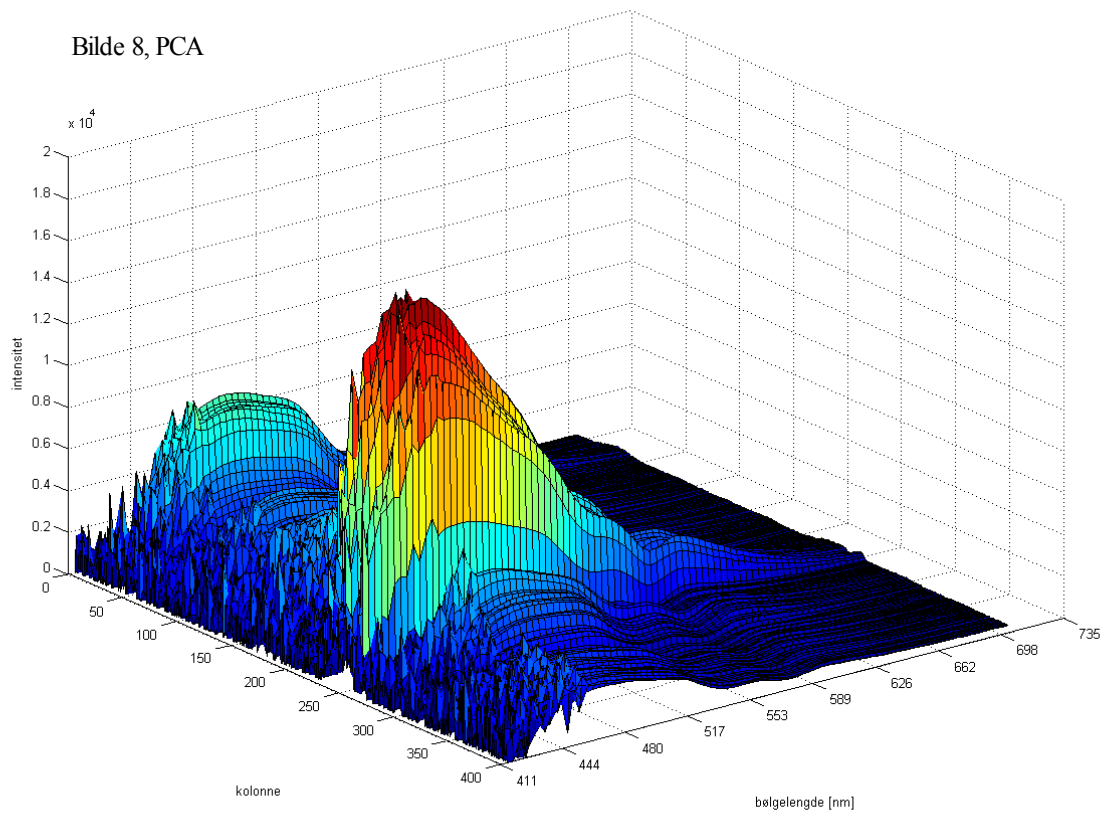
Bilde 7, Wavelet, 4 itterasjoner,
25% reduksjon



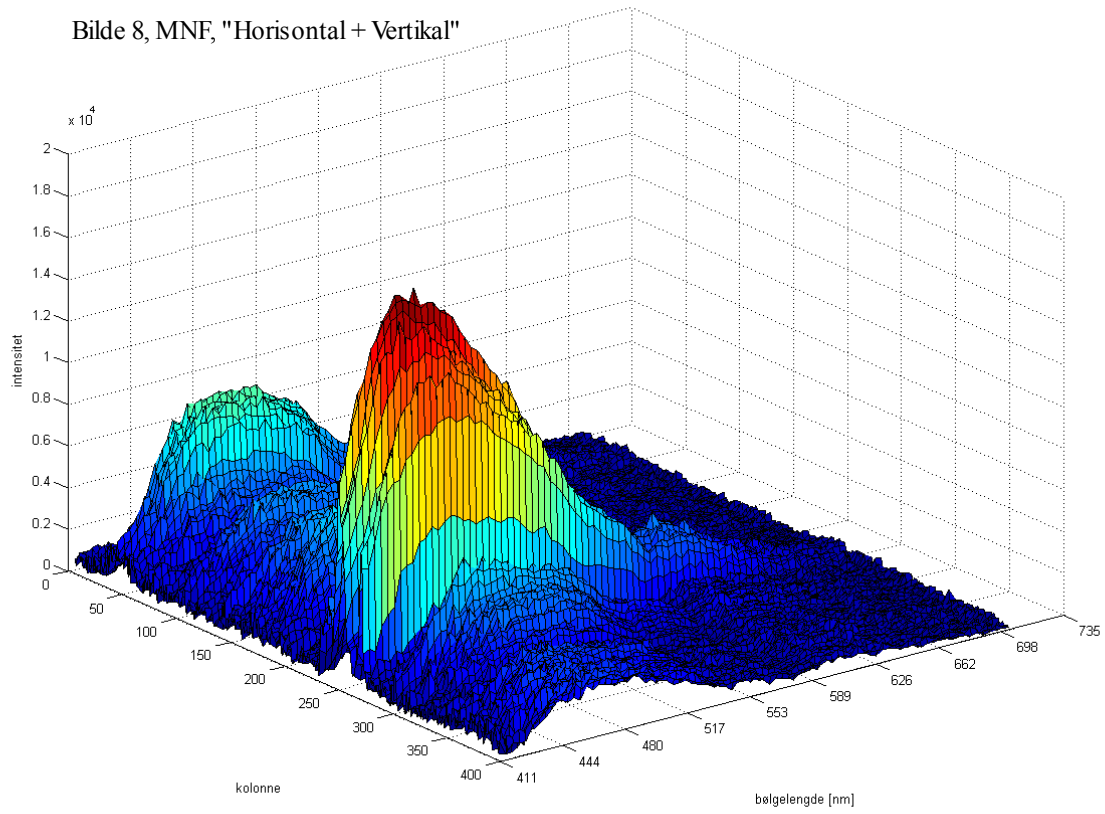
Bilde 8, Udempet spekter



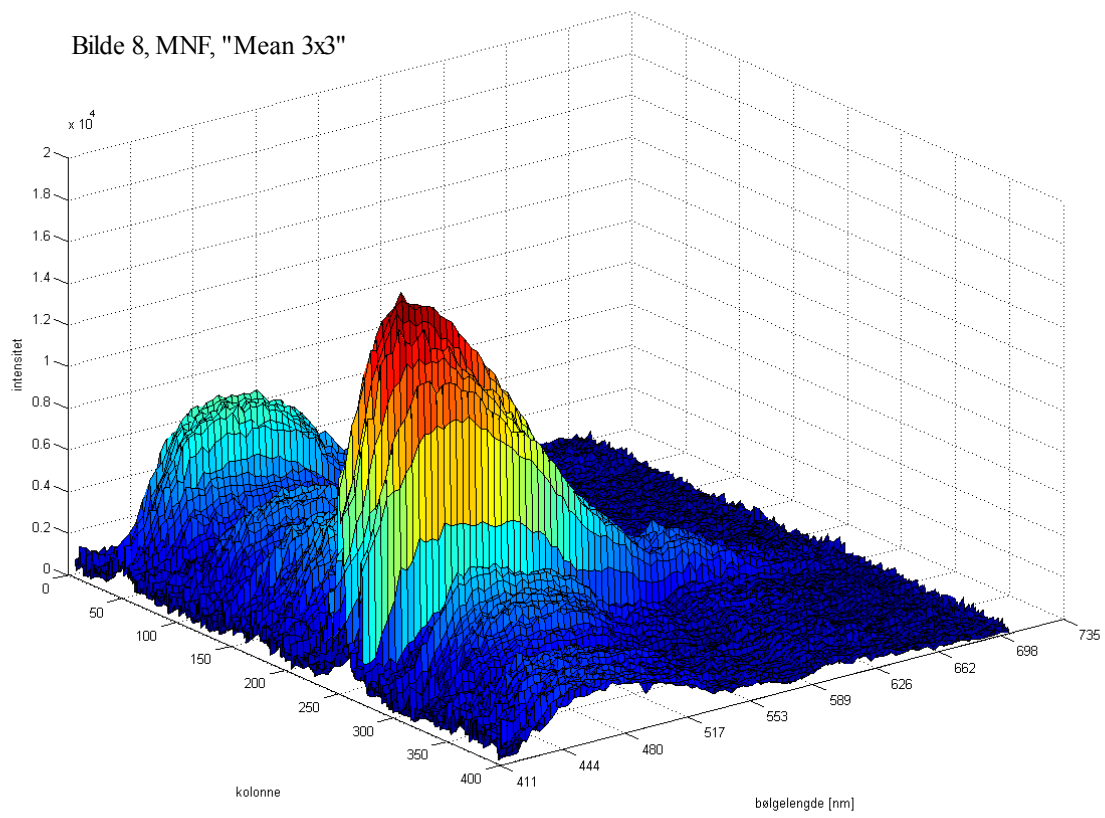
Bilde 8, PCA



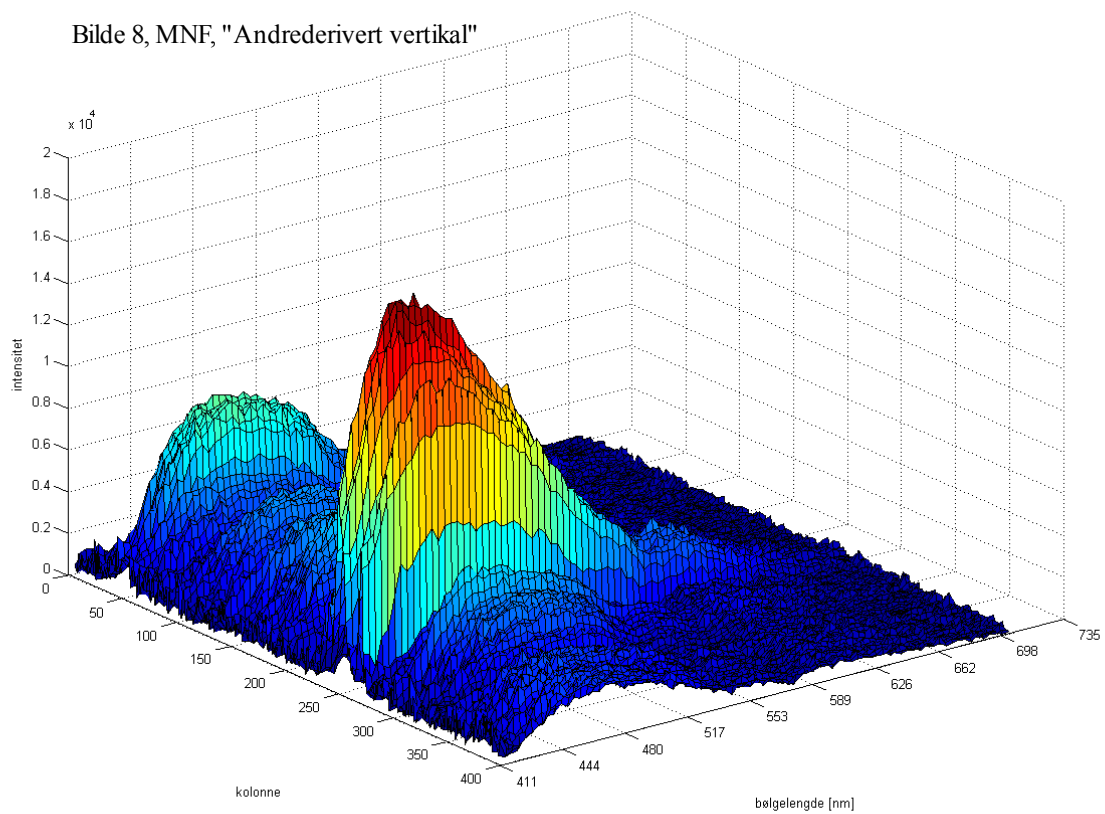
Bilde 8, MNF, "Horisontal + Vertikal"



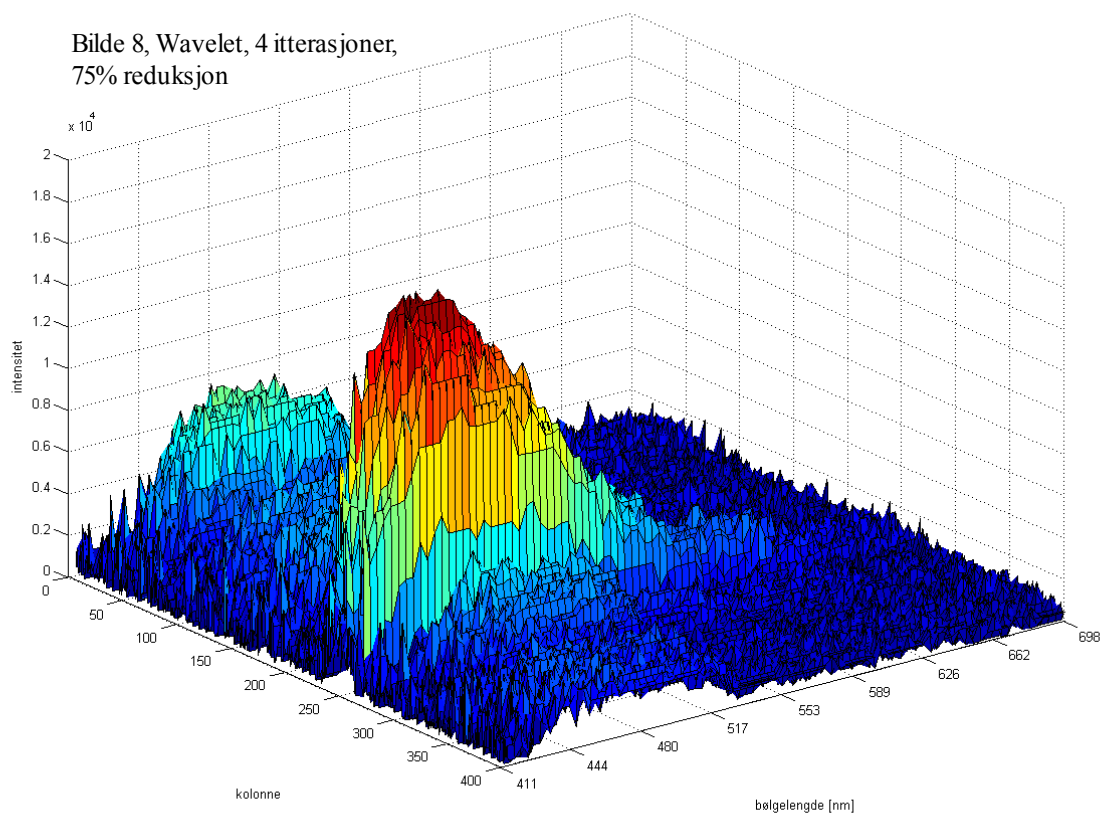
Bilde 8, MNF, "Mean 3x3"



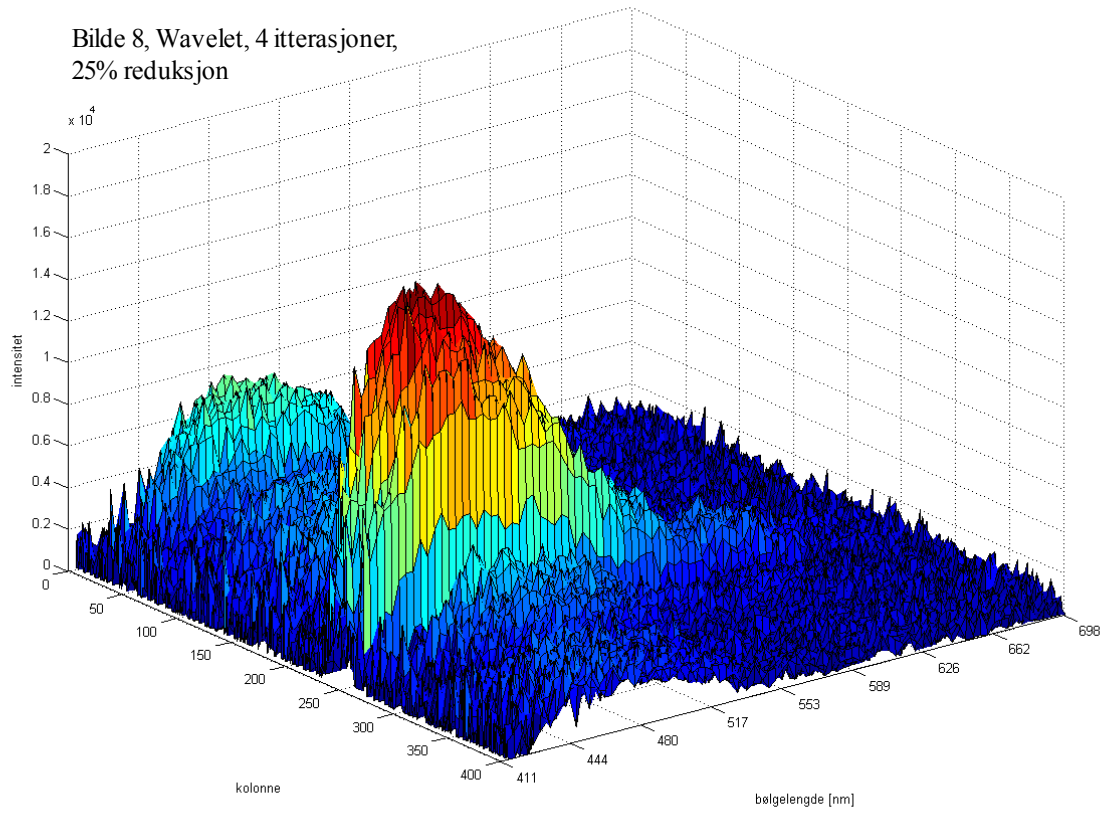
Bilde 8, MNF, "Andrederivert vertikalt"



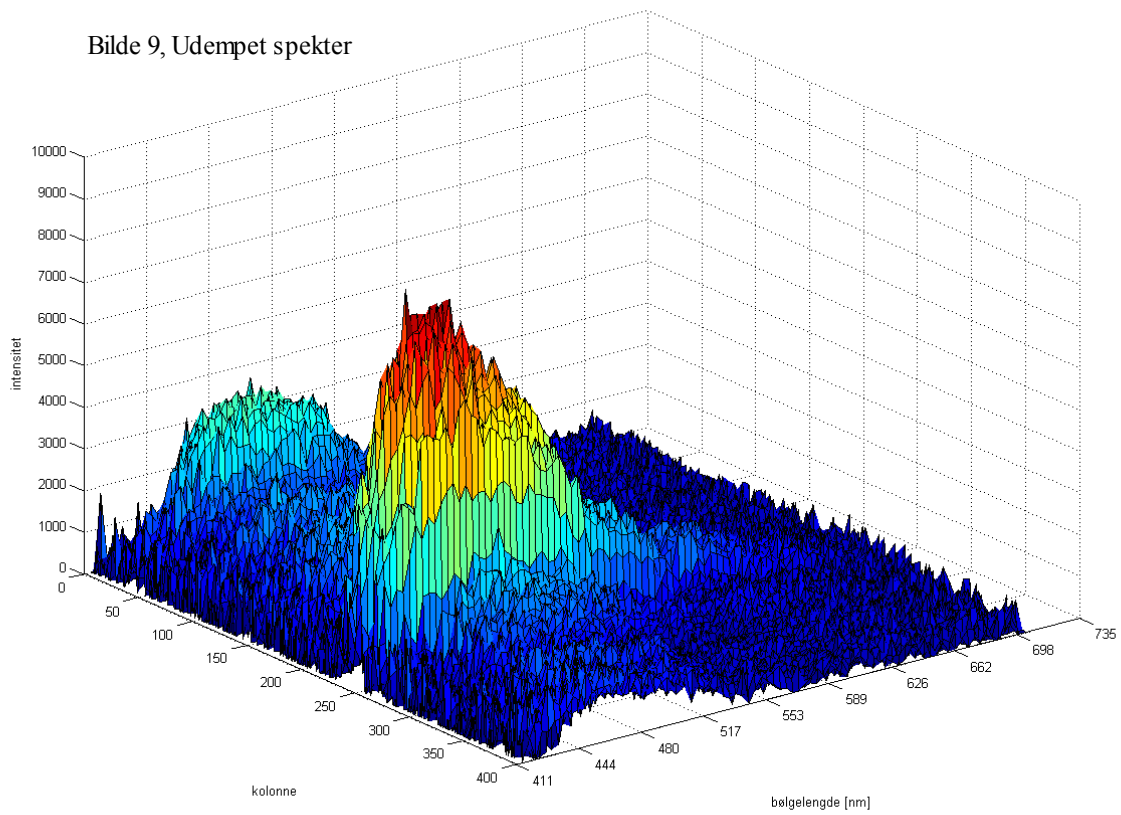
Bilde 8, Wavelet, 4 itterasjoner, 75% reduksjon



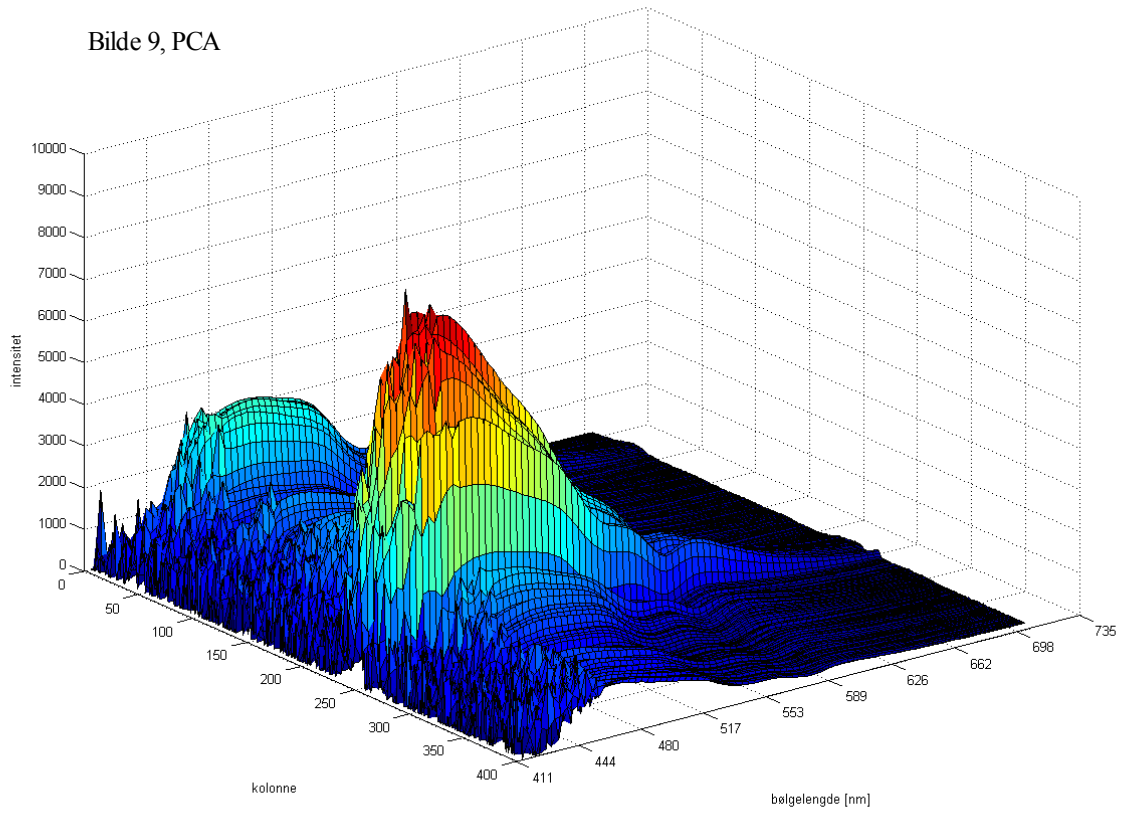
Bilde 8, Wavelet, 4 itterasjoner,
25% reduksjon



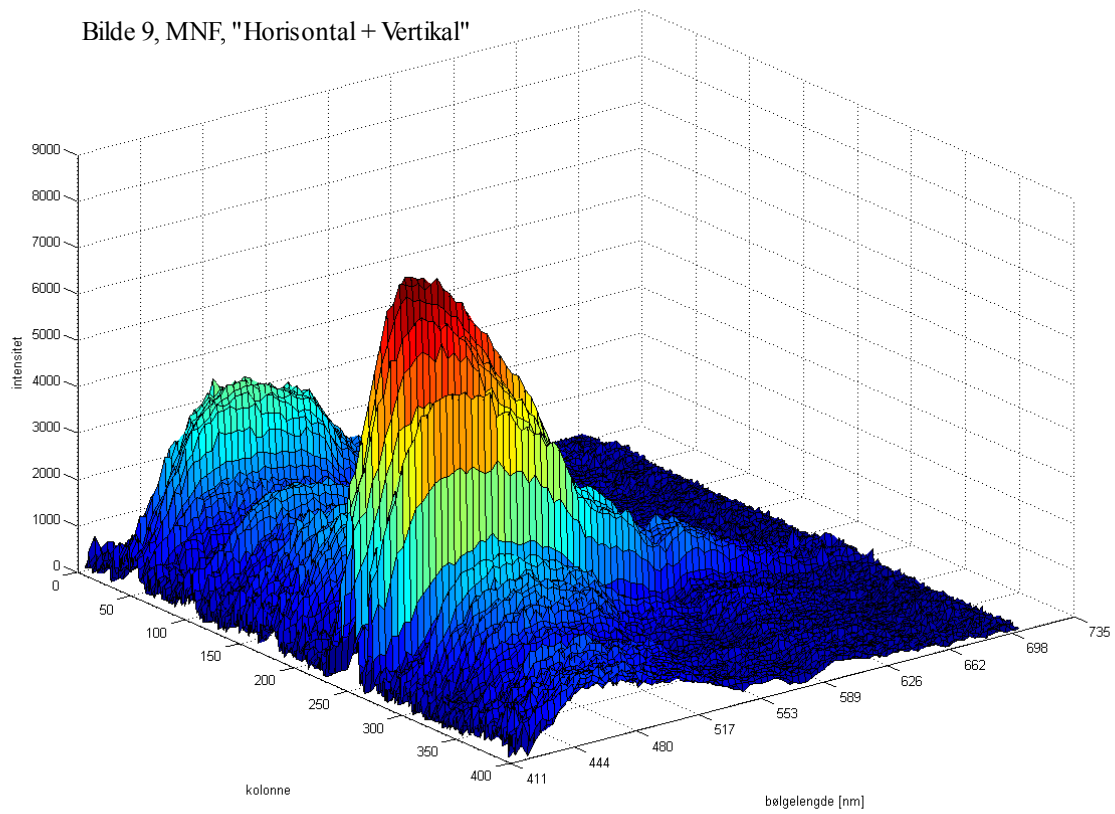
Bilde 9, Udempet spekter



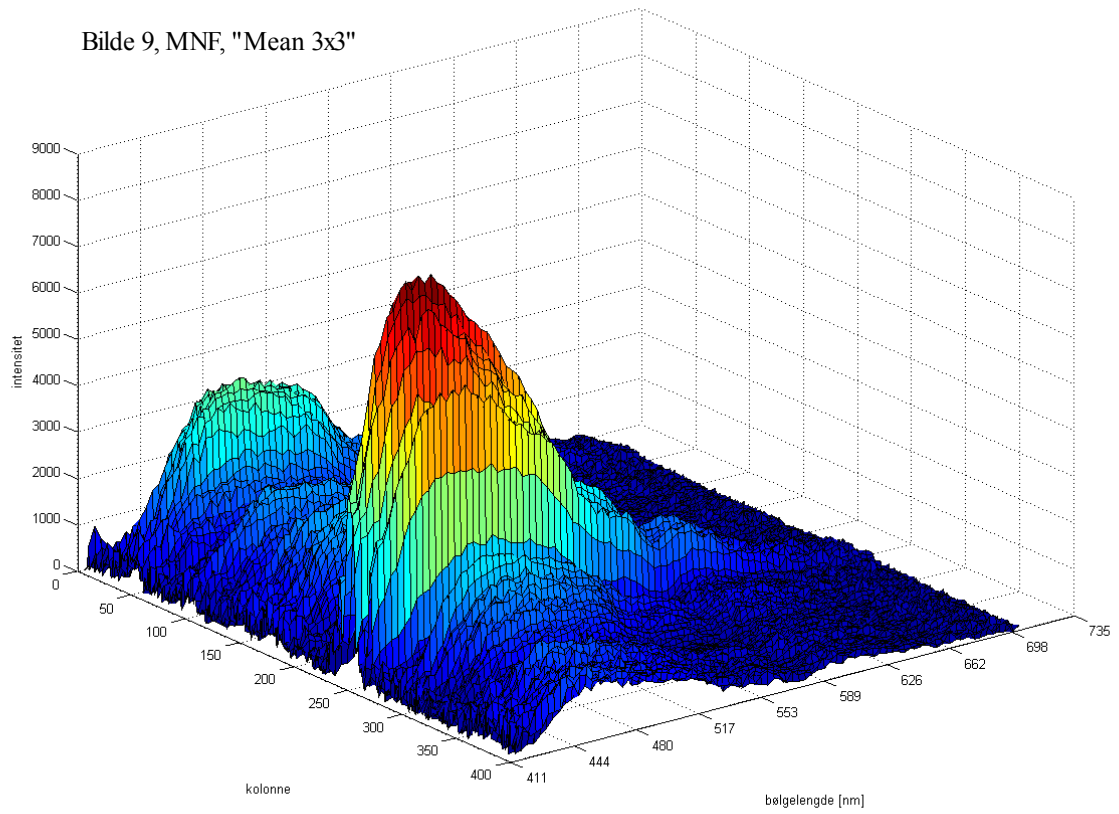
Bilde 9, PCA



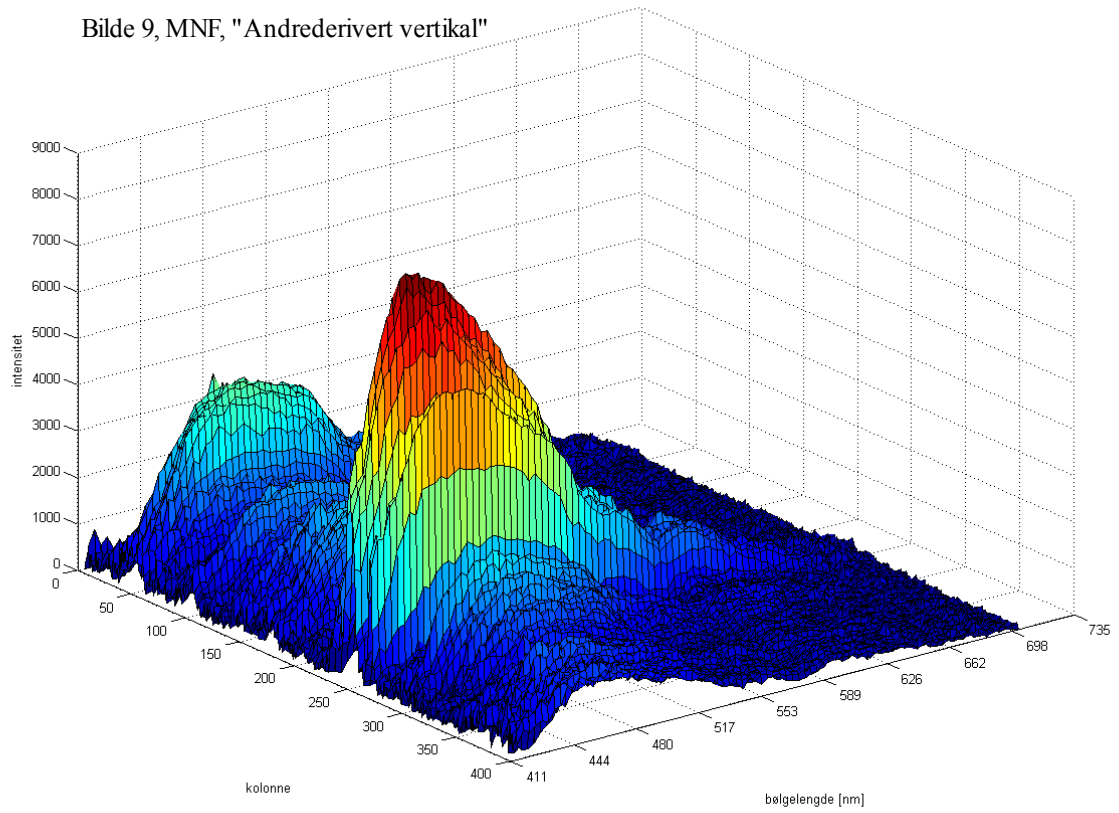
Bilde 9, MNF, "Horizontal + Vertikal"



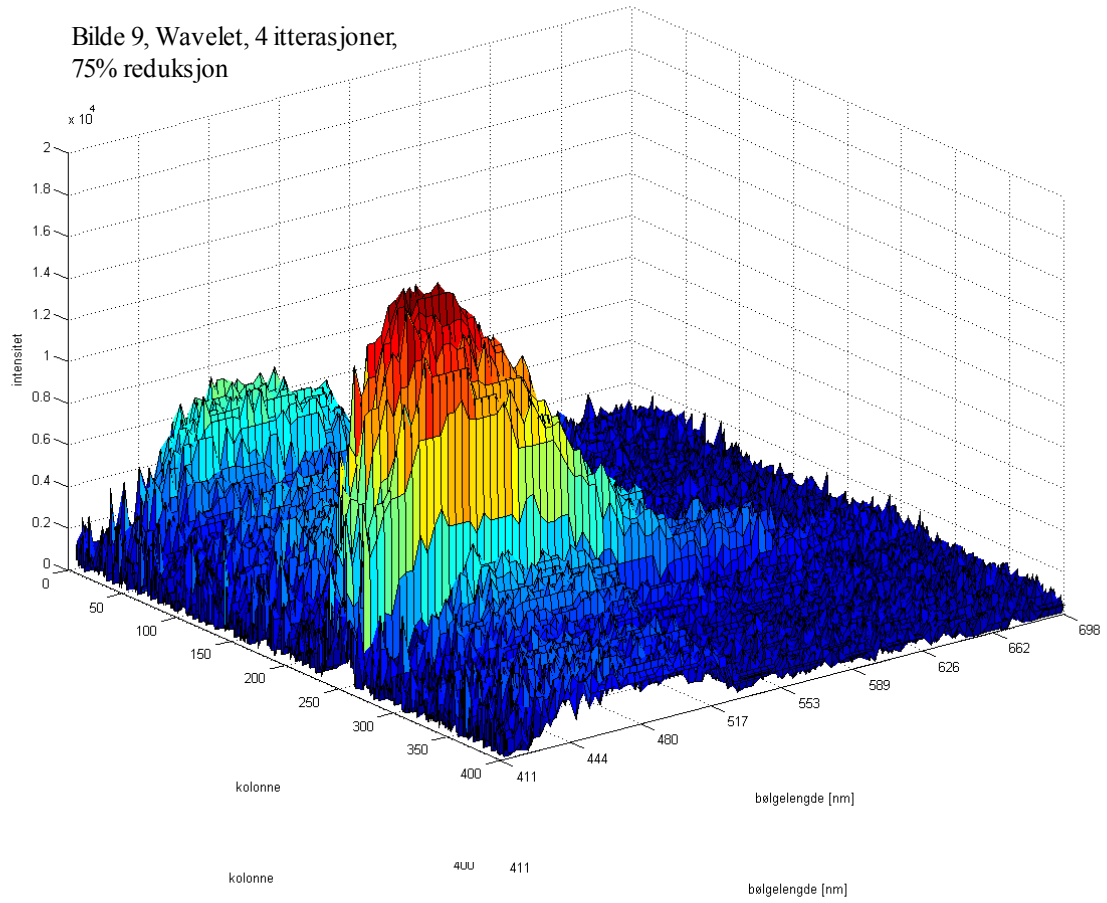
Bilde 9, MNF, "Mean 3x3"



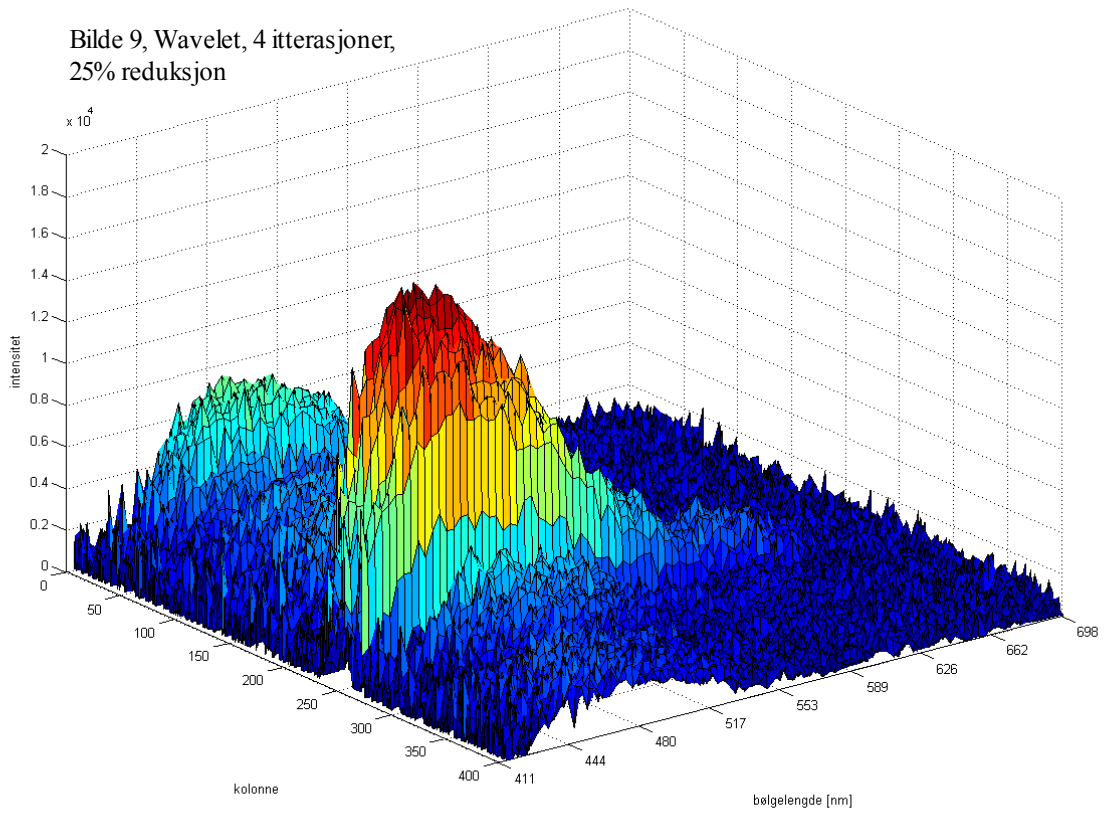
Bilde 9, MNF, "Andrederivert vertikalt"



Bilde 9, Wavelet, 4 itterasjoner,
75% reduksjon

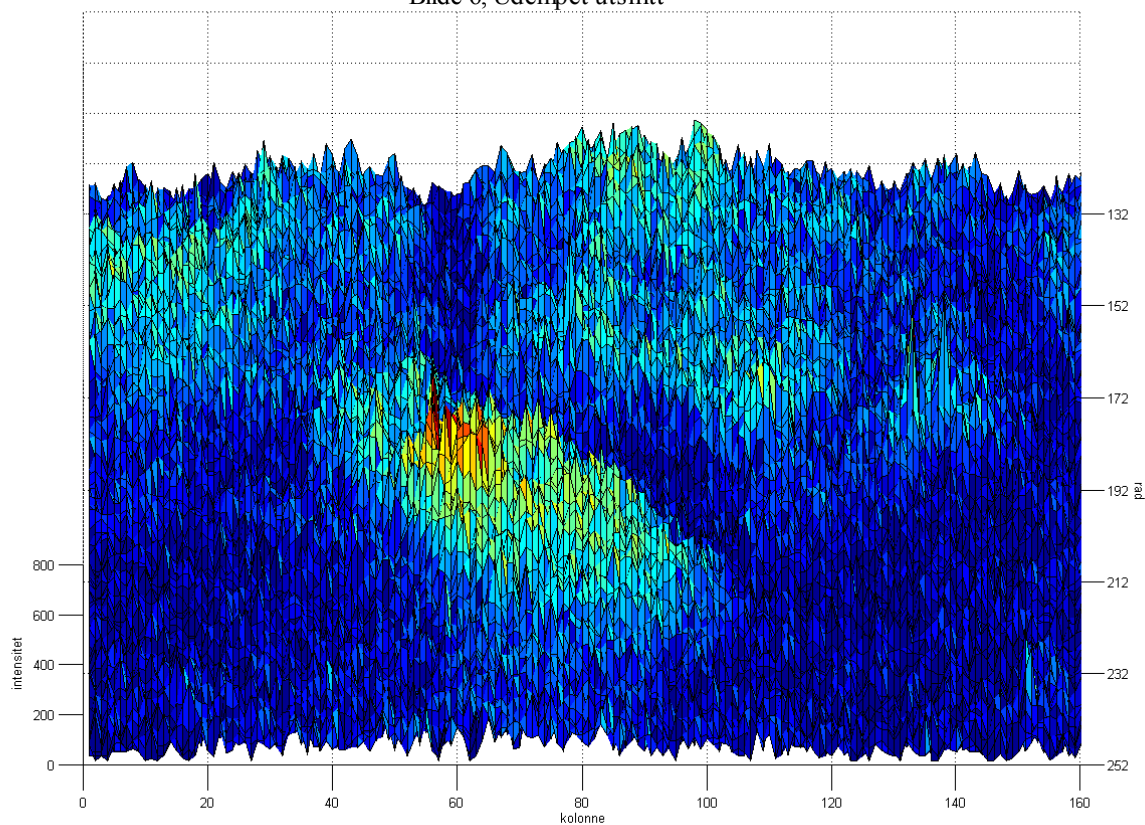


Bilde 9, Wavelet, 4 itterasjoner,
25% reduksjon

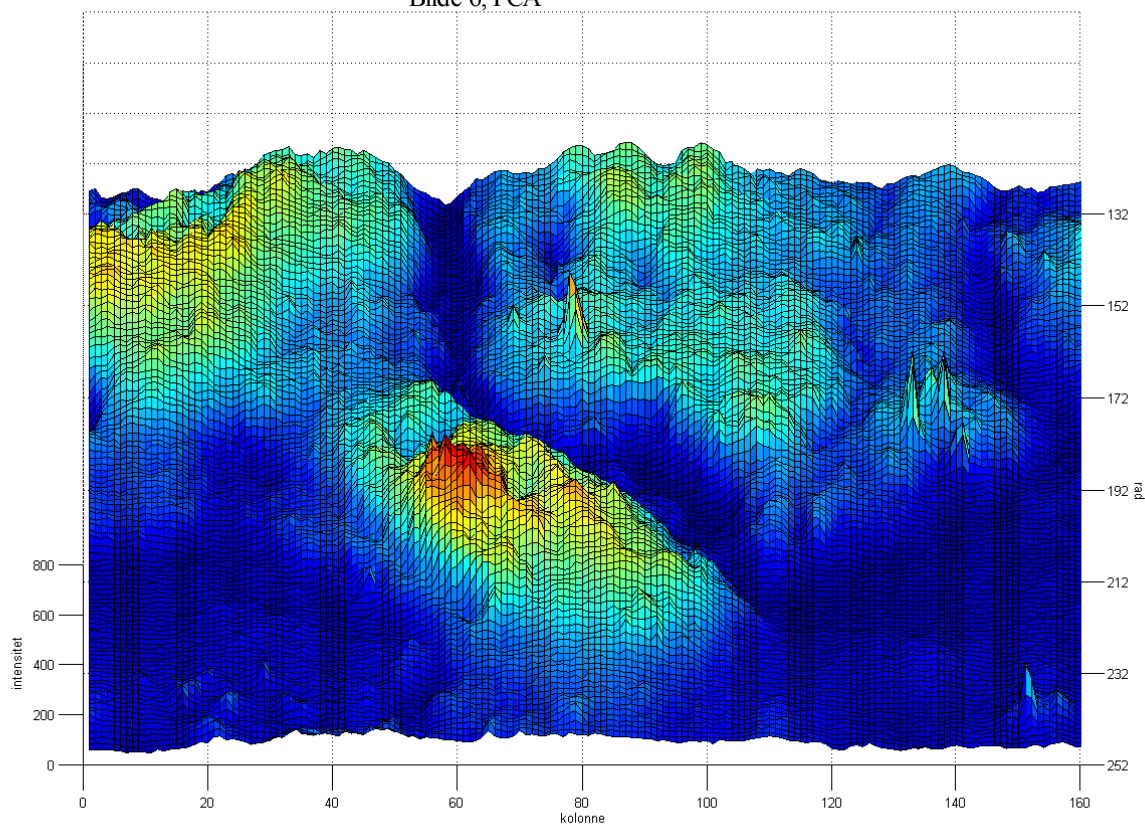


8.9 3D-utsnitt for spektralbånd 10 (444nm)

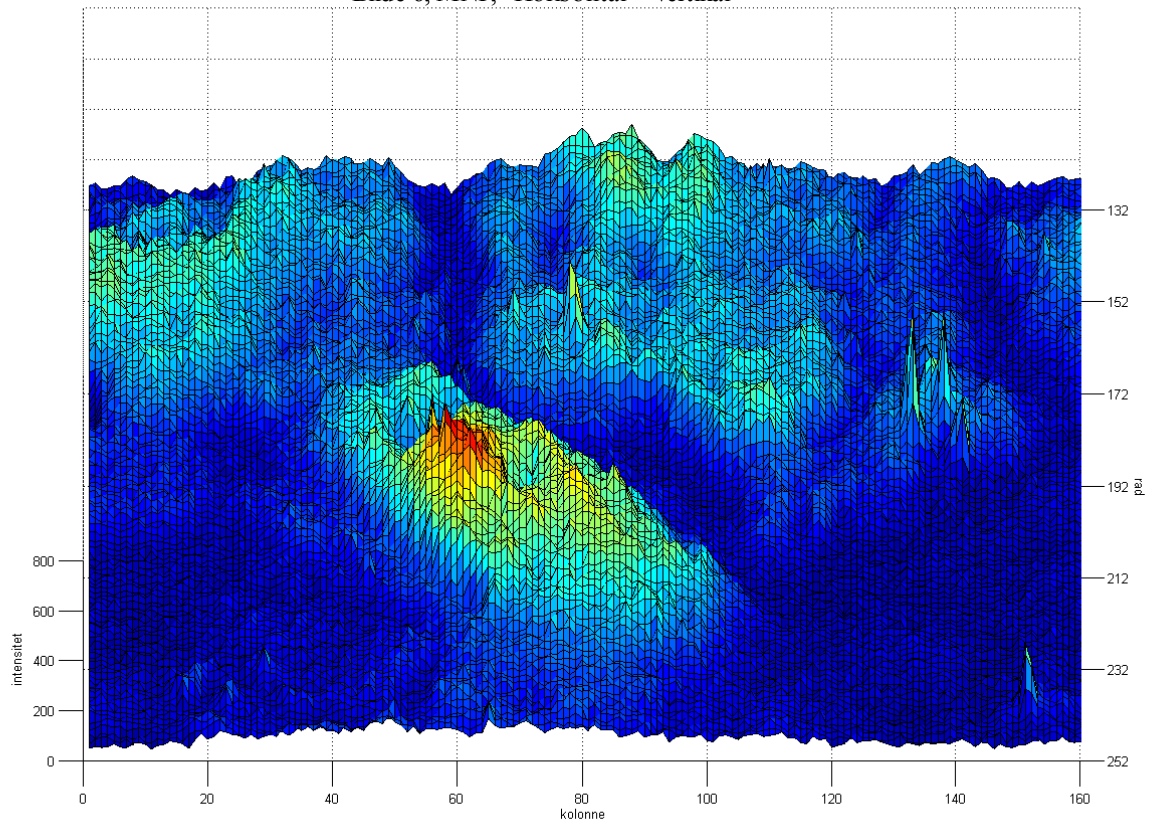
Bilde 6, Udempet utsnitt



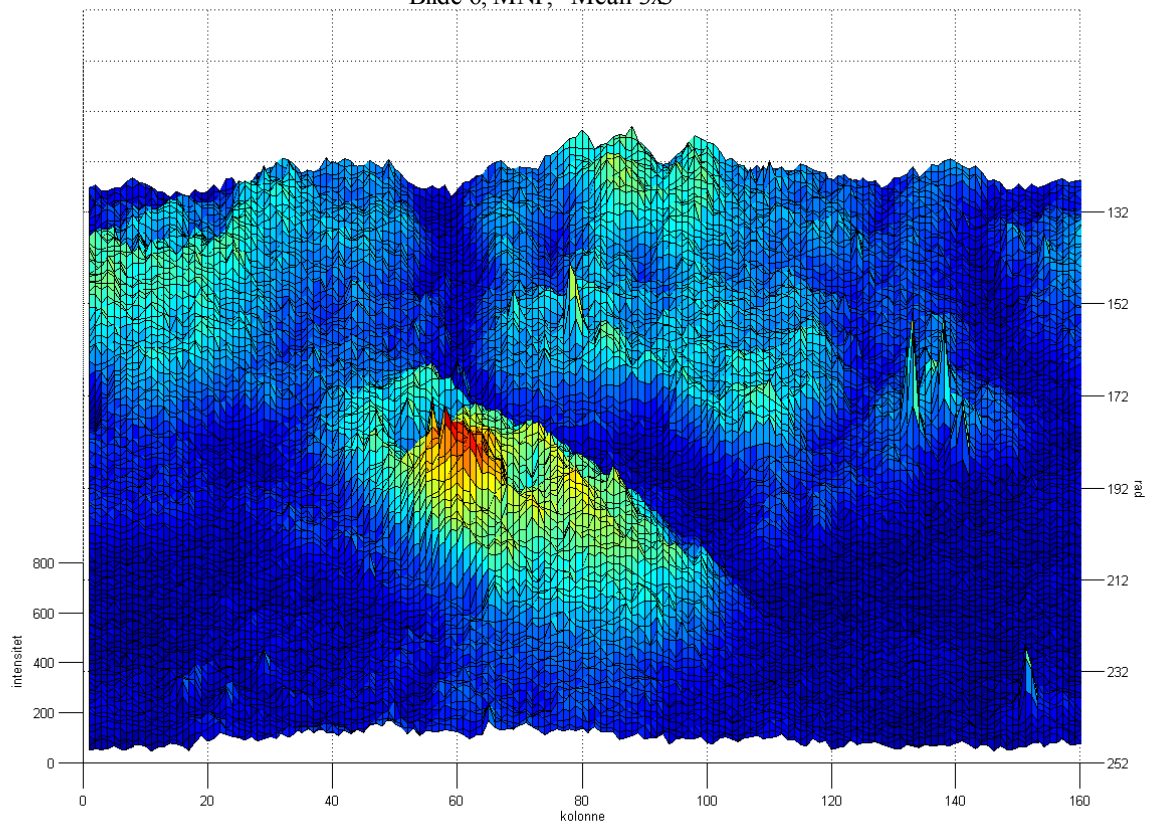
Bilde 6, PCA



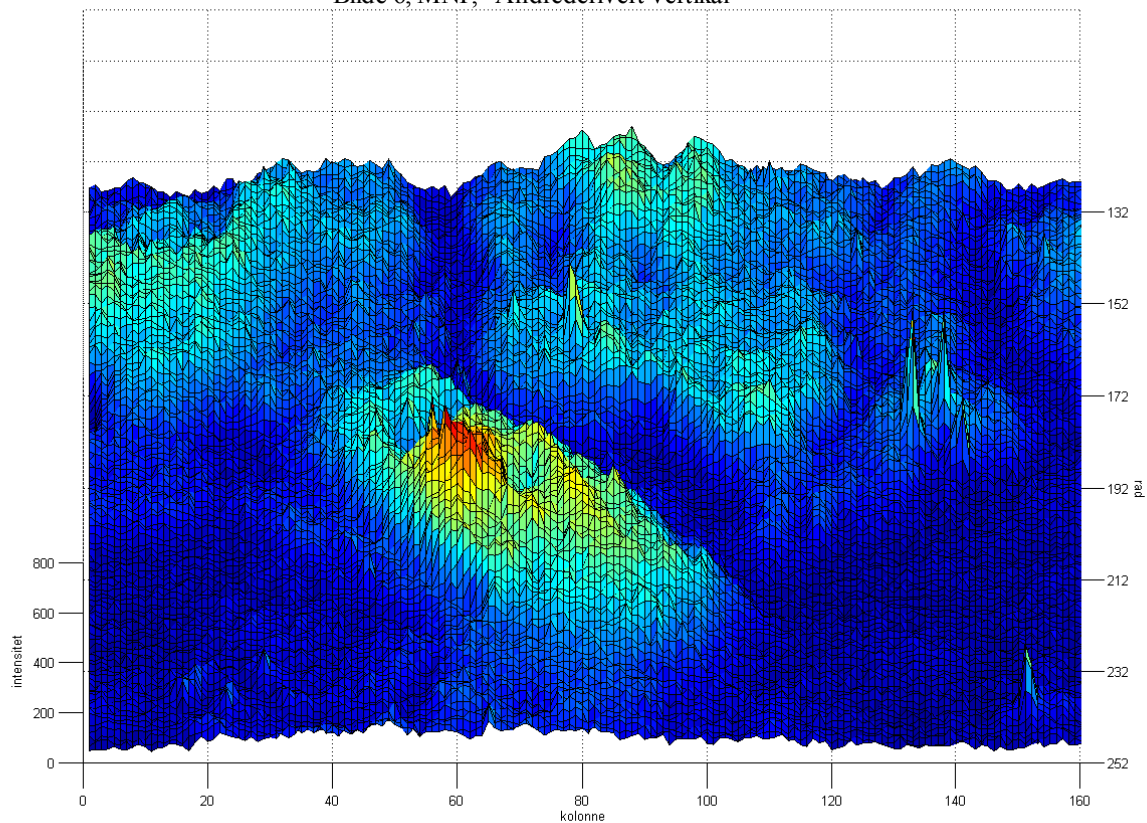
Bilde 6, MNF, "Horisontal + Vertikal"



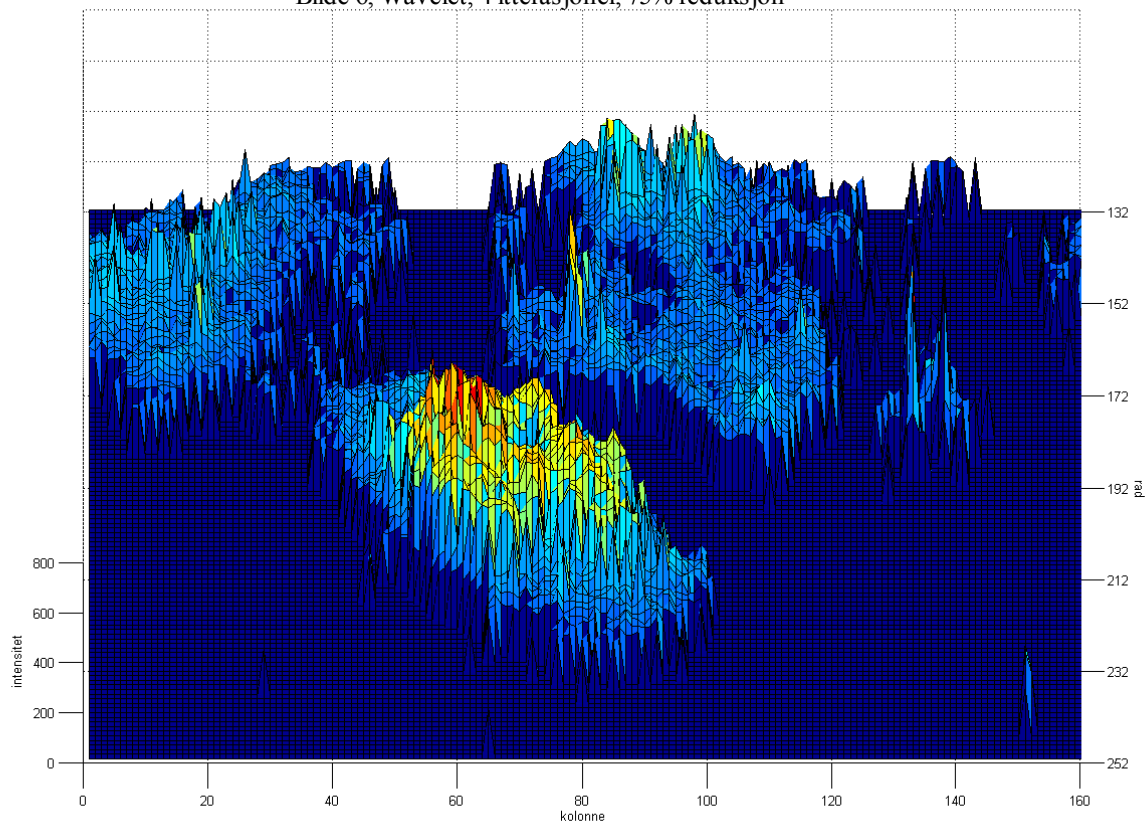
Bilde 6, MNF, "Mean 3x3"



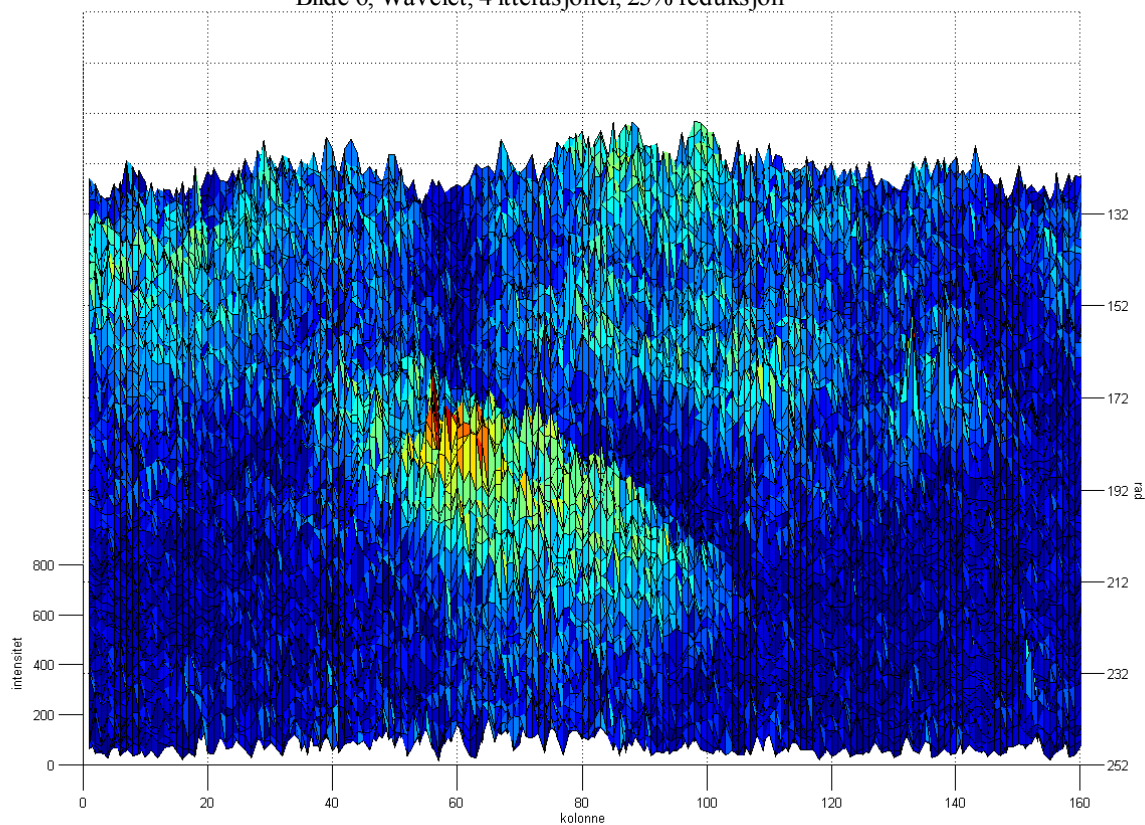
Bilde 6, MNF, "Andrederivert vertikal"



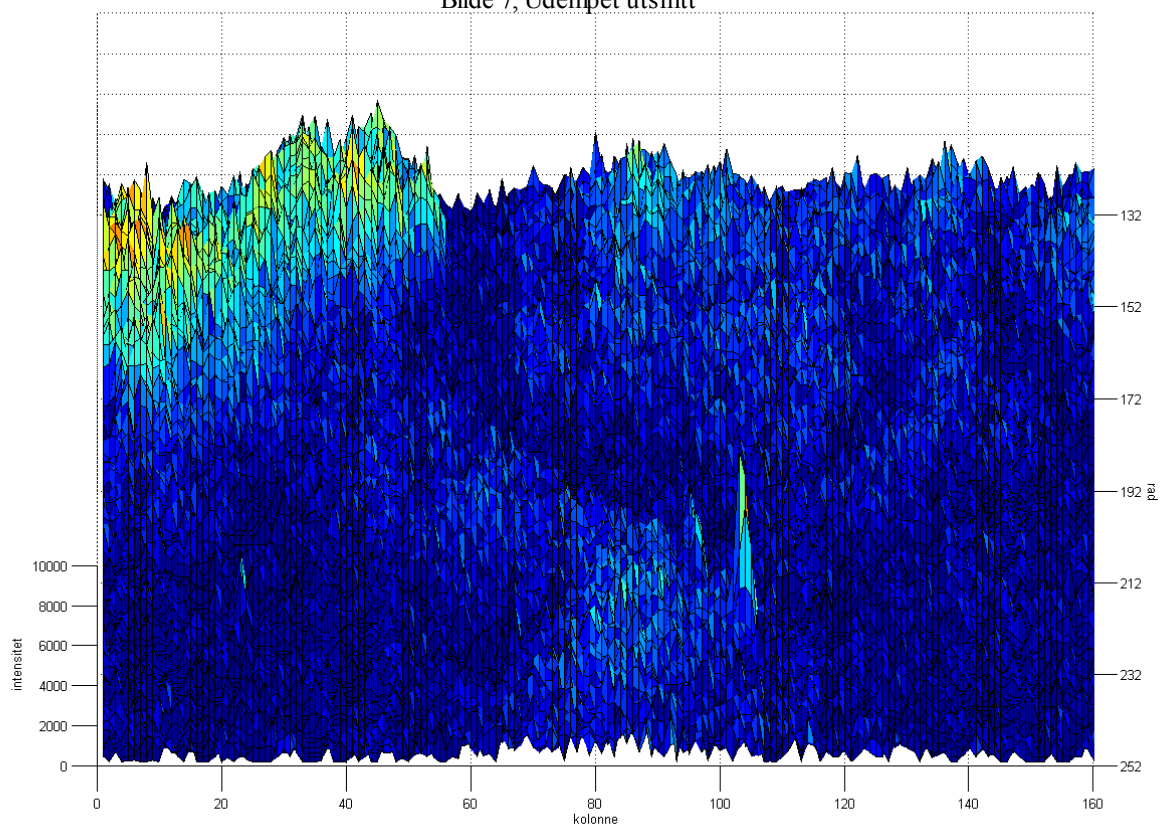
Bilde 6, Wavelet, 4 itterasjoner, 75% reduksjon



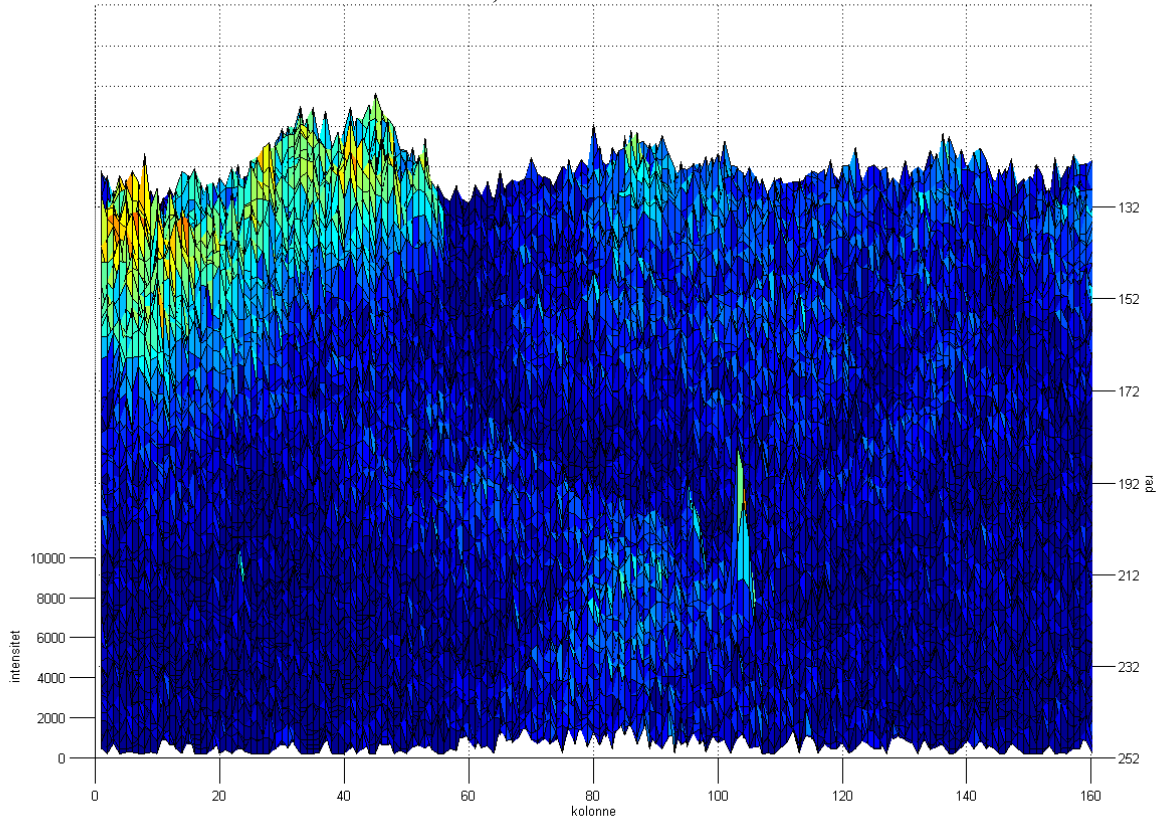
Bilde 6, Wavelet, 4 itterasjoner, 25% reduksjon



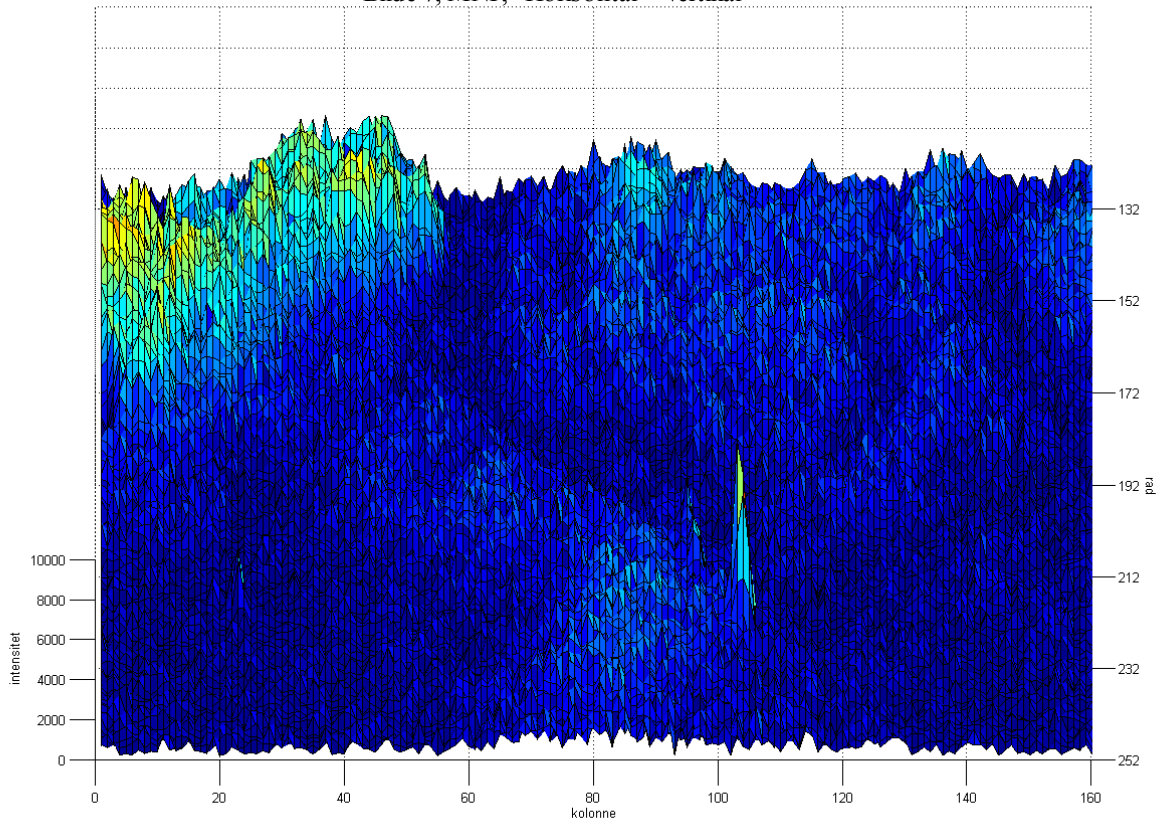
Bilde 7, Udempet utsnitt



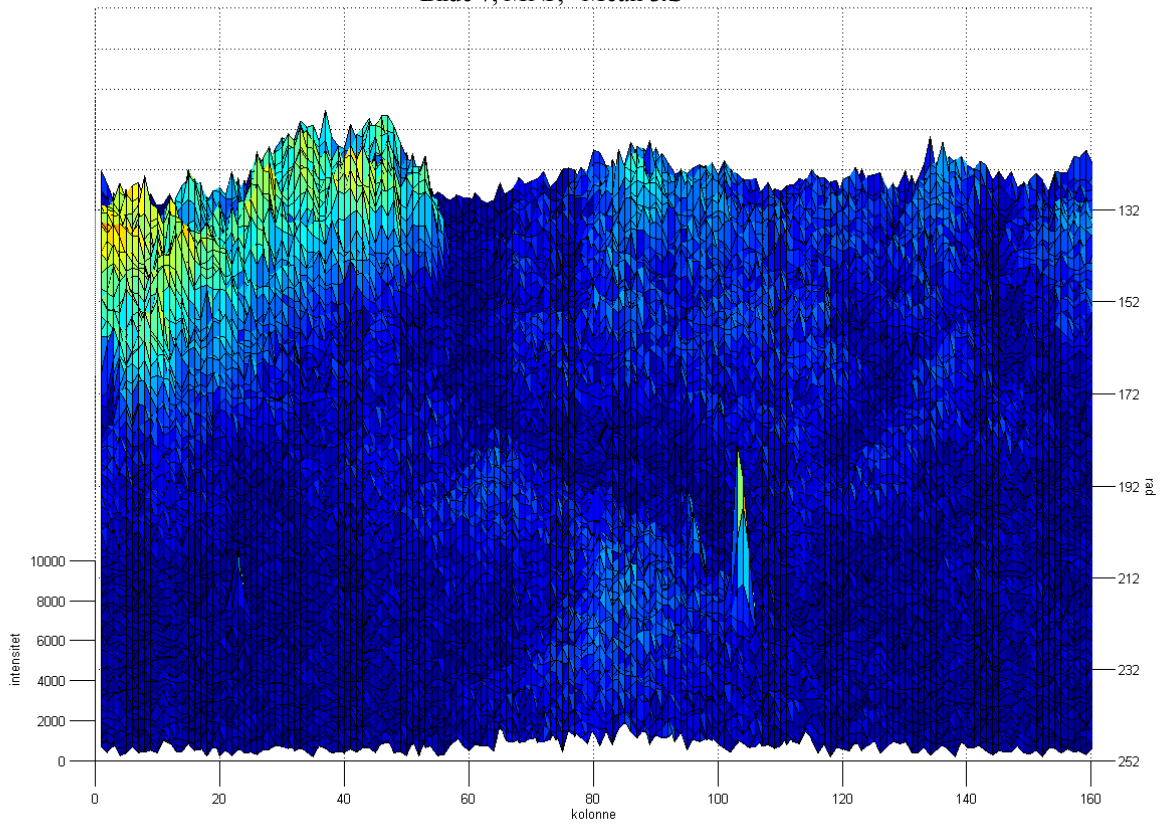
Bilde 7, PCA



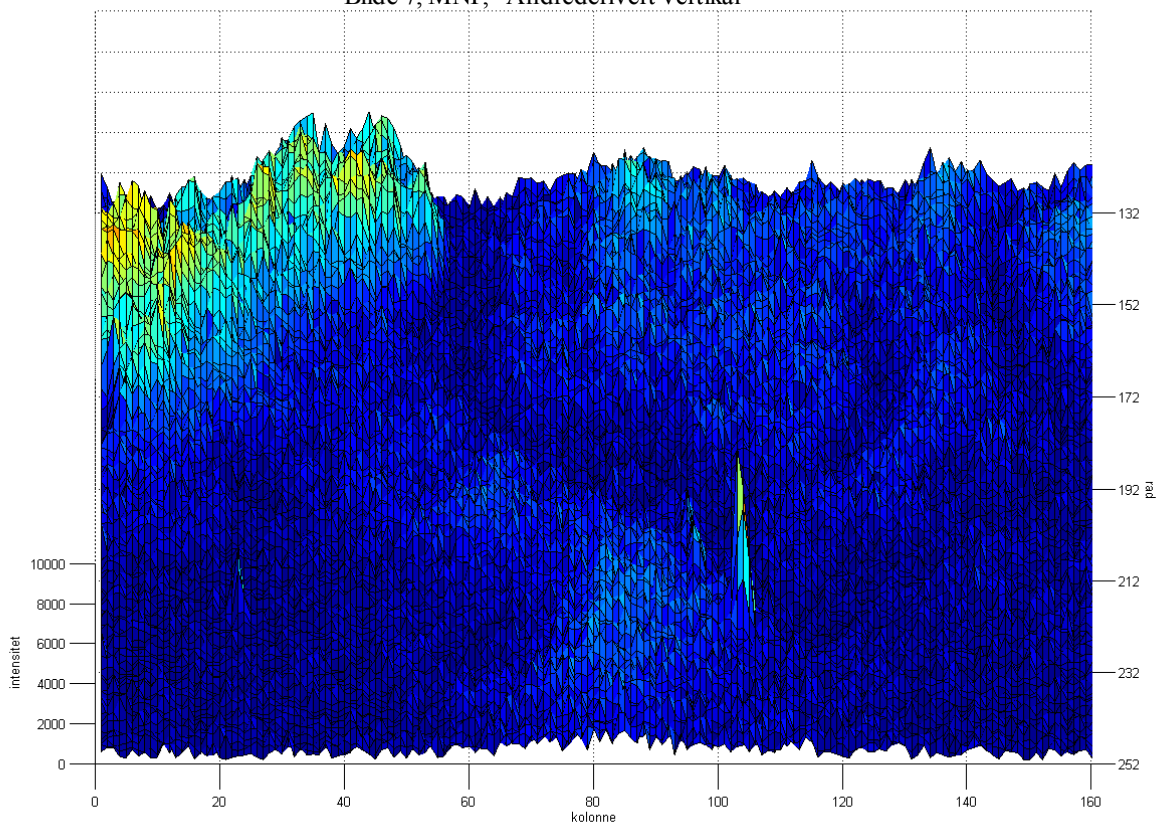
Bilde 7, MNF, "Horisontal + Vertikal"



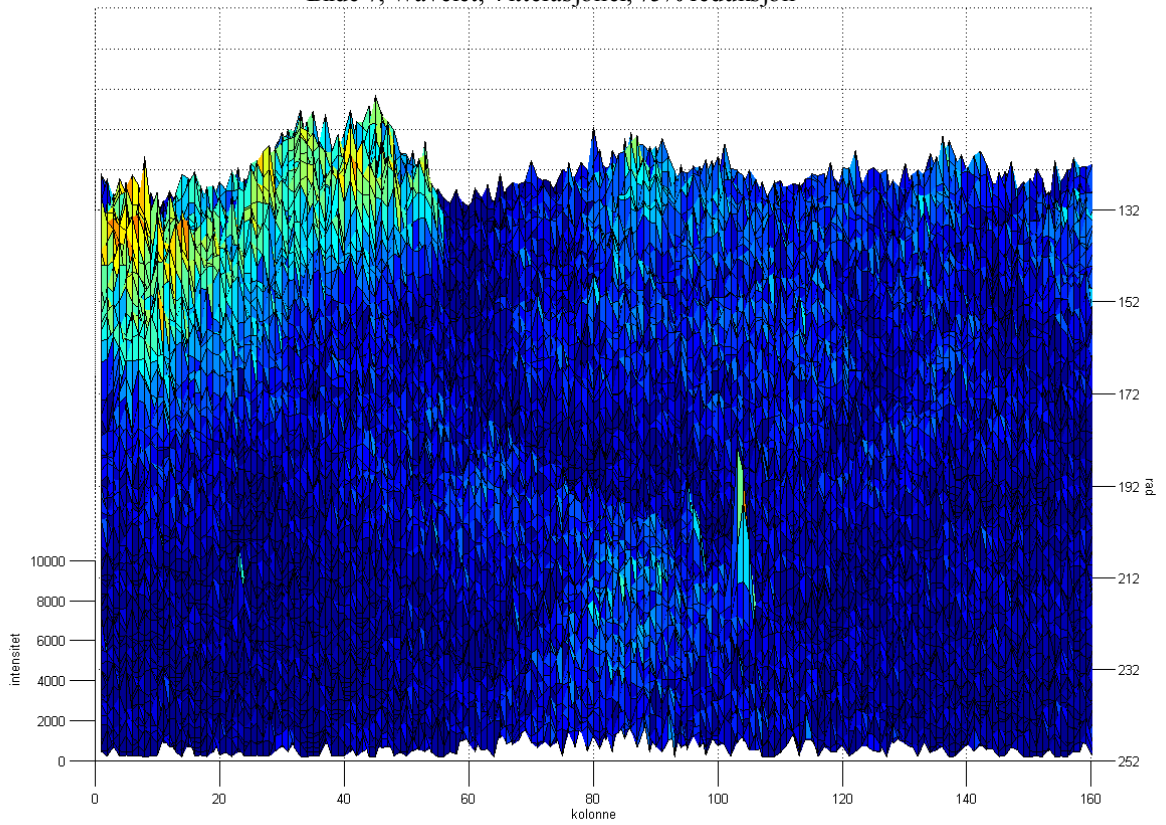
Bilde 7, MNF, "Mean 3x3"



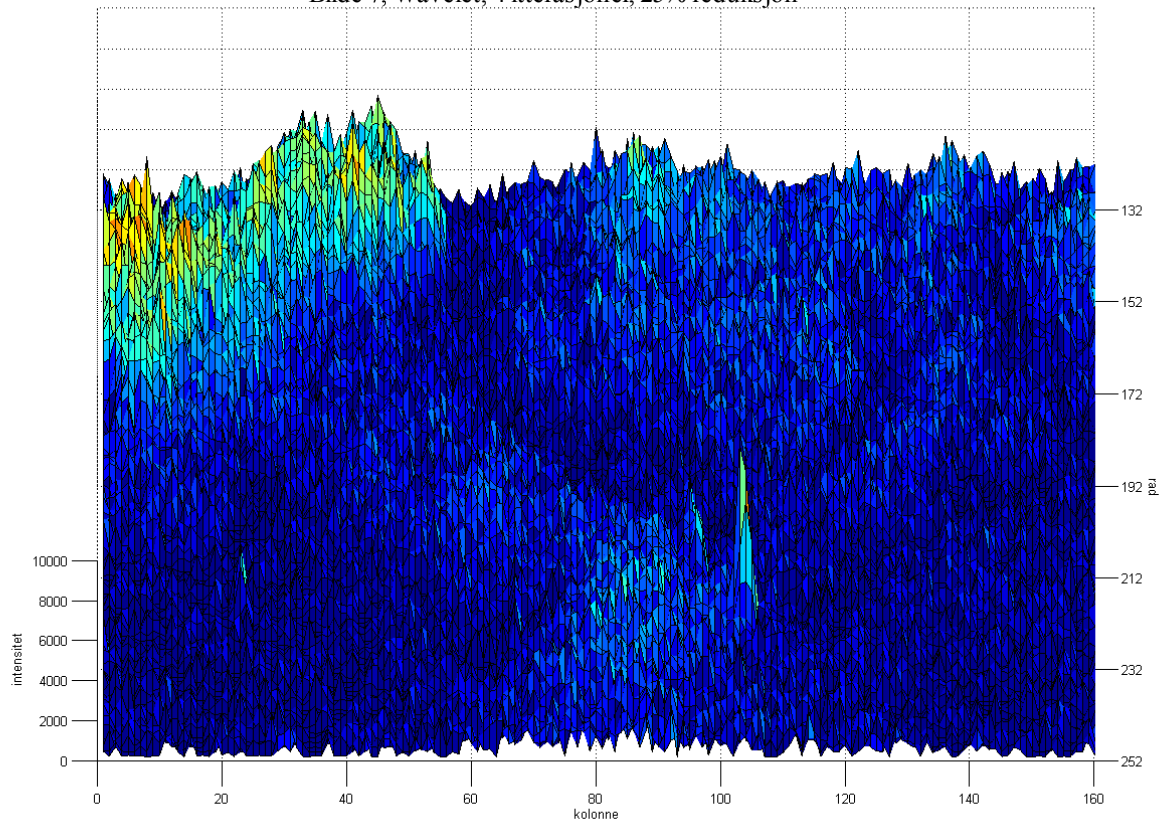
Bilde 7, MNF, "Andrederivert vertikalt"



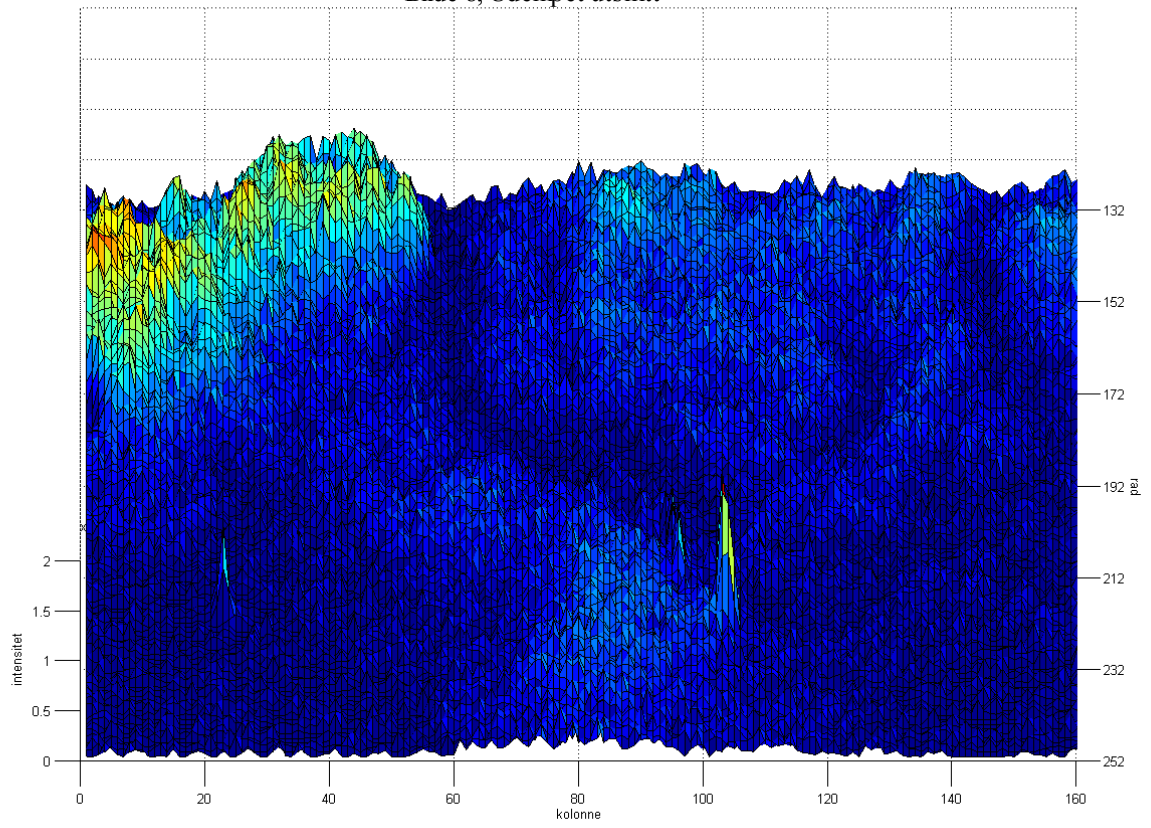
Bilde 7, Wavelet, 4 itterasjoner, 75% reduksjon



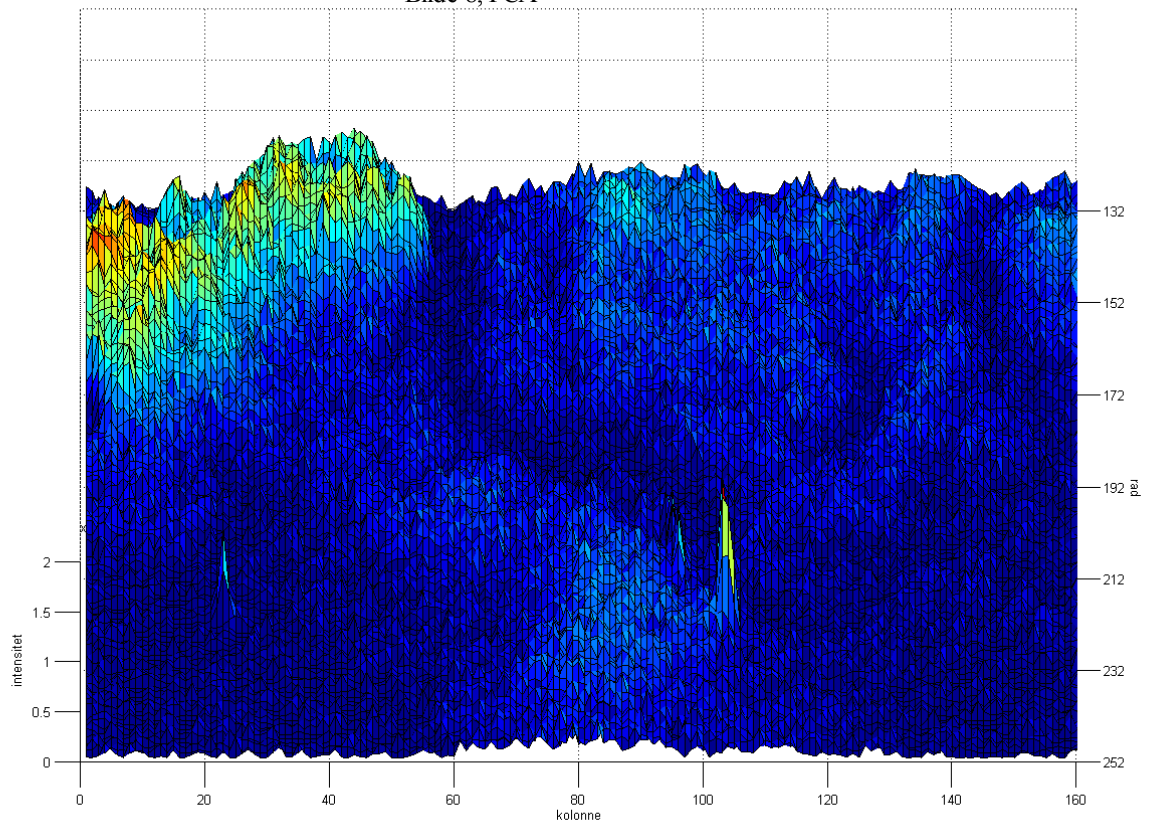
Bilde 7, Wavelet, 4 itterasjoner, 25% reduksjon



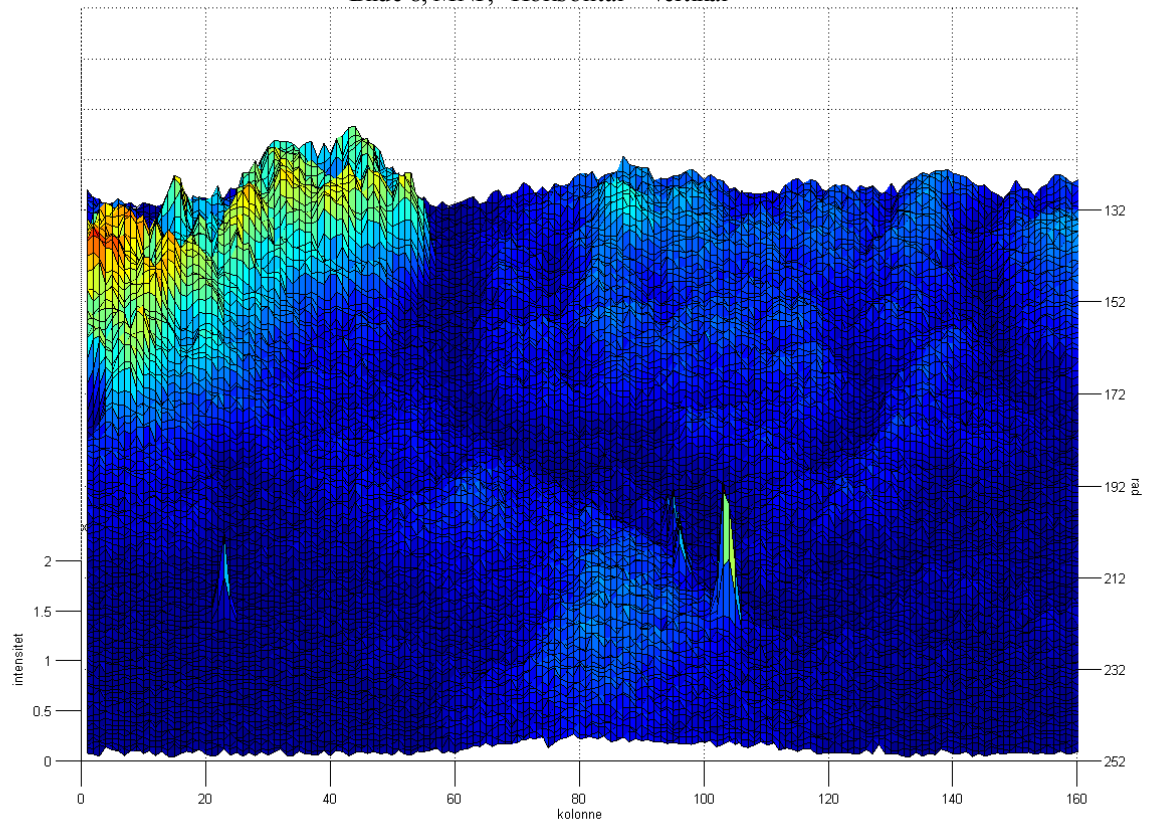
Bilde 8, Udempet utsnitt



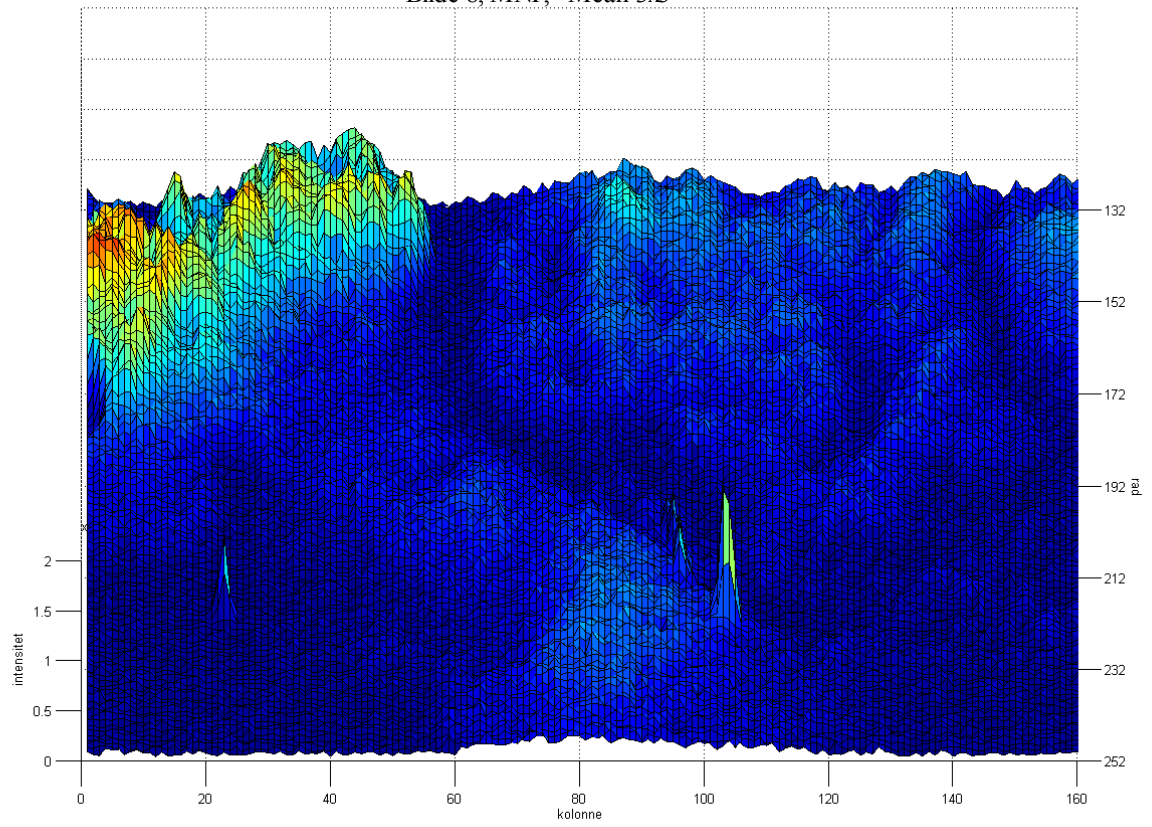
Bilde 8, PCA



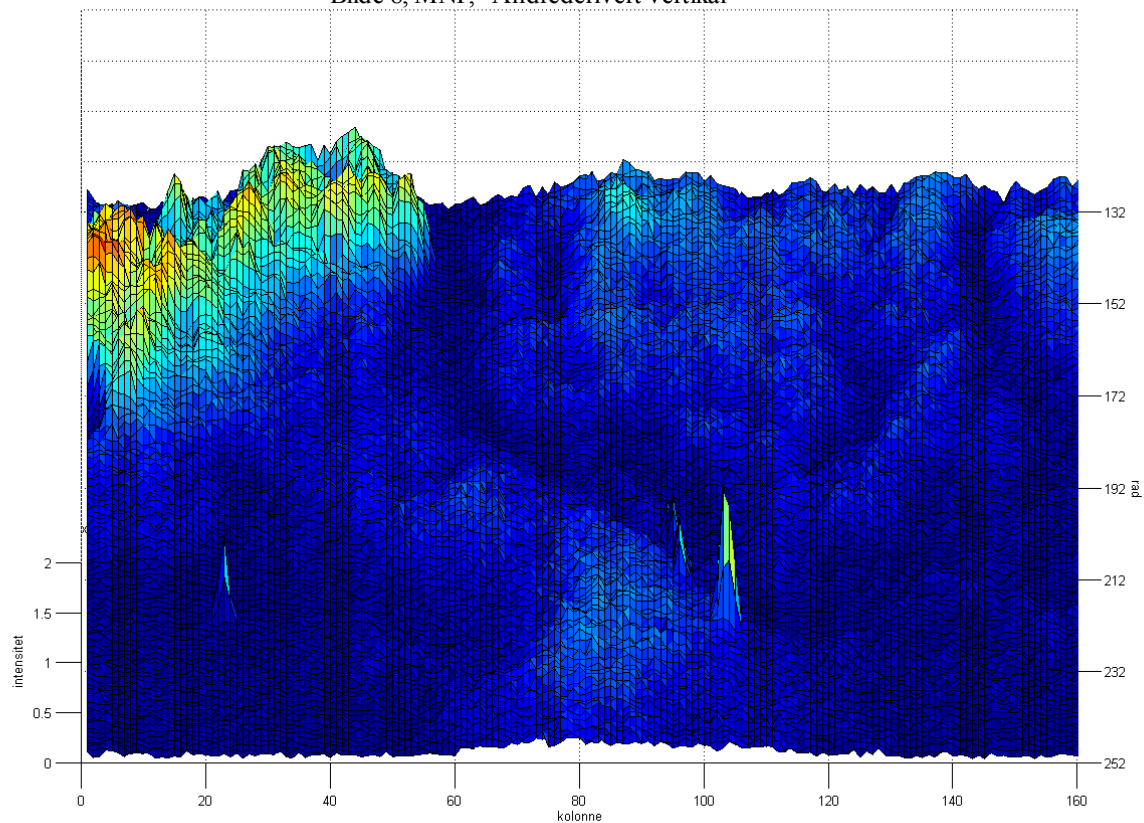
Bilde 8, MNF, "Horisontal + Vertikal"



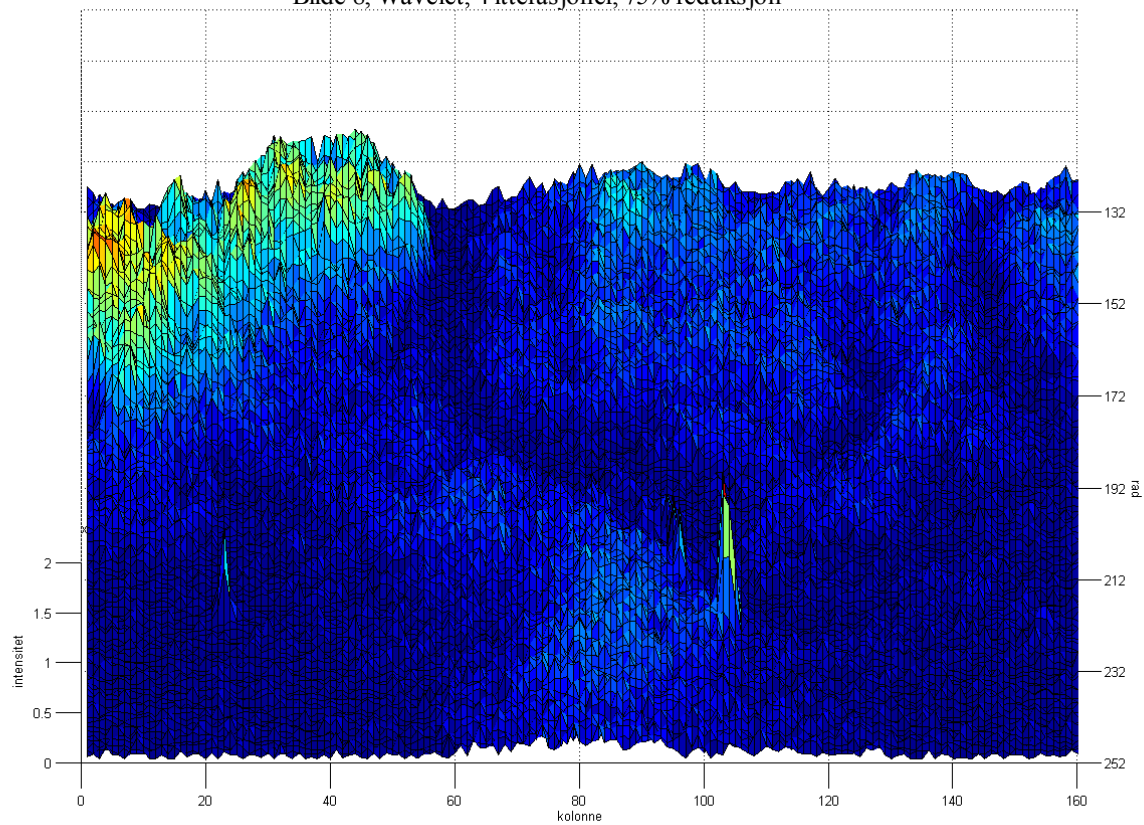
Bilde 8, MNF, "Mean 3x3"



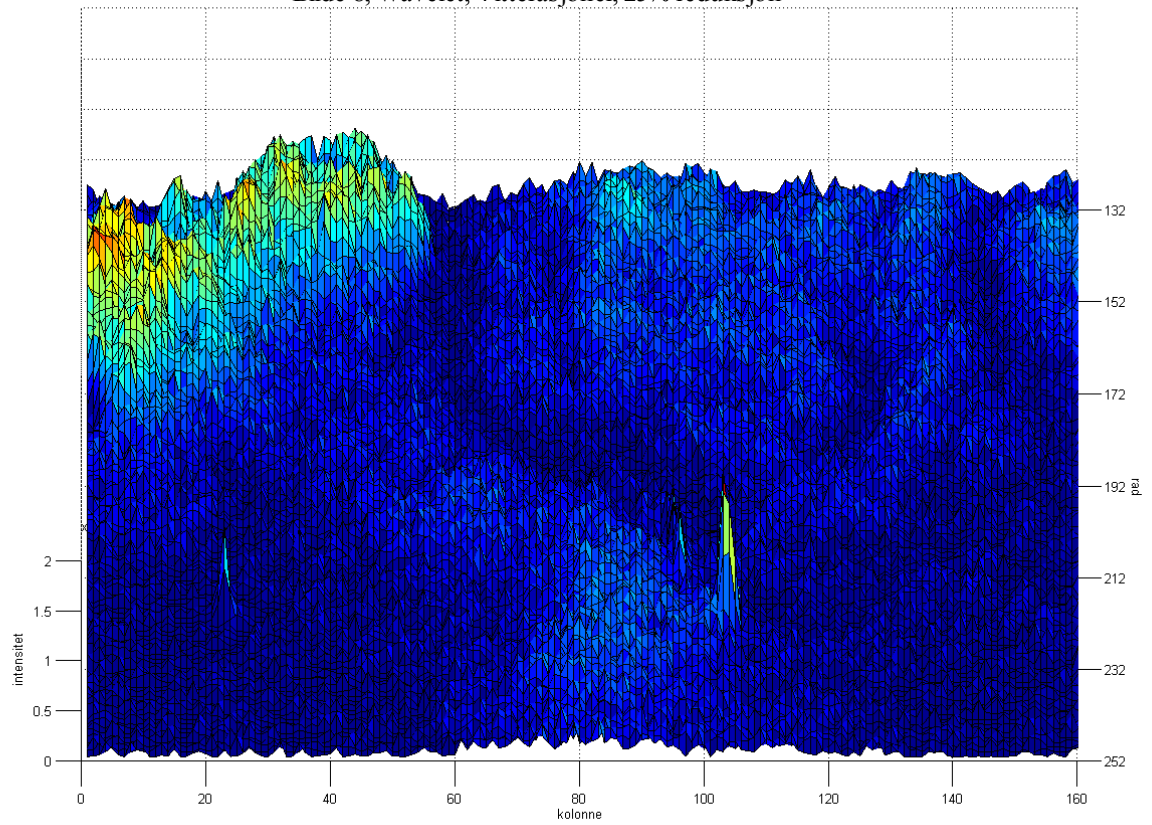
Bilde 8, MNF, "Andrederivert vertikalt"



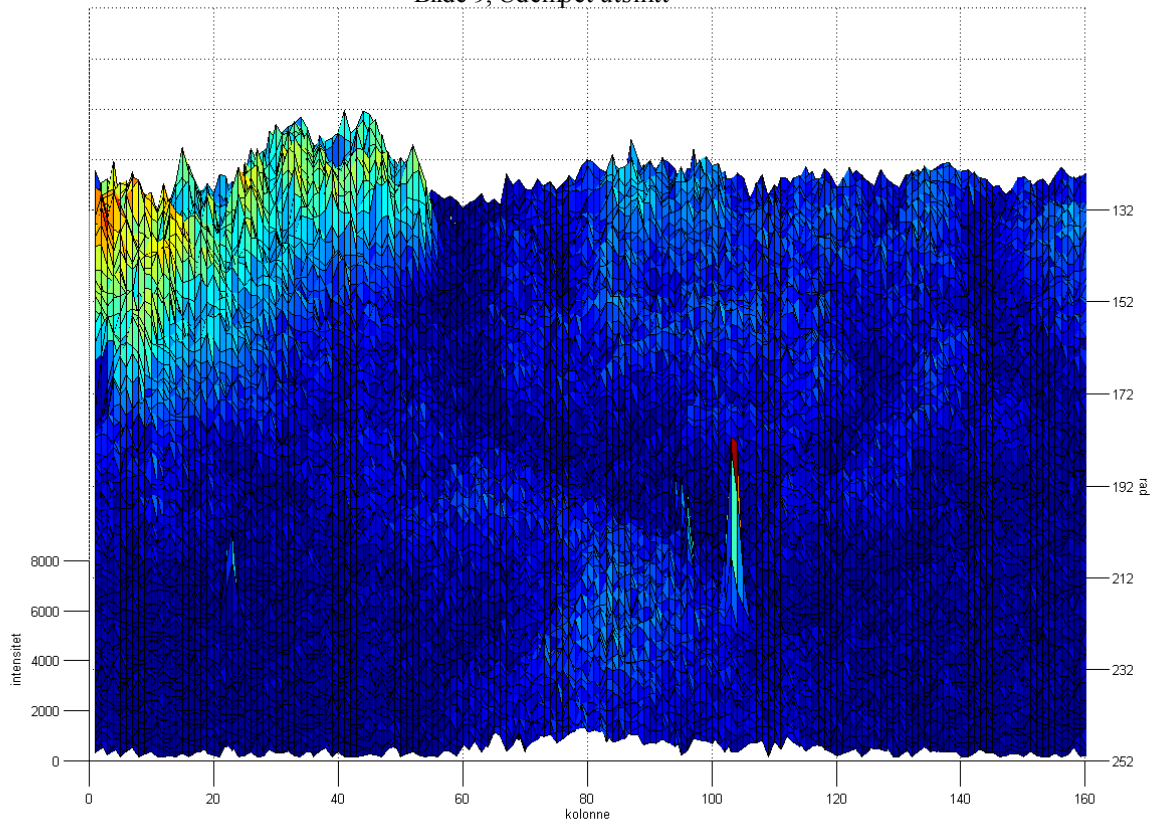
Bilde 8, Wavelet, 4 itterasjoner, 75% reduksjon



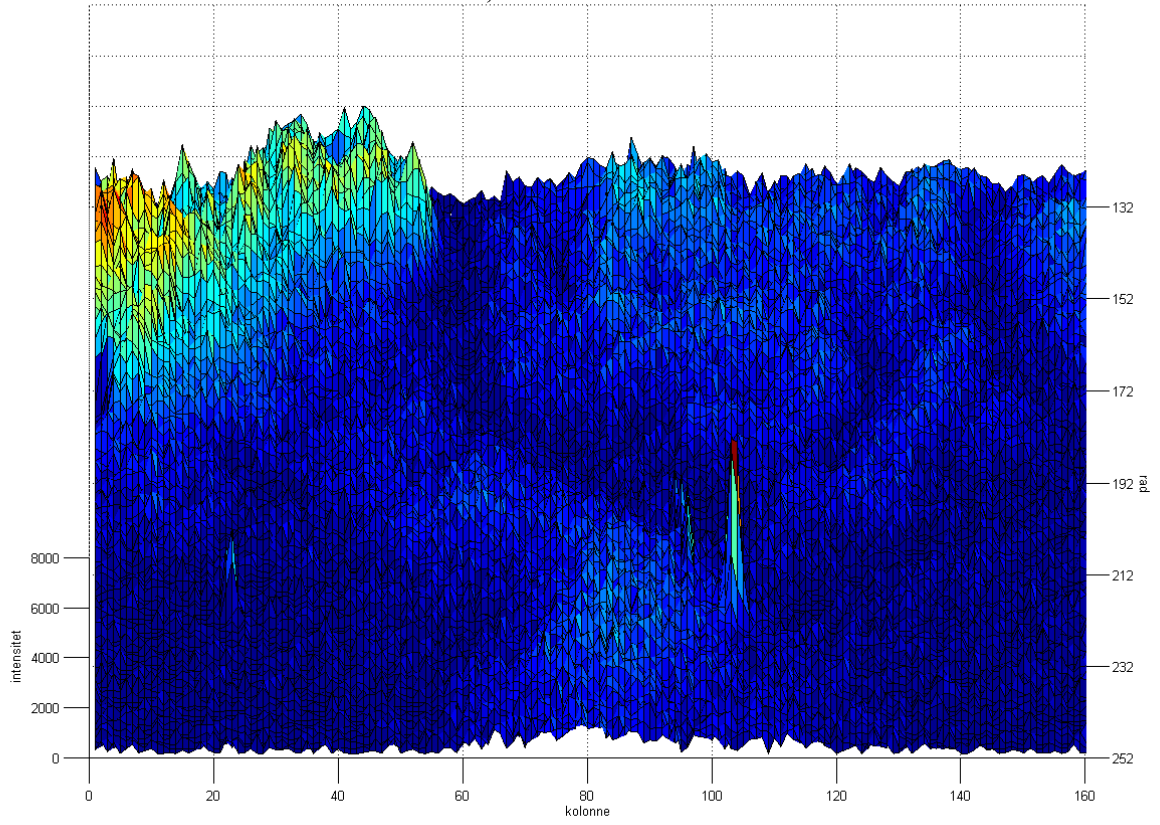
Bilde 8, Wavelet, 4 itterasjoner, 25% reduksjon



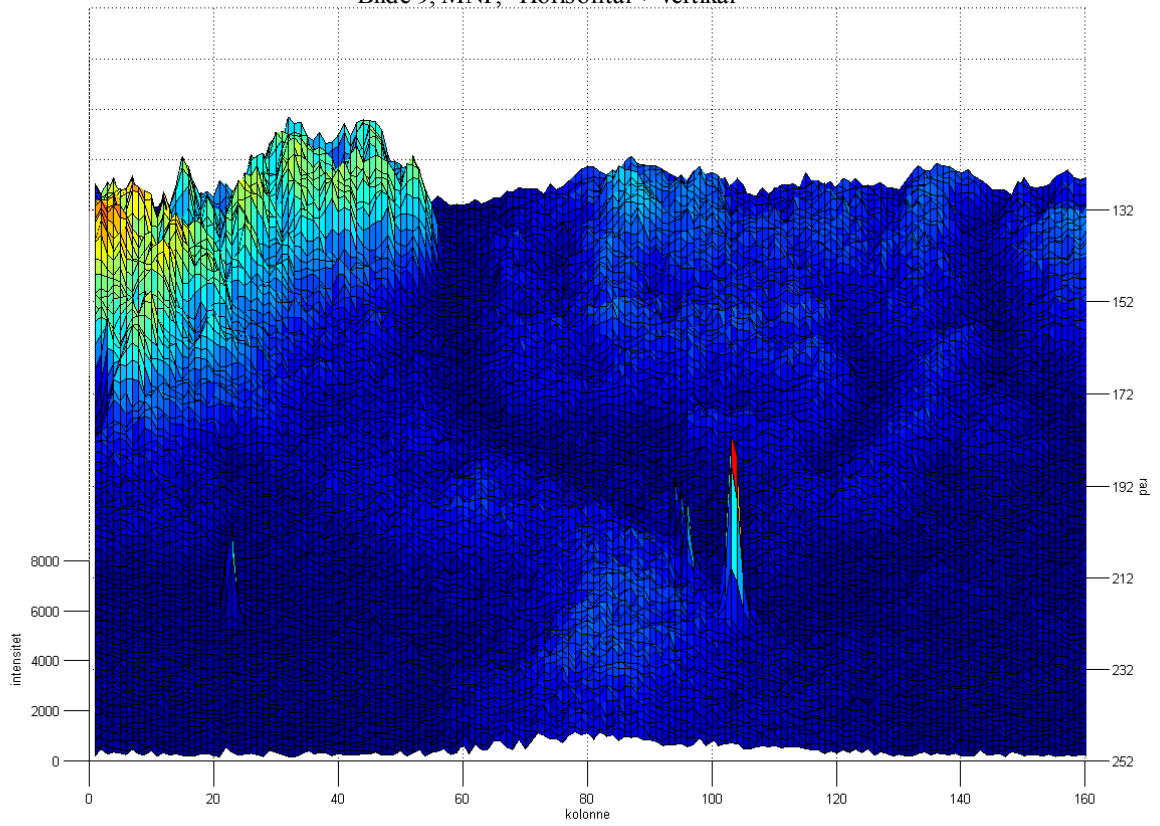
Bilde 9, Udempet utsnitt



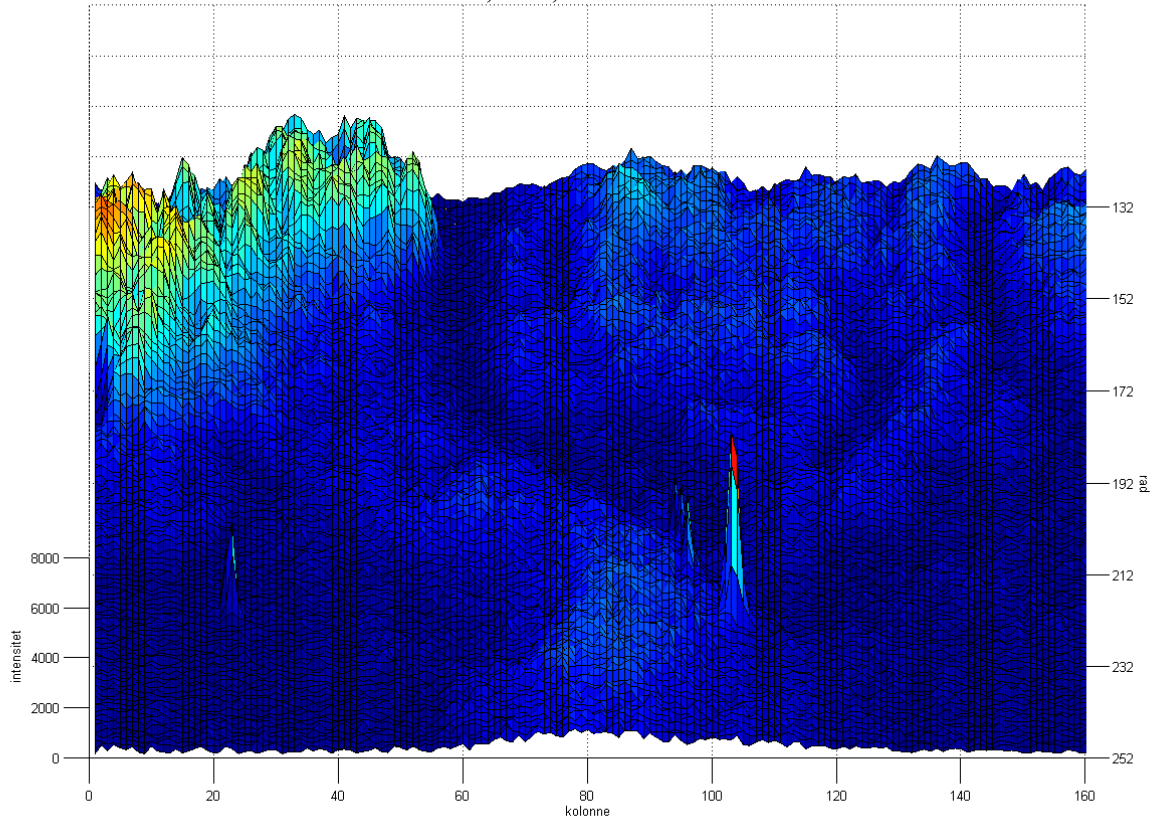
Bilde 9, PCA



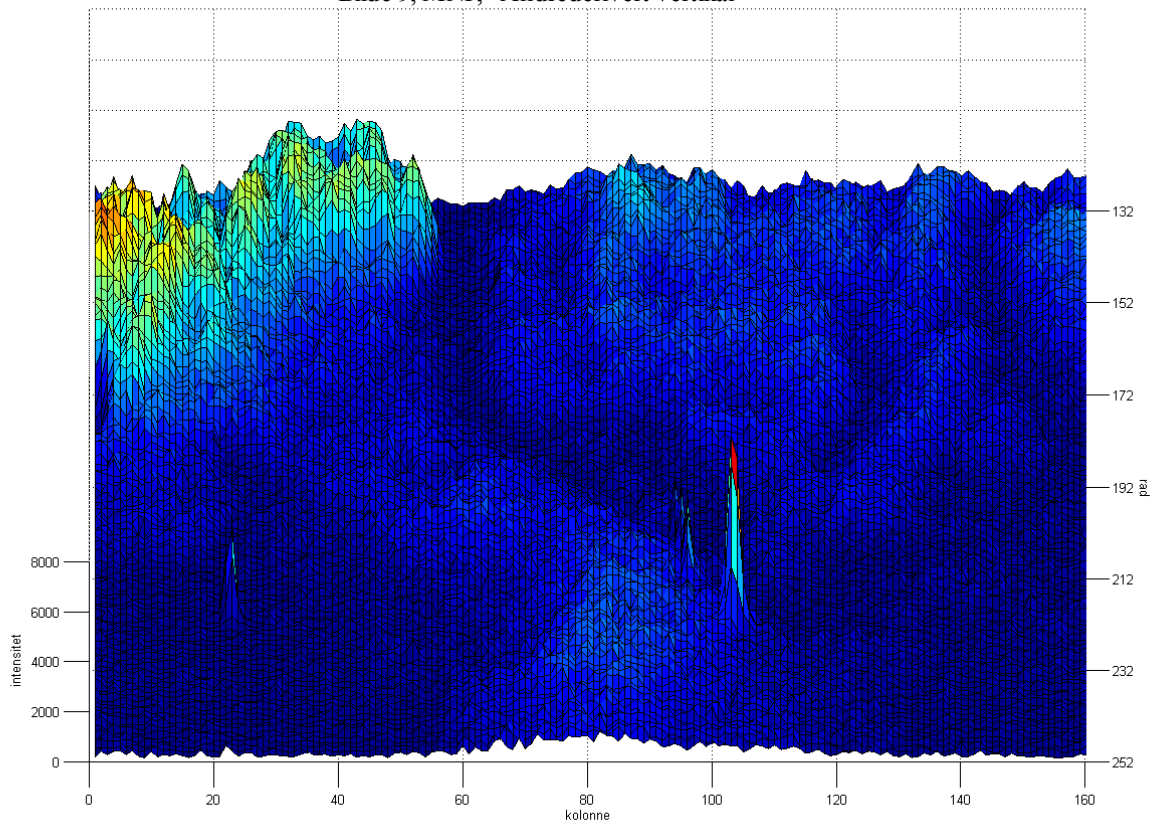
Bilde 9, MNF, "Horisontal + Vertikal"



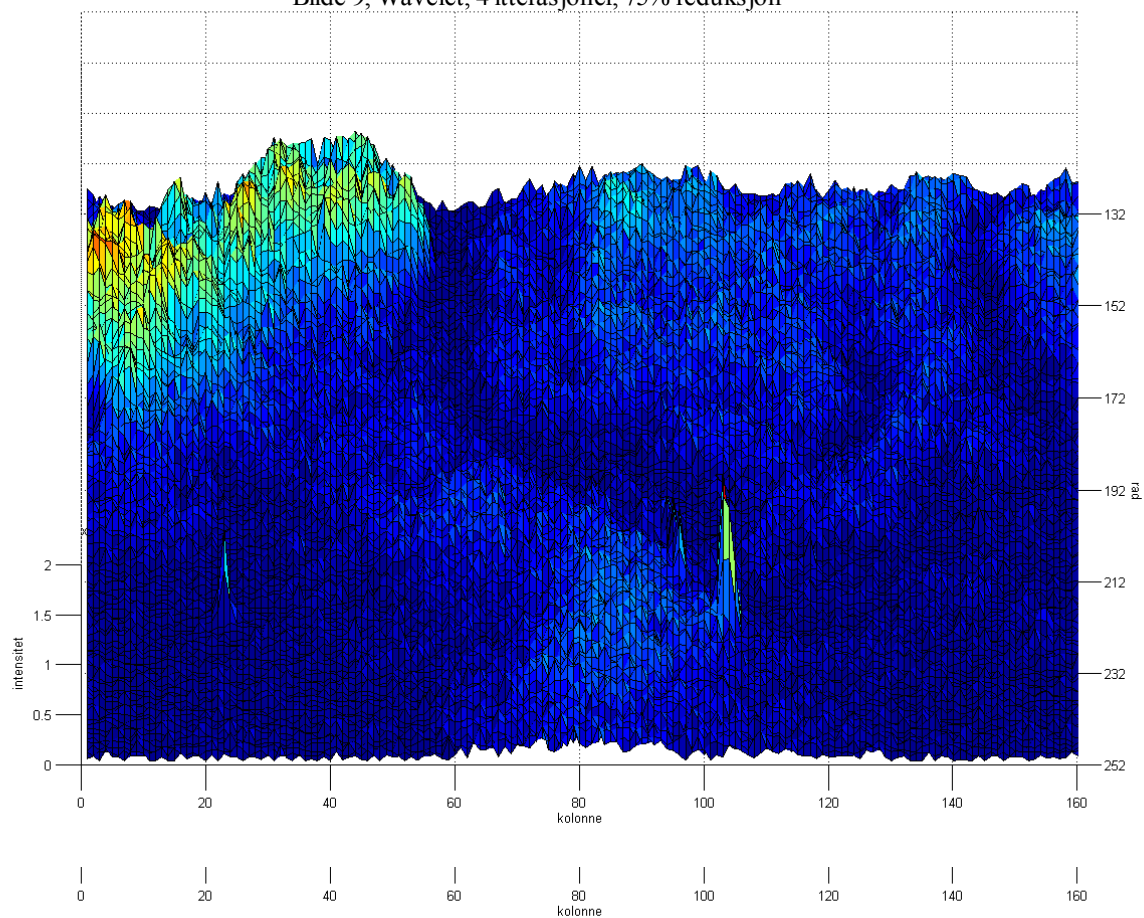
Bilde 9, MNF, "Mean 3x3"



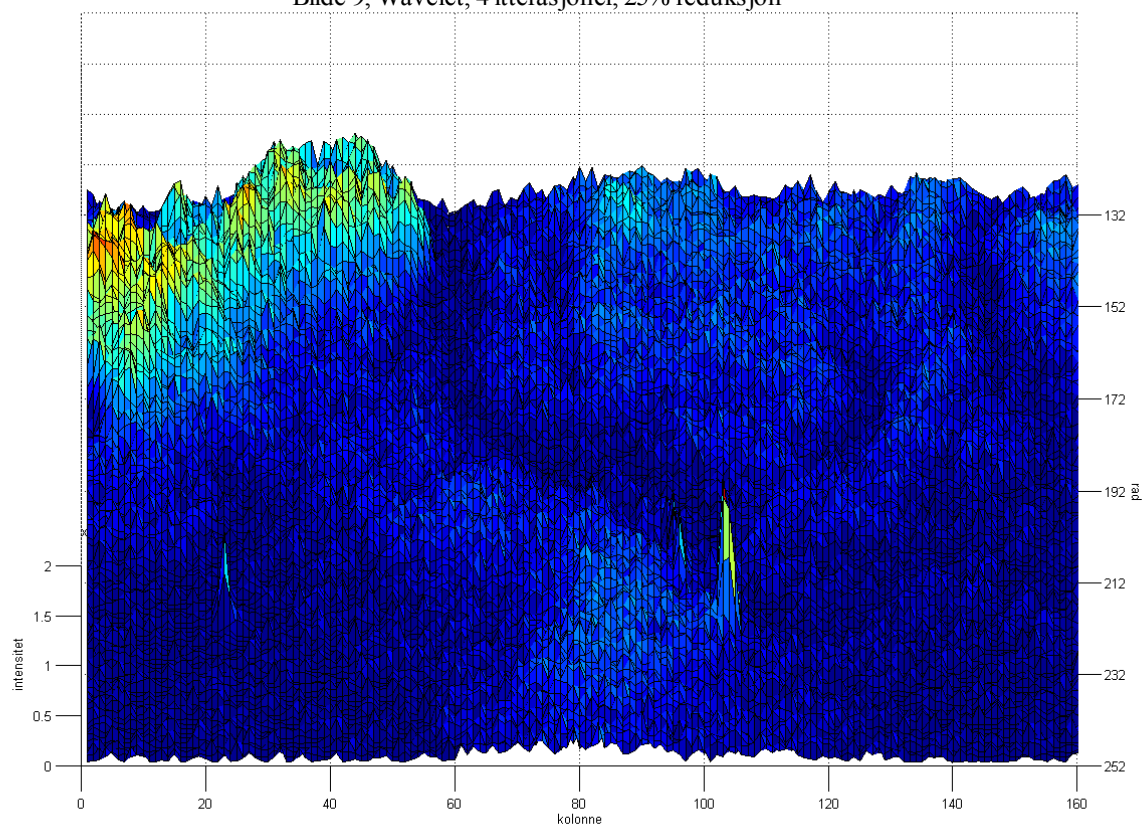
Bilde 9, MNF, "Andrederivert vertikalt"



Bilde 9, Wavelet, 4 itterasjoner, 75% reduksjon

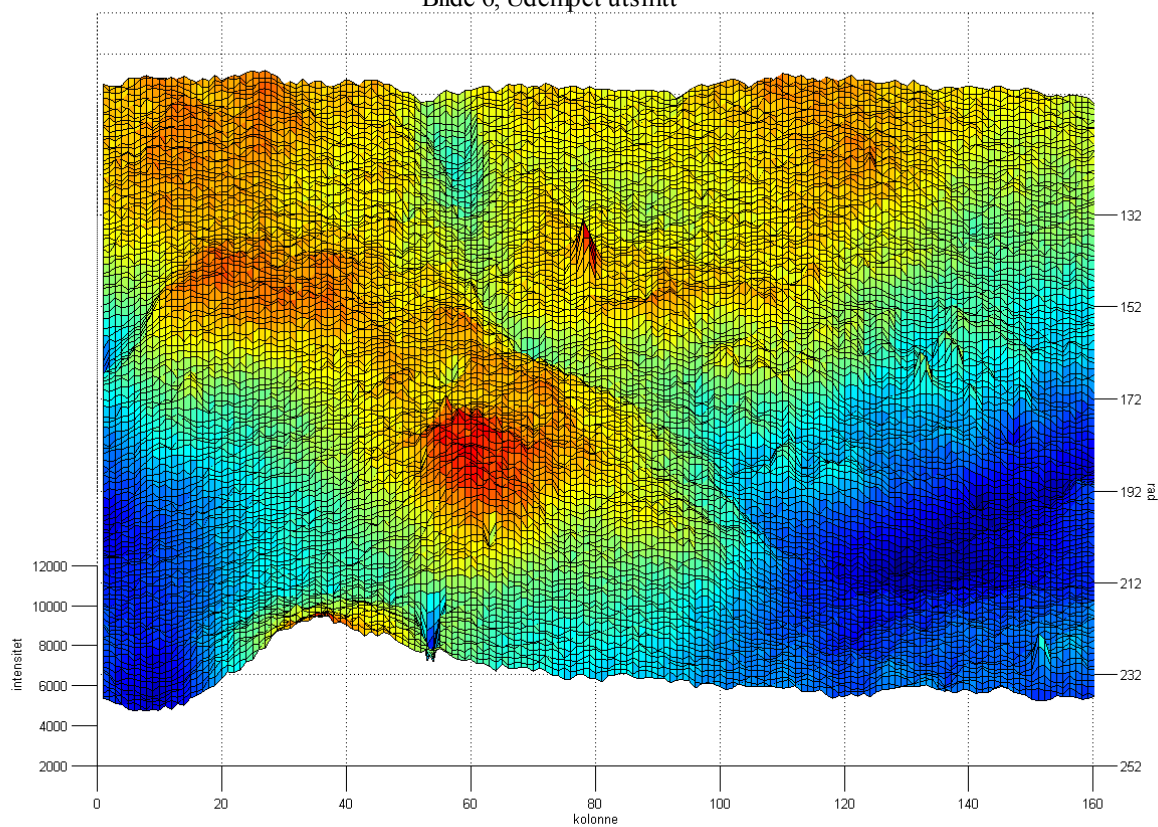


Bilde 9, Wavelet, 4 itterasjoner, 25% reduksjon

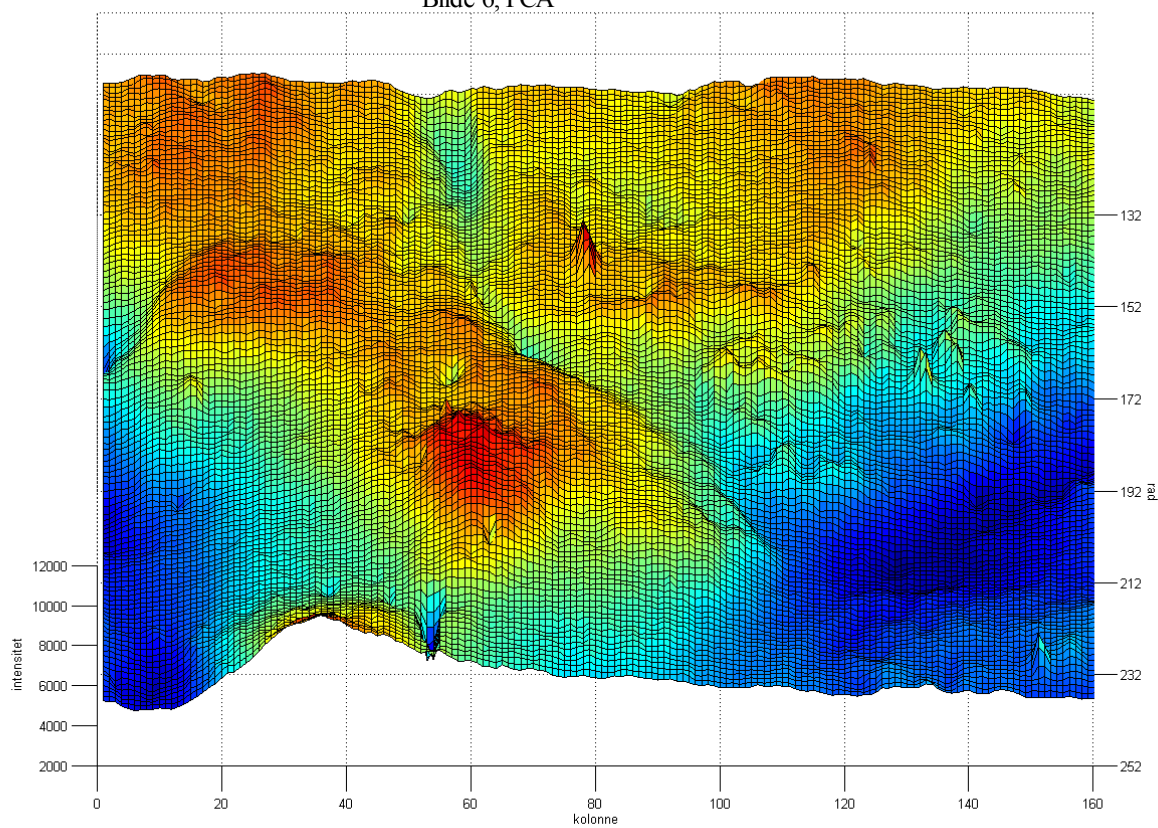


8.10 3D-utsnitt for spektralbånd 70 (662nm)

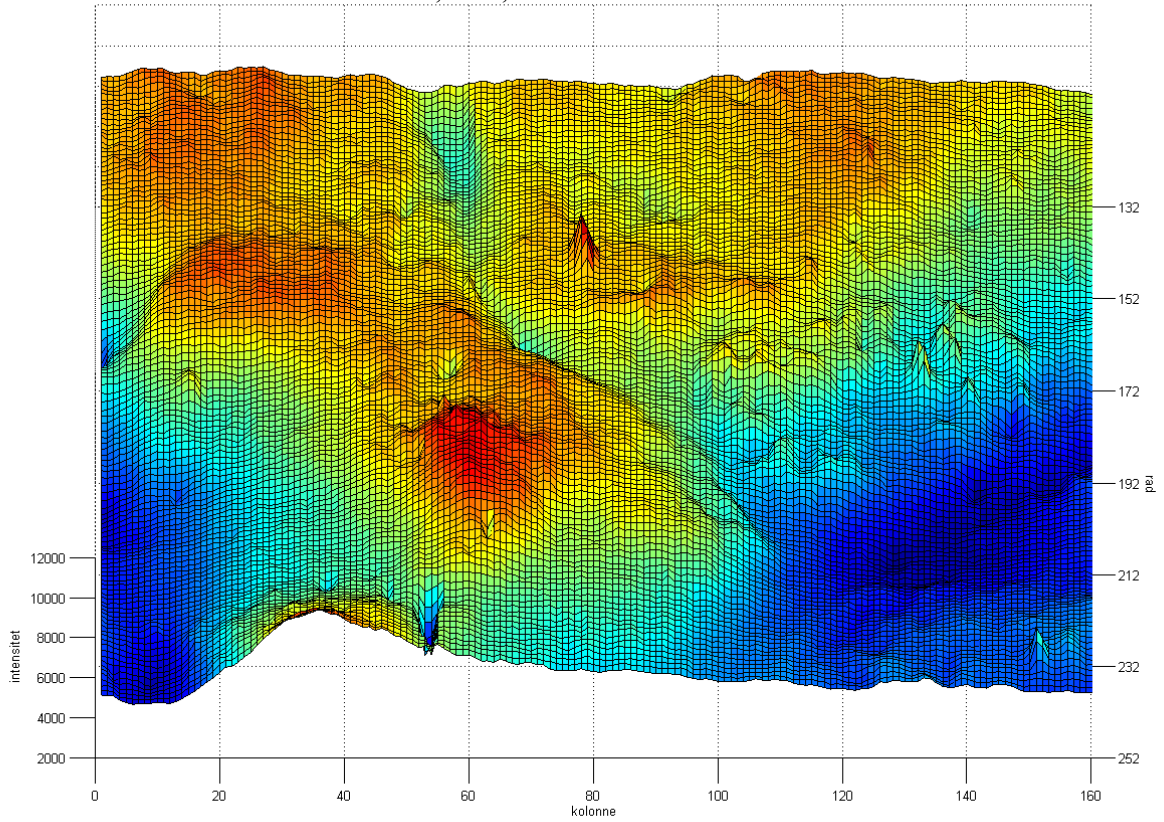
Bilde 6, Udempet utsnitt



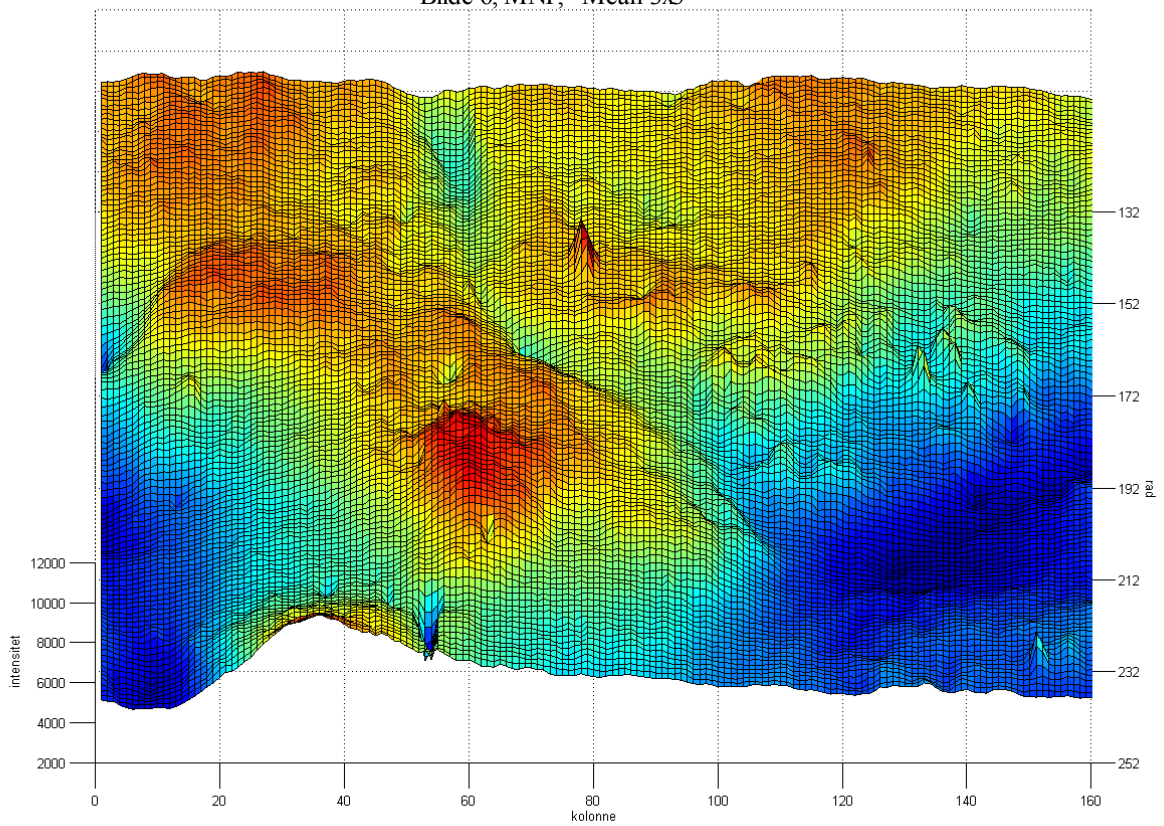
Bilde 6, PCA



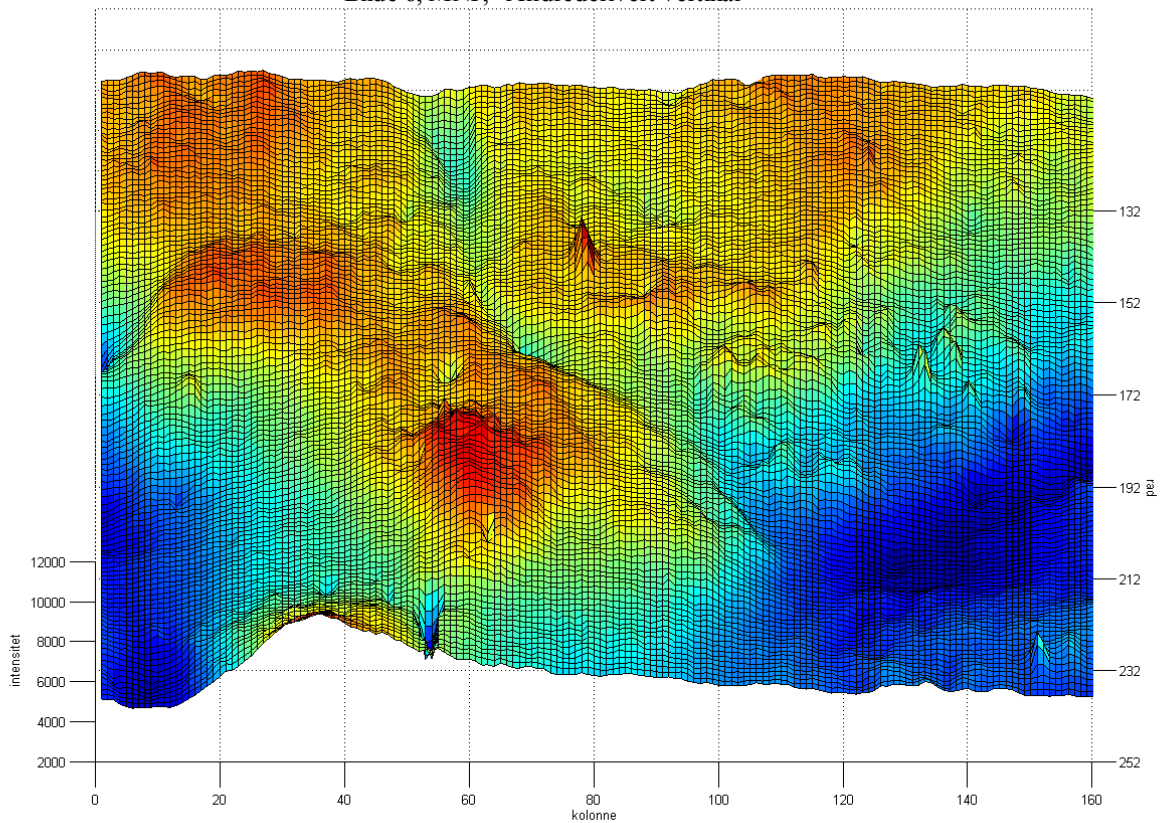
Bilde 6, MNF, "Horisontal + Vertikal"



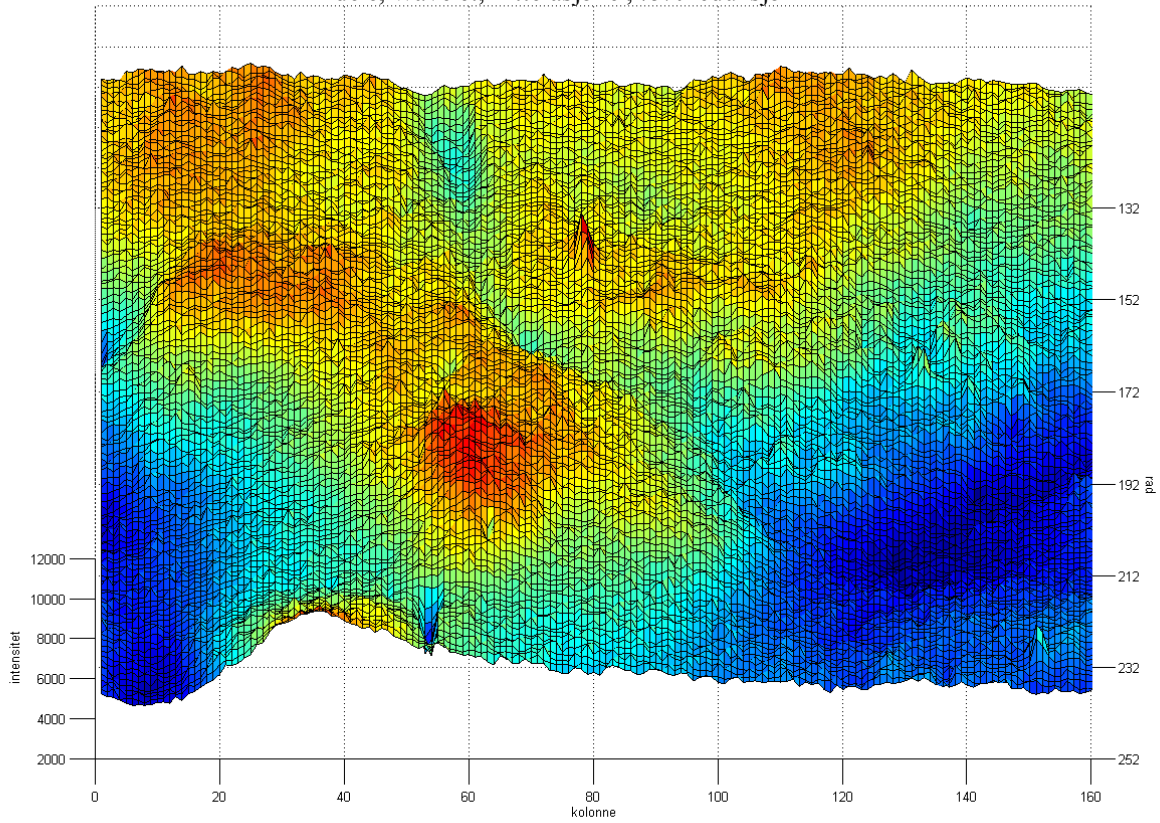
Bilde 6, MNF, "Mean 3x3"



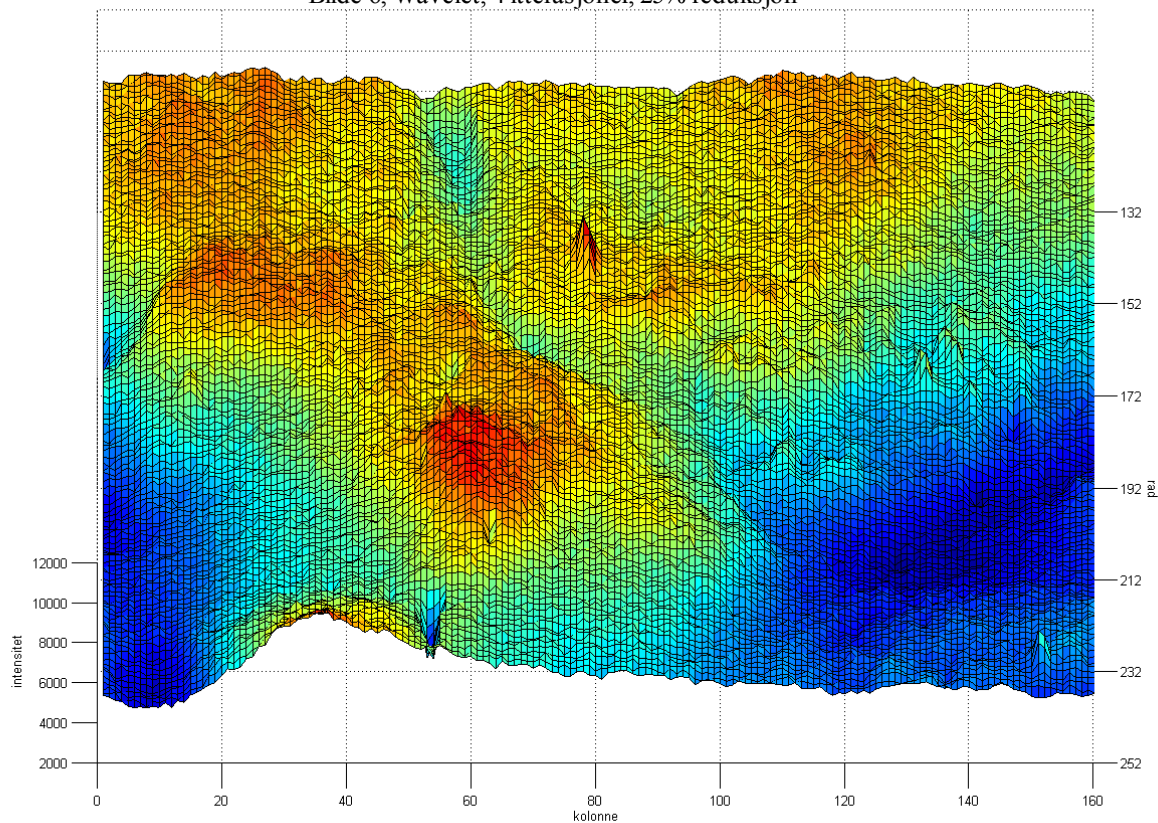
Bilde 6, MNF, "Andrederivert vertikalt"



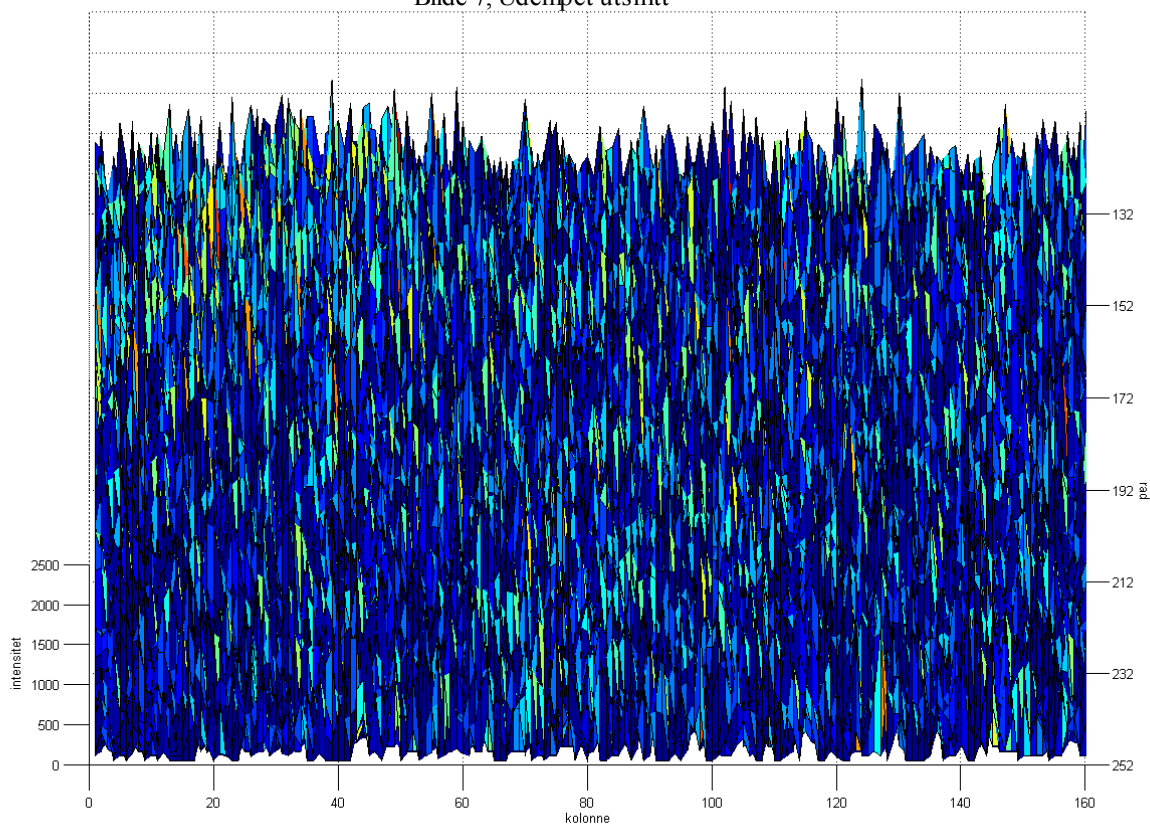
Bilde 6, Wavelet, 4 itterasjoner, 75% reduksjon



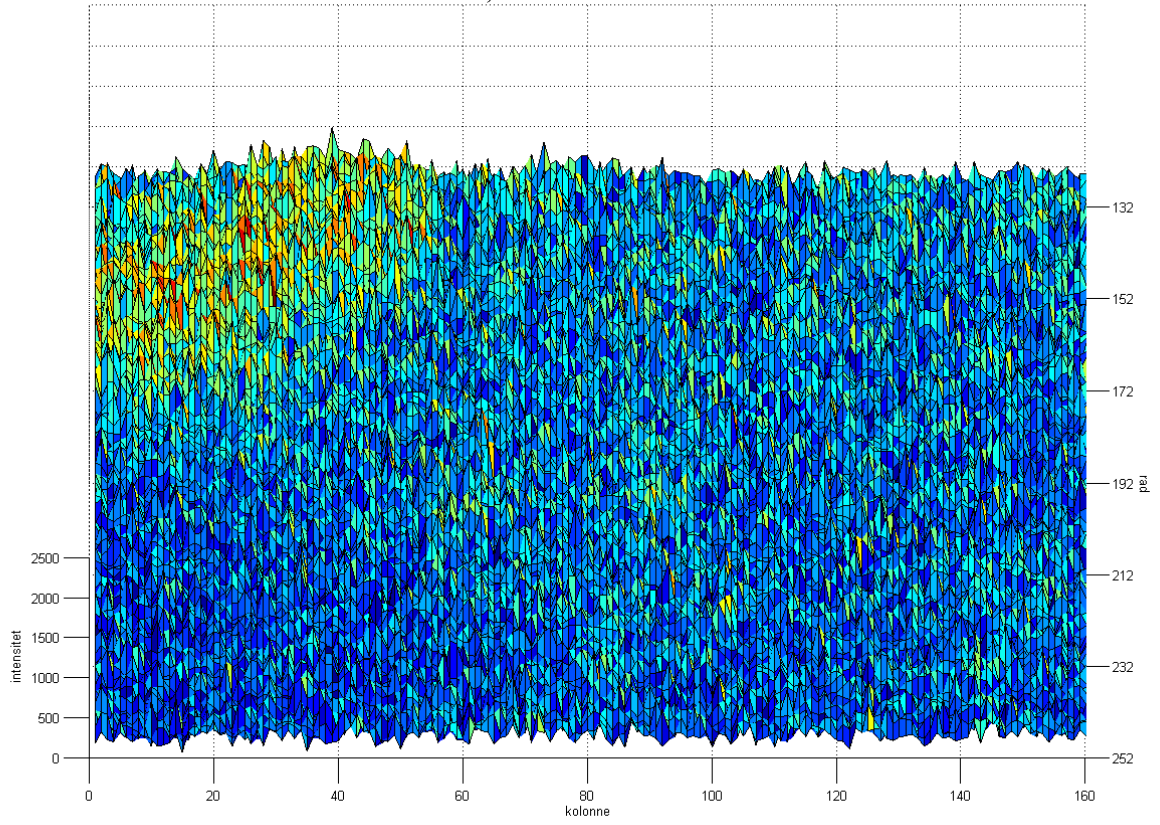
Bilde 6, Wavelet, 4 itterasjoner, 25% reduksjon



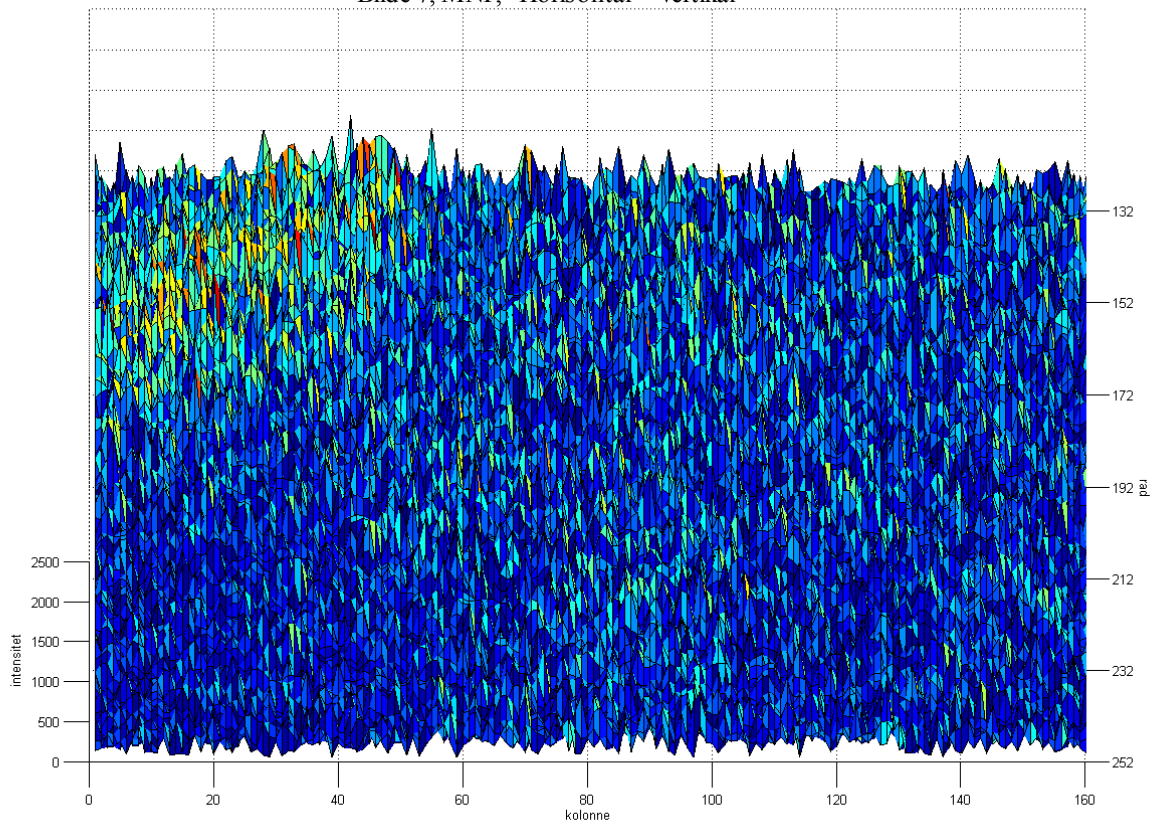
Bilde 7, Udempet utsnitt



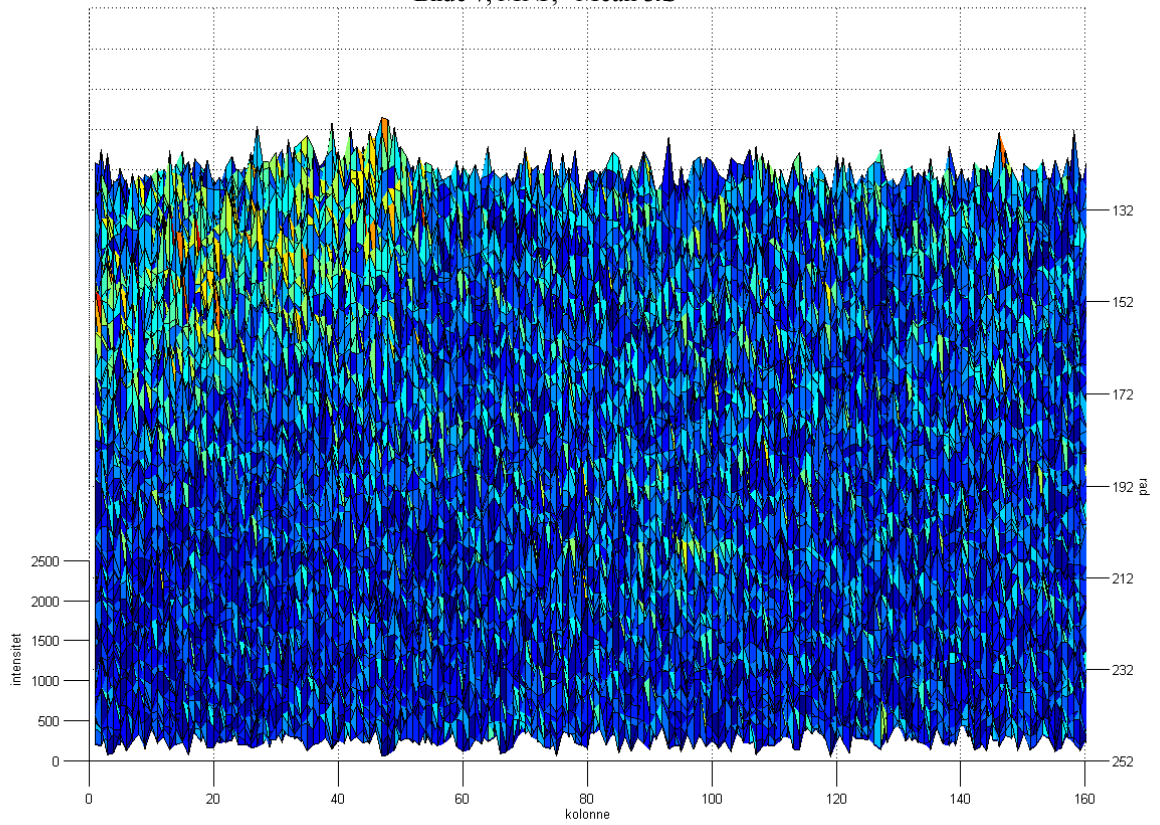
Bilde 7, PCA



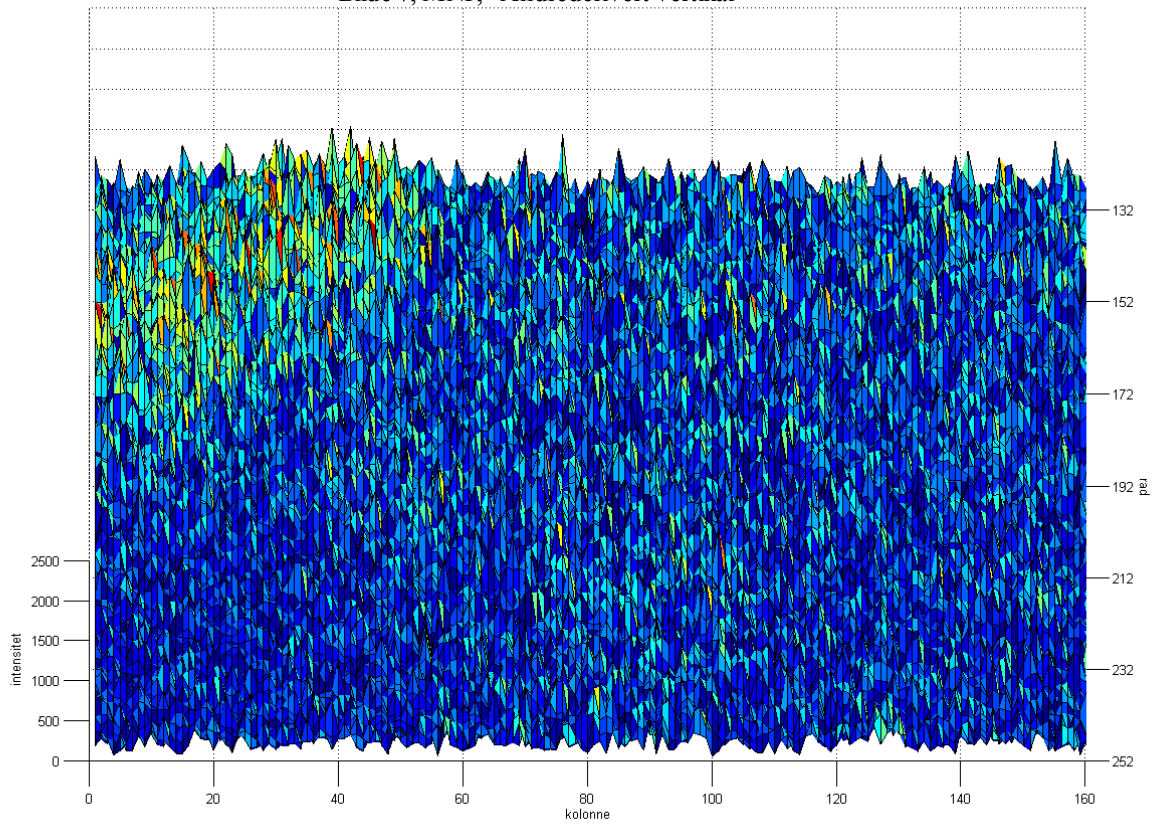
Bilde 7, MNF, "Horisontal + Vertikal"



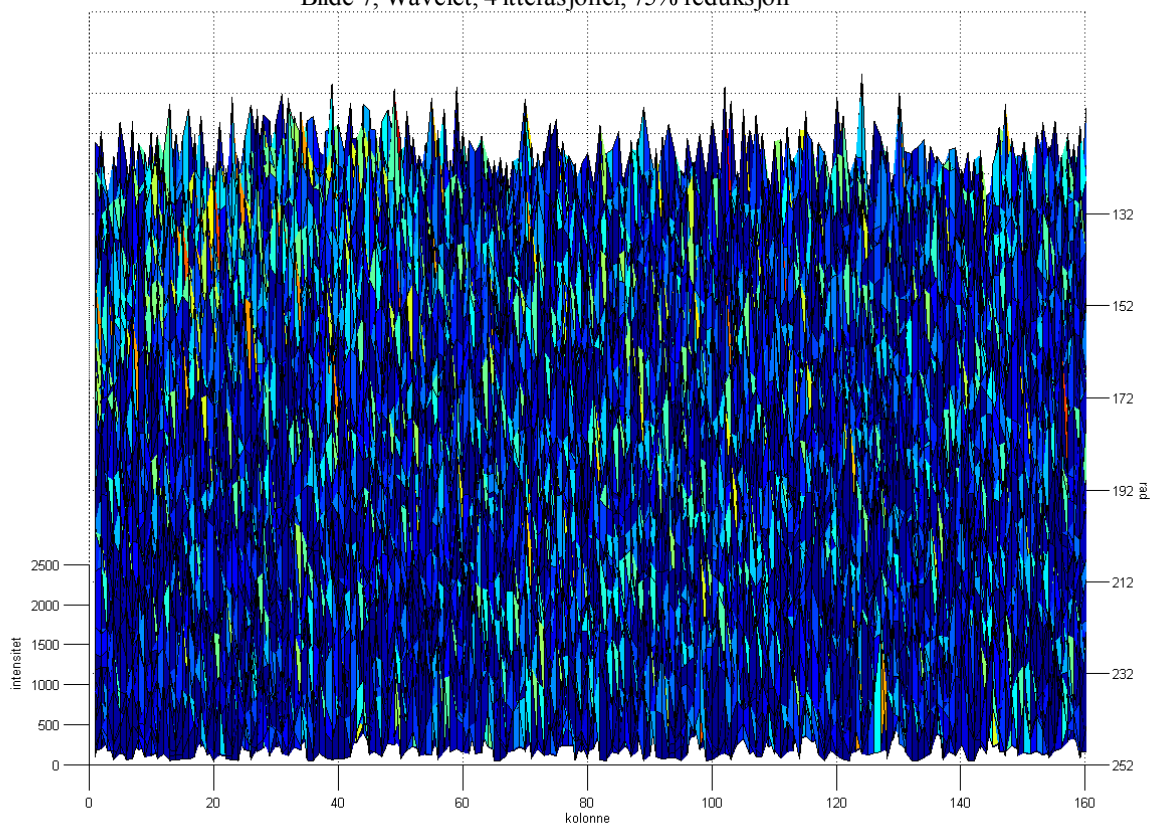
Bilde 7, MNF, "Mean 3x3"



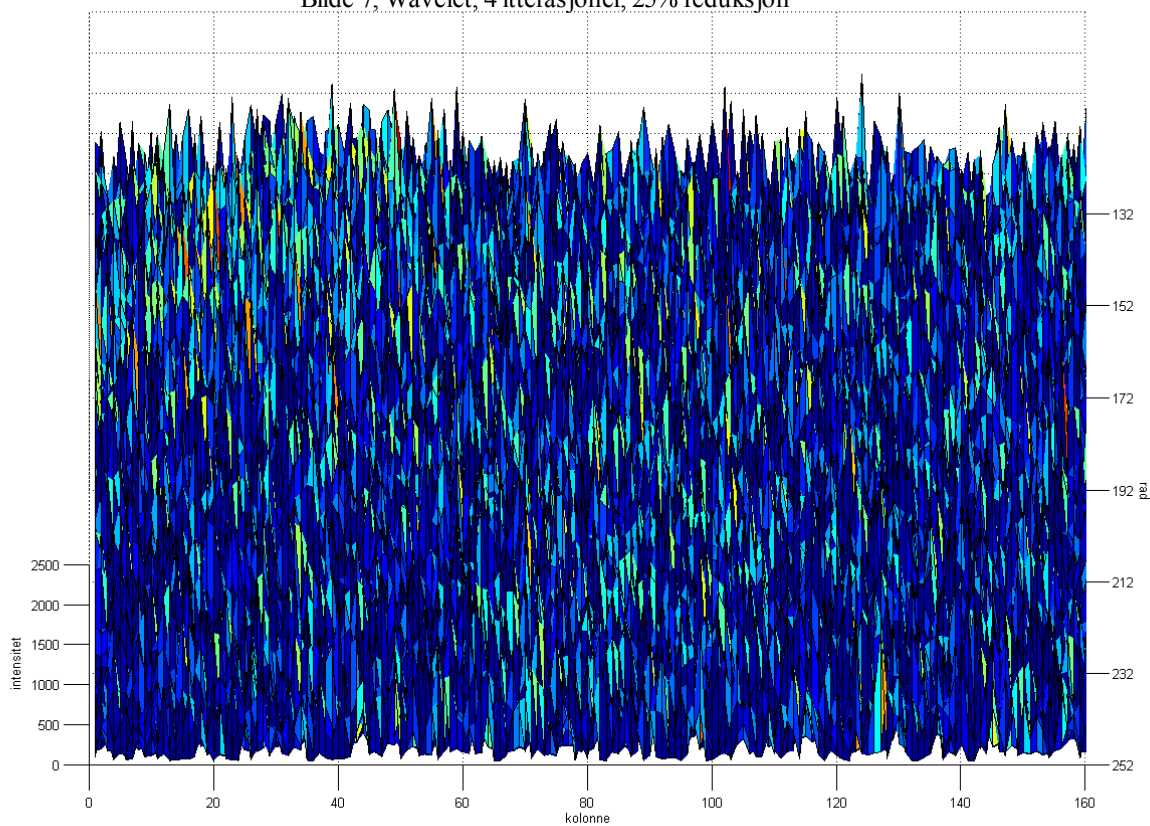
Bilde 7, MNF, "Andrederivert vertikalt"



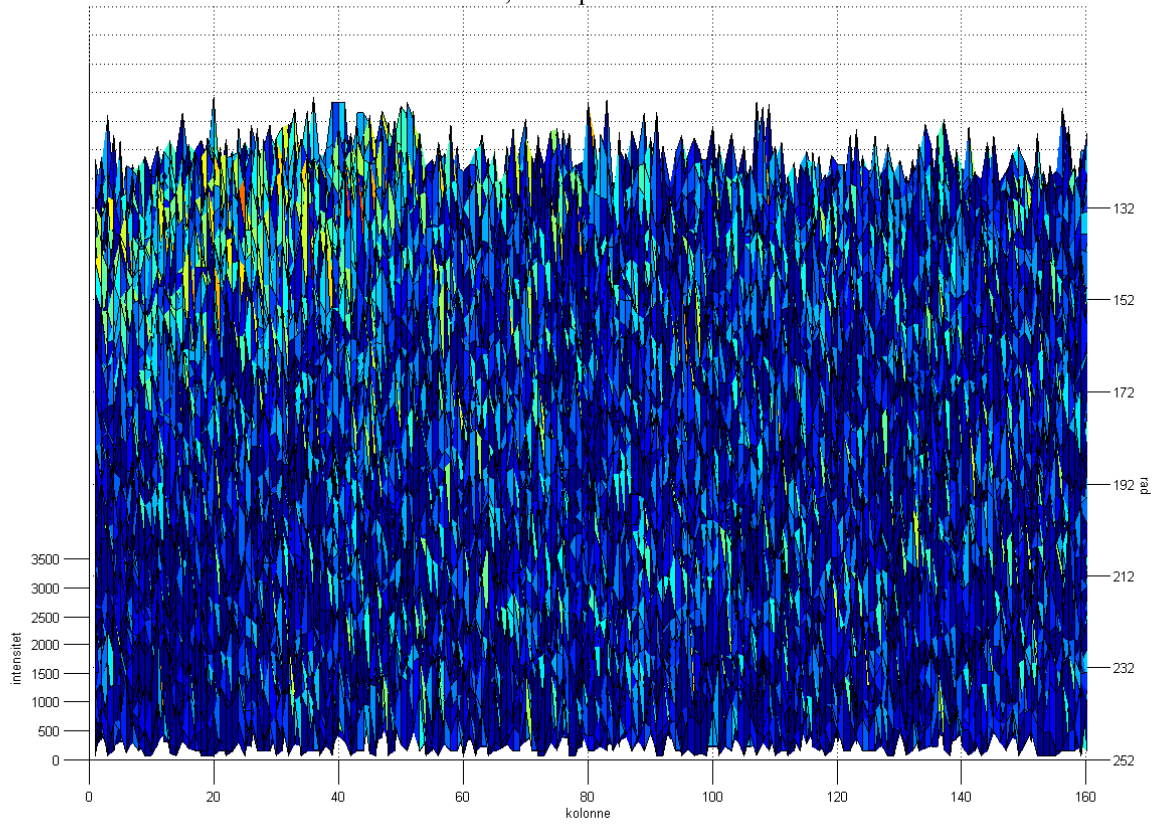
Bilde 7, Wavelet, 4 itterasjoner, 75% reduksjon



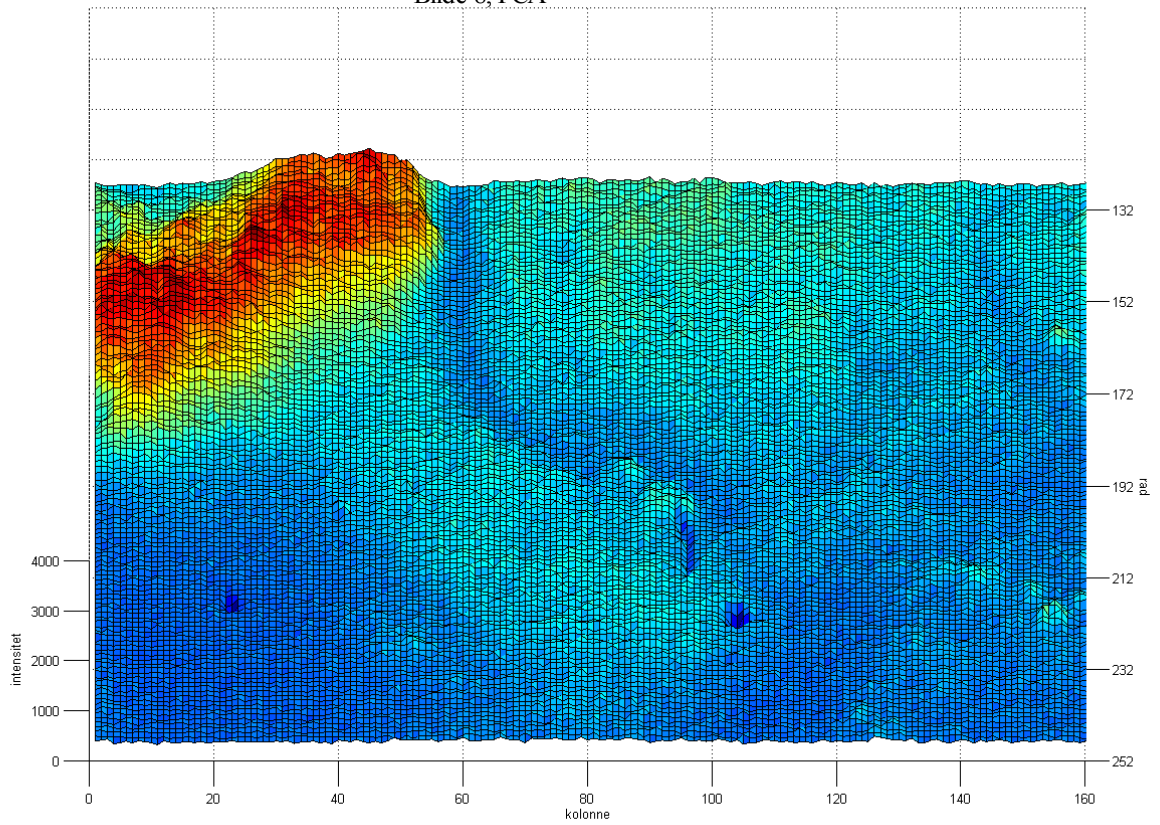
Bilde 7, Wavelet, 4 itterasjoner, 25% reduksjon



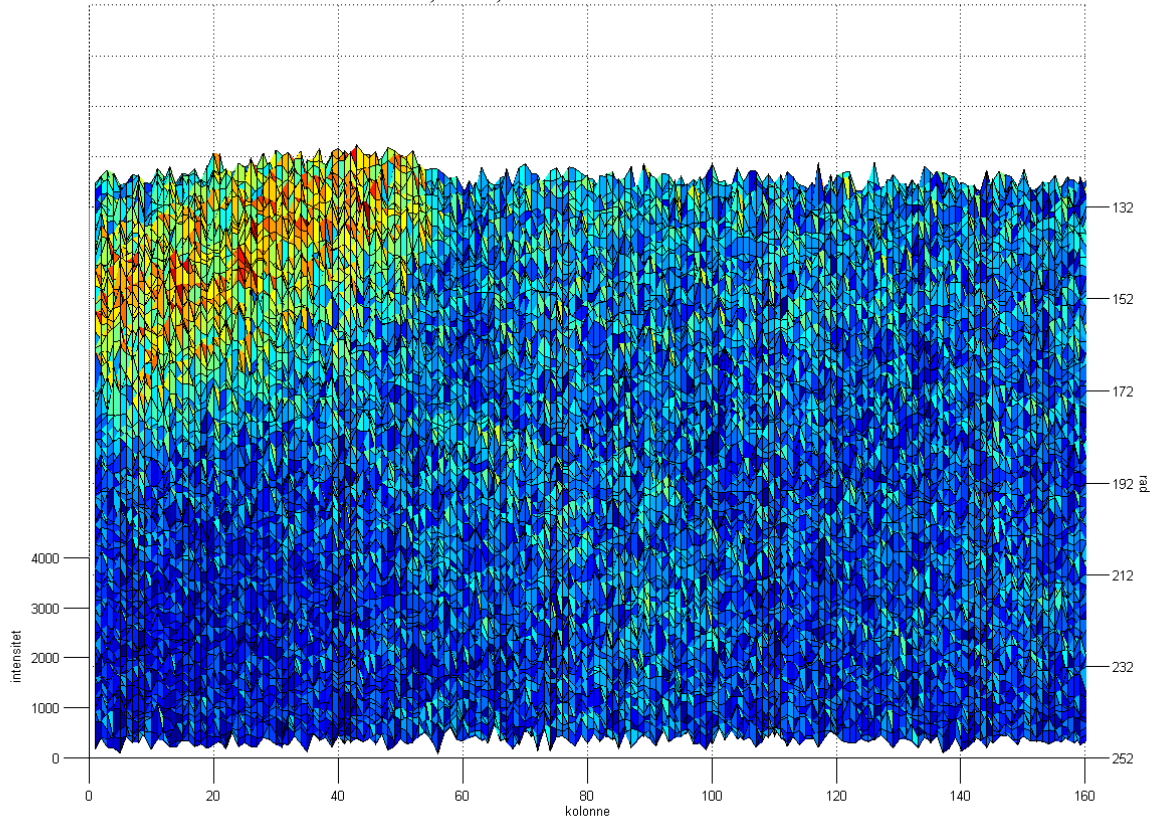
Bilde 8, Udempet utsnitt



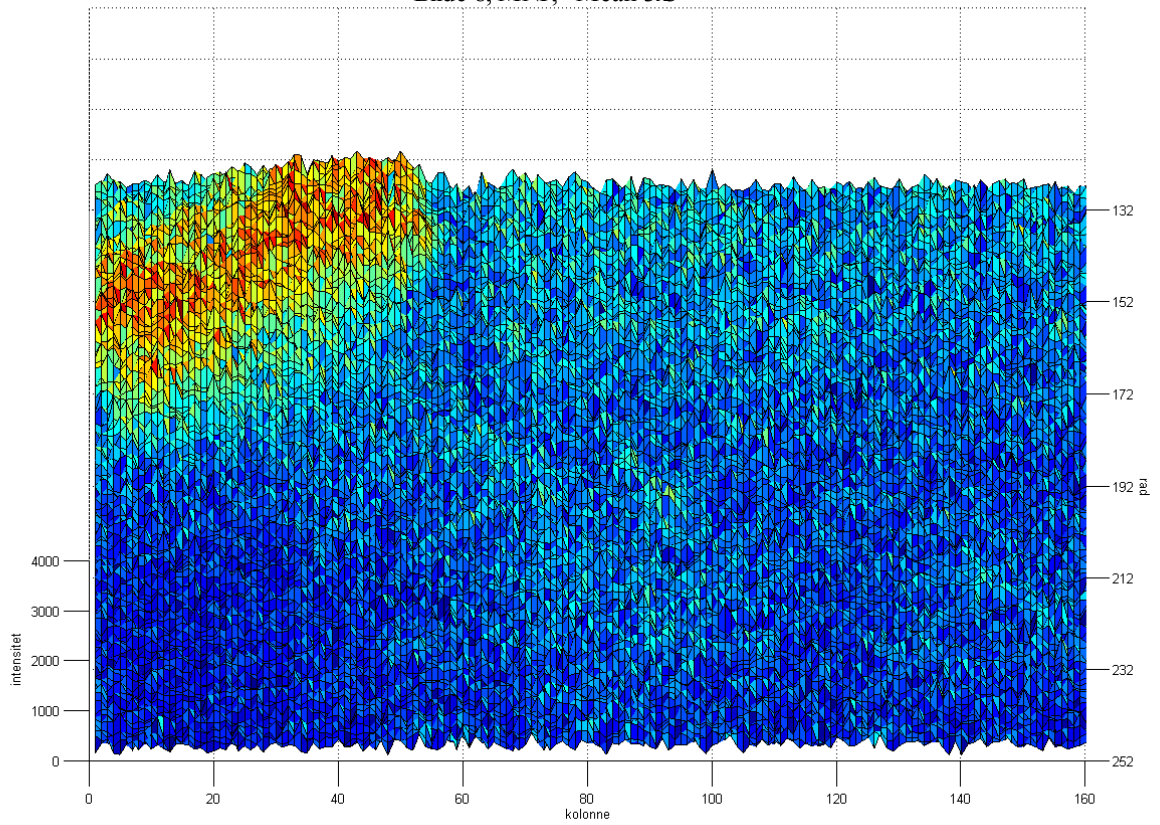
Bilde 8, PCA



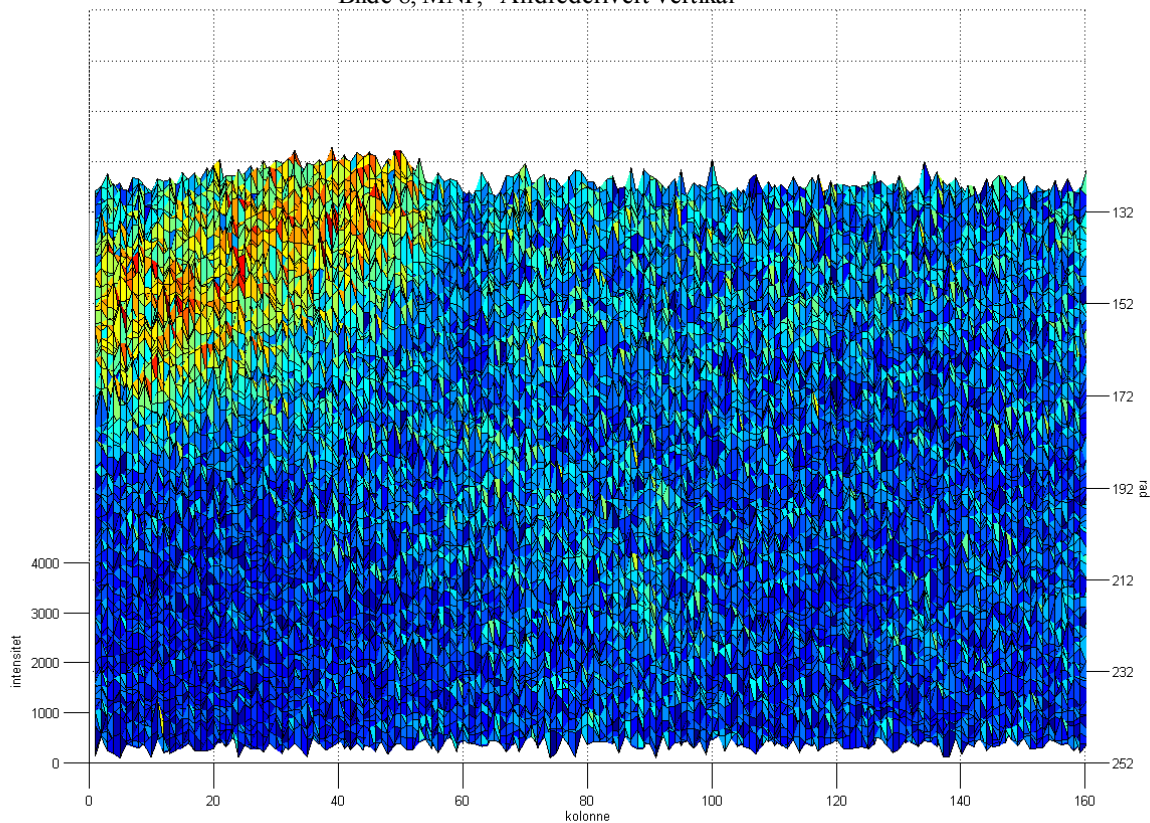
Bilde 8, MNF, "Horisontal + Vertikal"



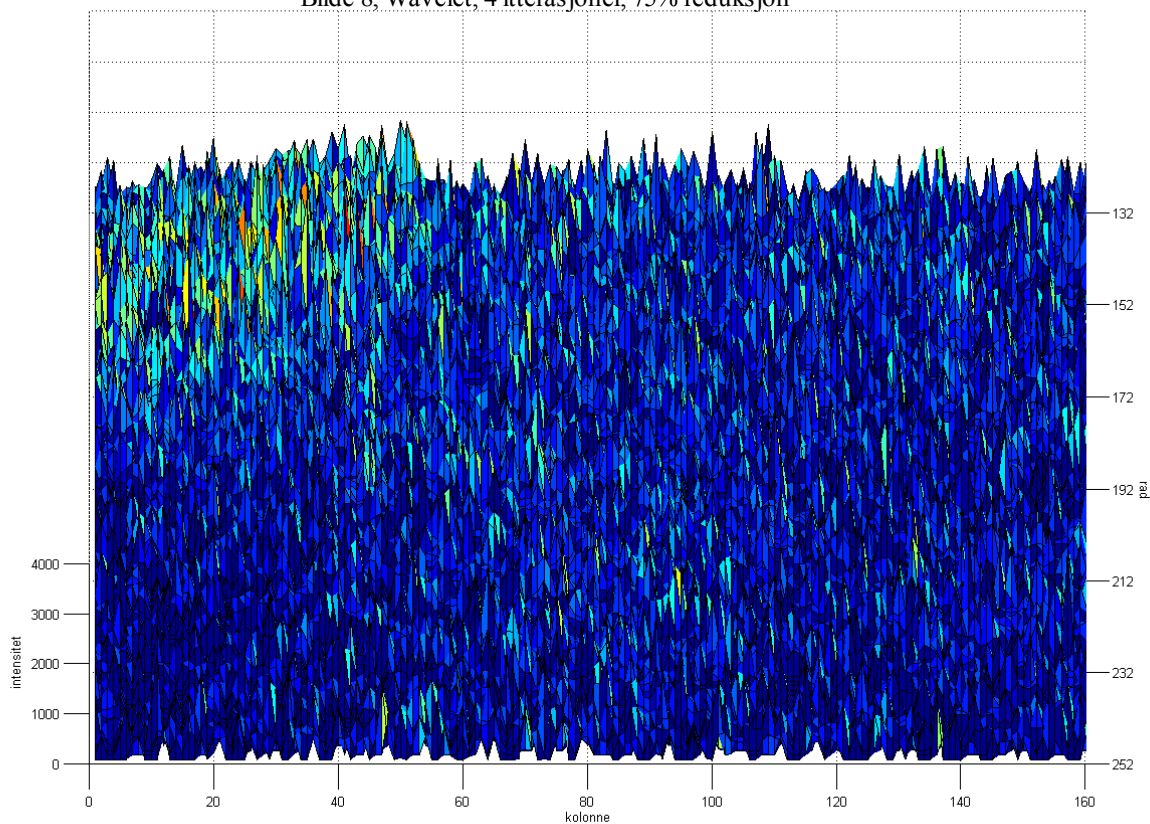
Bilde 8, MNF, "Mean 3x3"



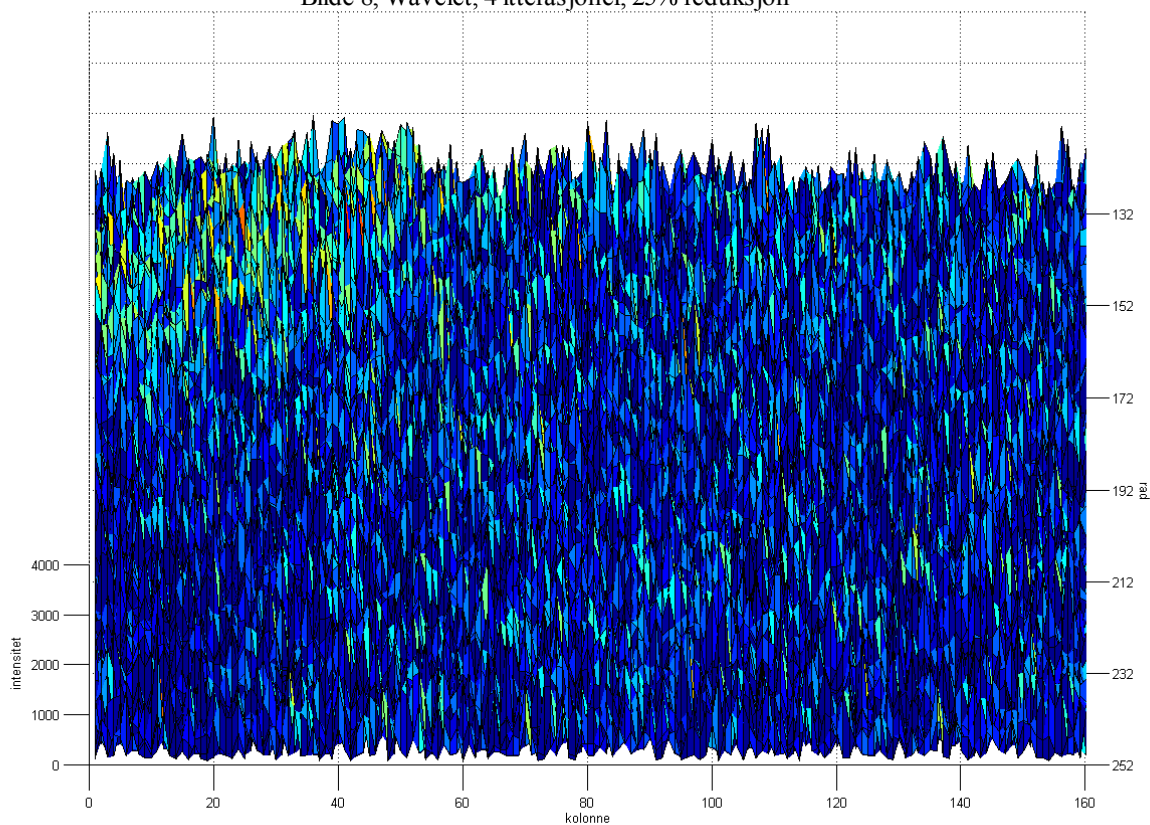
Bilde 8, MNF, "Andrederivert vertikalt"



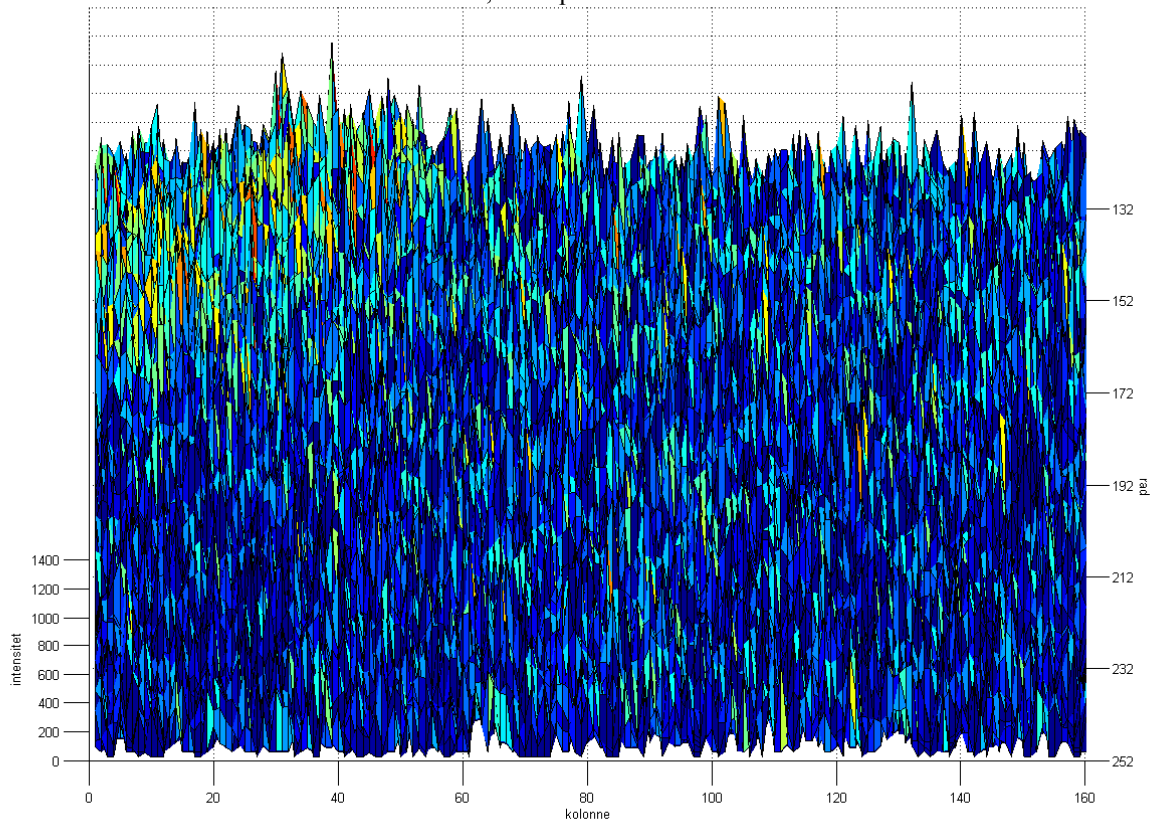
Bilde 8, Wavelet, 4 itterasjoner, 75% reduksjon



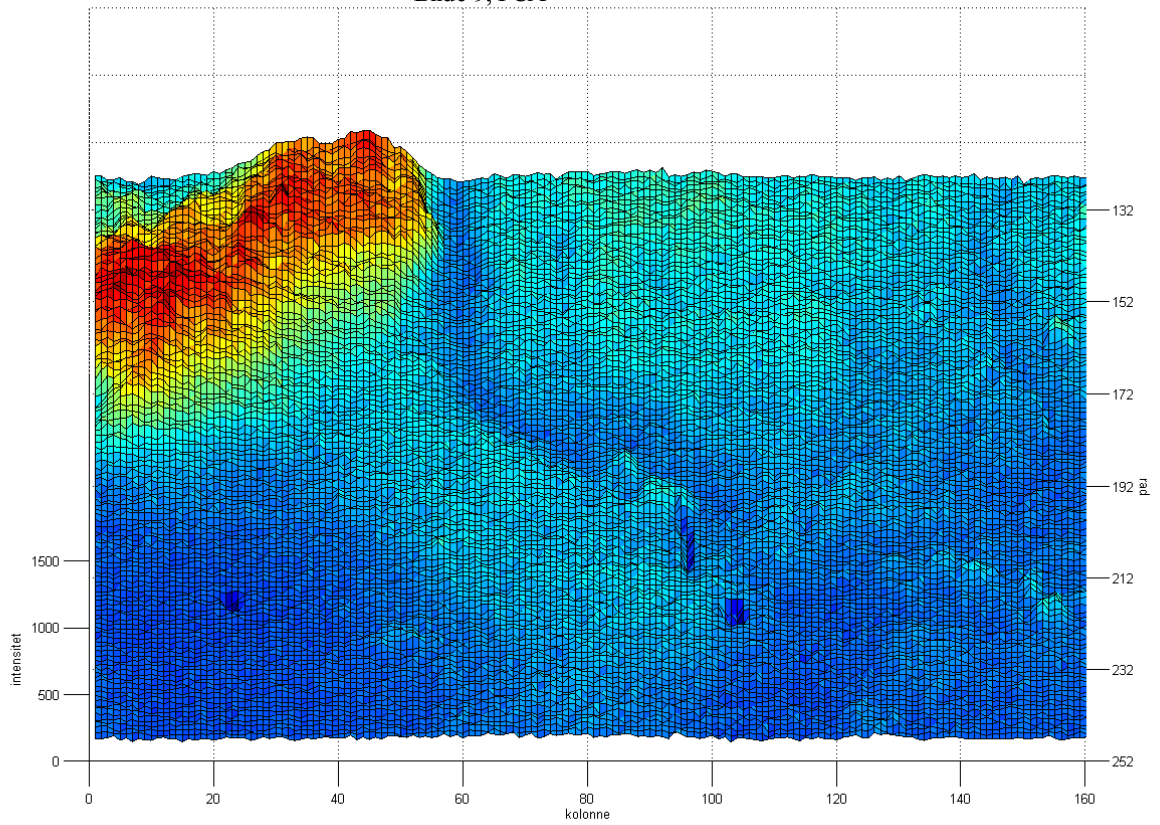
Bilde 8, Wavelet, 4 itterasjoner, 25% reduksjon



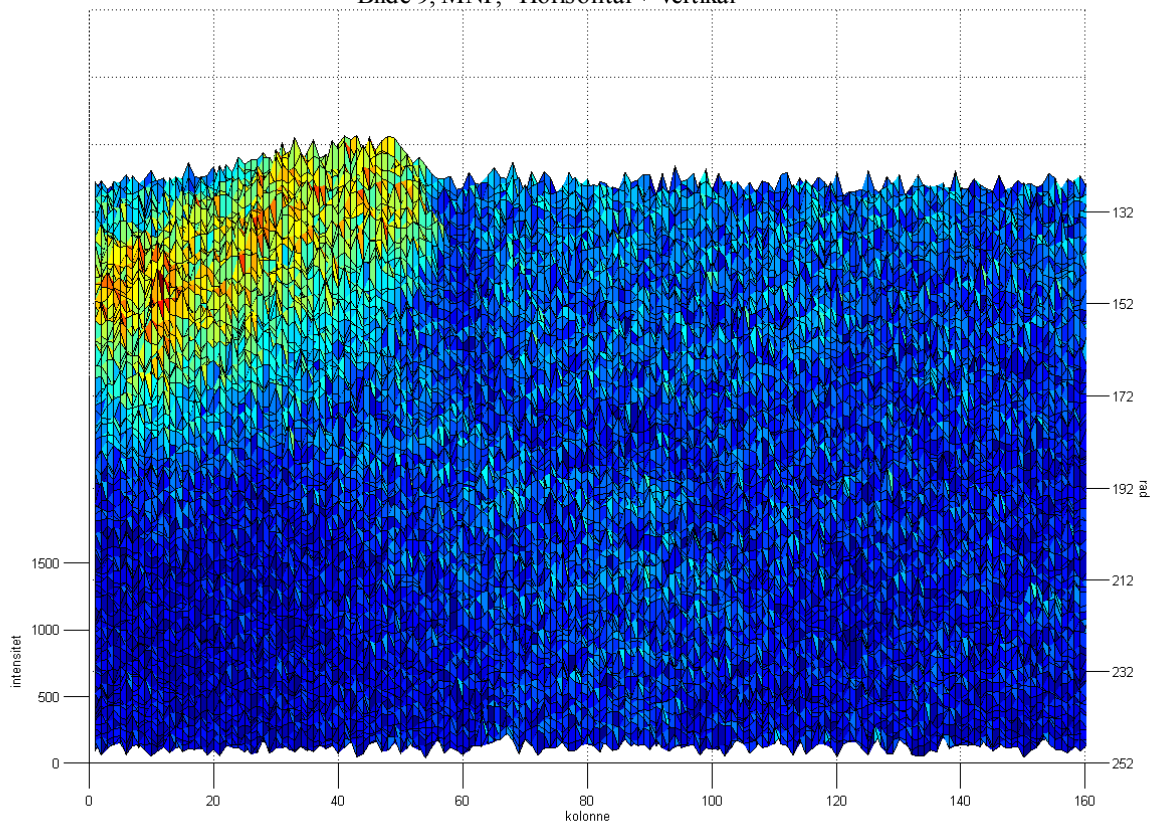
Bilde 9, Udempet utsnitt



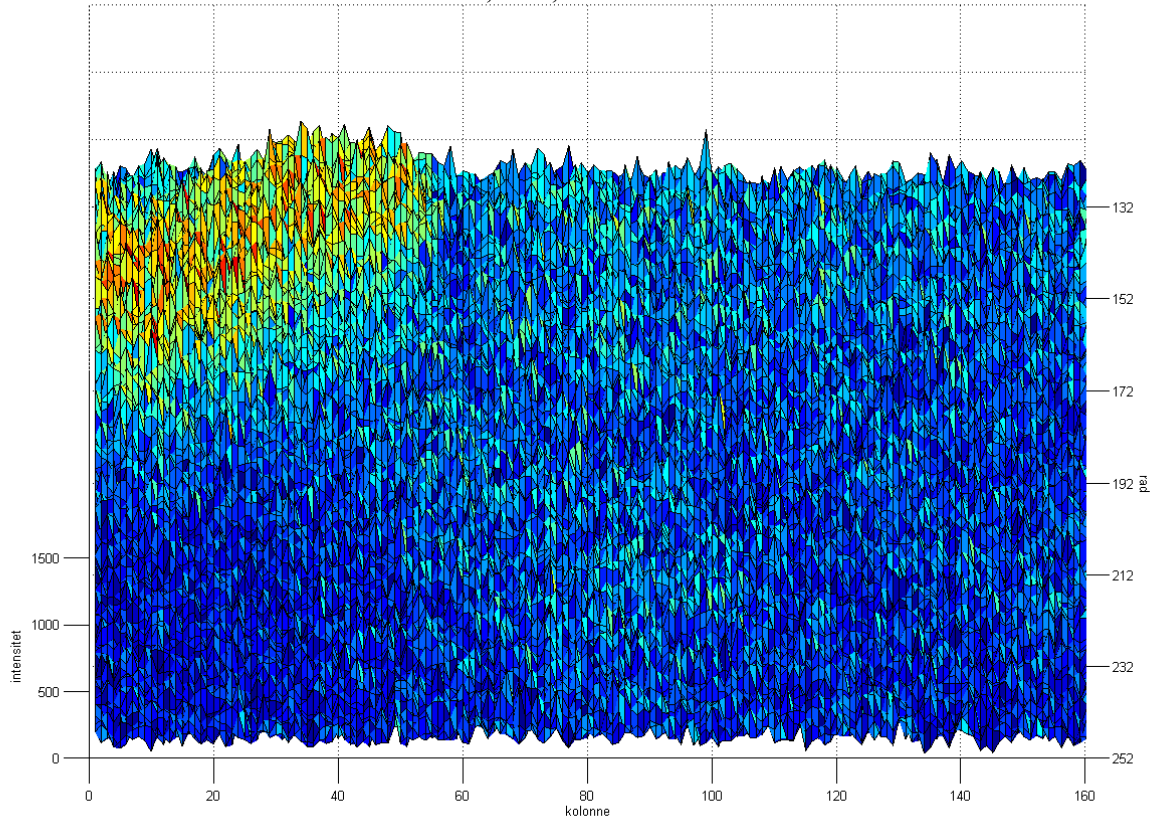
Bilde 9, PCA



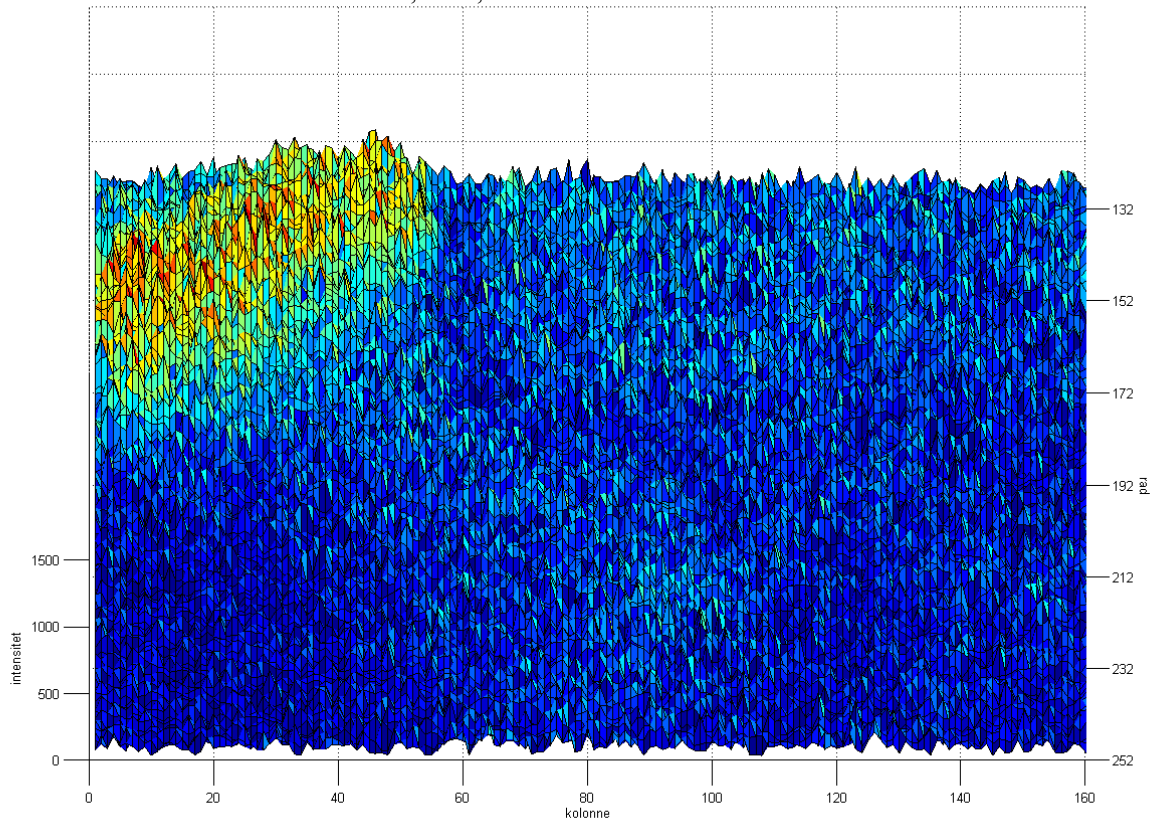
Bilde 9, MNF, "Horisontal + Vertikal"



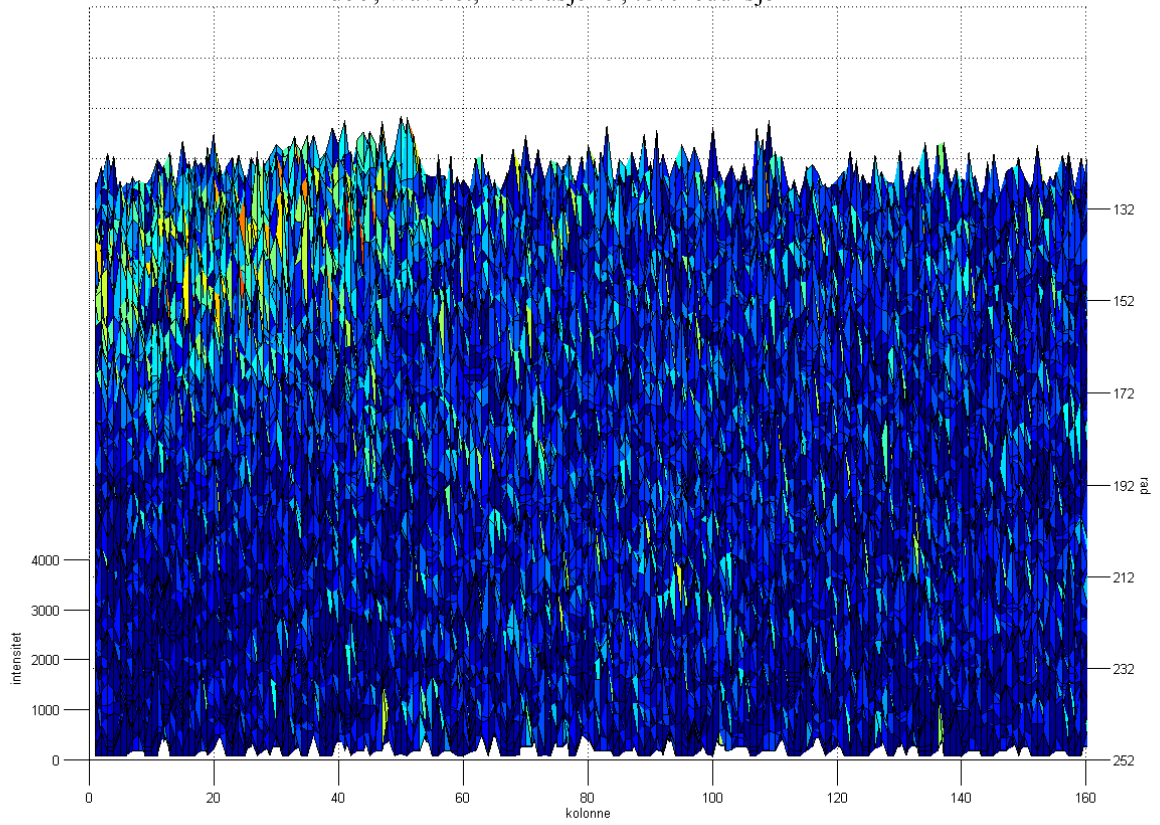
Bilde 9, MNF, "Mean 3x3"



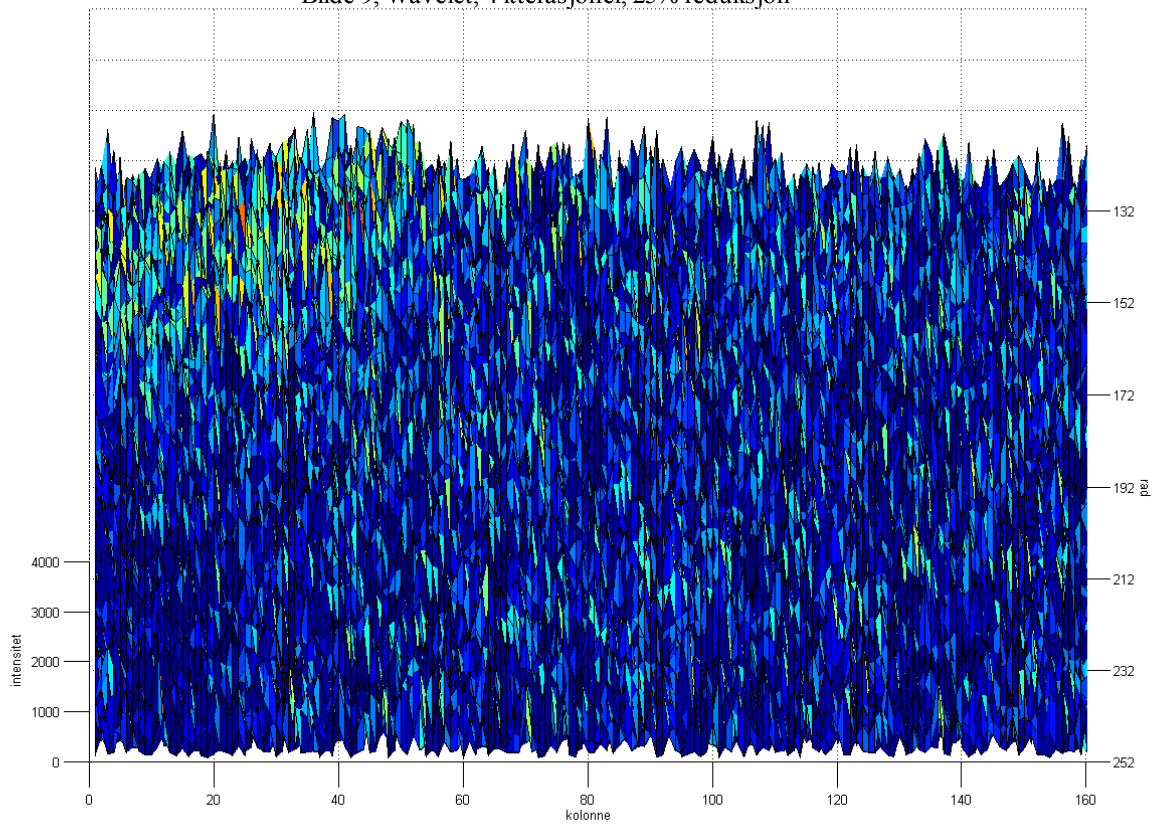
Bilde 9, MNF, "Andrederivert vertikalt"



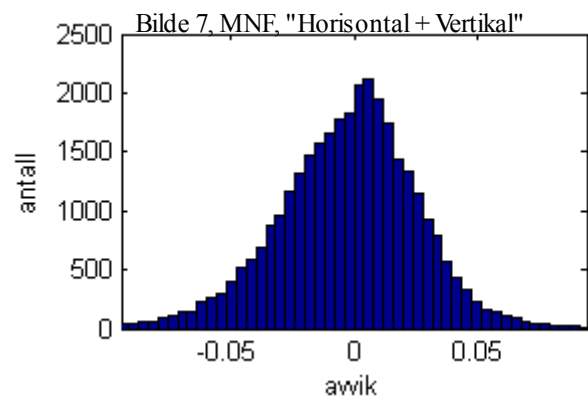
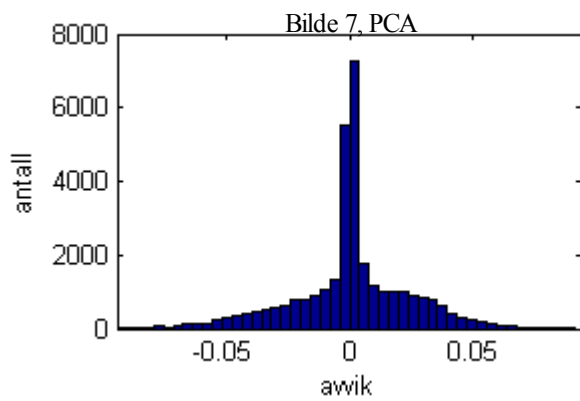
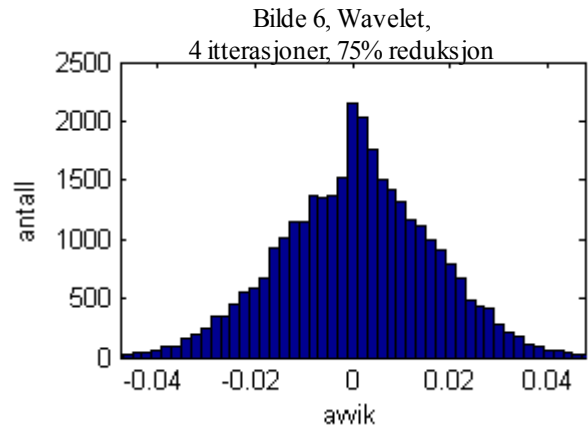
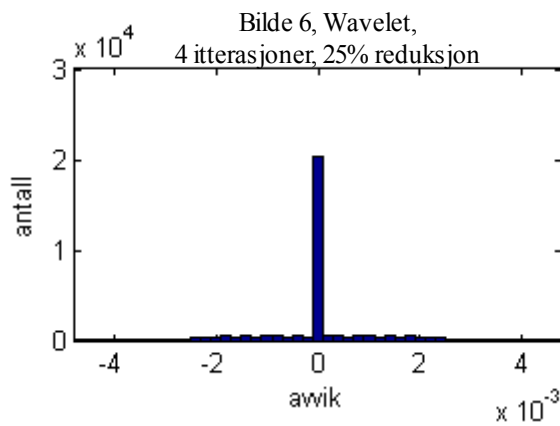
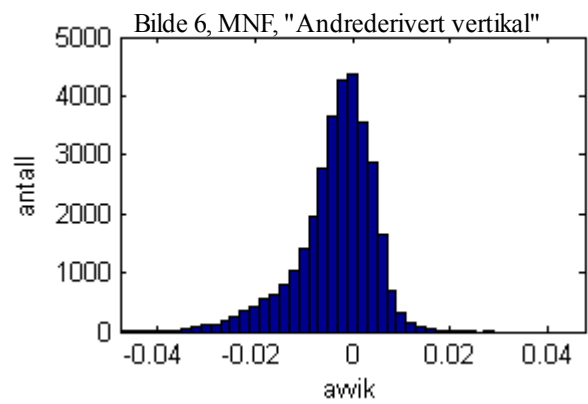
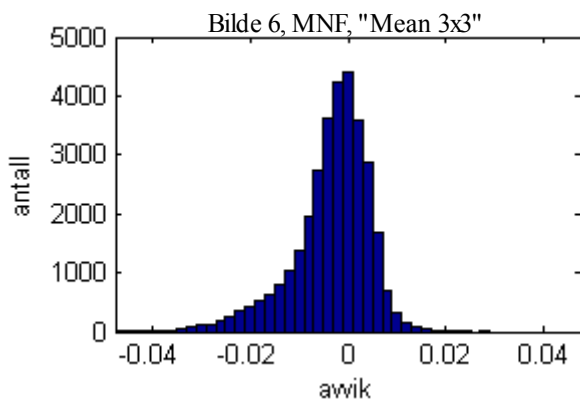
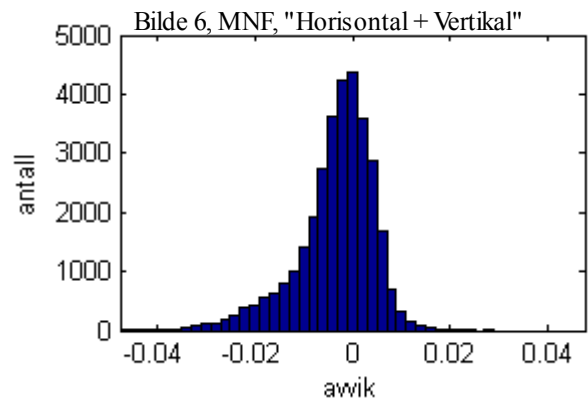
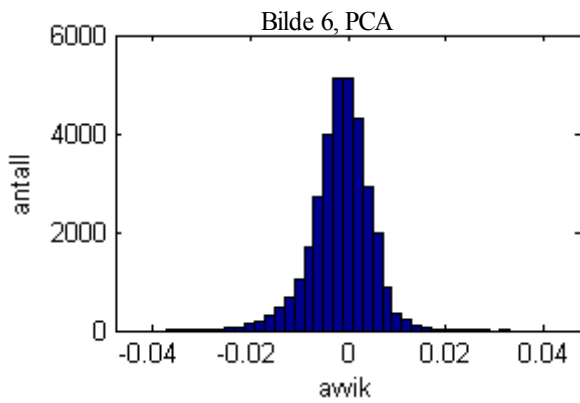
Bilde 9, Wavelet, 4 itterasjoner, 75% reduksjon

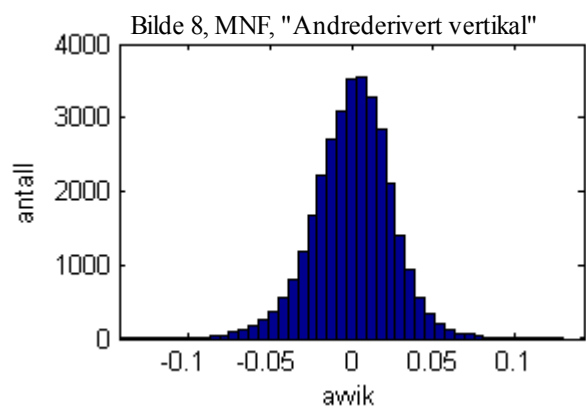
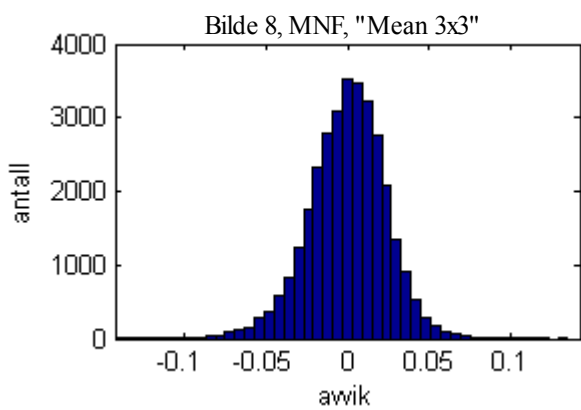
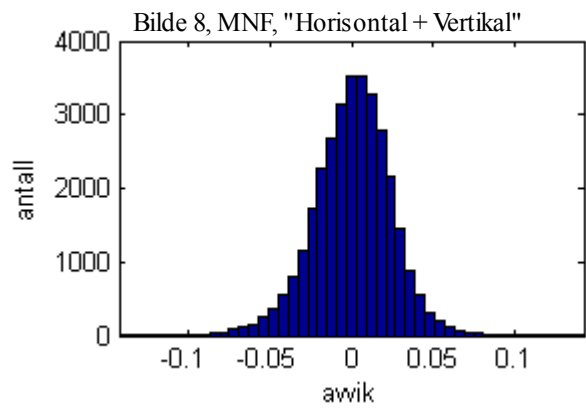
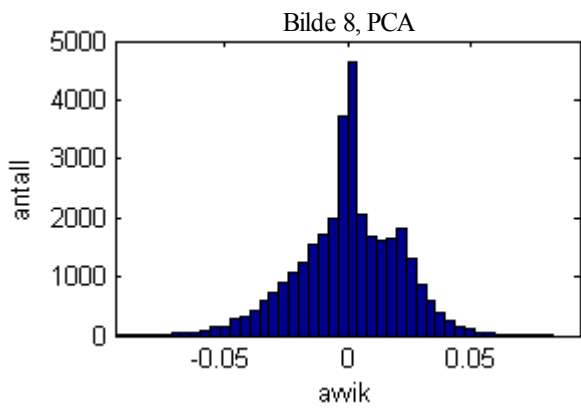
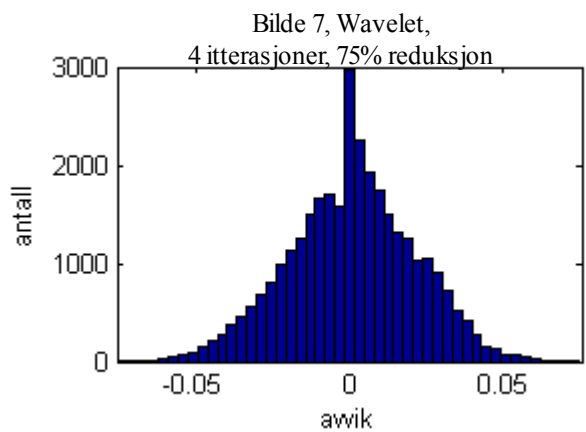
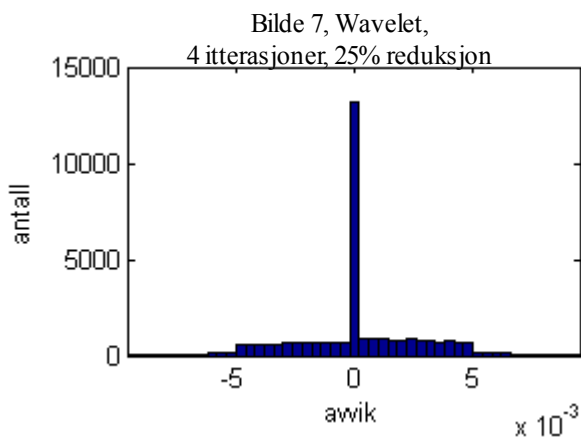
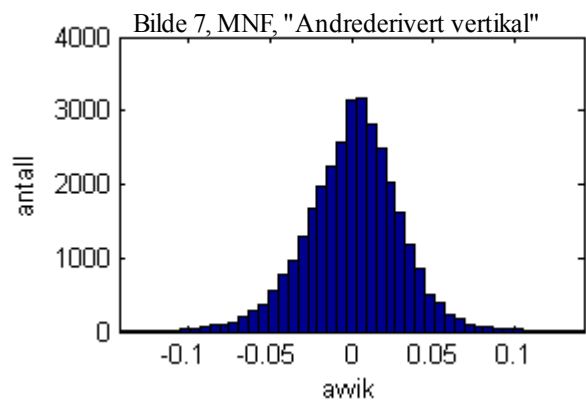
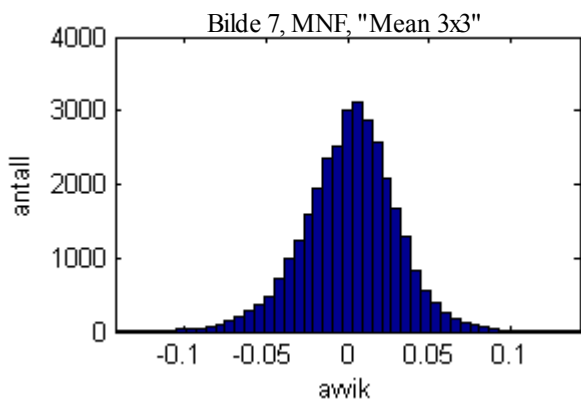


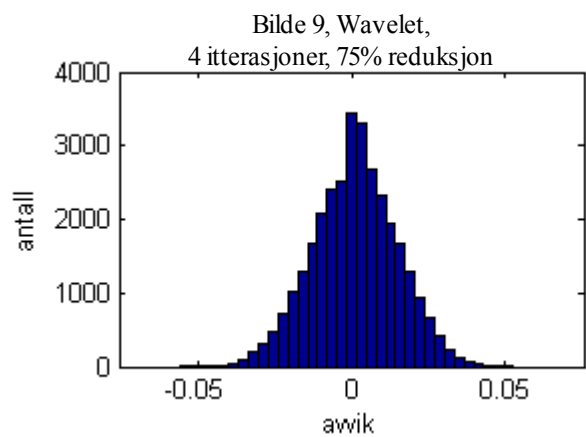
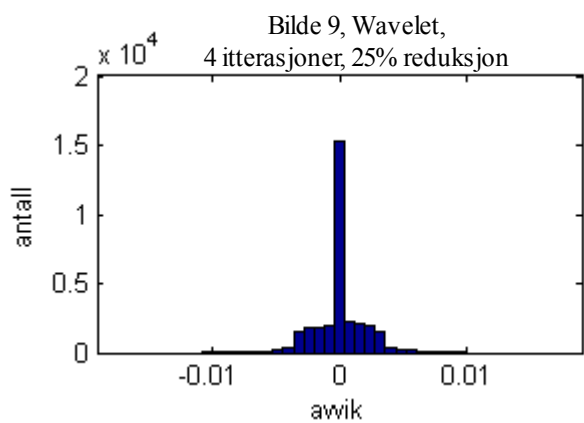
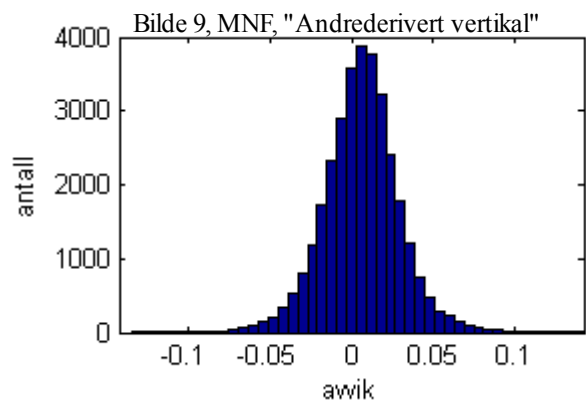
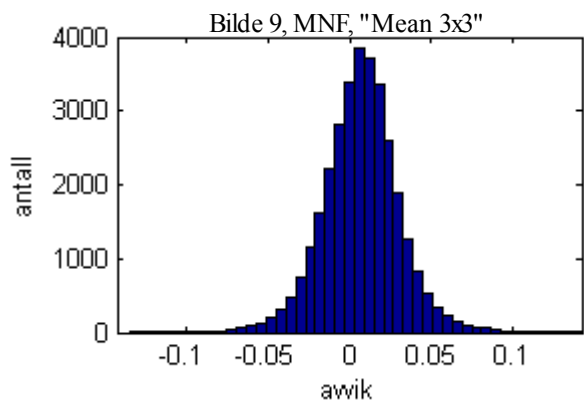
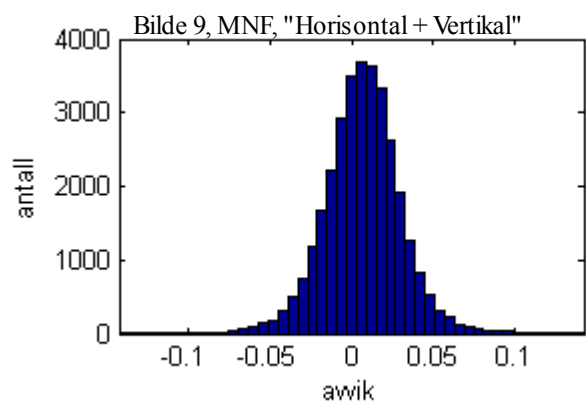
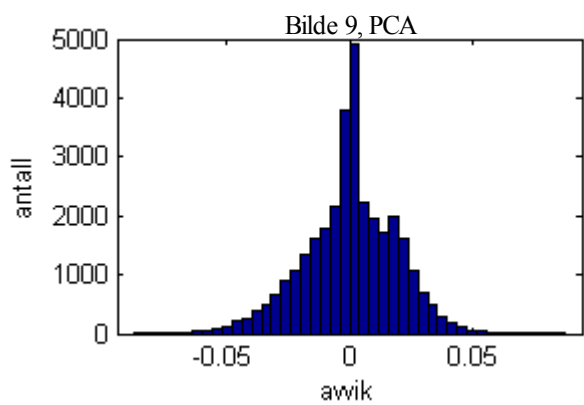
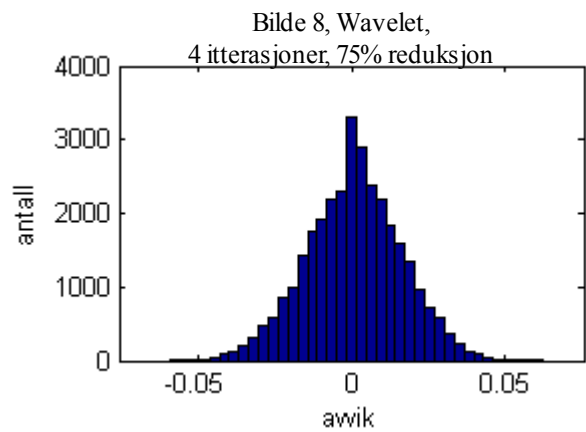
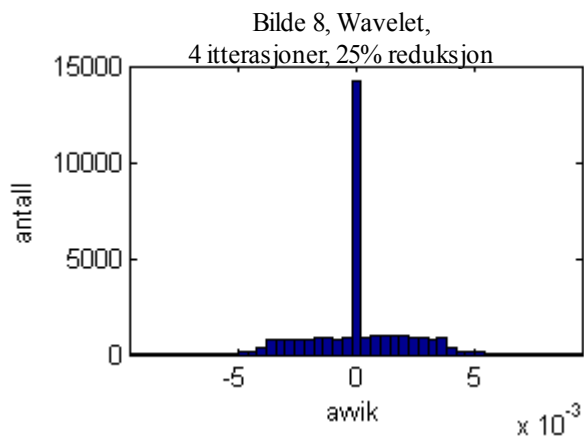
Bilde 9, Wavelet, 4 itterasjoner, 25% reduksjon



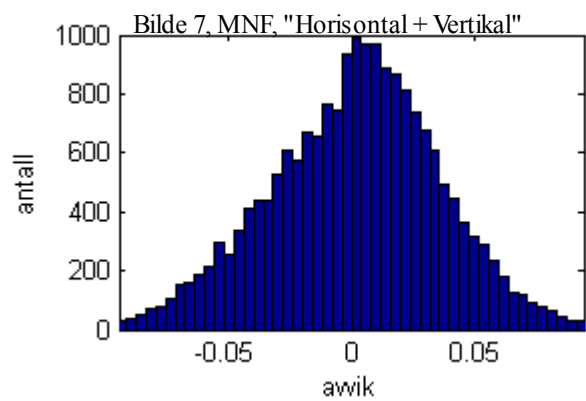
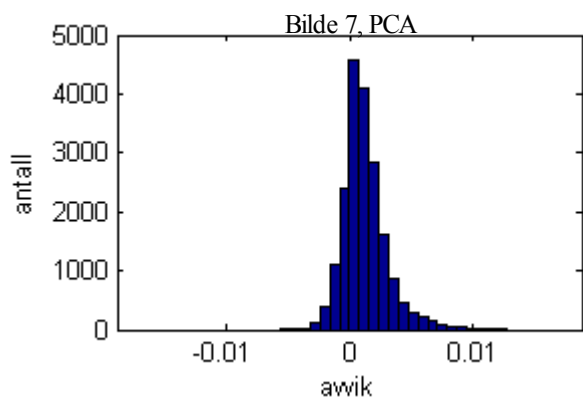
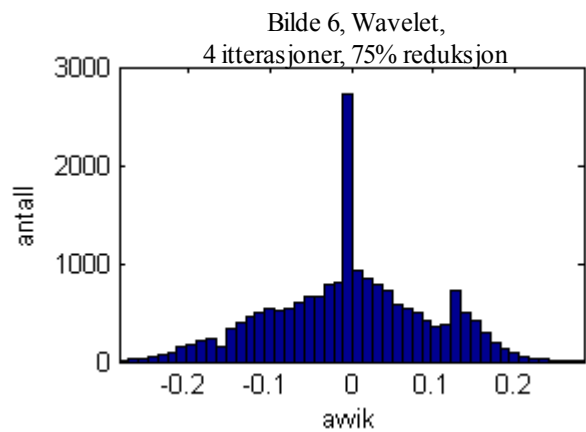
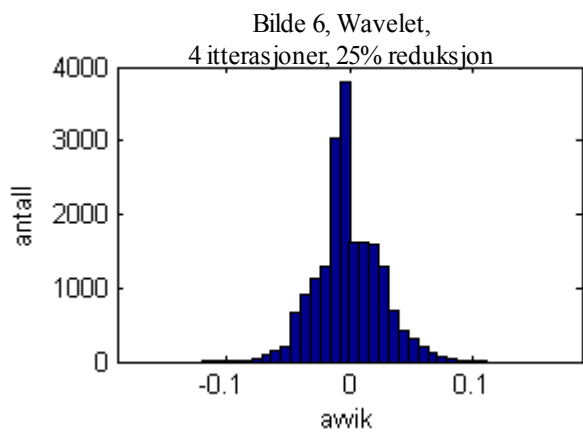
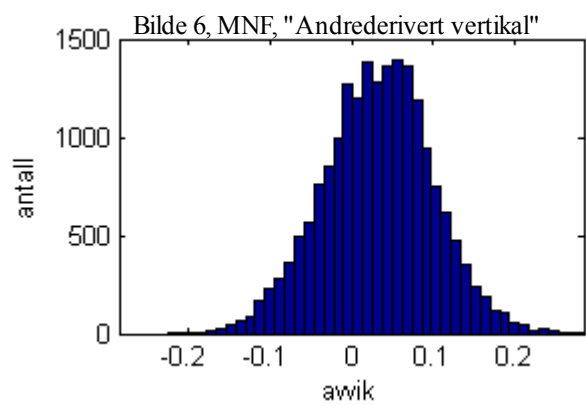
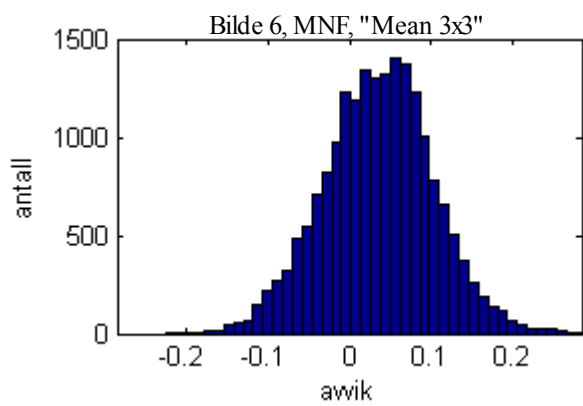
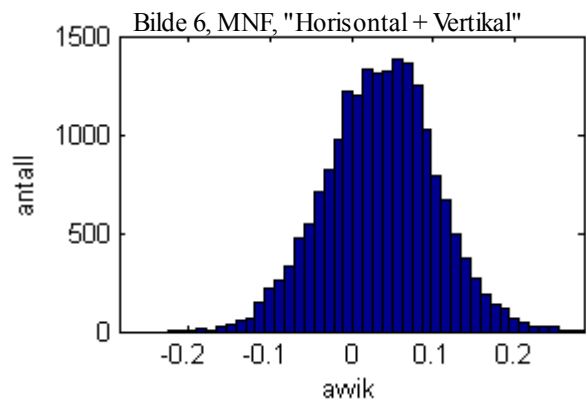
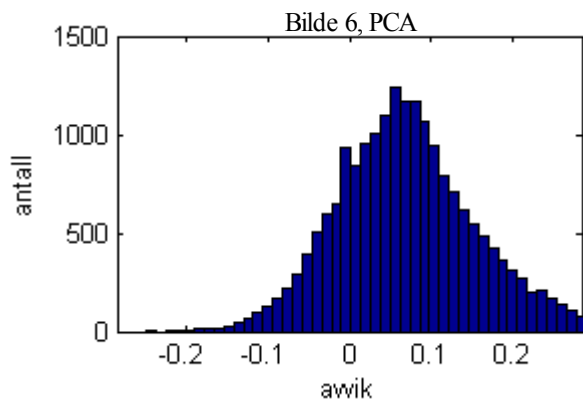
8.11 Støyfordeling til spekter for enkel rad

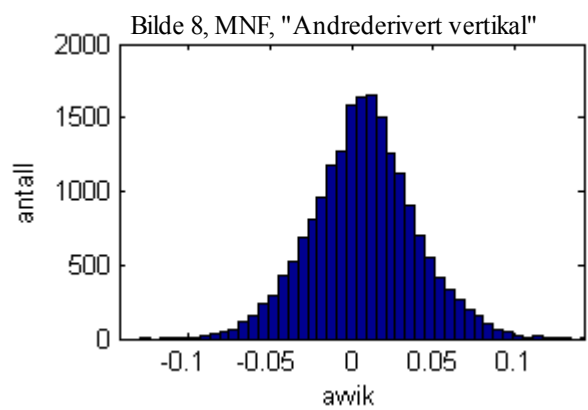
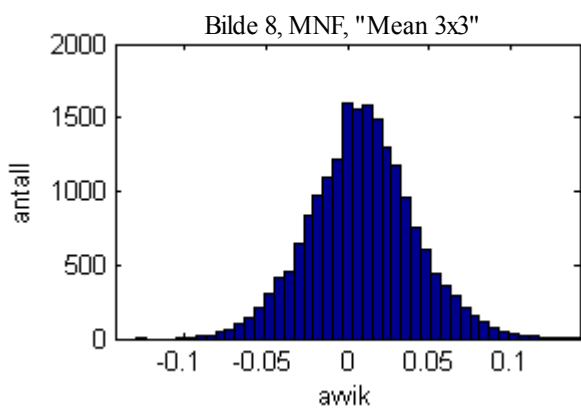
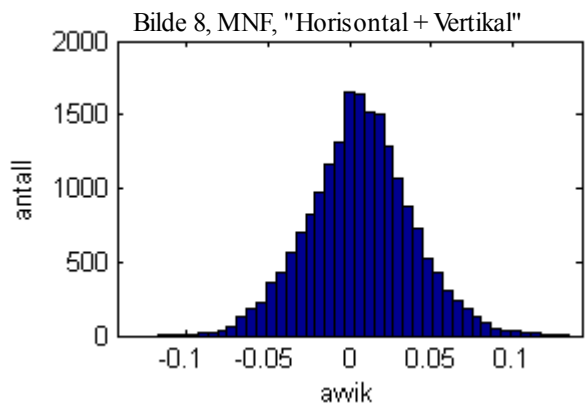
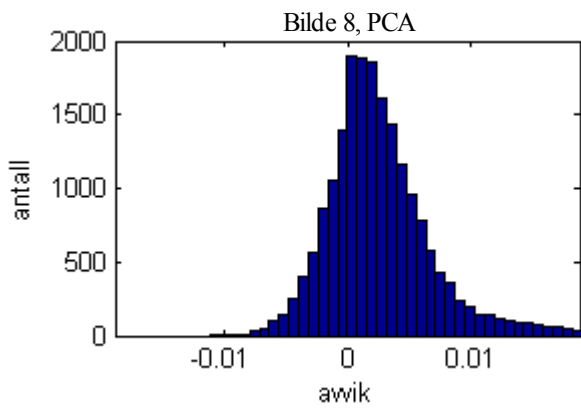
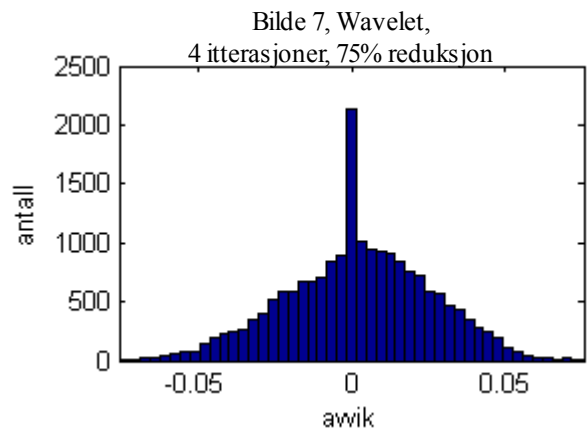
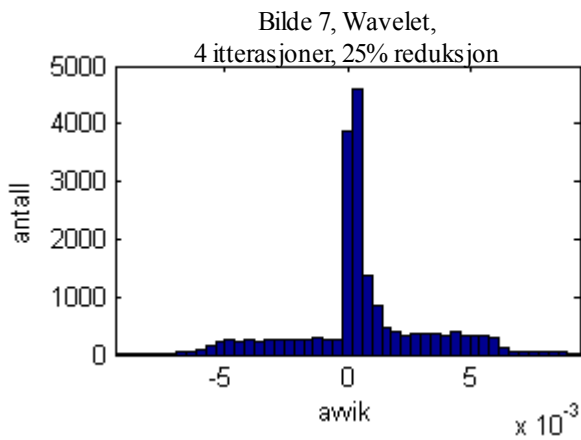
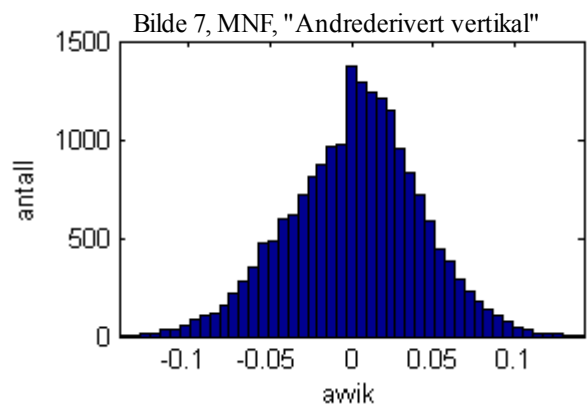
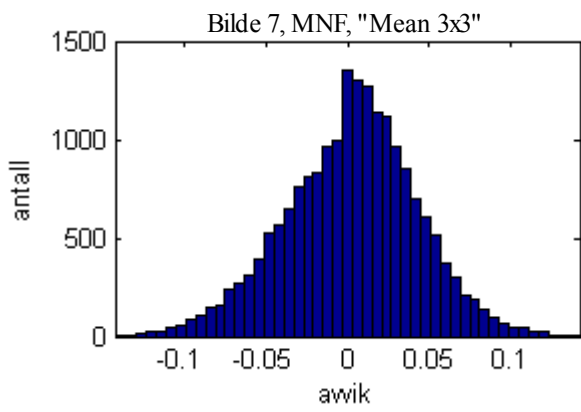


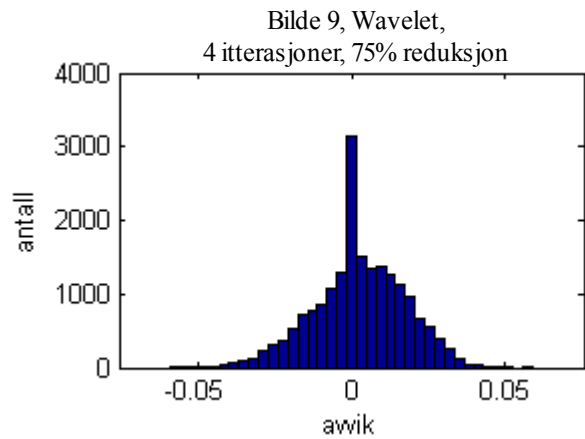
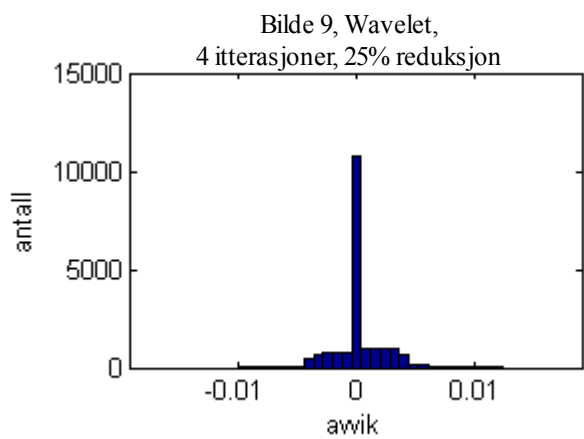
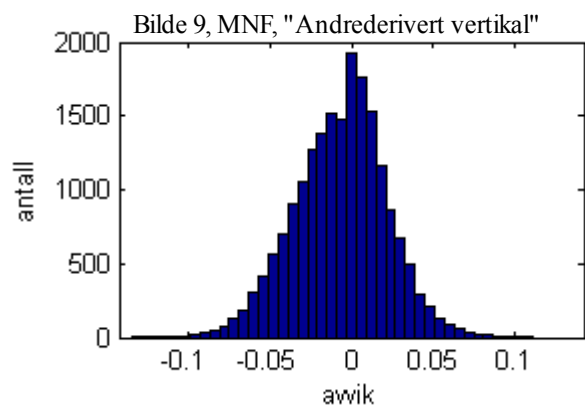
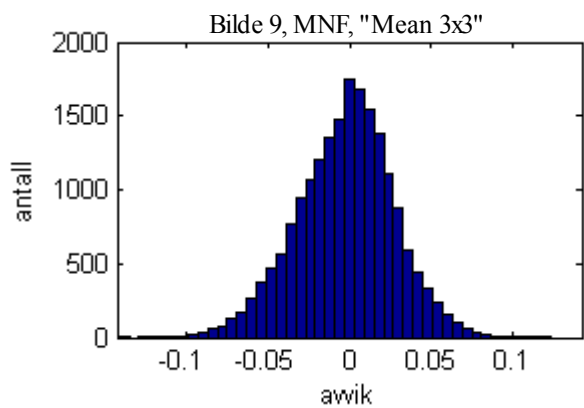
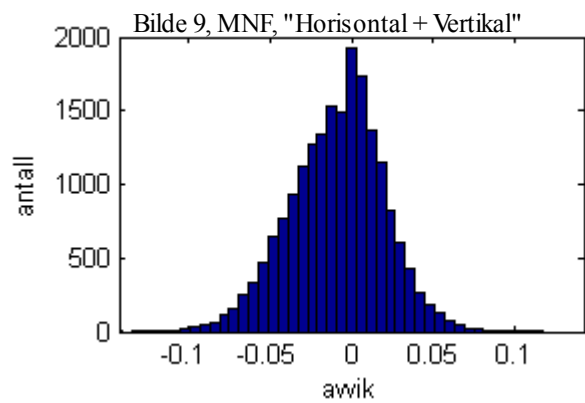
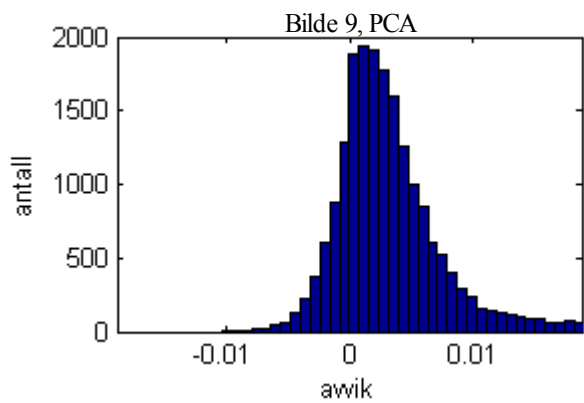
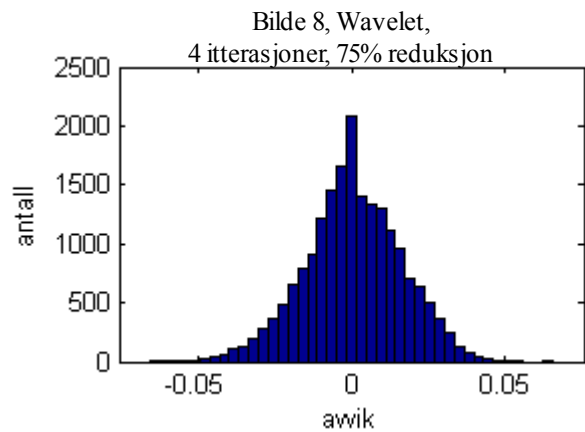
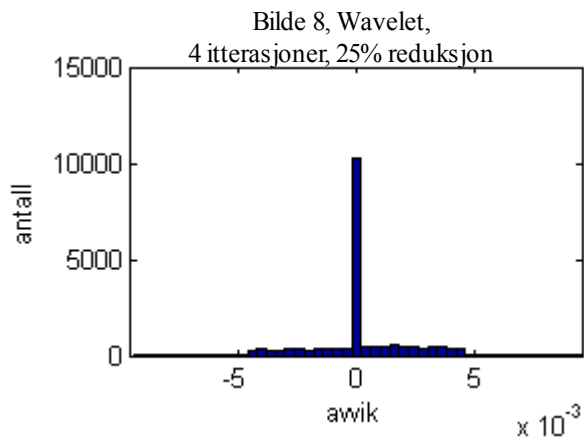




8.12 Støyfordeling til 3D-utsnitt for spektralbånd 10 (444nm)







8.13 Støyfordeling til 3D-utsnitt for spektralbånd 70 (662nm)

