




Universitetet  
i Stavanger

DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

## MASTEROPPGAVE

Studieprogram/spesialisering: Automatisering og signalbehandling	Vårsemesteret, 2015  Åpen
Forfatter: Daniel Myklatun Tveit	 ..... (signatur forfatter)
Fagansvarlig: Kjersti Engan  Veileder(e): Kjersti Engan	
Tittel på masteroppgaven: Deteksjon av pustefrekvens fra video  Engelsk tittel: Detection of respiratory rate from video	
Studiepoeng: 30	
Emneord: Bildebehandling, signalbehandling	Sidetall: .....60.....  + vedlegg/annet: .....7.....  Stavanger, .....15/06/2015..... dato/år

# Deteksjon av pustefrekvens fra video

Daniel Myklatun Tveit

15. juni 2015

# Innhold

<b>1</b>	<b>Innledning</b>	<b>4</b>
<b>2</b>	<b>Teori</b>	<b>5</b>
2.1	Hilbert transform . . . . .	5
2.1.1	Analytisk signal . . . . .	6
2.2	Global og lokal fase i bilder . . . . .	8
2.3	Riesz transform . . . . .	11
2.3.1	2D-generalisering av det analytiske signalet . . . . .	12
2.4	Bildepyramider . . . . .	13
2.4.1	Gaussisk pyramide . . . . .	13
2.4.2	Laplace-pyramide . . . . .	15
2.5	Tilnærmet Riesz transform . . . . .	16
2.6	Deteksjon av global bevegelse i video . . . . .	17
<b>3</b>	<b>Implementering</b>	<b>21</b>
3.1	Fase-metode . . . . .	21
3.2	Differanse-metode . . . . .	24
3.3	Deteksjon av forstyrrelser i biderammen . . . . .	26
3.4	Bestemming av pustefrekvens . . . . .	29
<b>4</b>	<b>Eksperiment</b>	<b>36</b>
4.1	Innstillinger . . . . .	36
4.2	Segmentering . . . . .	36
4.3	Gjennomsnittlig relativ feil . . . . .	36
4.4	Andel gyldige målinger . . . . .	37
4.5	Kjøretid . . . . .	37
<b>5</b>	<b>Resultat</b>	<b>38</b>
5.1	Video 1 . . . . .	38
5.2	Video 2 . . . . .	40
5.3	Video 3 . . . . .	43
5.4	Video 4 . . . . .	46
5.5	Video 5 . . . . .	49
5.6	Video 6 . . . . .	52
5.7	Samlet resultat . . . . .	55
<b>6</b>	<b>Diskusjon</b>	<b>57</b>
<b>7</b>	<b>Konklusjon</b>	<b>58</b>

## 1 Innledning

Hensikten med denne rapporten er å finne en robust metode for å detektere pustefrekvensen til en baby fra videoopptak. Deteksjon av pustefrekvens krever vanligvis en invasiv metode, men med videoovervåking kan dette gjøres uten kontakt med babyen. Det er ønskelig å lage en metode som er lite sensitiv for støy, selv for små bevegelser som kan være vanskelig å se med øyet. Det er tidligere blitt laget en metode for å detektere bevegelse i videoer [10]. Denne metoden er basert på lokal fase i bilderammene. Der det tidligere er blitt brukt en styrbar kompleks pyramide for å finne lokal fase brukes det her en tilnærming til Riesz transformen, som er en 2D-generalisering av Hilbert transformen [6]. Med en tilnærming til Riesz transformen kan det lages en metode som forsterker små bevegelser i videoer og kjører i sanntid [8]. I denne rapporten er det brukt en slik tilnærming sammen med deteksjon av bevegelse i video for å finne pustefrekvensen til en baby fra videoopptak.

## 2 Teori

I dette kapitlet presenteres det teoretiske grunnlaget for rapporten. Først presenteres Hilbert transformen og det analytiske signalet. Deretter forklares global og lokal fase i bilder. Etterpå presenteres Riesz transformen og 2D generaliseringen av det analytiske signalet. Videre skrives det om bildepyramider og til slutt presenteres en tilnærming av Riesz transformen og hvordan global bevegelse finnes fra video.

### 2.1 Hilbert transform

Hilbert transformen er en lineær operator som transformerer et signal til samme domene, men gir muligheten for å utvide signalet til det komplekse planet. Hilbert transformen til et reellt signal  $f(t)$  er definert som [1]:

$$f_H(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{f(\tau)}{\tau - t} d\tau \quad (1)$$

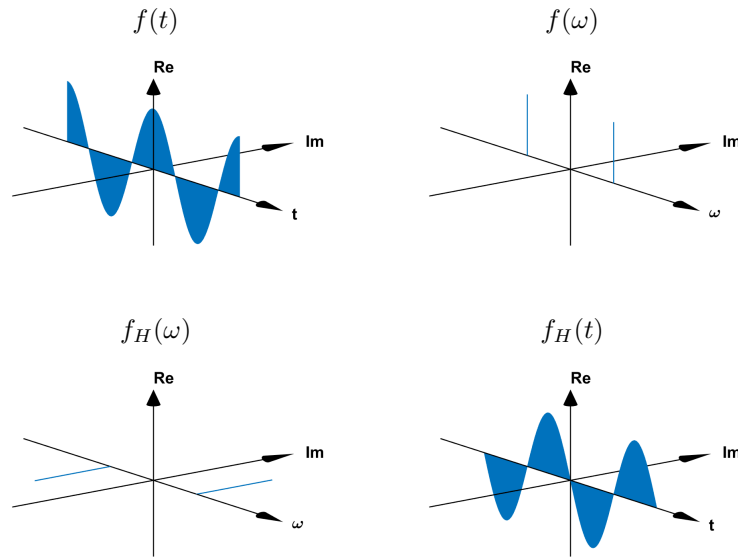
Det kan skrives som en konvolusjon med  $\frac{-1}{\pi t}$ :

$$f_H(t) = \frac{-1}{\pi t} * f(t) \quad (2)$$

Det kan vises at Fouriertransformen til  $\frac{-1}{\pi t}$  blir  $i \cdot \text{sign}(\omega)$  som er  $+i$  for positiv  $\omega$  og  $-i$  for negativ  $\omega$ . Dermed fås følgende sammenheng:

$$F_H(\omega) = i \cdot \text{sign}(\omega) \cdot F(\omega) \quad (3)$$

der  $F_H(\omega)$  er Fouriertransformen til  $f_H(t)$ ,  $\text{sign}(\cdot)$  er fortegnfunksjonen og  $F(\omega)$  er Fouriertransformen til det reelle signalet  $f(t)$ . Dermed kan det sees at Hilbert transformen fungerer som et filter med enhetsforsterkning der fasen forskyves med  $\pm \frac{\pi}{2}$  avhengig av fortegnet til  $\omega$ . Dette gjør at alle cosinus-komponenter blir transformert til negative sinus-komponenter og alle sinus-komponenter transformeres til cosinus-komponenter. Da vil alle like-funksjoner bli transformert til odde-funksjoner og odde-funksjoner bli like-funksjoner.



Figur 1: Oppe til venstre vises en cosinus-funksjon  $f(t)$ . Oppe til høyre vises Fouriertransformen til  $f(t)$ . Nede til venstre vises Fouriertransformen til Hilbert transformen av  $f(t)$ . Nede til høyre vises Hilbert transformen av  $f(t)$ .

I figur 1 kan det sees hvordan Hilbert transformen til et signal  $f(t)$  finnes ved å gå om frekvensdomenet. Med Fouriertransformen finnes  $F(\omega)$  fra det reelle signalet  $f(t)$ . Deretter finnes  $F_H(\omega)$  fra ligning 3 og den inverse Fouriertransformen til  $F_H(\omega)$  gir Hilbert transformen  $f_H(t)$  for det reelle signalet  $f(t)$ .

### 2.1.1 Analytisk signal

Et signal som ikke har negative frekvenskomponenter kalles et analytisk signal [2]. Det analytiske signalet til en reell funksjon  $f(t)$  er definert som:

$$f_A(t) = f(t) - i \cdot f_{Hi}(t) \quad (4)$$

der  $f_{Hi}(t)$  er Hilbert transformen av  $f(t)$  og kalles quadrature funksjonen til  $f(t)$ . Quadrature funksjonen til  $\cos(t)$  er  $-\sin(t)$  og det analytiske signalet til  $\cos(t)$  blir da  $e^{it}$ . Det kan sies at det analytiske signalet  $f_A(t)$  blir for  $f(t)$  på samme måte som  $e^{it}$  er for  $\cos(t)$ . Som et eksempel betraktes et amplitudemodulert signal:

$$f(t) = M(t) \cdot \cos(\omega_0 t)$$

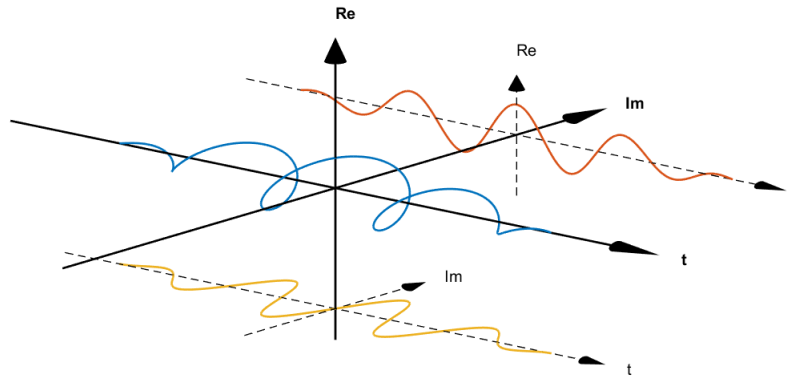
der  $\omega_0$  er bærefrekvensen og  $M(t)$  angir en amplitudemodulasjon. Da kan Hilbert transformen tilnærmes som:

$$f_{Hi}(t) \approx -M(t) \cdot \sin(\omega_0 t)$$

Det analytiske signalet finnes fra ligning 4:

$$f_A(t) \approx M(t) \cdot [\cos(\omega_0 t) + i \cdot \sin(\omega_0 t)] = M(t) \cdot e^{i\omega_0 t}$$

Dette kan sees i figur 2 der det analytiske signalet  $f_A(t)$  er en amplitude-modulert spiral i 3D-rommet utspent av det komplekse plan og tidsaksen. Prosjeksjonen av  $f_A(t)$  på planet som dannes av reell akse og tidsaksen viser det reelle signalet  $f(t)$ . Prosjeksjonen av  $f_A(t)$  på planet som dannes av imaginær akse og tidsaksen viser den imaginære delen av  $f_A(t)$  gitt av  $-i \cdot f_{Hi}(t)$ .



Figur 2: Blå kurve viser det analytiske signalet  $f_A(t)$ , rød kurve viser det reelle signalet  $f(t)$  og den gule kurven viser det analytiske signalets imaginære del  $-i \cdot f_{Hi}(t)$ .

Fra ligning 4 kan det sees at analytiske signalet kan skrives på polarkoordinat-form:

$$f_A(t) = A(t) \cdot e^{i\phi(t)} \quad (5)$$

Da er den momentane amplituden, eller envelopen, til  $f(t)$  gitt av:

$$A(t) = |f_A(t)| \quad (6)$$

Den momentane fasen er gitt av:

$$\phi(t) = \arg[f_A(t)] \quad (7)$$

Og den momentane frekvensen er gitt av raten som den momentane fasen endrer seg med:

$$\omega(t) = \frac{d\phi(t)}{dt} \quad (8)$$

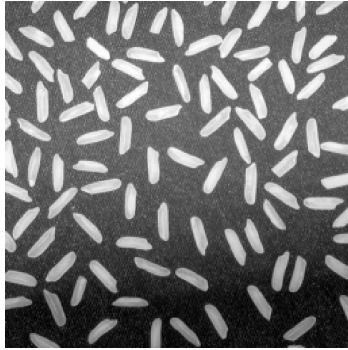
Det analytiske signalet kan da brukes til å finne momentane, eller lokale, egenskaper ved signalet  $f(t)$ .

## 2.2 Global og lokal fase i bilder

Med Fouriertransformen kan bilder representeres i form av en amplitude-komponent og en fase-komponent. Det er oppdaget at fase-komponenten inneholder mer av informasjonen i et bilde enn amplitude-komponenten [3]. Et eksempel på dette kan sees i figur 3.



Bilde 1



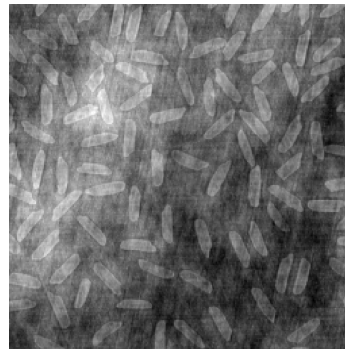
Bilde 2



Amplitude-komponenten til bilde 1  
med fase-komponenten til bilde 2

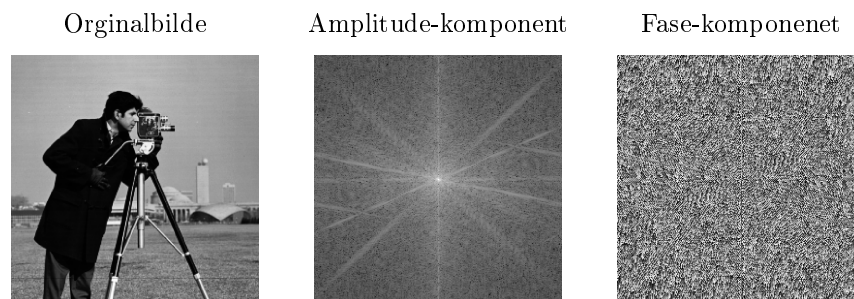


Amplitude-komponenten til bilde 2  
med fase-komponenten til bilde 1



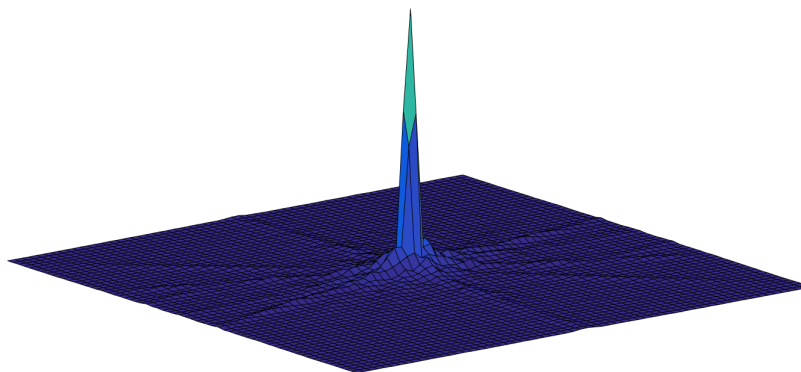
Figur 3: Øverst til venstre vises et bilde av noen riskorn. Oppe til høyre vises et bilde av en kameramann. Nede til venstre vises amplitude-komponenten til bildet av riskornene kombinert med fase-komponenten til bildet av kameramannen. Nede til høyre vises amplitude-komponenten til bildet av kameramannen med fase-komponenten til bildet av riskornene.

Fra figur 3 er det tydelig at det er fase-komponenten til et bilde som sier mest om innholdet i bildet. For å forklare hvorfor sees det nærmere på bildet av kameramannen:



Figur 4: Fra venstre til høyre vises orginalbilde av kameramannen, amplitude-komponenten med logaritmisk skala og fase-komponenten. Amplitude- og fase-komponenten vises med null-frekvens i midten.

Fra amplitude-komponenten i figur 4 kan det sees at den stort sett består av lave frekvens-komponenter. Dette er enda tydeligere fra figur 5 der amplitude-komponenten er gitt uten logaritmisk skala. Det kan sees at amplitude-komponenten vokser eksponensielt inn mot null-frekvensen i midten. Amplitude-komponenten til de fleste bilder antar en slik form og det er derfor vanskelig å skille bilder fra hverandre basert på bare amplitude-komponenten [4]. Fase-komponenten derimot består av frekvens-komponenter jevnt fordelt over hele frekvensspekteret. Dette gir en indikasjon på at mesteparten av informasjonen i bildet ligger i fase-komponenten.



Figur 5: 3D-visning av amplitude-komponenten i figur 4 uten logaritmisk skala. Null-frekvensen er i midten.

Fouriertransformen finner de globale egenskapene til et signal. Dette er egenskapene for hele signalet. For et ikke-stasjonært signal varierer frekvensinnholdet med tid/posisjon og det kan være hensiktsmessig å se på egenskapene i deler av signalet. Siden et bilde sjelden representerer et stasjonært signal gir ikke de globale egenskapene all informasjonen i bildet og det kan være nyttig å finne

de lokale egenskapene. Dette er egenskaper som uttrykker informasjonen i deler av bildet. En slik egenskap er lokale fase og gir informasjon om fasen i deler av bildet. Lokal fase inneholder mer informasjon om et bilde enn global fase og kan brukes til å rekonstruere et bilde helt uten amplitude-komponenten [5]. I kapittel 2.1 ble den lokale fasen til et 1D-signal funnet fra det analytiske signalet. Det ønskes å gjøre noe lignende for bilder og det er derfor nødvendig med en 2D-generalisering av det analytiske signalet.

### 2.3 Riesz transform

Riesz transformen er en tilnærmet 2D-generalisering av Hilbert transformen [6]. Riesz transformen er definert som følger [7]:

$$h_1(\mathbf{x}) = \frac{x_1}{2\pi|\mathbf{x}|^3} \quad (9)$$

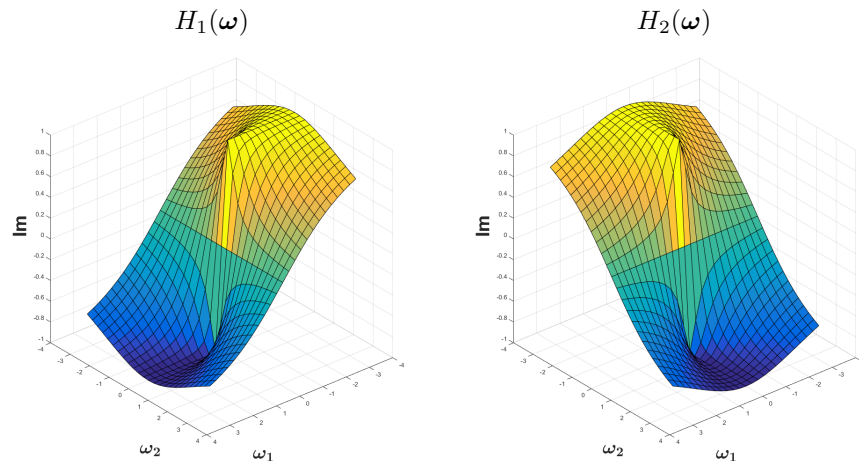
$$h_2(\mathbf{x}) = \frac{x_2}{2\pi|\mathbf{x}|^3} \quad (10)$$

der  $h_1$  og  $h_2$  er filtermaskene for Riesz transformen og  $\mathbf{x}$  er en vektor som representerer 2D-koordinater  $(x_1, x_2)$ . I frekvensdomenet blir dette:

$$H_1(\boldsymbol{\omega}) = -i \cdot \frac{\omega_1}{|\boldsymbol{\omega}|} \quad (11)$$

$$H_2(\boldsymbol{\omega}) = -i \cdot \frac{\omega_2}{|\boldsymbol{\omega}|} \quad (12)$$

der  $\boldsymbol{\omega}$  er en vektor som representerer romlig frekvens  $(\omega_1, \omega_2)$ . For DC-komponenten kan det vises at  $H_1(0) = H_2(0) = 0$  [6]. I figur 6 kan frekvensresponsen gitt av ligning 11 og 12 sees.



Figur 6: Frekvensresponsen til filtermaskene for Riesz transform.

Det kan sees av frekvensresponsen i figur 6 at filtrene har lik form, men er vridd  $90^\circ$  i forhold til hverandre.  $H_1(\boldsymbol{\omega})$  går i  $\omega_1$ -retning og  $H_2(\boldsymbol{\omega})$  går i  $\omega_2$ -retning.

### 2.3.1 2D-generalisering av det analytiske signalet

Det finnes en 2D-generalisering av det analytiske signalet som kalles *the monogenic signal* [7]. Generaliseringen gjør det mulig å finne lokal amplitude, lokal fase og lokal orientering. På samme måte som Hilbert transformen brukes for å danne det analytiske signalet, brukes Riesz transformen for å danne 2D-generaliseringen av det analytiske signalet. Riesz transformen til bildet  $f(\mathbf{x})$  finnes fra konvolusjon med filtermaskene for Riesz transform:

$$\mathbf{f}_R(\mathbf{x}) = \begin{pmatrix} R_1(\mathbf{x}) \\ R_2(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} (f * h_1)(\mathbf{x}) \\ (f * h_2)(\mathbf{x}) \end{pmatrix} \quad (13)$$

der  $\mathbf{x}$  er en vektor som representerer bildekoordinater,  $R_1$  og  $R_2$  er de to komponentene til Riesz transformen av bildet og  $h_1$  og  $h_2$  er filtermaskene fra ligning 9 og 10. Videre dannes 2D-generaliseringen av det analytiske signalet ved å kombinere bildet  $f(\mathbf{x})$  med Riesz transformen  $\mathbf{f}_R(\mathbf{x})$ :

$$\mathbf{f}_M(\mathbf{x}) = \begin{pmatrix} \mathbf{f}_R(\mathbf{x}) \\ f(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} R_1(\mathbf{x}) \\ R_2(\mathbf{x}) \\ f(\mathbf{x}) \end{pmatrix} \quad (14)$$

Dersom de tre komponentene av  $\mathbf{f}_M(\mathbf{x})$  sees på som kartesiske koordinater og konverteres til kulekoordinater kan lokal amplitude, lokal fase og lokal orientering finnes [8]:

$$A(\mathbf{x}) = \sqrt{f(\mathbf{x})^2 + R_1(\mathbf{x})^2 + R_2(\mathbf{x})^2} \quad (15)$$

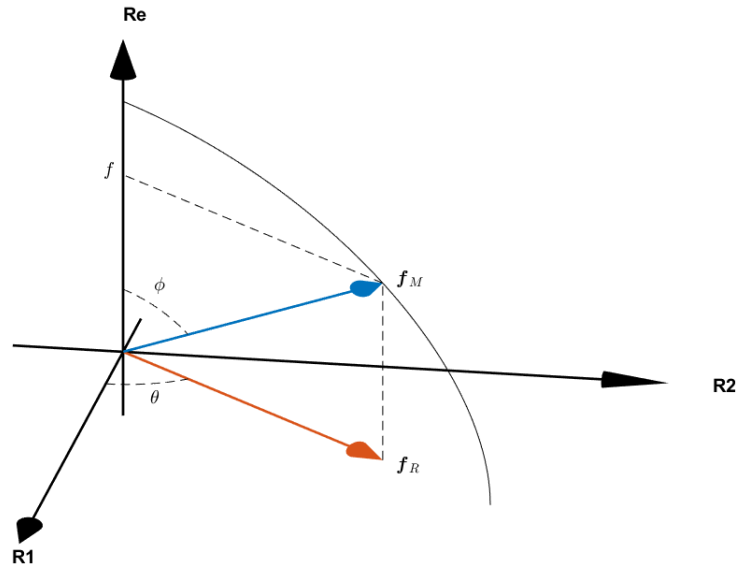
Den lokale amplituden  $A(\mathbf{x})$  til et bilde er gitt av lengden på vektoren som går fra origo til  $\mathbf{f}_M(\mathbf{x})$ . Den lokale fasen  $\phi(\mathbf{x})$  til et bilde er gitt av vinkelen mellom den reelle bilde-aksen og planet som dannes av Riesz transformen til bildet:

$$\phi(\mathbf{x}) = \tan^{-1} \left( \frac{\sqrt{R_1(\mathbf{x})^2 + R_2(\mathbf{x})^2}}{f(\mathbf{x})} \right) \quad (16)$$

der  $\tan^{-1}(\cdot) \in [0, \pi)$ . Den lokale orienteringen  $\theta(\mathbf{x})$  til et bilde er gitt av vinkelen mellom  $R_1$ -akse og vektoren som går fra origo til  $\mathbf{f}_R(\mathbf{x})$ :

$$\theta(\mathbf{x}) = \tan^{-1} \left( \frac{R_2(\mathbf{x})}{R_1(\mathbf{x})} \right) \quad (17)$$

I figur 7 kan det sees hvordan lokal amplitude, lokal fase og lokal orientering finnes fra 2D-generaliseringen av det analytiske signalet.



Figur 7: I figuren vises 3D-rommet utspent av  $R_1$ -,  $R_2$ - og den reelle bildeaksen. Bildet  $f(\mathbf{x})$  går langs den reelle bildeaksen. Blå pil viser vektoren som går fra origo til 2D-generaliseringen av det analytiske signalet  $f_M(\mathbf{x})$ . Rød pil viser vektoren som går fra origo til Riesz transformen  $f_R(\mathbf{x})$ . Lokal amplitude er gitt av lengden fra origo til  $f_M(\mathbf{x})$ ,  $\phi(\mathbf{x})$  angir lokal fase og  $\theta(\mathbf{x})$  angir lokal orientering.

## 2.4 Bildepyramider

En bildepyramide er en måte å representere et bilde over flere skalaer. Navnet kommer av at for hvert trinn i pyramiden nedsamples og filtreres bildet slik at trinnene danner en pyramidestruktur når de stables oppå hverandre. Det velges typisk en nedsampling med en faktor på to for hvert trinn. Det siste trinnet viser bildet i grov skala og det første trinnet viser bildet i fin skala. Med en bildepyramide kan det letes over skalaer for å finne interessante områder eller punkter i bildet.

### 2.4.1 Gaussisk pyramide

En Gaussisk pyramide er en pyramidestruktur der hvert trinn nedsamples og lavpassfiltreres som gitt av følgende [9]:

$$g_l(i, j) = \sum_{m=-2}^2 \sum_{n=-2}^2 w(m, n) \cdot g_{l-1}(2i + m, 2j + n) \quad (18)$$

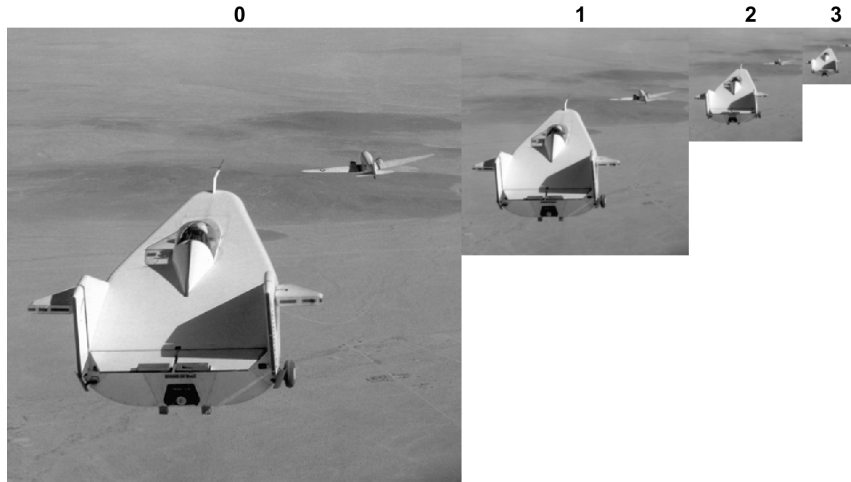
der  $g_l$  er bildet i pyramidenivå  $l$  med  $g_0$  som originalbildet og  $w$  er en filtermaske gitt ved:

$$w(m, n) = \hat{w}(m) \cdot \hat{w}(n) \quad (19)$$

der  $\hat{w}$  finnes fra:

$$\begin{aligned} \hat{w}(0) &= a \\ \hat{w}(-1) &= \hat{w}(1) = \frac{1}{4} \\ \hat{w}(-2) &= \hat{w}(2) = \frac{1}{4} - \frac{a}{2} \end{aligned}$$

Med  $a = 0.4$  fås en filtermaske som ligner Gaussisk form [9]. Det kan sees at for hvert trinn i den Gaussiske pyramiden nedsamples bildet fra trinnet før med faktor på to og filtreres med en Gaussisk filtermaske. I figur 8 kan det sees fire trinn i en Gaussisk pyramide.



Figur 8: Fire trinn i en Gaussisk pyramide.

En Gaussisk pyramide kan betraktes som en samling lavpassfiltrerte versjoner av originalbildet.

### 2.4.2 Laplace-pyramide

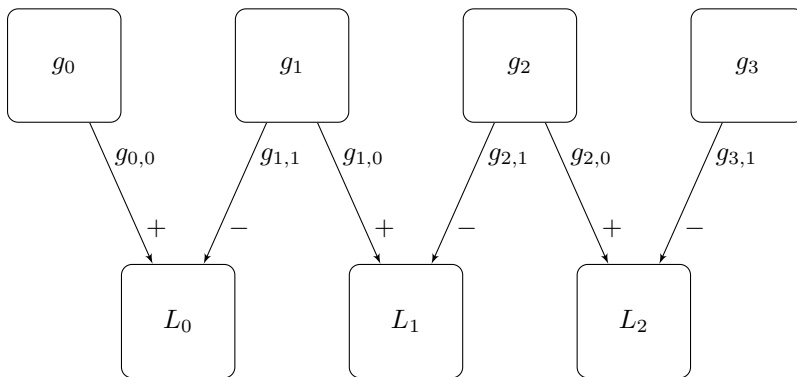
Laplace-pyramiden til et bilde finnes fra differansen mellom trinnene i den Gaussiske pyramiden til bildet. For å finne denne differansen må bildet i et trinn interpoleres til størrelsen av bildet i trinnet før. Denne interpolasjonen er gitt av følgende [9]:

$$g_{l,k}(i, j) = 4 \cdot \sum_{m=-2}^2 \sum_{n=-2}^2 w(m, n) \cdot g_{l,k-1} \left( \frac{i-m}{2}, \frac{j-n}{2} \right) \quad (20)$$

der  $g_{l,k}$  er bildet i pyramidenivå  $l$  utvidet til størrelsen på bildet i  $k$  nivå før og  $w$  er filtermasken gitt i ligning 19. Bare verdier der  $\frac{i-m}{2}$  og  $\frac{j-n}{2}$  er heltall blir tatt med i summen. Interpolasjon til ett trinn før øker størrelsen på et bilde fra  $(M+1) \times (N+1)$  til  $(2M+1) \times (2N+1)$ . Et trinn i Laplace-pyramiden finnes fra [9]:

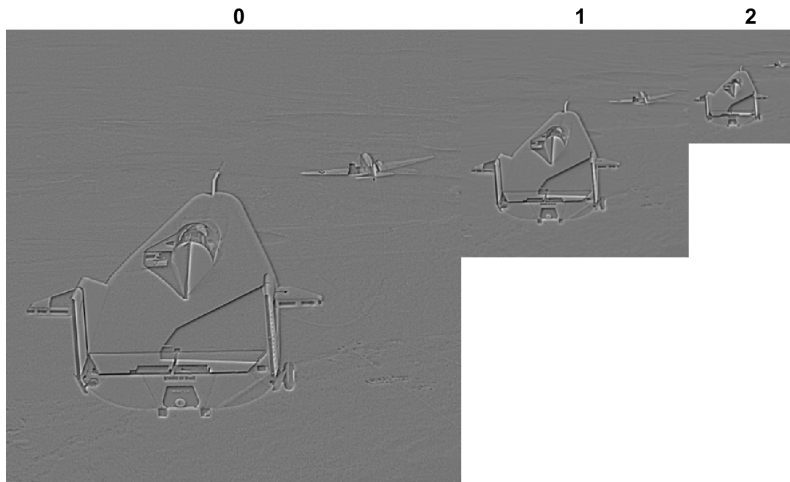
$$L_l = g_l - g_{l+1,1} \quad (21)$$

der  $g_l$  er et trinn i den Gaussiske pyramiden og  $g_{l+1,1}$  er det neste trinnet, men interpolert til størrelsen av  $g_l$ . I figur 9 kan det sees hvordan Laplace-pyramiden dannes fra den Gaussiske pyramiden.



Figur 9: Den øverste rekken viser fire trinn i en Gaussisk pyramide. Differansen mellom to trinn danner et trinn i Laplace-pyramiden. For å finne differansen må det minste bildet i de to trinnene interpoleres til størrelsen av det største.

Dersom den Gaussiske pyramiden fra figur 8 brukes til å generere en Laplace-pyramide som i figur 9 fås resultatet i figur 10.



Figur 10: Tre trinn i en Laplace-pyramide funnet fra den Gaussiske pyramiden i figur 8.

Siden en Gaussisk pyramide består av lavpassfiltrerte versjoner av orginalbildet vil Laplace-pyramiden bestå av båndpassfiltrerte bilder.

## 2.5 Tilnærmet Riesz transform

For å finne Riesz transformen til en bilderamme i en video er det ikke ønskelig å gå veien om frekvensdomenet siden dette medfører mye utregninger. I stedet brukes to filtermasker som gir en tilnærmet Riesz transform. Disse filtermaskene er gitt som [8]:

$$h_1 = (0.5, 0, -0.5) \quad (22)$$

$$h_2 = (0.5, 0, -0.5)^T \quad (23)$$

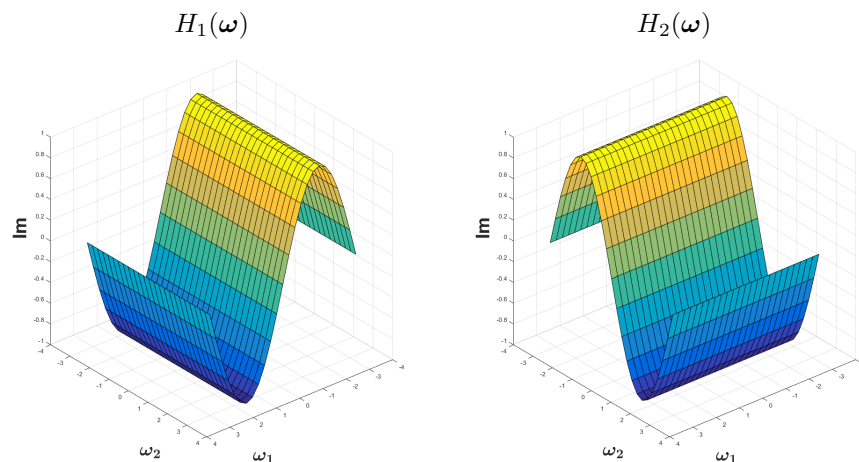
Fourier transformen til disse filtermaskene er gitt av:

$$H_1(\boldsymbol{\omega}) = -i \cdot \sin(\omega_1) \quad (24)$$

$$H_2(\boldsymbol{\omega}) = -i \cdot \sin(\omega_2) \quad (25)$$

der  $\boldsymbol{\omega}$  er en vektor som angir romlig frekvens  $(\omega_1, \omega_2)$ . For  $\omega_1, \omega_2 \approx \frac{\pi}{2}$  er frekvensresponsen tilnærmet lik Riesz transformen [8]. Det betyr at for båndpassfiltrerte bilder der senterfrekvensen er rundt  $\frac{\pi}{2}$  er responsen med filtermaskene i ligning 22 og 23 tilnærmet lik Riesz transformen. For en Laplace-pyramide der trinnene nedsamples med en faktor på to vil hvert trinn være et slikt båndpassfiltrert bilde. I figur 11 kan frekvensresponsen fra ligning 24 og 25 sees.





Figur 11: Frekvensresponsen av filtermaskene for tilnærmet Riesz transform.

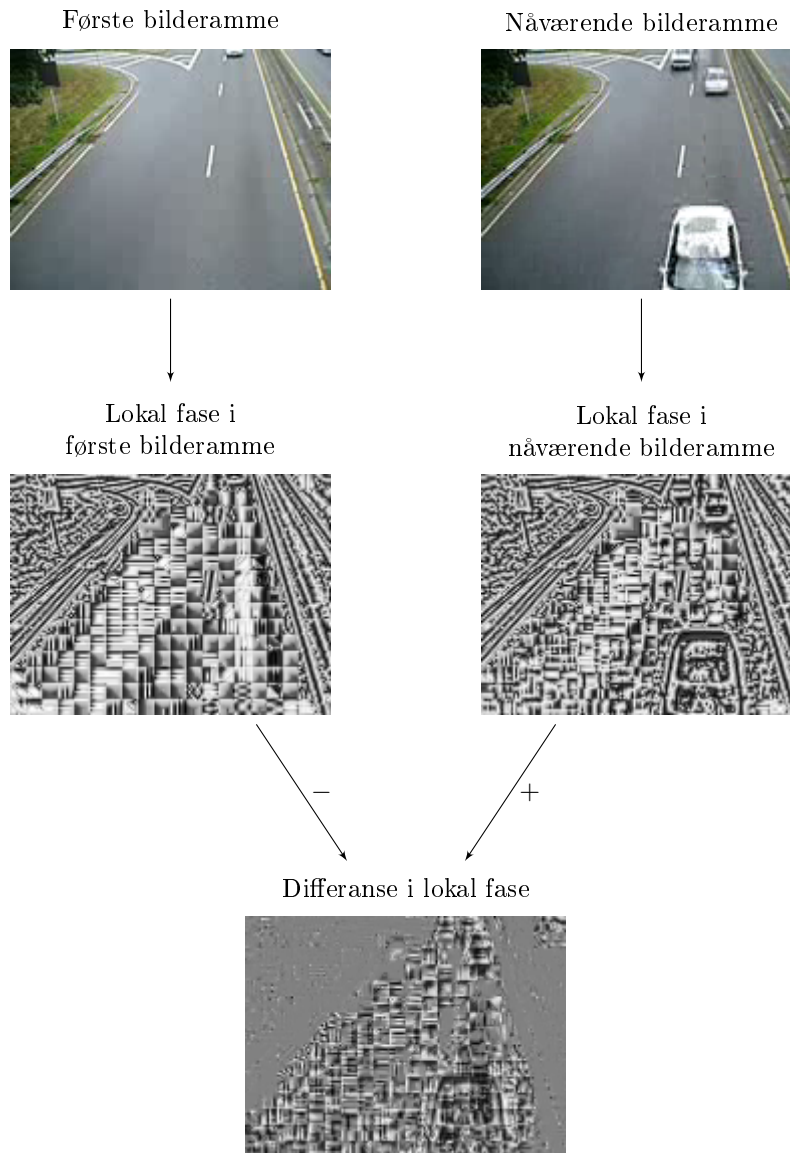
Dersom frekvensresponsen av filtermaskene for tilnærmet Riesz transform i figur 11 sammenlignes med frekvensresponsen av filtermaskene for Riesz transform i figur 6 kan det sees at formene ikke er helt like, men for  $\omega_1, \omega_2 \approx \frac{\pi}{2}$  er tilnærmingen ganske god.

## 2.6 Deteksjon av global bevegelse i video

Global bevegelse i en video kan finnes ved hjelp av lokal amplitude og lokal fase [10]. Global bevegelse uttrykker den bevegelsen som kan sees i videoen. Først finnes differansen i lokal fase mellom nåværende bilderamme og en fast bilderamme. Den faste bilderammen velges typisk til å være den første bilderammen i en videosekvens:

$$\phi_D(\mathbf{x}) = \phi(\mathbf{x}) - \phi_0(\mathbf{x}) \quad (26)$$

der  $\phi(\mathbf{x})$  er lokal fase i nåværende bilderamme og  $\phi_0(\mathbf{x})$  er lokal fase i den første bilderammen. Dette kan sees i figur 12.

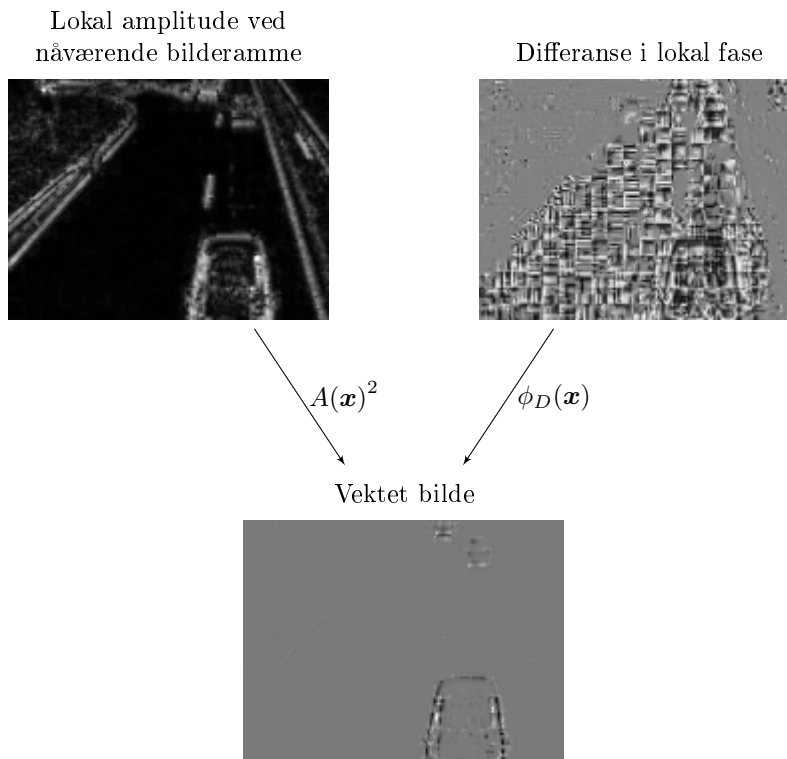


Figur 12: Øverst vises to bilderammer fra en video av en motorvei. I midten vises den lokale fasen til bilderammene. Nederst vises differansen i lokal fase. Lokal fase vises som gråtonebilder i området  $[0, \pi)$  og differansebildet vises som gråtonebilde i området  $(-\pi, \pi)$ .

Deretter vektes differansen i lokal fase med lokal amplitude  $A(\mathbf{x})$  til nåværende bilderamme:

$$f_W(\mathbf{x}) = \phi_D(\mathbf{x}) \cdot A(\mathbf{x})^2 \quad (27)$$

Grunnen til vektingen er at den lokale fasen ikke vil være entydig i områder der det er lite tekstur. Den lokale amplituden gir et mål for teksturstyrke slik at en vekting med  $A(\mathbf{x})^2$  gir et bilde der bare de strukturene som har endret seg fra første bilderamme fremkommer [10]. Dette kan sees i figur 13.



Figur 13: Oppe til venstre vises lokal amplitude til nåværende bilderamme, oppe til høyre vises differansen i lokal fase fra figur 12 og nederst vises vektet bilde funnet fra ligning 27.

Dersom det summeres over alle pikslene i hvert vektet bilde i en video fås et signal som uttrykker den globale bevegelsen i videoen [10]:

$$y_M(n) = \sum_{\mathbf{x}} f_W(\mathbf{x}) \quad (28)$$

der  $y_M(n)$  er et tidsdiskret signal som uttrykker den globale bevegelsen i videoen,  $\mathbf{x}$  er en vektor som representerer bildekoordinater og  $f_W(\mathbf{x})$  er et vektet

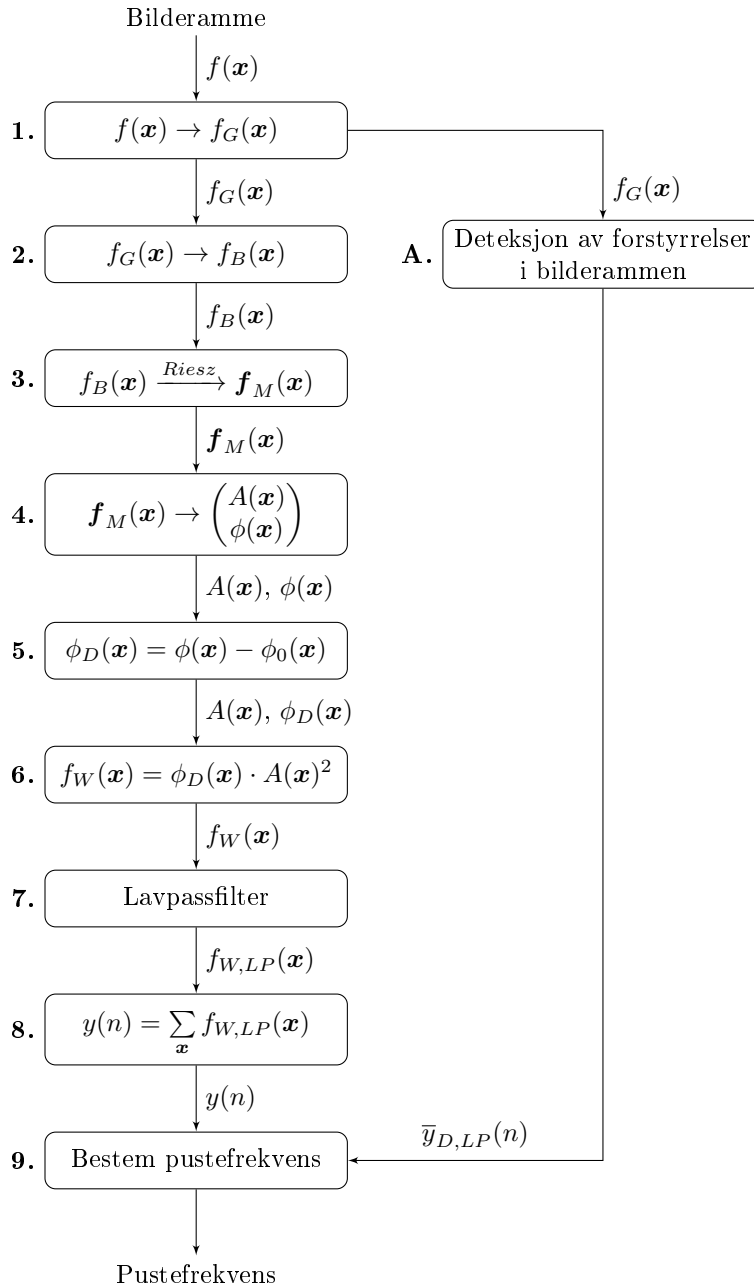
bilde. Ett vektet bilde gir ett punkt i signalet  $y_M(n)$ . For hele videoen dannes hele signalet  $y_M(n)$ .

## 3 Implementering

For å detektere pustefrekvensen til en baby fra videoopptak trengs det en metode som kan detektere små bevegelser og er lite sensitiv for støy. Metoden bør være nøyaktig, men ikke for omfattende siden det er ønskelig at metoden skal kunne kjøre i sanntid. I dette kapitlet presenteres de to metodene som er laget for å detektere pustefrekvens. Først er en metode basert på endring i lokal fase som blir kalt fase-metoden. Den andre metoden er basert på endring i intensitet og kalles differanse-metoden. Differanse-metoden er laget som et enklere alternativ for å sammenligne med fase-metoden. Fase-metoden er mer omfattende og krever mer regnekraft.

### 3.1 Fase-metode

Fase-metoden bruker tilnærmet Riesz transform for å danne en 2D-generalisering av det analytiske signalet for hver bilderamme. Deretter finnes lokal amplitude og lokal fase for bilderammene. Fase-metoden bruker endring i lokal fase mellom nåværende og første bilderamme. Denne differansen vektet med lokal amplitude for nåværende bilderamme. For hvert vektet bilde i videoen summeres det over alle pikslene og det dannes et signal som beskriver den globale bevegelsen i videoen. Siden lokal fase er lite sensitiv for støy er det ventet at en slik metode vil være robust. Fase-metoden er mer omfattende enn differanse-metoden, men implementert som ferdig kompilert kode er det ventet at den kan kjøre i sanntid. I figur 14 kan det sees et flytskjema for hvordan fase-metoden er implementert.



Figur 14: Flytskjema for fase-metoden.

Fase-metoden bruker et glidende tidsvindu med størrelse  $T_W$  sekund som flyttes med  $T_U$  sekund om gangen. Dette er gjort slik at pustefrekvensen bestemmes

for en del av videoen om gangen og oppdateres etterhvert som vinduet flyttes. Fase-metoden bruker antall bilderammer tilsvarende vindusstørrelsen  $T_W$  for å detektere pustefrekvens. Når tidsvinduet flyttes bruker fase-metoden like mange bilderammer, men forskjøvet med bilderammer tilsvarende oppdateringstiden  $T_U$ . Videre er punktene i flytskjemaet for fase-metoden i figur 14 beskrevet nærmere:

1. Bilderammen konverteres til et gråtonebilde:

$$f(\mathbf{x}) \rightarrow f_G(\mathbf{x})$$

der  $f(\mathbf{x})$  er nåværende bilderamme,  $\mathbf{x}$  er en vektor som representerer bildekoordinater og  $f_G(\mathbf{x})$  er et gråtonebilde av bilderammen.

2. Gråtonebildet båndpassfiltreres ved å utvikle en Laplace-pyramide. Det brukes ett trinn i Laplace-pyramiden for å lage et båndpassfiltrert bilde som er av samme størrelse som bilderammen:

$$f_G(\mathbf{x}) \rightarrow f_B(\mathbf{x})$$

der  $f_G(\mathbf{x})$  er et gråtonebilde av bilderammen og  $f_B(\mathbf{x})$  er gråtonebildet båndpassfiltrert.

3. Riesz transformen til det båndpassfiltrerte bildet finnes med filtermaskene for tilnærmet Riesz transform i ligning 22 og 23. Deretter dannes 2D-generaliseringen av det analytiske signalet med ligning 14:

$$f_B(\mathbf{x}) \xrightarrow{\text{Riesz}} \mathbf{f}_M(\mathbf{x})$$

der  $f_B(\mathbf{x})$  er båndpassfiltrert bilderamme og  $\mathbf{f}_M(\mathbf{x})$  er 2D-generaliseringen av det analytiske signalet.

4. Fra 2D-generaliseringen av det analytiske signalet finnes lokal amplitude med ligning 15 og lokal fase med ligning 16:

$$\mathbf{f}_M(\mathbf{x}) \rightarrow \begin{pmatrix} A(\mathbf{x}) \\ \phi(\mathbf{x}) \end{pmatrix}$$

der  $\mathbf{f}_M(\mathbf{x})$  er 2D-generaliseringen av det analytiske signalet,  $A(\mathbf{x})$  er lokal amplitude og  $\phi(\mathbf{x})$  er lokal fase.

5. Differansen i lokal fase mellom nåværende og første bilderamme finnes som:

$$\phi_D(\mathbf{x}) = \phi(\mathbf{x}) - \phi_0(\mathbf{x})$$

der  $\phi(\mathbf{x})$  er lokal fase i nåværende bilderamme og  $\phi_0(\mathbf{x})$  er lokal fase i den første bilderammen.

6. Differansen i lokal fase vektes med lokal amplitude kvadrert for å danne et vektet bilde:

$$f_W(\mathbf{x}) = \phi_D(\mathbf{x}) \cdot A(\mathbf{x})^2$$

der  $\phi_D(\mathbf{x})$  er differansen i lokal fase og  $A(\mathbf{x})$  er lokal amplitude til nåværende bilderamme.

7. Vektet bilde lavpassfiltreres i tid etter differanseligningen:

$$y(n) = \alpha \cdot x(n) + (1 - \alpha) \cdot y(n - 1)$$

der  $y(n)$  er filtrert utgang,  $\alpha$  er en veiefaktor i området  $[0, 1]$ ,  $x(n)$  er ufiltrert inngang og  $y(n - 1)$  er filtrert utgang ved forrige tidsskritt. Veiefaktoren  $\alpha$  er satt av en bruker. Dette gir et lavpassfiltrert vektet bilde  $f_{W,LP}(\mathbf{x})$ .

8. Ett punkt i signalet for global bevegelse i videoen dannes ved å summere over alle pikslene i lavpassfiltrert vektet bilde:

$$y(n) = \sum_{\mathbf{x}} f_{W,LP}(\mathbf{x})$$

der  $y(n)$  er signalet for global bevegelse og  $f_{W,LP}(\mathbf{x})$  er lavpassfiltrert vektet bilde. For alle bilderammene i videoen fås hele signalet  $y(n)$ .

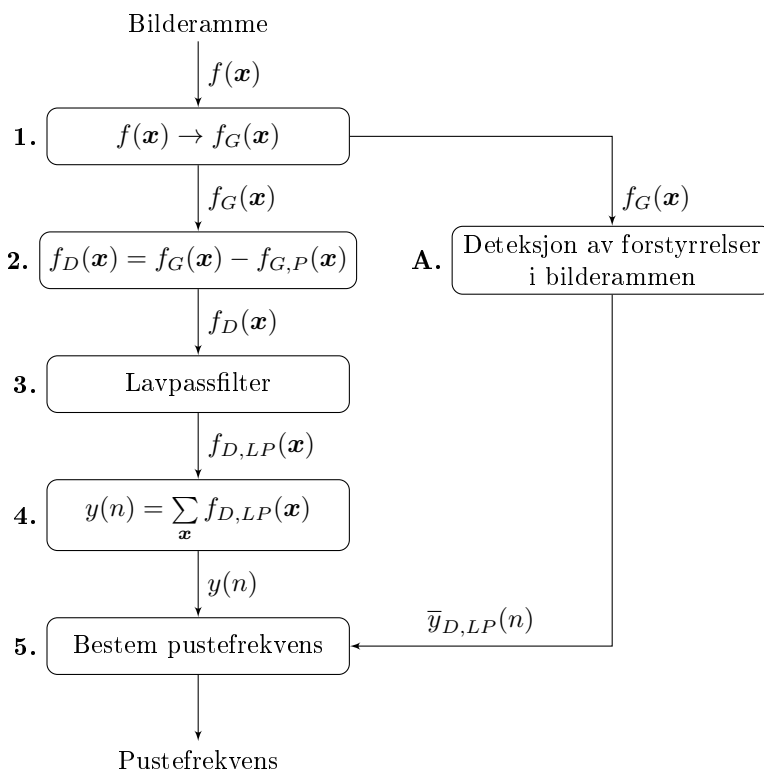
9. Se delkapittel 3.4.

- A. Se delkapittel 3.3.

### 3.2 Differanse-metode

Differanse-metoden er en enklere metode som fase-metoden kan sammenlignes med. Differanse-metoden bruker differansen i intensitet mellom bilderammene i videoen. Denne differansen brukes for å danne et signal som blir brukt for å detektere pustefrekvens. Siden intensitet er utsatt støy er det ventet at denne metoden gir noe dårligere resultat dersom det er små pustebevegelser og dårlige lysforhold. I figur 15 vises flytskjemaet for differanse-metoden.





Figur 15: Flytskjema for differanse-metoden.

Differanse-metoden bruker også et glidende tildsvindu med vindusstørrelse  $T_W$  sekund og oppdateringstid  $T_U$  sekund på samme måte som fase-metoden. Punktene i flytskjemaet for differanse-metoden i figur 15 er forklart nærmere i det følgende:

1. Bilderammen konverteres til et gråtonebilde:

$$f(\mathbf{x}) \rightarrow f_G(\mathbf{x})$$

der  $f(\mathbf{x})$  er nåværende bilderamme,  $\mathbf{x}$  er en vektor med bildekoordinater og  $f_G(\mathbf{x})$  er et gråtonebilde av bilderammen.

2. Differansen i intensitet mellom nåværende og forrige bilderamme finnes:

$$f_D(\mathbf{x}) = f_G(\mathbf{x}) - f_{G,P}(\mathbf{x})$$

der  $f_G(\mathbf{x})$  er et gråtonebilde av nåværende bilderamme og  $f_{G,P}(\mathbf{x})$  er et gråtonebilde av forrige bilderamme.

3. Differansebildet lavpassfiltreres i tid etter differanseligningen:

$$y(n) = \alpha \cdot x(n) + (1 - \alpha) \cdot y(n - 1)$$

Der  $y(n)$  er filtrert utgang,  $\alpha$  er en veiefaktor i området  $[0, 1]$ ,  $x(n)$  er ufiltrert inngang og  $y(n - 1)$  er filtrert utgang ved forrige tidsskritt. Veiefaktoren  $\alpha$  er satt av en bruker. Dette gir et lavpassfiltrert differansebilde  $f_{D,LP}(\mathbf{x})$ .

4. Det summeres over alle pikslene i det lavpassfiltrerte differansebildet:

$$y(n) = \sum_{\mathbf{x}} f_{D,LP}(\mathbf{x})$$

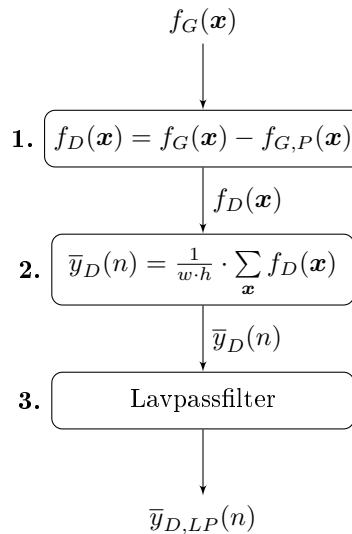
Dette gir ett punkt i signalet  $y(n)$  som blir brukt for å bestemme pustefrekvens. For alle bilderammene dannes hele signalet  $y(n)$ .

5. Se delkapittel 3.4.

- A. Se delkapittel 3.3.

### 3.3 Deteksjon av forstyrrelser i biderammen

For å detektere om noen er i veien for kameraet brukes differansen mellom påfølgende bilderammer. Gjennomsnittlig endring i intensitet finnes og lavpassfiltreres for å fjerne noe av støyen. Dersom gjennomsnittlig endring i intensitet overskrider en terskelverdi  $D_{TH}$  detekteres det en forstyrrelse i bilderammen. Terskelverdien  $D_{TH}$  er satt av en bruker. I figur 16 kan flytskjema av implementeringen sees.



Figur 16: Flytskjema for deteksjon av forstyrrelser i bilderammen.

Punktene i flytskjemaet i figur 16 forklares nærmere i det følgende:

1. Fra gråtonebilde av bilderammene finnes differansen i intensitet mellom nåværende og forrige bilderamme:

$$f_D(\mathbf{x}) = f_G(\mathbf{x}) - f_{G,P}(\mathbf{x})$$

der  $\mathbf{x}$  er en vektor med bildekoordinater,  $f_G(\mathbf{x})$  er et gråtonebilde av nåværende bilderamme og  $f_{G,P}(\mathbf{x})$  er et gråtonebilde av forrige bilderamme.

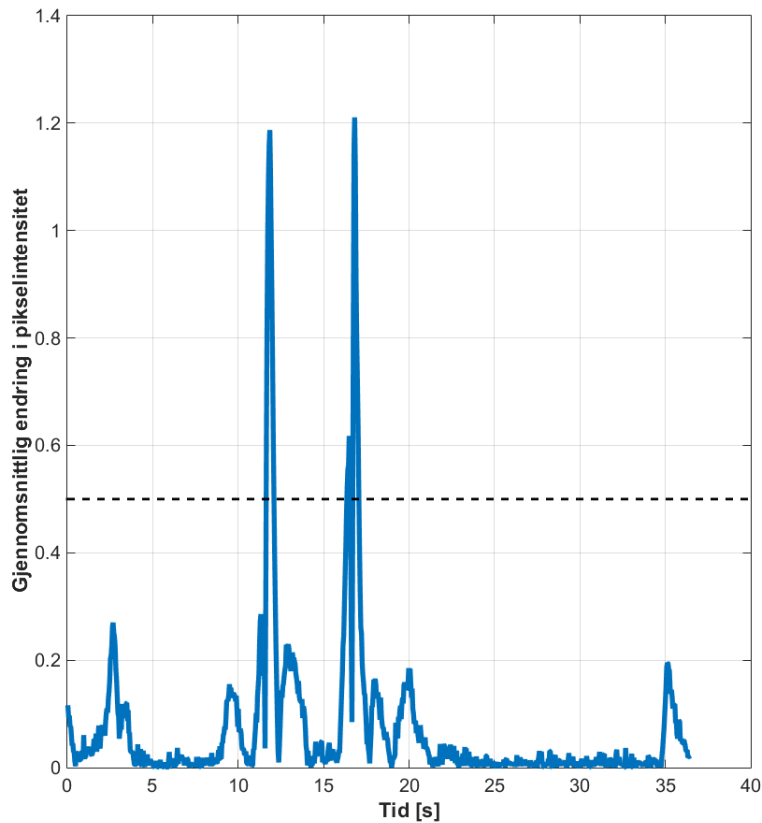
2. Fra differansebildet finnes gjennomsnittlig endring i intensitet mellom påfølgende bilderammer:

$$\bar{y}_D(n) = \frac{1}{w \cdot h} \cdot \sum_{\mathbf{x}} f_D(\mathbf{x})$$

der  $\bar{y}_D(n)$  er gjennomsnittlig endring i pikselintensitet for tidsskritt  $n$ ,  $w$  er bredden til bilderammene,  $h$  er høyden til bilderammene og  $f_D(\mathbf{x})$  er differansebildet.

3.  $\bar{y}_D(n)$  lavpassfiltreres for å fjerne noe av støyen i signalet. Dersom absoluttverdien av det lavpassfiltrerte signalet  $\bar{y}_{D,LP}(n)$  overskrider terskelverdien  $D_{TH}$  satt av en bruker, detekteres det en forstyrrelse.

Når endringen mellom bilderammene blir stor kan dette tyde på at det er en forstyrrelse i bilderammen. Da vil lavpassfiltrert gjennomsnittlig endring i pikselintensitet  $\bar{y}_{D,LP}(n)$  anta en stor absoluttverdi. For et glidende tidsvindu med størrelse  $T_W$  sekund som flyttes  $T_U$  sekund om gangen blir det detektert en ugyldig måling dersom  $|\bar{y}_{D,LP}(n)|$  ligger over terskelverdien  $D_{TH}$  i vinduet. Vindusstørrelsen  $T_W$ , oppdateringstiden  $T_U$  og terskelverdien  $D_{TH}$  er satt av en bruker. Når en måling blir detektert som ugyldig kan det ikke bestemmes en pustefrekvens siden det er detektert for mye bevegelse som ikke kommer av åndedrag.



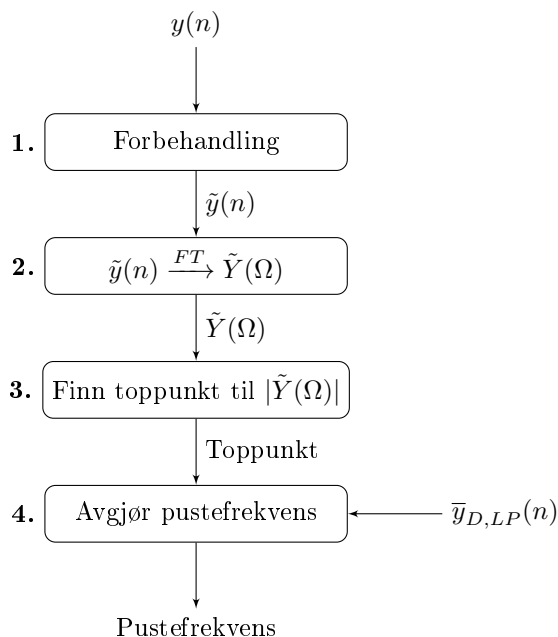
Figur 17: Blå kurve viser absoluttverdien av lavpassfiltrert gjennomsnittlig endring i pikselintensitet  $|\bar{y}_{D,LP}(n)|$ . Stiplet linje viser terskelverdien  $D_{TH}$ .

I figur 17 vises absoluttverdien av signalet  $\bar{y}_{D,LP}(n)$  for en video av en baby som sover. Fra 11 – 17s inn i videoen går det noen i veien for kameraet. For et glidende tidsvindu blir det detektert en ugyldig måling dersom  $|\bar{y}_{D,LP}(n)|$  ligger over terskelverdien  $D_{TH}$ , indikert med stiplet linje, i vinduet. For eksempel med

en vindusstørrelse på 5s ville det bli detektert en ugyldig måling for vinduet 10 – 15s, men ikke for vinduet 20 – 25s i figur 17. Det kan merkes at dersom noen går foranfor kameraet og blir stående helt stille, fortsetter deteksjon av pustefrekvens etter litt tid selv om babyen ikke kan sees. Derfor er det viktig at terskelverdien  $D_{TH}$  er satt liten nok til å detektere forstyrrelser, men stor nok til at pustebevegelser ikke detekteres som forstyrrelser.

### 3.4 Bestemming av pustefrekvens

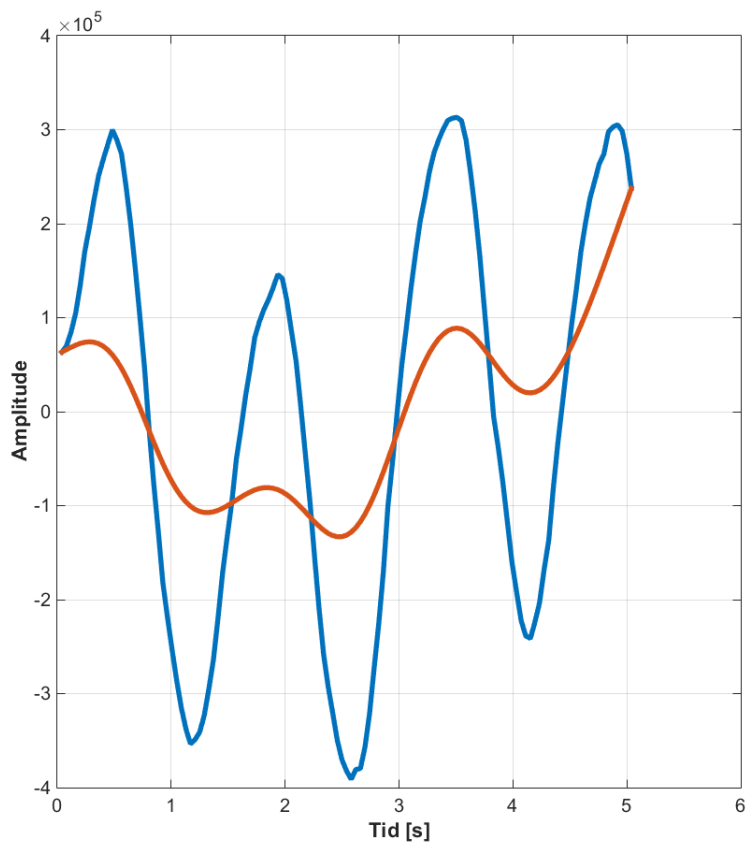
For å bestemme pustefrekvens brukes signalet for bevegelse i videoen  $y(n)$  fra enten fase- eller differanse-metoden. Det blir brukt et glidende tidsvindu med størrelse  $T_W$  sekund som flyttes  $T_U$  sekund om gangen. En bruker spesifiserer vindusstørrelsen  $T_W$  og oppdateringstiden  $T_U$ . Fra signalet for bevegelse i videoen plukkes det ut et vindu fra nåværende tidsskritt til  $T_W$  sekunder før. Dette vinduet oppdateres etter  $T_U$  sekund. Dermed finnes pustefrekvensen fra en del av signalet om gangen. Dette del-signalet går gjennom en forbehandling før det transformeres til frekvensdomenet. I frekvensdomenet finnes de frekvensene som i hovedsak utgjør del-signalet. Med disse frekvens-komponentene kan en pustefrekvens bestemmes. I tillegg brukes deteksjon av forstyrrelser i bilderammen beskrevet i delkapittel 3.3 for å avgjøre om pustefrekvensen er forårsaket av åndedrag. Flytskjema kan sees i figur 18.



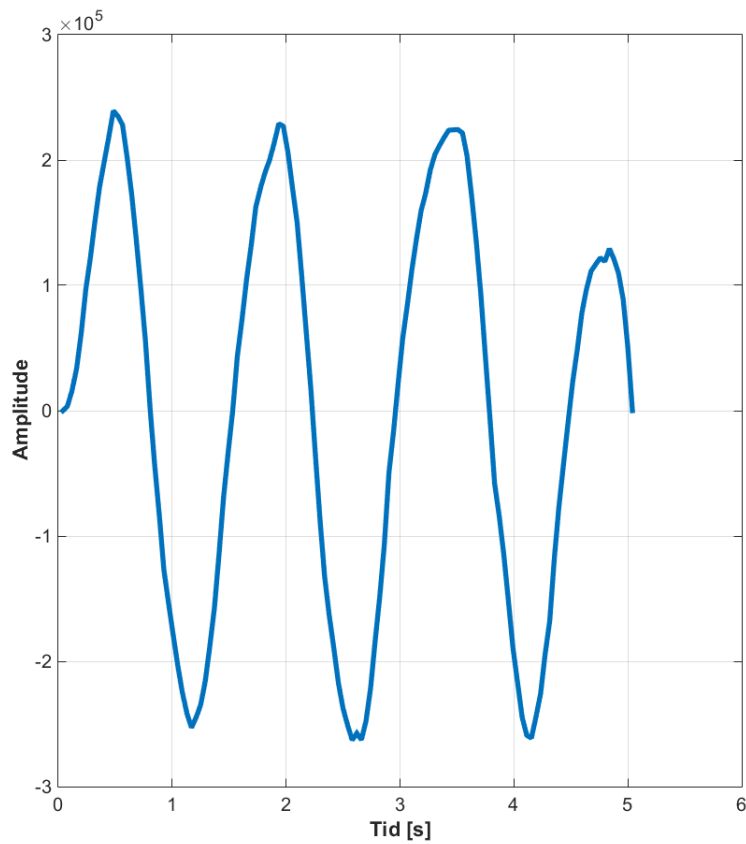
Figur 18: Flytskjema for deteksjon av pustefrekvens.

Videre beskrives punktene i figur 18 nærmere:

1. Signalet for bevegelse i videoen  $y(n)$  brukes for å plukke ut et  $T_W$  sekund langt del-signal. Fra del-signalet trekkes det fra en lavpassfiltrert versjon av del-signalet selv. Dette kan sees i figur 19 og 20.

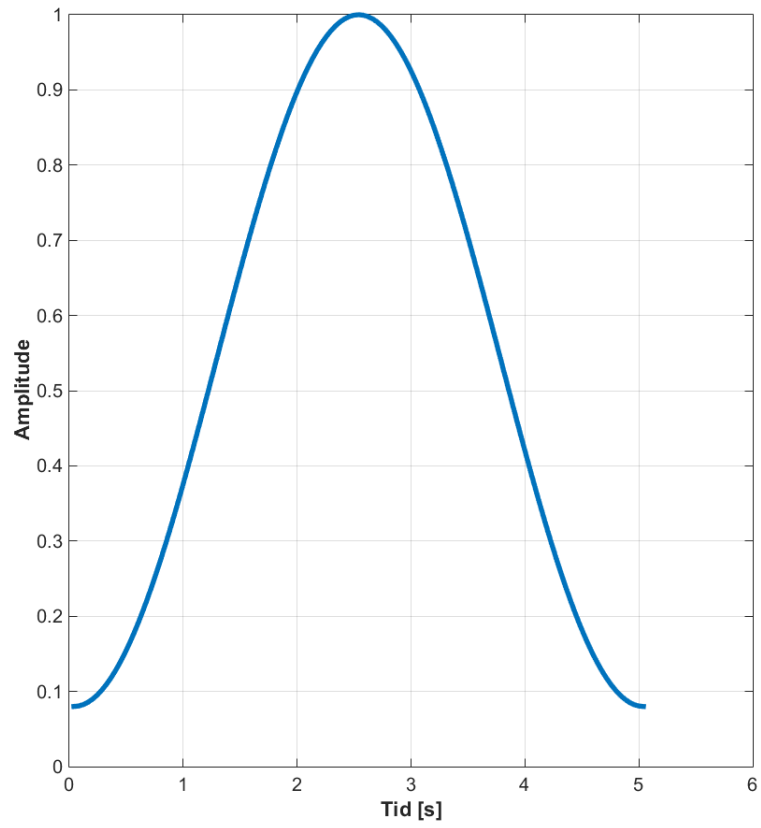


Figur 19: Blå kurve viser 5s av signalet for bevegelse i en video av en sovende baby. Rød kurve viser del-signalet lavpassfiltrert.



Figur 20: Kurven viser resultatet av å trekke en lavpassfiltrert versjon av del-signalet i figur 19 fra del-signalet selv.

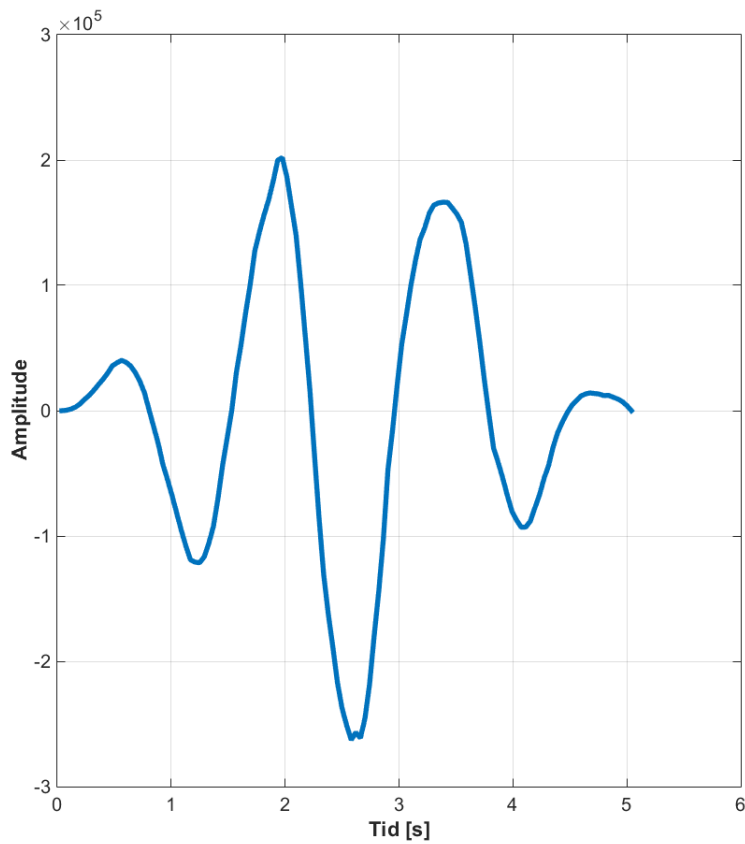
Fra figur 20 kan det sees at det dannes et del-signal som er sentrert rundt null. Videre legges det på et Hamming-vindu med størrelse  $T_W$  sekund. Dette gjør at del-signalet får en rolig overgang til null utenfor vinduet. Et Hamming-vindu kan sees i figur 21.



Figur 21: Hamming-vindu med vindusstørrelse 5s.

I figur 22 kan del-signalet fra figur 20 med Hamming-vinduet fra figur 21 sees.





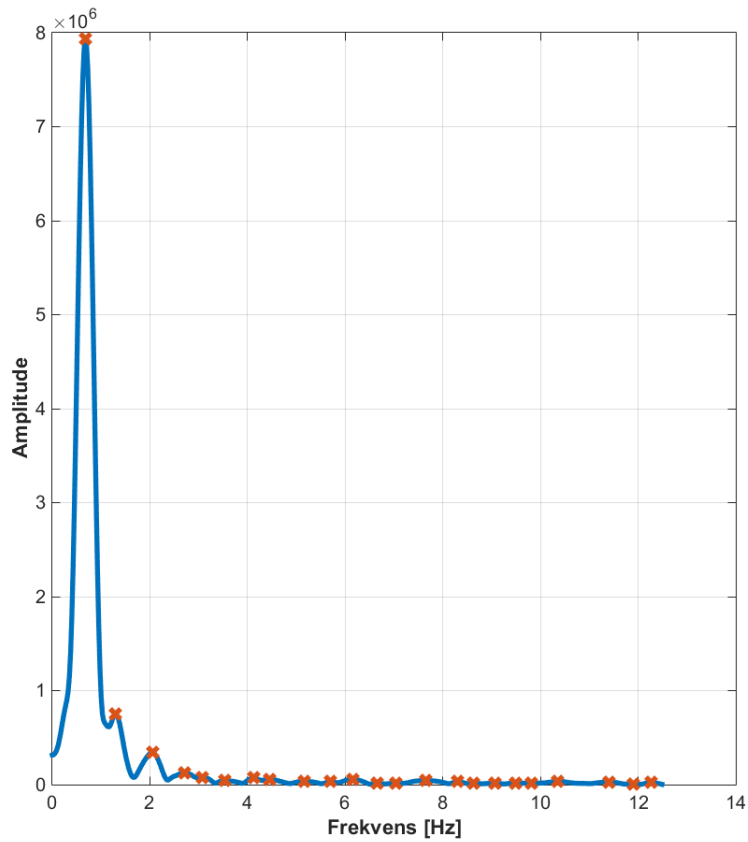
Figur 22: Kurven viser del-signalet fra figur 20 med et Hamming-vindu som er vist i figur 21.

2. Etter forbehandlingen finnes Fouriertransformen til del-signalet:

$$\tilde{y}(n) \xrightarrow{FT} \tilde{Y}(\Omega)$$

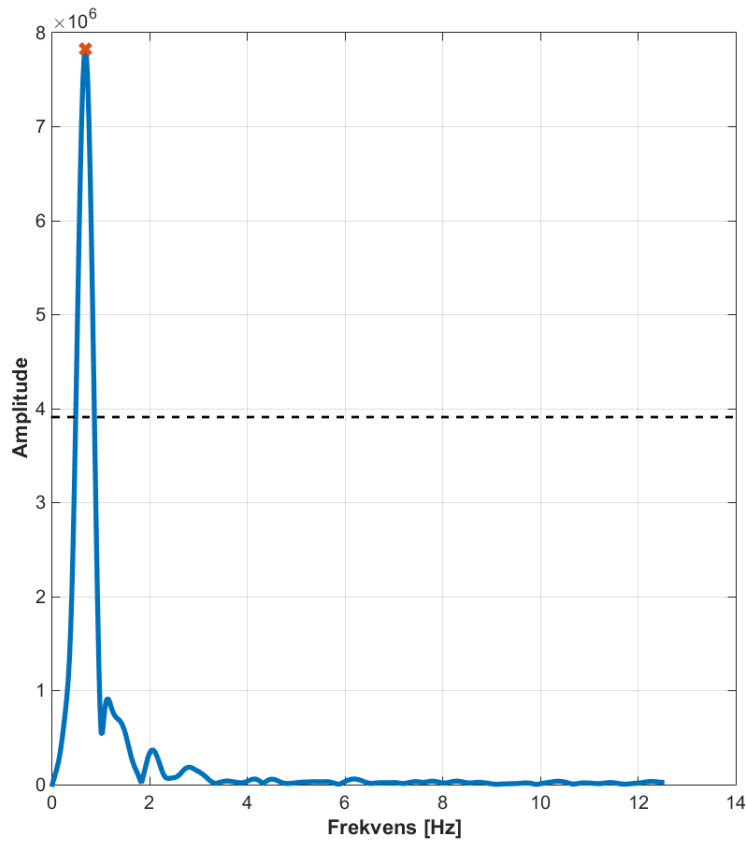
der  $\tilde{y}(n)$  er forbehandlet del-signal,  $\tilde{Y}(\Omega)$  er Fouriertransformen til forbehandlet del-signal og  $\Omega$  er diskret frekvens.

3. Videre finnes toppunktene til absoluttverdien av  $\tilde{Y}(\Omega)$ . I figur 23 kan absoluttverdien sees sammen med toppunktene som er funnet.



Figur 23: Blå kurve viser absoluttverdien av Fouriertransformen til del-signalet i figur 22. Røde kryss angir topppunkter.

4. For det første tidsvinduet og del-signalet velges frekvensen ved det største toppunktet i  $|\tilde{Y}(\Omega)|$  som pustefrekvens. Når vinduet oppdateres fås nye del-signal. For disse del-signalene brukes forrige valgte frekvens for å gjøre et smartere valg av pustefrekvens. Først velges kun de toppunktene som ligger over 50% av det største toppunktet i  $|\tilde{Y}(\Omega)|$  som potensielle pustefrekvenser. Av de potensielle pustefrekvensene velges den frekvensen som ligger nærmest forrige valgte pustefrekvens som nåværende pustefrekvens. Dette kan sees i figur 24.



Figur 24: Blå kurve viser absoluttverdien av Fourier transformen til et del-signal. Dette del-signalet er ikke det første og pustefrekvensen avgjøres basert på forrige valgte pustefrekvens. Stiplet linje viser terskelverdien som toppunktene må ligge over for å kunne betraktes som potensielle pustefrekvenser. Røde kryss angir potensielle pustefrekvenser. Siden det her bare er ett toppunkt som ligger over terskelverdien blir frekvensen ved dette toppunktet valgt som pustefrekvens.

Dersom det i figur 24 var flere toppunkt over den stiplede linjen ville det vært flere potensielle pustefrekvenser. Av de potensielle pustefrekvensene ville den frekvensen som var nærmest forrige valgte pustefrekvens bli valgt som nåværende pustefrekvens. Ved deteksjon av en ugyldig måling, som er beskrevet i delkapittel 3.3, blir neste gyldige måling av pustefrekvens bestemt på samme måte som for det første del-signalet.

## 4 Eksperiment

I dette kapittelet blir det forklart hvordan fase- og differanse-metoden sammenlignes. Det redegjøres for framgangsmåten i resultat-kapitlet, kapittel 5, og de målene som er brukt. For å teste hvor gode metodene er brukes seks videoer. Videoene er av to forskjellige friske babyer som sover. For hver video er det valgt tre nivå av segmentering. Disse nivåene går fra ingen segmentering av babyen til nøyaktig segmentering av babyens mage. Dette er gjort for å finne ut om metodene gir bedre resultat ved segmentering av interesseområdet. Det presenteres et godhetsmål for metodene samt mål for hvor godt forstyrrelser i bilderammen detekteres og kjøretid.

### 4.1 Innstillinger

For hver video må noen innstillinger for metodene settes. Disse innstillingene er som følger:

- $T_W$ : Vindusstørrelsen til et glidende tidsvindu.
- $T_U$ : Oppdateringstiden til det glidende tidsvinduet.
- $D_{TH}$ : Terskelverdien for deteksjon av forstyrrelse i video.
- $\alpha$ : Veiefaktor som bestemmer båndbredden til lavpassfilteret som brukes.

Innstillingene er forklart nærmere i kapittel 3 og er like for fase- og differanse-metoden.

### 4.2 Segmentering

For å se om fase- og differanse-metoden gir et bedre resultat når det på forhånd er gjort en segmentering av babyen velges det tre nivå av segmentering. Disse nivåene er ingen, grov og nøyaktig segmentering. For ingen segmentering brukes hele bilderammen. For grov segmentering velges et rektangel rundt babyen og litt av underlaget. For nøyaktig segmentering velges et rektangel rundt magen til babyen. For alle videoene kjøres fase- og differanse-metoden med alle nivåene av segmentering. Segmenteringen er gjort manuelt og representerer en førprosessering av bilderammene.

### 4.3 Gjennomsnittlig relativ feil

Det er på forhånd gjort en sannhetsmerking av pustefrekvensen for alle videoene. Sannhetsmerkingen er gjort ved å gå gjennom alle bilderammene og merke topp- og bunnpunkt for åndedrag. For å kunne sammenligne med fase- og differanse-metoden regnes det ut en ekte pustefrekvens ved å telle topp- og bunnpunkt i et glidende tidsvindu med størrelse  $T_W$  sekund og oppdateringstid  $T_U$  sekund. Det kan merkes at siden topp- og bunnpunktene vil falle enten innenfor eller utenfor det glidende vinduet har den ekte pustefrekvensen en tendens til å hoppe

mellom faste verdier. Som et godhetsmål for pustefrekvensen funnet med fase- eller differanse-metoden regnes det ut en relativ feil for hver måling. Relativ feil regnes ut som:

$$\text{Relativ feil} = \frac{|\text{Målt verdi} - \text{Ekte verdi}|}{\text{Ekte verdi}} \quad (29)$$

For hele videoen regnes det ut en gjennomsnittlig relativ feil fra alle målingene. Dette gir et mål på hvor gode metodene er.

#### 4.4 Andel gyldige målinger

For noen av videoene går det en person fremfor kameraet og det detekteres ugyldige målinger. En ugyldig måling er en målinger der pustefrekvensen ikke kan bestemmes siden babyen ikke er synlig eller at det er for mye bevegelse i videoen som ikke kommer av åndedrag. I sannhetsmerkingen er disse bilderammene merket og i utregningen av ekte pustefrekvens settes en måling som ugyldig dersom det glidende vinduet inneholder en eller flere bilderammer der det er noen i veien for kameraet. For fase- og differanse-metoden måles det da andel gyldige målinger som vil si den andelen gyldige målinger som finnes med fase- eller differanse metoden i forhold til ekte pustefrekvens. For 100% andel gyldige målinger finner fase- eller differanse-metoden en gyldig måling for alle gyldige målinger i ekte pustefrekvens. For andel gyldige målinger mindre enn dette finner fase- eller differanse-metoden en eller flere målinger som ugyldige der ekte pustefrekvens finner gyldige. Ugyldige målinger blir ikke tatt med i utregningen av gjennomsnittlig relativ feil siden de representerer en manglende måling.

#### 4.5 Kjøretid

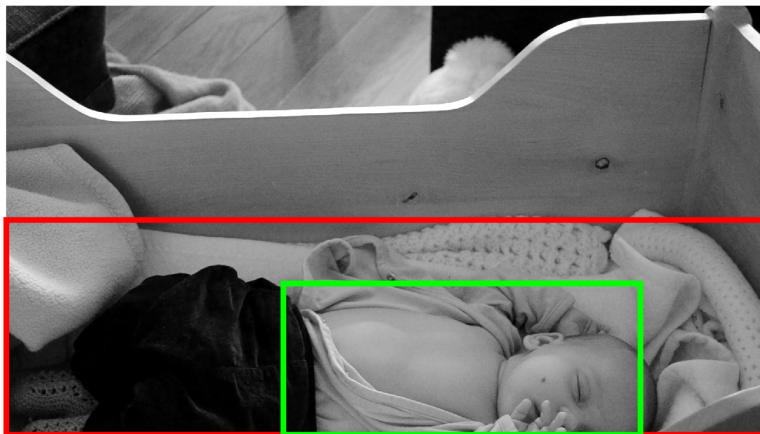
For begge metodene måles også kjøretiden. Siden metodene kjøres i *Matlab* og ikke som ferdig compilert kode gir ikke dette en meningsfull indikasjon på om metodene kan kjøre i sanntid eller ikke. I stedet vil forskjellen i kjøretid mellom metodene gi en indikasjon på hvor mye mer regnekraft fase-metoden trenger.

## 5 Resultat

I dette kapitlet presenteres resultatet for fase- og differanse-metoden. For hver video presenteres litt informasjon om videoen, nivåene av segmentering, innstillinger for fase- og differanse-metoden og resultatet for videoen. Til slutt i kapitlet vises et samlet resultat for alle videoene der fase- og differanse-metoden sammenlignes.

### 5.1 Video 1

Denne videoen viser en frisk baby med åpen skjorte som ligger i en krybbe. Babyens pustebevegelser kan tydelig sees, men beveger seg ikke mye ellers. Videoen er tatt opp med uprofesjonelt opptaksutstyr og det er mye støy i videoen. Kameraet justerer seg automatisk etter lysforholdene. I figur 25 kan en bilde-ramme fra videoen og nivåene av segmentering sees.



Figur 25: Rødt rektangel viser grov segmentering og grønt rektangel viser nøyaktig segmentering.

Teknisk informasjon om videoen:

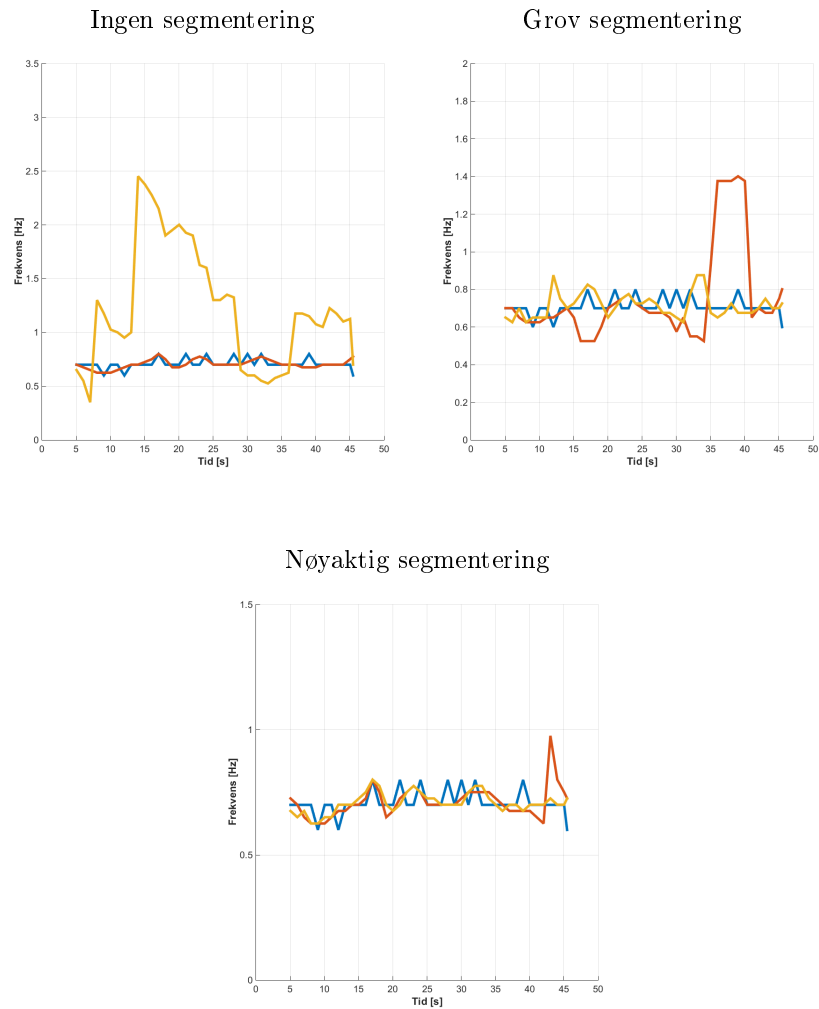
Filnavn	Varighet	Bildefrekvens	Oppløsning
DSC_0900.MOV	45.48s	25Hz	1920 × 1080

Innstillingene for fase- og differanse-metoden velges som følger:

- Vindusstørrelse  $T_W = 5.0s$
- Oppdateringstid  $T_U = 1.0s$
- Terskelverdi  $D_{TH} = 1.0$

- Veiefaktor  $\alpha = 0.1$

I figur 26 kan resultatet for fase- og differanse-metoden sees.



Figur 26: Resultat for video 1. Resultatet er delt opp i egne figurer for hvert nivå av segmentering. For alle figurene viser blå kurve ekte frekvens, rød kurve viser frekvens funnet med fase-metoden og gul kurve viser frekvens funnet med differanse-metoden.

Totalt resultat for fase- og differanse-metoden er gitt i tabell 1.

<b>Fase-metode</b>			
<b>Segmentering</b>	<b>Gjennomsnittlig relativ feil</b>	<b>Andel gyldige målinger</b>	<b>Kjøretid</b>
Ingen	5.26%	100%	448s
Grov	21.07%	100%	272s
Nøyaktig	7.00%	100%	145s
<b>Differanse-metode</b>			
<b>Segmentering</b>	<b>Gjennomsnittlig relativ feil</b>	<b>Andel gyldige målinger</b>	<b>Kjøretid</b>
Ingen	82.26%	100%	117s
Grov	8.85%	100%	99s
Nøyaktig	5.32%	100%	85s

Tabell 1: Totalt resultat for video 1.

For fase-metoden kan det sees at gjennomsnittlig relativ feil øker betydelig med grov segmentering. En grunn til dette kan være at kameraet justerer seg automatisk etter lysforholdet. Video 2 og 3 er svært like video 1, men i disse videoene er den automatiske justeringen slått av og resultatet bedres betydelig. Det kan merkes at siden pustefrekvensen bestemmes delvis basert på hva den var ved forrige måling må en pustefrekvens som avviker mye fra forrige måling framkomme svært tydelig for at det registreres en endring i pustefrekvens. Dette kan sees fra resultatet for fase-metoden med grov segmentering oppe til høyre i figur 26. For den røde kurven kan det sees at pustefrekvensen blir liggende rundt 1.4Hz fra 35 – 40s før den plutselig detter ned til rundt 0.7Hz. Dette tyder på at riktig pustefrekvens er funnet, men likevel registreres feil pustefrekvens siden forrige måling var feil.

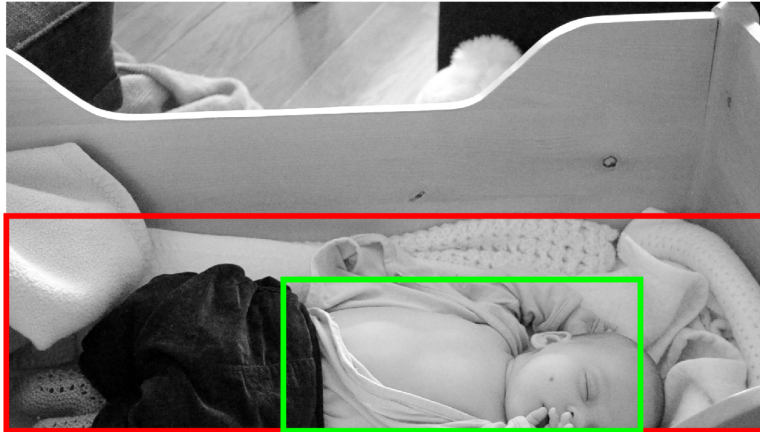
Resultatet for differanse-metoden blir bedre med segmentering av interesseområdet. Det kan sees at det trengs nøyaktig segmentering for å få et resultat i samme område som fase-metoden uten segmentering.

I denne videoen er det ingen forstyrrelser og metodene får 100% andel gyldige målinger. For kjøretiden kan det sees at fase-metoden bruker omtrent fire ganger så lang tid som differanse-metoden for ingen segmentering, men bare litt under dobbelt så lang tid for nøyaktig segmentering. Det er ventet at med ferdig kompilert kode vil begge metodene kunne kjøre i sanntid, muligens med restriksjoner på oppløsning og bilderate.

## 5.2 Video 2

Denne videoen er av en frisk baby med åpen skjorte som ligger i en krybbe. Pustebevegelsen kan tydelig sees i videoen, men ellers er det lite bevegelse i babyen. Videoen er tatt opp med uprofesjonelt opptaksutstyr og det er mye støy i videoen, men lysforholdene er stabile. I figur 27 kan det sees en bilderamme fra videoen og hvordan segmentering er valgt.





Figur 27: Rødt rektangel viser grov segmentering og grønt rektangel viser nøyaktig segmentering.

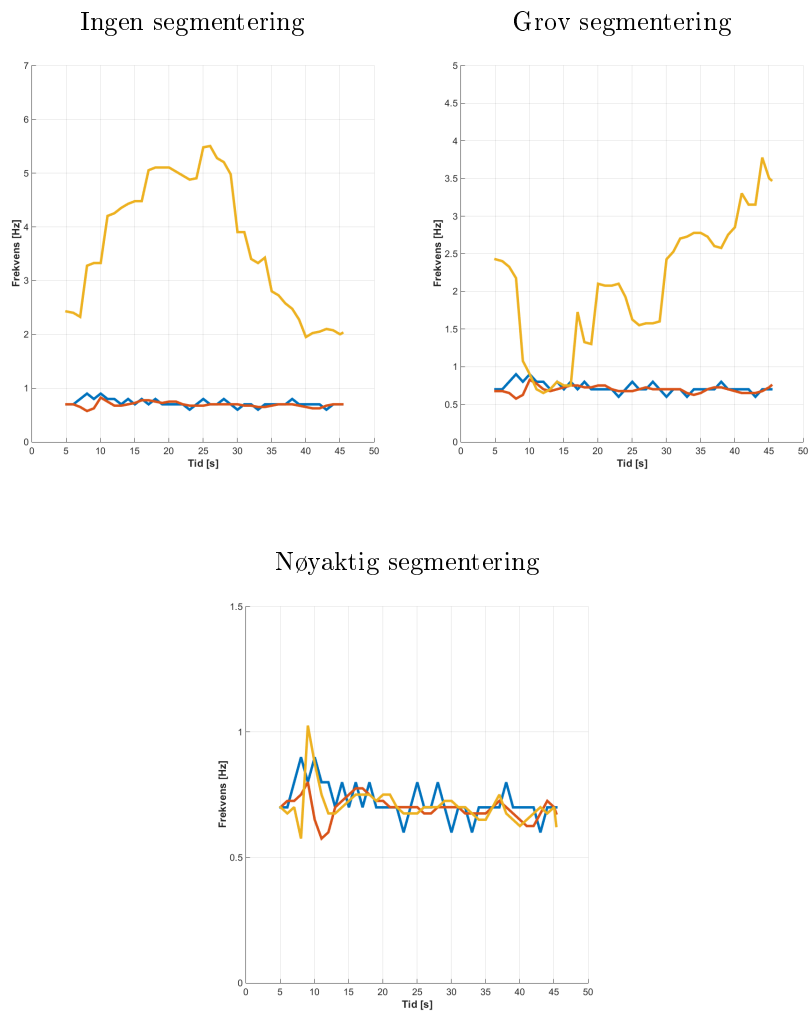
Informasjon om videoen:

<b>Filnavn</b>	<b>Varighet</b>	<b>Bildefrekvens</b>	<b>Oppløsning</b>
DSC_0901.MOV	45.36s	25Hz	1920 × 1080

Innstillinger for fase- og differanse-metoden:

- Vindusstørrelse  $T_W = 5.0s$
- Oppdateringstid  $T_U = 1.0s$
- Terskelverdi  $D_{TH} = 1.0$
- Veiefaktor  $\alpha = 0.1$

I figur 28 kan resultatet for fase- og differanse-metoden sees.



Figur 28: Resultat for video 2. Resultatet er delt opp i egne figurer for hvert nivå av segmentering. For alle figurene viser blå kurve ekte frekvens, rød kurve viser frekvens funnet med fase-metoden og gul kurve viser frekvens funnet med differanse-metoden.

Totalt resultat er gitt i tabell 2.

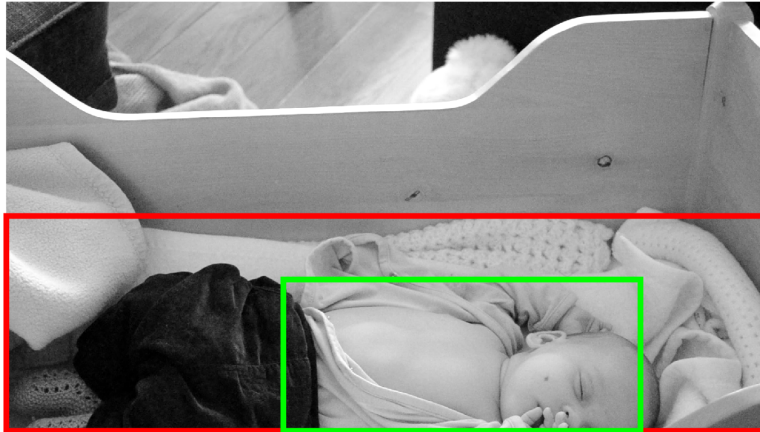
<b>Fase-metode</b>			
<b>Segmentering</b>	<b>Gjennomsnittlig relativ feil</b>	<b>Andel gyldige målinger</b>	<b>Kjøretid</b>
Ingen	7.35%	100%	454s
Grov	7.43%	100%	271s
Nøyaktig	7.41%	100%	147s
<b>Differanse-metode</b>			
Ingen	412.57%	100%	109s
Grov	199.18%	100%	93s
Nøyaktig	8.09%	100%	83s

Tabell 2: Totalt resultat for video 2.

For fase-metoden kan det sees at segmentering ikke gir et bedre resultat. Resultatet for differanse-metoden er svært avhengig av segmentering. Det kan også sees at differanse-metoden trenger en nøyaktig segmentering av babyen for å få et resultat i samme område som fase-metoden uten segmentering. Begge metodene får 100% andel gyldige målinger da det ikke er forstyrrelser i videoen. Det kan sees at kjøretiden for fase-metoden er omtrent fire ganger lengre enn differanse-metoden uten segmentering og litt under dobbelt så lang med nøyaktig segmentering.

### 5.3 Video 3

Denne videoen er av en frisk baby som ligger i en krybbe med åpen skjorte. Babyens pustebevegelser kan tydelig sees, men beveger seg ikke mye ellers. Videoen er tatt opp med uprofesjonelt kamera og har en del støy, men lysforholdene er stabile. Fra 11 – 17s inn i videoen går det en person framfor kameraet. I figur 29 kan det sees en bilderamme fra videoen og hvordan segmentering er valgt.



Figur 29: Rødt rektangel viser grov segmentering og grønt rektangel viser nøyaktig segmentering.

Teknisk informasjon om videoen:

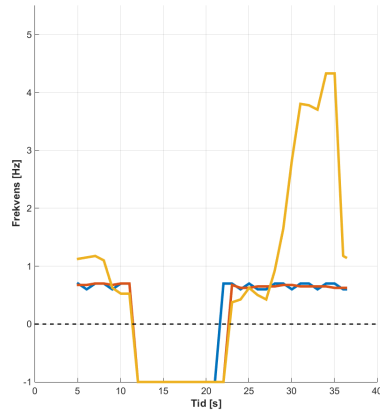
Filnavn	Varighet	Bildefrekvens	Oppløsning
DSC_0902.MOV	36.36s	25Hz	1920 × 1080

For fase- og differanse-metoden er følgende innstillinger valg:

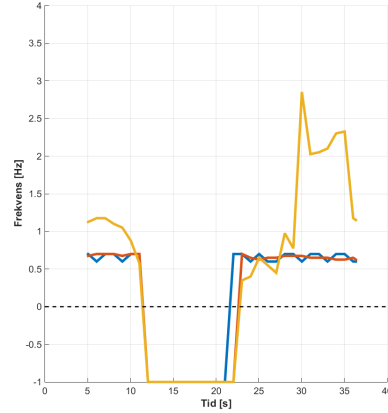
- Vindusstørrelse  $T_W = 5.0s$
- Oppdateringstid  $T_U = 1.0s$
- Terskelverdi  $D_{TH} = 0.5$
- Veiefaktor  $\alpha = 0.1$

I figur 30 vises resultatet for de to metodene.

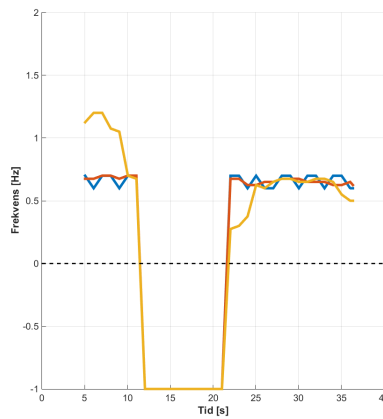
Ingen segmentering



Grov segmentering



Nøyaktig segmentering



Figur 30: Resultat for video 3. Resultatet er delt opp i egne figurer for hvert nivå av segmentering. For alle figurene viser blå kurve ekte frekvens, rød kurve viser frekvens funnet med fase-metoden og gul kurve viser frekvens funnet med differanse-metoden. Stiplet linje indikerer 0Hz. Alle målinger under denne linjen anses som ugyldige.

I tabell 3 kan totalt resultat for fase- og differanse-metoden sees.

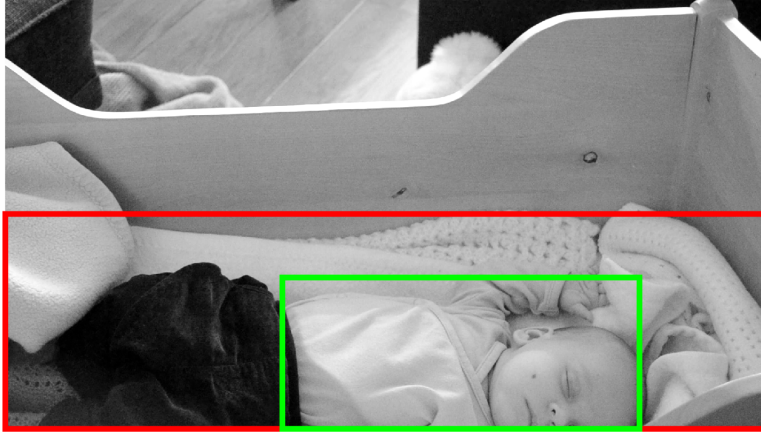
<b>Fase-metode</b>			
<b>Segmentering</b>	<b>Gjennomsnittlig relativ feil</b>	<b>Andel gyldige målinger</b>	<b>Kjøretid</b>
Ingen	6.19%	95.65%	360s
Grov	6.61%	95.65%	221s
Nøyaktig	6.26%	100%	117s
<b>Differanse-metode</b>			
Ingen	164.95%	95.65%	85s
Grov	101.45%	95.65%	74s
Nøyaktig	27.81%	100%	66s

Tabell 3: Totalt resultat for video 3.

Igjen kan det sees at segmentering ikke gir et bedre resultat for fase-metoden. Selv med en nøyaktig segmentering av babyen er ikke differanse-metoden i stand til å gi et like godt resultat som fase-metoden uten segmentering. Det kan sees at resultatet for differanse-metoden er svært avhengig av segmentering av interesse-området. For ingen og grov segmentering detekterer begge metodene en ugyldig måling der ekte frekvens fant en gyldig måling og dermed er ikke andel gyldige målinger 100%. Kjøretiden for fase metoden er i området 2 – 4 ganger lengre enn for differanse-metoden, avhengig av segmentering.

## 5.4 Video 4

I denne videoen vises en frisk baby med igjenkneppet skjorte. Det er svært lite bevegelse i babyen og det er vanskelig å se pustebevegelsen. Videoen er tatt opp med uprofesjonelt opptaksutstyr og det er en del støy i videoen, men lysforholdene er stabile. Det kan sees at noen beveger seg bak kameraet fra skyggene i videoen. Fra 21 – 26s inn i videoen går det noen framfor kameraet. I figur 31 kan det sees en bilderamme fra videoen og hvordan segmentering av babyen er valgt.



Figur 31: Rødt rektangel viser grov segmentering og grønt rektangel viser nøyaktig segmentering.

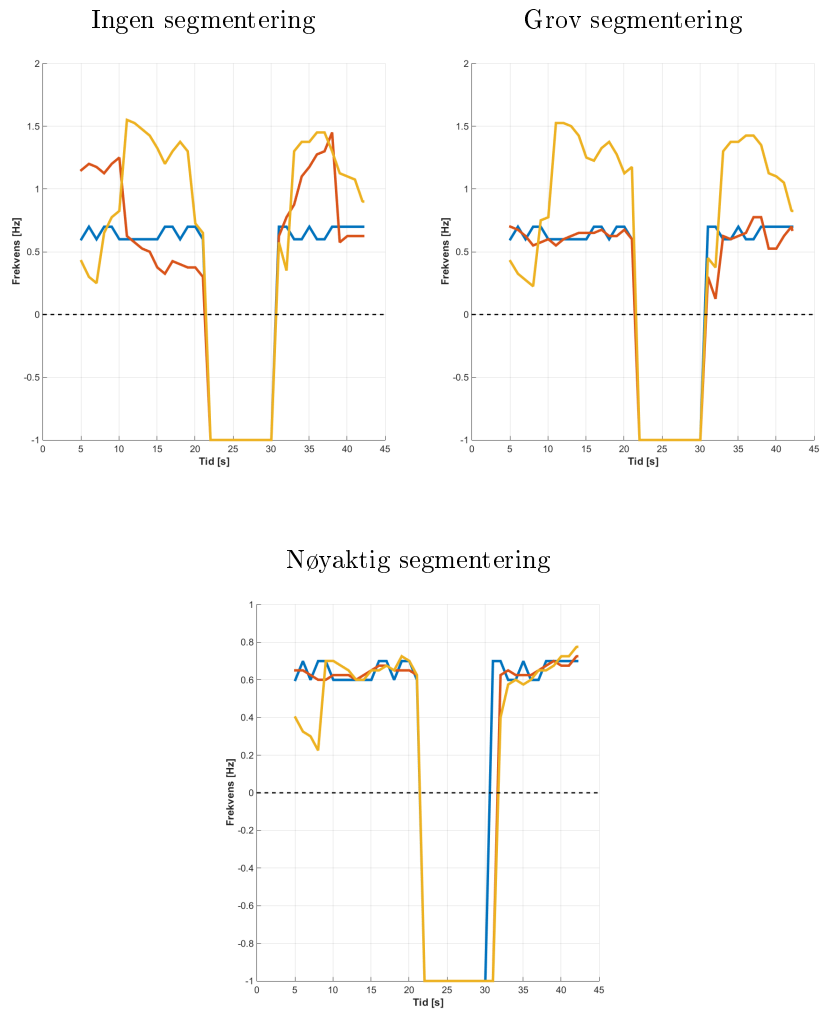
Teknisk informasjon om videoen:

<b>Filnavn</b>	<b>Varighet</b>	<b>Bildefrekvens</b>	<b>Oppløsning</b>
DSC_0904.MOV	42.12s	25Hz	1920 × 1080

Innstillingene for fase- og differanse-metode er valgt som følger:

- Vindusstørrelse  $T_W = 5.0s$
- Oppdateringstid  $T_U = 1.0s$
- Terskelverdi  $D_{TH} = 0.5$
- Veiefaktor  $\alpha = 0.1$

I figur 32 kan resultatet for fase- og differanse-metoden sees.



Figur 32: Resultat for video 4. Resultatet er delt opp i egne figurer for hvert nivå av segmentering. For alle figurene viser blå kurve ekte frekvens, rød kurve viser frekvens funnet med fase-metoden og gul kurve viser frekvens funnet med differanse-metoden. Stiplet linje indikerer 0Hz. Alle målinger under denne linjen anses som ugyldige.

I tabell 4 kan totalt resultat for fase- og differanse-metoden sees.



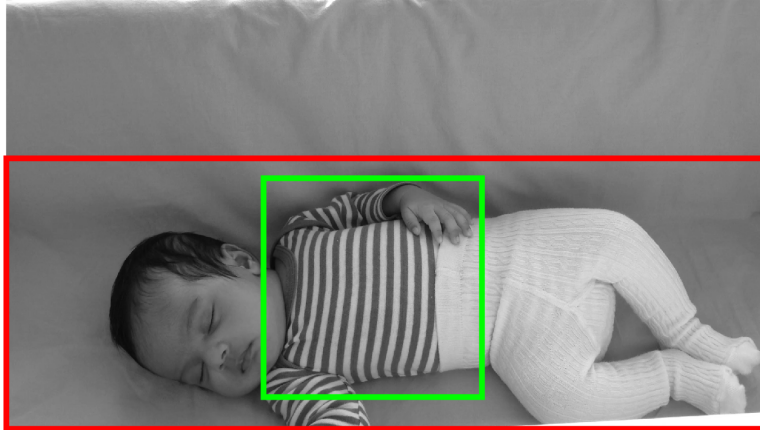
<b>Fase-metode</b>			
<b>Segmentering</b>	<b>Gjennomsnittlig relativ feil</b>	<b>Andel gyldige målinger</b>	<b>Kjøretid</b>
Ingen	48.67%	100%	424s
Grov	12.97%	100%	250s
Nøyaktig	5.93%	96.67%	133s
<b>Differanse-metode</b>			
Ingen	77.08%	100%	98s
Grov	82.58%	100%	85s
Nøyaktig	13.48%	96.67%	74s

Tabell 4: Totalt resultat for video 4.

For denne videoen er hverken fase- eller differanse-metoden i stand til å gi et godt resultat uten segmentering. Siden det er svært liten pustebevegelse i videoen er fase-metoden avhengig av segmentering av babyen. For grov og nøyaktig segmentering bedres resultatet betraktelig for fase-metoden. Differanse-metoden er avhengig av en nøyaktig segmentering av babyen for å gi et resultat i størrelsesorden med fase-metoden med grov segmentering. For nøyaktig segmentering i figur 32 kan det sees at begge metodene detekterer en ugyldig måling der ekte pustefrekvens detekterer en gyldig måling. Derfor er ikke andel gyldige målinger 100% for metodene med nøyaktig segmentering. For kjøretiden kan det sees at fase-metoden bruker betraktelig lengre tid, men at denne forskjellen blir mindre med segmentering. Grunnen til dette er at med segmentering behandles færre piksler.

## 5.5 Video 5

Denne videoen viser en frisk baby som ligger på en sofa. Det er lite bevegelse i babyen, men det er mulig å se pustebevegelser. Videoen er tatt opp med et uprofesjonelt kamera. Det er stabile lysforhold og lite støy i videoen. I figur 33 kan det sees en bilderamme fra videoen og hvordan segmentering av babyen er valgt.



Figur 33: Rødt rektangel viser grov segmentering og grønt rektangel viser nøyaktig segmentering.

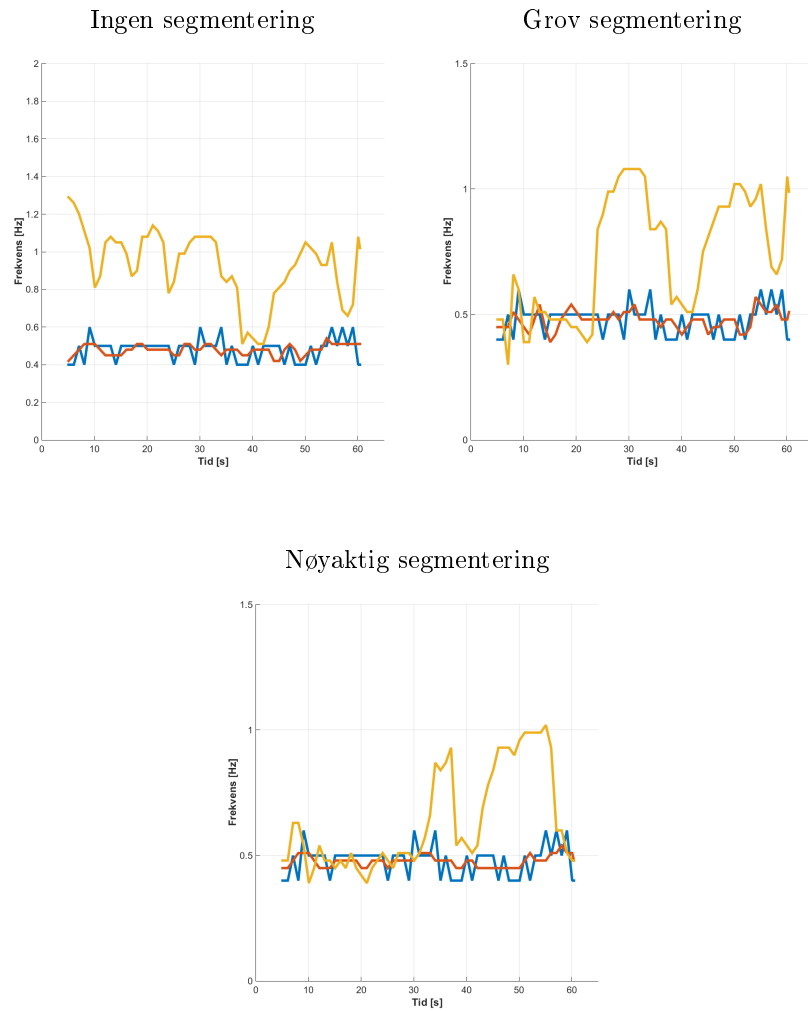
Informasjon om videoen:

Filnavn	Varighet	Bildefrekvens	Oppløsning
IMG_0203.MOV	60.42s	29.97Hz	1920 × 1080

Innstillinger for fase- og differanse-metoden:

- Vindusstørrelse  $T_W = 5.0s$
- Oppdateringstid  $T_U = 1.0s$
- Terskelverdi  $D_{TH} = 1.0$
- Veiefaktor  $\alpha = 0.1$

Resultatet for fase- og differanse-metoden kan sees i figur 34.



Figur 34: Resultat for video 5. Resultatet er delt opp i egne figurer for hvert nivå av segmentering. For alle figurene viser blå kurve ekte frekvens, rød kurve viser frekvens funnet med fase-metoden og gul kurve viser frekvens funnet med differanse-metoden.

I tabell 5 kan totalt resultat for metodene sees.

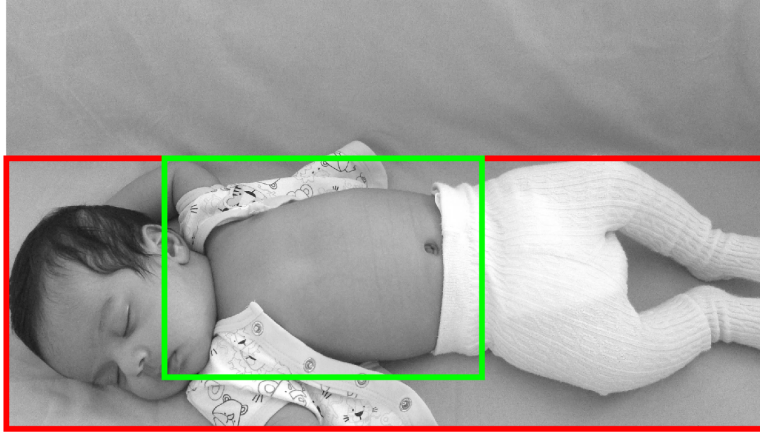
<b>Fase-metode</b>			
<b>Segmentering</b>	<b>Gjennomsnittlig relativ feil</b>	<b>Andel gyldige målinger</b>	<b>Kjøretid</b>
Ingen	10.11%	100%	708s
Grov	11.94%	100%	511s
Nøyaktig	10.39%	100%	215s
<b>Differanse-metode</b>			
Ingen	97.11%	100%	178s
Grov	62.50%	100%	162s
Nøyaktig	41.23%	100%	132s

Tabell 5: Totalt resultat for video 5.

Det kan sees at resultatet for fase-metoden ikke er avhengig av segmentering i denne videoen. Resultatet for differanse-metoden derimot er svært avhengig av segmentering av interesseområdet. Dette gjelder spesielt dersom pustebevegelsene er små. For fase-metoden fås et godt resultat uten segmentering, men differanse-metoden er ikke i stand til å gi et akseptabelt resultat for noen av segmenteringsnivåene. Begge metodene får 100% andel gyldige målinger da det ikke er forstyrrelser i videoen. For denne videoen er bildefrekvensen 29.97Hz og det kan merkes at en raskere bilderate har en betydelig effekt på kjøretiden. Igjen kan det sees at fase-metoden bruker 2 – 4 ganger lengre tid enn differanse-metoden, avhengig av segmentering.

## 5.6 Video 6

Denne videoen er av en frisk baby med åpen skjorte som ligger på en sofa. Babyens pustebevegelser kan tydelig sees, men beveger seg ikke mye ellers. Videoen er tatt opp med uprofesjonelt opptaksutstyr og det er noe støy i videoen. Kameraet justerer lysstyrken automatisk etter lysforholdene. I figur 35 kan det sees en bilderamme fra videoen og hvordan segmentering er valgt.



Figur 35: Rødt rektangel viser grov segmentering og grønt rektangel viser nøyaktig segmentering.

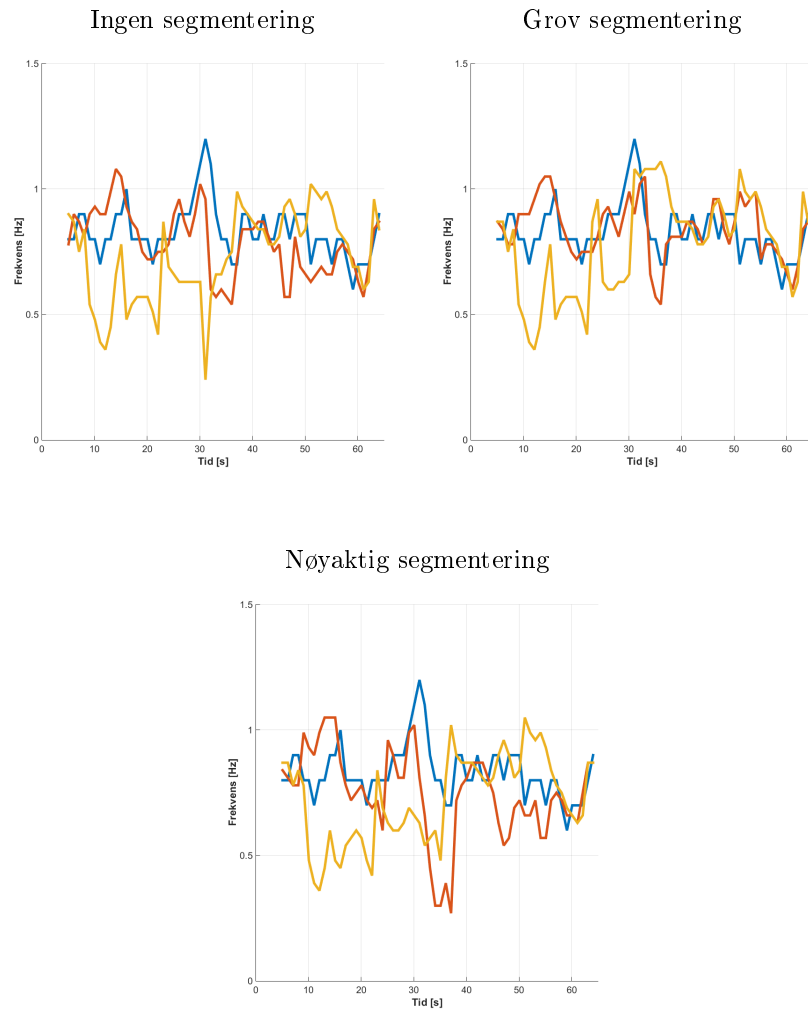
Teknisk informasjon om videoen:

<b>Filnavn</b>	<b>Varighet</b>	<b>Bildefrekvens</b>	<b>Oppløsning</b>
IMG_0209.MOV	64.06s	29.97Hz	1920 × 1080

Innstillinger for fase- og differanse-metoden:

- Vindusstørrelse  $T_W = 5.0s$
- Oppdateringstid  $T_U = 1.0s$
- Terskelverdi  $D_{TH} = 1.0$
- Veiefaktor  $\alpha = 0.1$

I figur 36 kan resultatet for fase- og differanse-metoden sees.



Figur 36: Resultat for video 6. Resultatet er delt opp i egne figurer for hvert nivå av segmentering. For alle figurene viser blå kurve ekte frekvens, rød kurve viser frekvens funnet med fase-metoden og gul kurve viser frekvens funnet med differanse-metoden.

Fra tabell 6 kan totalt resultat for fase- og differanse-metoden sees.

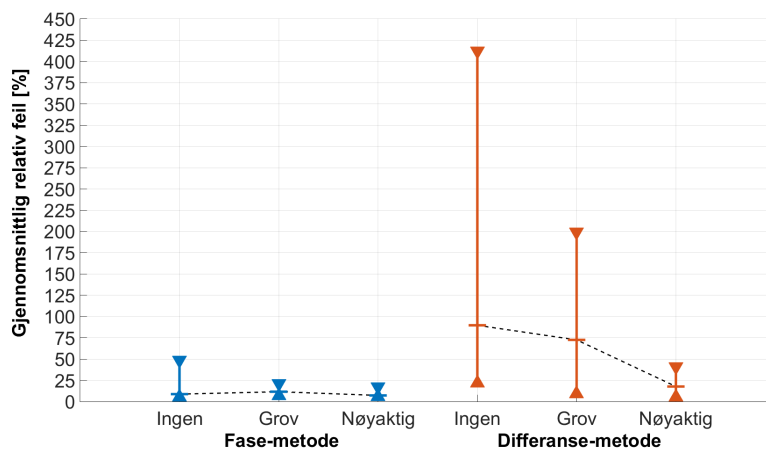
<b>Fase-metode</b>			
<b>Segmentering</b>	<b>Gjennomsnittlig relativ feil</b>	<b>Andel gyldige målinger</b>	<b>Kjøretid</b>
Ingen	12.67%	100%	750s
Grov	11.19%	100%	551s
Nøyaktig	17.29%	100%	274s
<b>Differanse-metode</b>			
Ingen	21.60%	100%	192s
Grov	21.87%	100%	159s
Nøyaktig	21.88%	100%	135s

Tabell 6: Totalt resultat for video 6.

Fase-metoden er i stand til å finne et godt resultat uten segmentering. Med en nøyaktig segmentering fås et noe dårligere resultat. Årsaken kan være at kameraet automatisk justerer lysstyrken, noe som påvirker resultatet. Pustebevegelsen til babyen kan tydelig sees fra videoen og differanse-metoden finner et brukbart resultat for alle nivåene av segmentering, men også her vil den automatiske lysjusteringen gjøre resultatet noe dårligere. Andel gyldige målinger er 100% for begge metodene da det ikke er forstyrrelser i videoen. Kjøretiden viser at fase-metoden bruker lengre tid enn differanse-metoden, men at denne forskjellen minker med segmentering. Dette kommer av at metodene da behandler færre piksler.

## 5.7 Samlet resultat

Det presenteres et samlet resultat for å kunne sammenligne fase- og differanse-metoden for alle videoene. For hver video finnes gjennomsnittlig relativ feil. Medianverdien samt største og minste verdi av gjennomsnittlig relativ feil kan da brukes for å sammenligne metodene. Dette kan sees i figur 37.



Figur 37: Samlet resultat for fase- og differanse-metoden. Blå kurver viser gjennomsnittlig relativ feil for fase-metoden for alle nivåene av segmentering. Røde kurver viser dette for differanse-metoden. Horisontal strek viser medianverdi og trekanten indikerer største og minste verdi.

Samlet resultat er også oppsummert i tabell 7. Der vises medianverdien av gjennomsnittlig relativ feil for fase- og differanse-metode for alle nivåene av segmentering. I parentes vises differansen mellom største og minste verdi av gjennomsnittlig relativ feil.

Segmentering	Fase-metode	Differanse-metode
Ingen	8.73(43.41)%	89.69(390.97)%
Grov	11.57(14.46)%	72.54(190.33)%
Nøyaktig	7.21(11.36)%	17.68(35.91)%

Tabell 7: Medianverdi av gjennomsnittlig relativ feil for alle videoene med differansen mellom største og minste verdi i parentes.

For fase-metoden kan det sees at medianverdien av gjennomsnittlig relativ feil ikke blir mindre med segmentering. Derimot blir differansen mellom største og minste verdi mindre. For differanse-metoden fås et bedre resultat med bedre segmentering av babyen. Det samme gjelder differansen mellom største og minste verdi. Det kan sees at medianverdien for fase-metoden uten segmentering er lavere enn medianverdien til differanse-metoden med nøyaktig segmentering. Det kan også sees at differansen mellom største og minste verdi av gjennomsnittlig relativ feil er omtrent lik for de to tilfellene. Dette betyr at ved bruk av fase-metoden uten segmentering fås et resultat som er bedre enn differanse-metoden med nøyaktig segmentering.



## 6 Diskusjon

Som det kan sees fra resultatet i kapittel 5 er det en trend at fase-metoden gir et bedre resultat enn differanse-metoden. Fase-metoden krever mer regnekraft, men for å få et tilsvarende resultat som fase-metoden trenger differanse-metoden en nøyaktig segmentering av interesseområdet i bilderammen. Denne segmentering krever da mer regnekraft slik at for tilsvarende resultat kan det tenkes at fase- og differanse-metoden trenger tilsvarende regnekraft.

Det er blitt påpekt at i noen av videoene bruker kameraet en automatisk lysjustering. Dette var tilfellet i video 1 og 6. I disse videoene merkes det et betydelig dårligere resultat for begge metodene som kan komme av lysjusteringen. Denne funksjonen bør derfor deaktiveres dersom det er mulig.

Det er også blitt nevnt at siden den ekte målingen av pustefrekvens er regnet ut fra en merking av bilderammer med topp- og bunnpunkt for åndedrag vil den ekte pustefrekvensen variere unaturlig. Ekte pustefrekvens burde derfor blitt målt for eksempel med maske over babyens munn som var synkronisert med videoopptak. Dette ville trolig gitt en bedre sammenligning mellom de to metodene og ekte pustefrekvens.

Fase-metoden finner global bevegelse fra en vekting av bilderammene der vektingen blir gjort med lokal amplitude. Lokal amplitude er et mål for tekststyrke og det trengs derfor en tydelig tekstur i bilderammen som beveger seg. Dette er spesielt viktig for videoopptak der åndedragene er små og ikke kan sees tydelig. Det kan i disse tilfellene være nødvendig med en segmentering av interesseområdet eller annen førprosessering for å framheve pustefrekvensen.

I denne rapporten er bilderammene konvertert til gråtonebilder. Det er mulig at ved bruk av fargebilder fås et bedre resultat, men da kreves mer regnekraft.

Det ble sett at fase-metoden bruker 2 – 4 ganger lengre tid enn differanse-metoden, avhengig av segmentering. Det er klart at fase-metoden krever mer regnekraft, men med ferdig compilert kode er det fortsatt ventet at begge metodene vil kunne kjøre i sanntid.

Det er også viktig å nevne at på grunn av et lite datasett er trening av de to metodene blitt gjort på samme datasett som de testes med. Det burde strengt tatt blitt brukt separate datasett for trening og testing. Derfor bør resultatet for fase-metoden betraktes som en indikasjon på at fase-metoden kan gi gode resultat og at den bør utforskes mer.

## 7 Konklusjon

Med lokal fase og lokal amplitude er det laget en metode som detekterer pustefrekvensen til en baby fra videoopptak på en robust måte. Der det tidligere er blitt brukt en kompleks styrbar pyramide for å finne global bevegelse i en video, er det her blitt laget en metode som bruker en tilnærming til Riesz transformen. Ved å bruke en slik tilnærming krever metoden mindre regnekraft og det er dermed ventet at en effektiv implementering av metoden har mulighet til å kjøre i sanntid. En slik metode er ikke avhengig av en segmentering av interesseområdet såfremt bevegelsen er stor nok til å detekteres og kan skilles fra støy. Metoden kan detektere svært små pustebevegelser under de rette forholdene og er lite sensitiv for støy. Det er ventet at metoden ville gitt en nøyaktig pustefrekvens dersom den ble brukt i et overvåkingssystem.

## Referanser

- [1] Ronald N. Bracewell, *The Fourier Transform and Its Applications (Third Edition)*. Boston, Massachusetts: McGraw-Hill, 2000.
- [2] Julius O. Smith III. (2007). *Mathematics of the Discrete Fourier Transform (DFT) with Audio Applications (Second Edition)* [E-bok]. Hentet fra: <https://ccrma.stanford.edu/~jos/mdft/>
- [3] Alan V. Oppenheim og Jae S. Lim, "The Importance of Phase in Signals", *Proceedings of the IEEE*, vol. 69, nr. 5, s. 529-541, 1981.
- [4] David A. Forsyth og Jean Ponce, *Computer Vision A Modern Approach (Second Edition)*. Harlow: Pearson, 2012.
- [5] Nikolay Skarbnik, Yehoshua Y. Zeevi og Chen Sagiv. (2010, Aug). The Importance of Phase in Image Processing, *CCIT Report #773* [Online]. Hentet fra: <http://webee.technion.ac.il/publication-link/index/id/572>
- [6] Michael Felsberg og Gerald Sommer, "The Monogenic Signal", *IEEE Transactions on Signal Processing*, vol. 49, nr. 12, s. 3136-3144, 2001.
- [7] M. Felsberg og G. Sommer, "The Monogenic Scale-Space: A Unifying Approach to Phase-Based Image Processing in Scale-Space", *Journal of Mathematical Imaging and Vision*, vol. 21, s. 5-26, 2004.
- [8] Neal Wadhwa, Michael Rubinstein, Frédo Durand og William T. Freeman. (2014). Riesz Pyramids for Fast Phase-Based Video Magnification, *2014 IEEE International Conference on Computational Photography (ICCP)* [Online]. Hentet fra: <http://people.csail.mit.edu/nwadhwa/riesz-pyramid/>
- [9] Peter J. Burt og Edward H. Andelson, "The Laplacian Pyramid as a Compact Image Code", *IEEE Transactions on Communications*, vol. COM-31, nr. 4, s. 532-540, 1983.
- [10] Abe Davis, et al., "The Visual Microphone: Passive Recovery of Sound from Video", *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2014*, vol. 33, nr. 4, s. 79:1-79:10, 2014.

## Vedlegg

*Matlab*-kode:

1. `rrFromVid.m`
2. `laplacianPyr.m`

```
% I koden brukes metoden laplacianPyr(...), se laplacianPyr.m
close all;
clear all;

% Navn på video (.MOV fil) og sannehetsmerking (.mat fil)
videoName = 'DSC_0902';
% Vekt for lavpassfilter
alpha = 0.1;
% Vindusstørrelse
timeWindow = 5.0;
% Oppdateringstid
updateTime = 1.0;
% Terskelverdi
yd_th = 0.5;
% Metode (1: Differanse-metode, 2: Fase-metode)
method = 2;

% Filtermaskene for tilnærmet Riesz transform
h1 = [0.5, 0, -0.5];
h2 = [0.5, 0, -0.5]';

V = VideoReader([videoName, '.MOV']);
numFrames = round(V.FrameRate*V.Duration);
y = zeros(1, numFrames-1);
framesInTimeWindow = round(timeWindow*V.FrameRate);
framesInUpdateTime = round(updateTime*V.FrameRate);
fPeaks_num = ceil((numFrames-framesInTimeWindow)/framesInUpdateTime) + 1;
%Nf = ceil(timeWindow/updateTime);
Nf = 1;
t = zeros(1, fPeaks_num);
fPeaks = zeros(1, fPeaks_num);
histEqOn = 0;
%Nh = floor(framesInTimeWindow/3);
Nh = round(framesInTimeWindow/5);
%Nh = round(framesInTimeWindow/20);
yd = zeros(1, numFrames-1);
l = 0;

tic;
for k = 1:numFrames

    n = k-1;

    I = readFrame(V);
    I = rgb2gray(I);

    %I = I(end/2:end, 1:end);
    %I = I(700:end, 700:1600);

    %I = I(400:end, 1:end);
    %I = I(450:1000, 650:1200);
    %I = I(375:925, 650:1200);

    %I = I(400:end, 1:end);
    %I = I(400:950, 400:1200);
```

```
if (n >= 1)

    Id = double(I)-double(Iprev);

    if (n > 1)

        yd(n) = 0.1*mean(Id(:)) + 0.9*yd(n-1);

    else

        yd(n) = mean(Id(:));

    end

end

Iprev = I;

switch method

    % Differanse-metode
    case 1

        if (n >= 1)

            Iw = Id;

            if (n > 1)

                Iw = alpha*Iw + (1-alpha)*Iw_prev;

            end

            Iw_prev = Iw;

        end

    % Fase-metode
    case 2

        if (histEqOn)

            I = double(histeq(I));

        else
```

```
I = double(I);

end

L = laplacianPyr(I, 1);
Iorig = L{1};
R1 = imfilter(Iorig, h1);
R2 = imfilter(Iorig, h2);
%[Io,Ip,Ia] = cart2sph(R1,R2,Iorig);
Ia = sqrt(Iorig.^2 + R1.^2 + R2.^2); % Lokal amplitude
Ip = atan2(sqrt(R1.^2 + R2.^2), Iorig); % Lokal fase
%Io = atan2(R2, R1);

if (n >= 1)

    Ip_diff = Ip-Ip_prev;
    Iw = Ip_diff.*(Ia.^2);

    if(n > 1)

        Iw = alpha*Iw + (1-alpha)*Iw_prev;

    end

    Iw_prev = Iw;

end

if (n==0)
    Ip_prev = Ip;
end
%Ip_prev = Ip;

otherwise

    break;

end

if (n >= 1)

    y(n) = sum(Iw(:));

end
```

```
check = (n >= framesInTimeWindow);  
check = check && (mod(n-framesInTimeWindow, framesInUpdateTime) == 0);  
check = check || (k == numFrames);
```

```
if (check)
```

```
    l = l + 1;  
    t(l) = k/V.FrameRate;
```

```
    y1 = y(n-framesInTimeWindow+1:n);  
    yf = filtfilt(ones(1,Nh)/Nh, 1, y1);  
    y1 = y1-yf;  
    y1 = y1.*hamming(length(y1))';
```

```
    Y = abs(fft(y1, 1000));  
    f = linspace(0, V.FrameRate, length(Y));
```

```
    [pks, locs] = findpeaks(Y(1:end/2));
```

```
    if (l == 1)
```

```
        mnFrom = 1;
```

```
        [maxPeakVal, idx] = max(pks);  
        fPeaks(l) = f(locs(idx));
```

```
    else
```

```
        if (fPeaks(l-1) < 0)
```

```
            mnFrom = 1;
```

```
            [maxPeakVal, idx] = max(pks);  
            fPeaks(l) = f(locs(idx));
```

```
        else
```

```
            mnFrom = mnFrom + (l-Nf>mnFrom);
```

```
            possibleFreq = f(locs(pks>=(max(pks)*0.5)));  
            mnVal = mean(fPeaks(mnFrom:l-1));  
            [~, minLoc] = min(abs(possibleFreq - mnVal));  
            fPeaks(l) = possibleFreq(minLoc);
```

```
        end
```

```
    end
```



```
    if ((sum(abs(yd(n-framesInTimeWindow+1:n)) > yd_th)) ~= 0)
        fPeaks(1) = -1;
    end

    display(100*k/numFrames);
end

end

runTime = toc;

%%

load([videoName, '.mat']);

realFreqTop = zeros(1, fPeaks_num);
realFreqBot = zeros(1, fPeaks_num);
l = 0;
for k = 1:numFrames

    n = k-1;

    check = (n >= framesInTimeWindow);
    check = check && (mod(n-framesInTimeWindow, framesInUpdateTime) == 0);
    check = check || (k == numFrames);

    if (check)

        l = l+1;

        if ((sum(topPkt(k-framesInTimeWindow+1:k)<0)) ~= 0)

            realFreqTop(l) = -1;

        else

            realFreqTop(l) = sum(topPkt(k-framesInTimeWindow+1:k)...
                /timeWindow);

        end

        if ((sum(bunnPkt(k-framesInTimeWindow+1:k)<0)) ~= 0)
```

```
    realFreqBot(l) = -1;

    else

        realFreqBot(l) = sum(bunnPkt(k-framesInTimeWindow+1:k)...
            /timeWindow);

    end

end

end

realFreq = (realFreqTop + realFreqBot)/2;
realFreq(realFreqBot<0) = -1;
realFreq(realFreqTop<0) = -1;

figure;
hold on;
plot(t, realFreq);
plot(t, fPeaks);

ylabel('Frekvens [Hz]');
xlabel('Tid [s]');
legend({'Ekte frekvens', 'Målt frekvens'});
yMin = min([min(realFreq), min(fPeaks)]);
yMax = 1.5*max([max(realFreq), max(fPeaks)]);
ylim([yMin*(yMin<0), 1.5*yMax]);
xlim([0, ceil(t(end))]);
grid;

hold off;

okMes = (fPeaks > -1).*(realFreq > -1);
relErr = abs(fPeaks.*okMes - realFreq.*okMes)./realFreq;
avgRelErr = sum(relErr)/sum(okMes);
andOkMes = sum(okMes)/sum(realFreq > -1);

display(['Average relative error: ', num2str(avgRelErr*100)]);
display(['Andel gyldige målinger: ', num2str(andOkMes*100)]);
display(['Runtime: ', num2str(runTime)]);
```

```
function pyr = laplacianPyr(Im, scales)
% Input
% Im: Bilde som det vil dannes en Laplace-pyramide av. Dersom det er et
% fargebilde dannes Laplace-pyramiden for hver fargekomponent.
% scales: Antall trinn i Laplace-pyramiden
% Output
% pyr: Celle-vektor der hver indeks er et trinn i pyramiden. Indeks 1 er
% fin skala og siste indeks er grov skala.

pyr = cell(1, scales);
I0 = Im;

for i=1:scales

    I1 = impyramid(I0, 'reduce');
    imSize = [size(I0, 1), size(I0, 2)];
    pyr{i} = I0-imresize(impyramid(I1, 'expand'), imSize);
    I0 = I1;

end

end
```