

DAFx - Digital Audio Effects [Zölzer 2002]

Capítulo 7: Processamento em tempo-frequência

André Jucovsky Bianchi
ajb@ime.usp.br

Departamento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo

15 de maio de 2012

Estrutura da apresentação

Introdução

Modelos básicos de Phase Vocoder

- Soma de banco de filtros

- Análise/síntese bloco-a-bloco

Implementações

- Banco de filtros - soma de senóides

- FFT - IFFT

- FFT - soma de senóides

- Gaborets

- Phase unwrapping

Efeitos

- Filtros em tempo-frequência

- Dispersão

- Robotização

- Murmurização

- Compressão/expansão temporal

- Alteração de altura

- Separação de componentes estáveis e transitórios

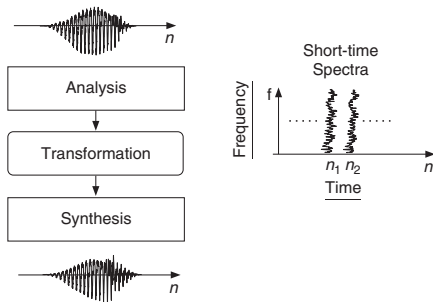
- Mutação entre dois sons

- Denoising

- Panorâmica espectral

Conclusões

Introdução



Esquema geral dos efeitos digitais baseados na representação tempo-frequência:

1. **Análise:** $x(n) \leftrightarrow X(k)$.
2. **Transformação:** $\tilde{X}(k) = f(X(k))$.
3. **Ressíntese:** $\tilde{x}(n) \leftrightarrow \tilde{X}(k)$.

Introdução

Phase Vocoder:

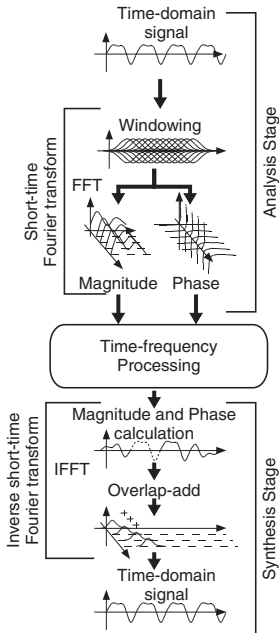
1. Análise:

- ▶ Janela deslizante de tamanho finito.
- ▶ FFTs consecutivas do sinal $x(n)$.
- ▶ Espectro variante no tempo: $X(n, k) = |X(n, k)|e^{i\varphi(n, k)}$,
 $k = 0, 1, \dots, N - 1$.

2. Transformação:

modificações em $|X(n, k)|$ e $\varphi(n, k)$.

- ### 3. Ressíntese:
- IFFT, soma de senóides, overlap-add, etc.



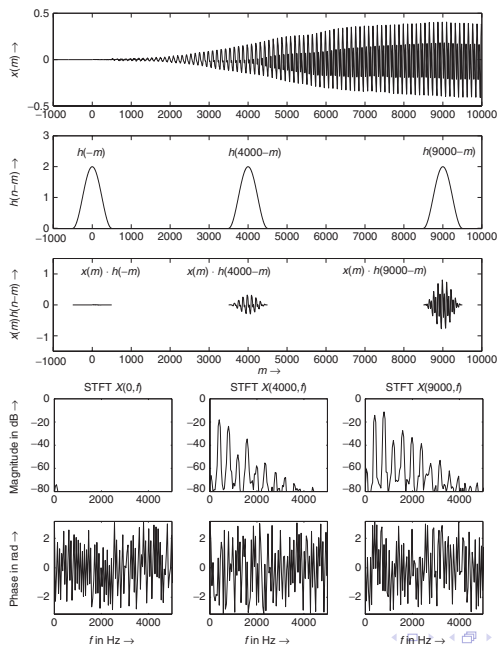
Short-Time Fourier Transform (STFT)

Se $x(n) \in \mathbb{C}^{\mathbb{Z}}$ é um **sinal** e $h(n) \in \mathbb{C}^{\mathbb{Z}}$ uma **janela** centrada em $n = 0$, a **STFT** é definida por:

$$\begin{aligned} X(n, k) &= \sum_{m=-\infty}^{\infty} x(m)h(n-m)W_N^{mk} \\ &= X_R(n, k) + iX_I(n, k) = |X(n, k)| \cdot e^{i\varphi(n, k)}, \end{aligned}$$

com $W_N = e^{-i2\pi/N}$.

Janela de análise deslizante e STFT



Estrutura da apresentação

Introdução

Modelos básicos de Phase Vocoder

Soma de banco de filtros

Análise/síntese bloco-a-bloco

Implementações

Banco de filtros - soma de senóides

FFT - IFFT

FFT - soma de senóides

Gaborets

Phase unwrapping

Efeitos

Filtros em tempo-frequência

Dispersão

Robotização

Murmurização

Compressão/expansão temporal

Alteração de altura

Separação de componentes estáveis e transitórios

Mutação entre dois sons

Denoising

Panorâmica espectral

Conclusões

Modelo de soma de banco de filtros

Análise

Definimos, para $k = 0, \dots, N - 1$:

$$h_k(n) = h(n)W_N^{-nk}$$

$$H_k(e^{i\omega}) = H(e^{i(\omega - \omega_k)}), \quad \omega_k = \frac{2\pi}{N}k.$$

Assim, obtemos N sinais passa-banda:

$$\begin{aligned} y_k(n) = \tilde{X}(n, k) &= \sum_{m=-\infty}^{\infty} x(m)h_k(n - m) = \\ &= \sum_{m=-\infty}^{\infty} x(m)h(n - m)W_N^{-(n-m)k} \end{aligned} \quad (2.1)$$

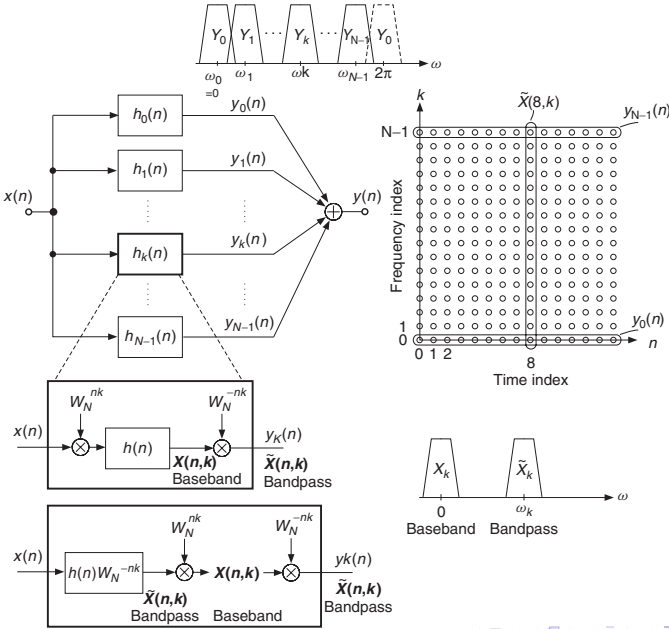
$$\begin{aligned} &= W_N^{-nk} \sum_{m=-\infty}^{\infty} x(m)h(n - m)W_N^{mk} = \\ &= W_N^{-nk} X(n, k). \end{aligned} \quad (2.2)$$

Modelo de soma de banco de filtros

Síntese

$$y(n) = \sum_{k=0}^{N-1} y_k(n) = \sum_{k=0}^{N-1} \tilde{X}(n, k) = \sum_{k=0}^{N-1} X(n, k) W_N^{-nk}$$

Modelo de soma de banco de filtros



Modelo de soma de banco de filtros

Simetria em sinais reais (1/2)

Se $x(n) \in \mathbb{R}$, vale:

$$y_k(n) = \tilde{X}(n, k) = \tilde{X}^*(n, N - k) = y_{N-k}^*(n),$$

e podemos definir:

$$\begin{aligned}\hat{y}_k &= \tilde{X}(n, k) + \tilde{X}(n, N - k) = \tilde{X}(n, k) + \tilde{X}^*(n, k) \\ &= |X(n, k)| \cdot \left[e^{i\tilde{\varphi}(n, k)} + e^{-i\tilde{\varphi}(n, k)} \right] = \\ &= 2|X(n, k)| \cdot \cos(\tilde{\varphi}(n, k)),\end{aligned}$$

para $k = 1, \dots, \frac{N}{2} - 1$.

Modelo de soma de banco de filtros

Simetria em sinais reais (2/2)

O que leva às seguintes equações de análise:

$$\hat{y}_0(n) = y_0(n), \quad k = 0.$$

$$\hat{y}_k(n) = 2|X(n, k)| \cos(\omega_k n + \varphi(n, k)), \quad k = 1, \dots, \frac{N}{2} - 1.$$

$$\hat{y}_{\frac{N}{2}}(n) = y_{\frac{N}{2}}(n), \quad k = \frac{N}{2}.$$

A equação de síntese fica:

$$y(n) = \sum_{k=0}^{N/2} \hat{y}_k(n).$$

Estrutura da apresentação

Introdução

Modelos básicos de Phase Vocoder

- Soma de banco de filtros

- Análise/síntese bloco-a-bloco**

Implementações

- Banco de filtros - soma de senóides

- FFT - IFFT

- FFT - soma de senóides

- Gaborets

- Phase unwrapping

Efeitos

- Filtros em tempo-frequência

- Dispersão

- Robotização

- Murmurização

- Compressão/expansão temporal

- Alteração de altura

- Separação de componentes estáveis e transitórios

- Mutação entre dois sons

- Denoising

- Panorâmica espectral

Conclusões

Modelo de análise/síntese bloco-a-bloco

Análise

Se o tamanho do salto de análise é R_a , a análise do s -ésimo bloco é dada por:

$$\begin{aligned} X(sR_a, k) &= \sum_{m=-\infty}^{\infty} x(m)h(sR_a - m)W_N^{mk} \\ &= W_N^{sR_a k} \sum_{m=-\infty}^{\infty} x(m)h(sR_a - m)W_N^{-(sR_a - m)k} \\ &= W_N^{sR_a k} \cdot \tilde{X}(sR_a, k) \\ &= X_R(sR_a, k) + iX_I(sR_a, k) = |X(sR_a, k)| \cdot e^{i\varphi(sR_a, k)}, \end{aligned}$$

com $k = 0, 1, \dots, N - 1$.

Modelo de análise/síntese bloco-a-bloco

Síntese

Se $Y(sR_s, k)$ é o espectro modificado (sendo que R_s é o tamanho do salto de síntese), a síntese do s -ésimo bloco é dada por:

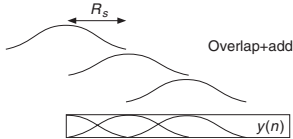
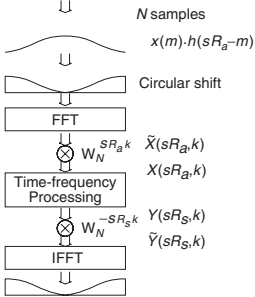
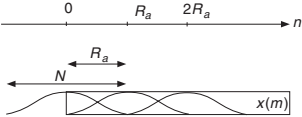
$$y(n) = \sum_{s=-\infty}^{\infty} f(n - sR_s)y_s(n - sR_s),$$

sendo que y_s é dado por:

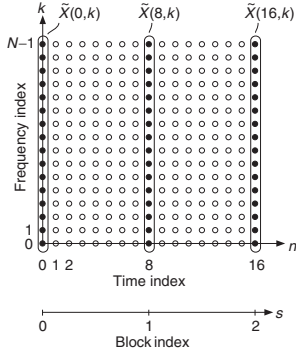
$$y_s(n) = \frac{1}{N} \sum_{k=0}^{N-1} \left[W_N^{-sR_s k} Y(sR_s, k) \right] W_N^{-nk}$$

com $k = 0, 1, \dots, N - 1$.

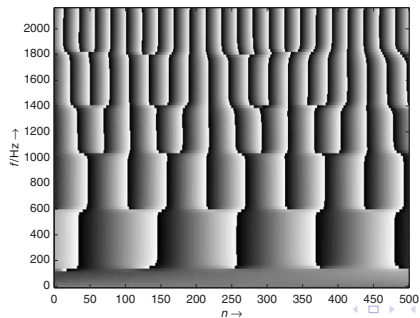
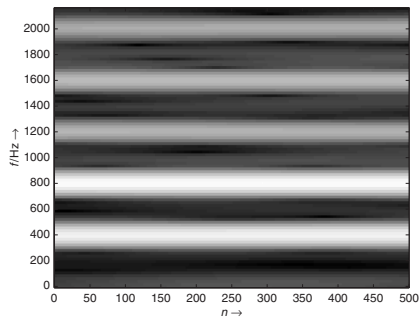
Modelo de análise/síntese bloco-a-bloco



$x(m) \cdot h(-m)$ $x(m) \cdot h(8-m)$ $x(m) \cdot h(16-m)$



Representações de amplitude e fase



Estrutura da apresentação

Introdução

Modelos básicos de Phase Vocoder

Soma de banco de filtros

Análise/síntese bloco-a-bloco

Implementações

Banco de filtros - soma de senóides

FFT - IFFT

FFT - soma de senóides

Gaborets

Phase unwrapping

Efeitos

Filtros em tempo-frequência

Dispersão

Robotização

Murmurização

Compressão/expansão temporal

Alteração de altura

Separação de componentes estáveis e transitórios

Mutação entre dois sons

Denosing

Panorâmica espectral

Conclusões

Implementação: banco de filtros - soma de senóides

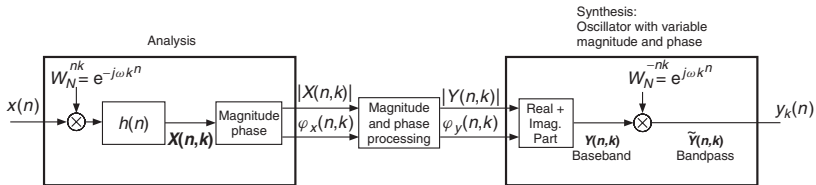
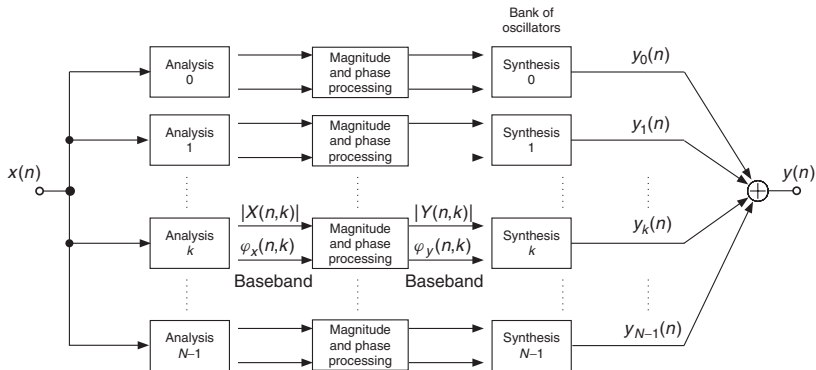
A análise pode ser vista como fruto de uma convolução:

$$\begin{aligned} X(n, k) &= [x(n) \cdot e^{-i\omega_k n}] * h(n) = |X(n, k)| e^{i\varphi(n, k)} \\ &= X_R(n, k) + iX_I(n, k), \end{aligned}$$

com:

$$\begin{aligned} X_R(n, k) &= |X(n, k)| \cos(\varphi(n, k)) \\ X_I(n, k) &= |X(n, k)| \sin(\varphi(n, k)). \end{aligned}$$

Implementação: banco de filtros - soma de senóides



Implementação: banco de filtros - soma de senóides

Frequência instantânea

A frequência instantânea é dada pela derivada da fase na saída do filtro passa-banda:

$$\begin{aligned}\omega_i(n, k) &= \frac{2\pi}{f_S} f_i(n, k) = \frac{d}{dn} \tilde{\varphi}(n, k) \\ &= \frac{d}{dn} (\omega_k n + \varphi(n, k)) \\ &= \omega_k + \frac{d}{dn} \varphi(n, k) \\ &= \omega_k + \varphi(n, k) - \varphi(n-1, k)\end{aligned}$$

O que dá:

$$f_i(n, k) = \left(\frac{k}{N} + \frac{\omega(n, k) - \omega(n-1, k)}{2\pi} \right) \cdot f_S.$$

Implementação: banco de filtros - soma de senóides

Síntese

Cálculo da fase a partir de uma frequência instantânea modificada:

$$\tilde{\varphi}(n, k) = \tilde{\varphi}(0, k) + \int_0^{nT} 2\pi f_i(\tau, k) d\tau$$

O sinal de saída é dado por:

$$\begin{aligned}\tilde{Y}(n, k) &= |Y(n, k)| e^{i\varphi_Y(n, k)} \cdot e^{i\omega_k n} \\ &= Y(n, k) e^{i\omega_k n}\end{aligned}$$

E a ressíntese é a soma dos sinais passa-banda modificados:

$$\begin{aligned}y(n) &= \sum_{k=0}^{N-1} \tilde{Y}(n, k) = \sum_{k=0}^{N-1} Y(n, k) \cdot e^{i\omega_k n} \\ &= \sum_{k=0}^{N/2} A(n, k) \cos[\omega_k n + \varphi_Y(n, k)]\end{aligned}$$

Estrutura da apresentação

Introdução

Modelos básicos de Phase Vocoder

Soma de banco de filtros

Análise/síntese bloco-a-bloco

Implementações

Banco de filtros - soma de senóides

FFT - IFFT

FFT - soma de senóides

Gaborets

Phase unwrapping

Efeitos

Filtros em tempo-frequência

Dispersão

Robotização

Murmurização

Compressão/expansão temporal

Alteração de altura

Separação de componentes estáveis e transitórios

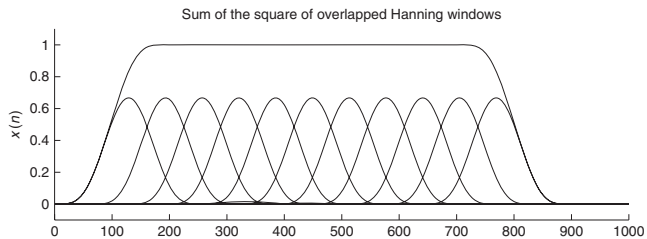
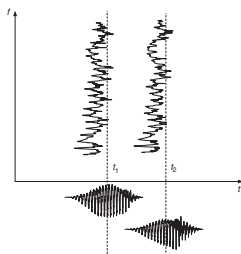
Mutação entre dois sons

Denoising

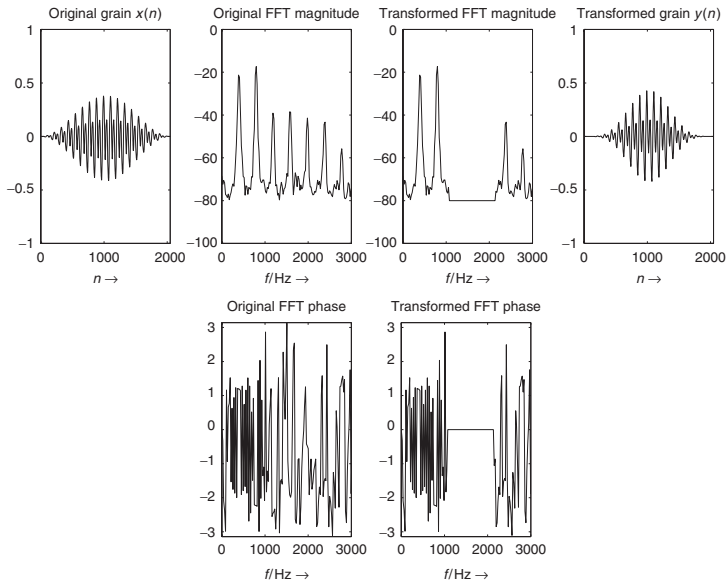
Panorâmica espectral

Conclusões

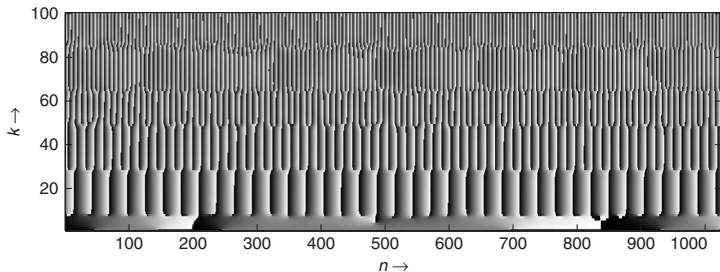
Implementação: FFT - IFFT



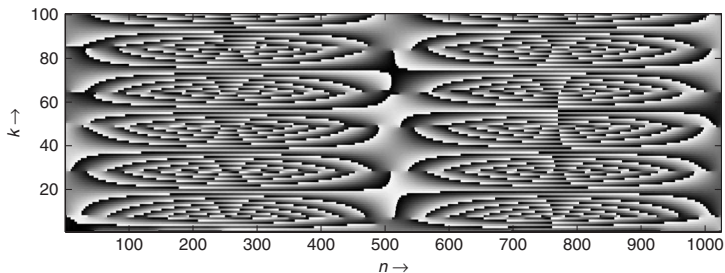
Implementação: FFT - IFFT



Implementação: FFT - IFFT



(a)



(b)

Estrutura da apresentação

Introdução

Modelos básicos de Phase Vocoder

Soma de banco de filtros

Análise/síntese bloco-a-bloco

Implementações

Banco de filtros - soma de senóides

FFT - IFFT

FFT - soma de senóides

Gaborets

Phase unwrapping

Efeitos

Filtros em tempo-frequência

Dispersão

Robotização

Murmurização

Compressão/expansão temporal

Alteração de altura

Separação de componentes estáveis e transitórios

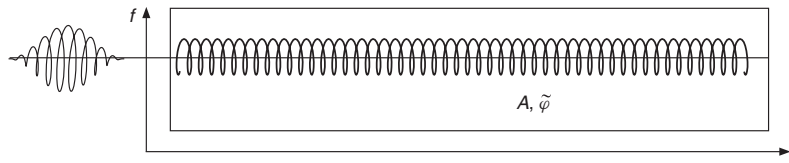
Mutação entre dois sons

Denoising

Panorâmica espectral

Conclusões

Implementação: FFT - soma de senóides



Estrutura da apresentação

Introdução

Modelos básicos de Phase Vocoder

Soma de banco de filtros

Análise/síntese bloco-a-bloco

Implementações

Banco de filtros - soma de senóides

FFT - IFFT

FFT - soma de senóides

Gaborets

Phase unwrapping

Efeitos

Filtros em tempo-frequência

Dispersão

Robotização

Murmurização

Compressão/expansão temporal

Alteração de altura

Separação de componentes estáveis e transitórios

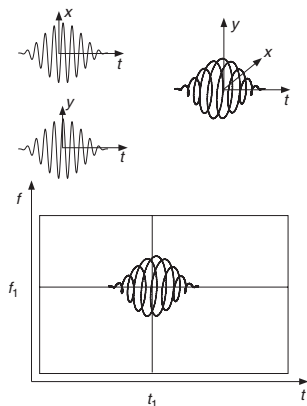
Mutação entre dois sons

Denoising

Panorâmica espectral

Conclusões

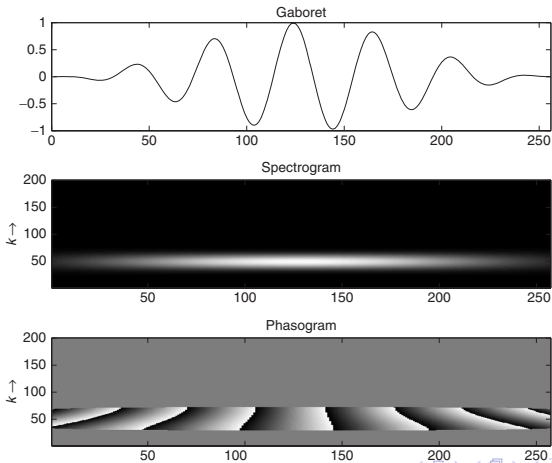
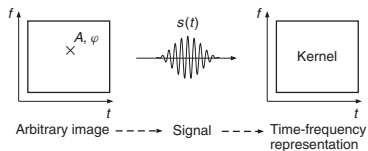
Implementação: Gaborets



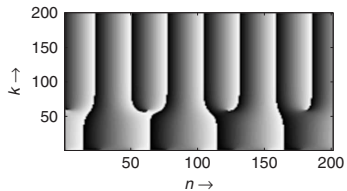
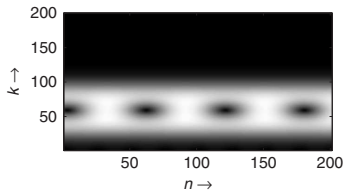
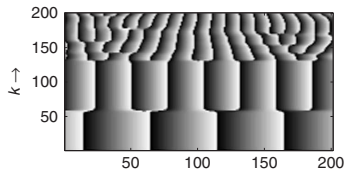
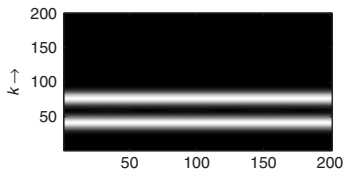
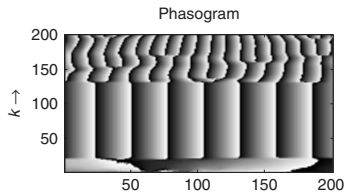
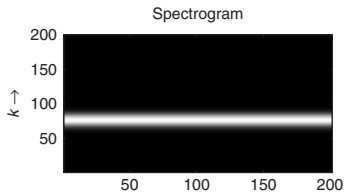
A síntese é dada por:

$$y(n) = \sum_{s=-\infty}^{\infty} \sum_{k=0}^{N-1} Y(sR_s, k) f(n - sR_s) W_N^{-nk}$$

Implementação: Gaborets



Implementação: Gaborets



Estrutura da apresentação

Introdução

Modelos básicos de Phase Vocoder

Soma de banco de filtros

Análise/síntese bloco-a-bloco

Implementações

Banco de filtros - soma de senóides

FFT - IFFT

FFT - soma de senóides

Gaborets

Phase unwrapping

Efeitos

Filtros em tempo-frequência

Dispersão

Robotização

Murmurização

Compressão/expansão temporal

Alteração de altura

Separação de componentes estáveis e transitórios

Mutação entre dois sons

Denoising

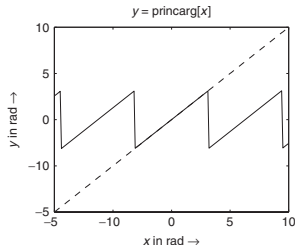
Panorâmica espectral

Conclusões

Implementação: phase unwrapping

A n -ésima fase do k -ésimo sinal de saída do banco de filtros é:

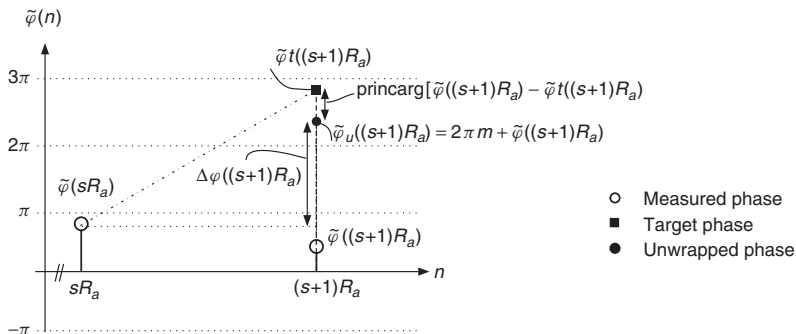
$$\tilde{\varphi}(n, k) = \varphi(n, k) + \underbrace{\frac{2\pi}{N}}_{\omega_k} n$$



```
1 function phase = princarg(phase_in)
2   phase = mod(phase_in+pi, -2*pi) + pi
```

princarg.m

Implementação: phase unwrapping



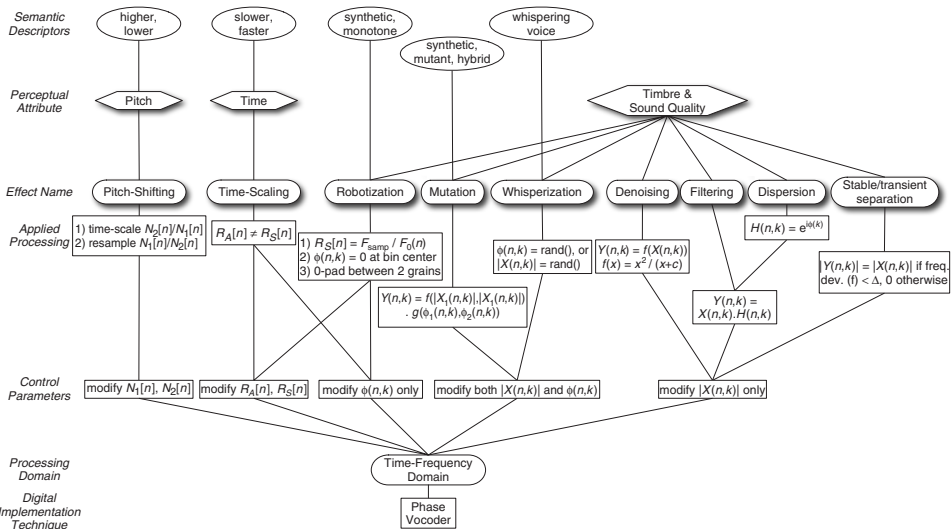
Diferença de fase desenrolada:

$$\Delta\varphi((s+1)R_a) = \omega_k R_a + \text{princarg}(\tilde{\varphi}((s+1)R_a) - \tilde{\varphi}(sR_a) - \omega_k R_a)$$

E a frequência instantânea para o índice k no instante $(s+1)R_a$:

$$f_i((s+1)R_a) = \frac{1}{2\pi} \frac{\Delta\varphi((s+1)R_a)}{R_a} f_S$$

Efeitos, operações e atributos perceptuais



Estrutura da apresentação

Introdução

Modelos básicos de Phase Vocoder

Soma de banco de filtros

Análise/síntese bloco-a-bloco

Implementações

Banco de filtros - soma de senóides

FFT - IFFT

FFT - soma de senóides

Gaborets

Phase unwrapping

Efeitos

Filtros em tempo-frequência

Dispersão

Robotização

Murmurização

Compressão/expansão temporal

Alteração de altura

Separação de componentes estáveis e transitórios

Mutação entre dois sons

Denoising

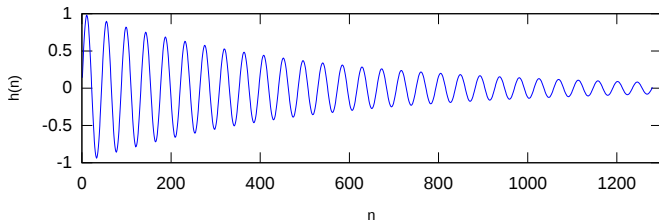
Panorâmica espectral

Conclusões

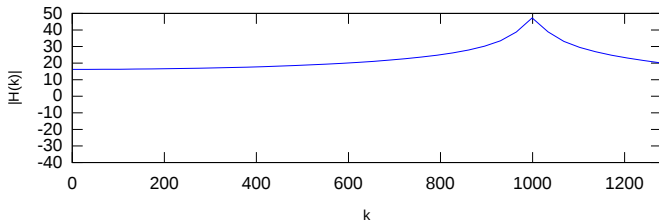
Filtros em tempo-frequência

$$h(n) = e^{\alpha n} \sin(2\pi fn) \Leftrightarrow H(k) = FFT(h(n))$$

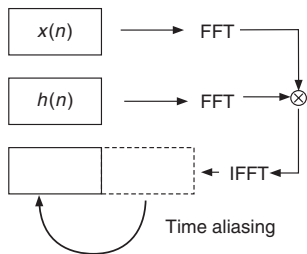
Impulse response of the FIR



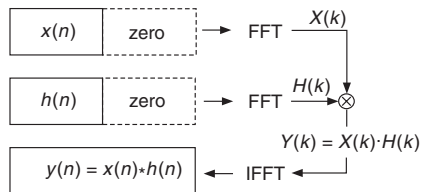
Magnitude spectrum of the FIR



Filtros em tempo-frequência



Circular convolution



Fast convolution

Convolução rápida:

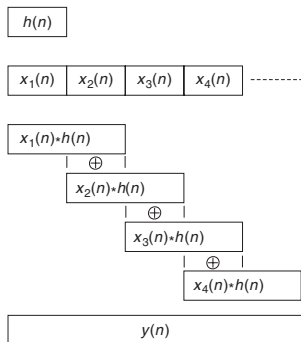
1. Preencha os sinais $x(n)$ e $h(n)$ com zeros até o tamanho $2N$.
2. Calcule $X(k)$ e $H(k)$, as FFTs de $2N$ pontos dos dois sinais.
3. Calcule $Y(k) = X(k)H(k)$ para $k = 0, \dots, 2N - 1$.
4. $y(n) = \text{IFFT}(Y(k))$.

Filtros em tempo-frequência

```
1 %—— user data ——
2 s_FIR = 1280; % length of the fir [samples]
3 s_win = 2*s_FIR; % window size [samples] for zero padding
4 % ...
5 %—— initialize calculation of fir ——
6 x = (1:s_FIR);
7 fr = 1000/FS;
8 alpha = -0.002;
9 fir = (exp(alpha*x).* sin(2*pi*fr*x))'; % FIR coeffs
10 fir2 = [fir; zeros(s_win-s_FIR,1)];
11 fcorr = fft(fir2);
12 % ...
13 while pin<pend
14     grain = [DAFx_in(pin+1:pin+s_FIR); vec_pad];
15 %=====
16     ft = fft(grain) .* fcorr;
17     grain = (real(ifft(ft)));
18 %=====
19     DAFx_out(pin+1:pin+s_win) = ...
20         DAFx_out(pin+1:pin+s_win) + grain;
21     pin = pin + s_FIR;
22 end
```

VX_filter.m

Overlap
+ add



Estrutura da apresentação

Introdução

Modelos básicos de Phase Vocoder

Soma de banco de filtros

Análise/síntese bloco-a-bloco

Implementações

Banco de filtros - soma de senóides

FFT - IFFT

FFT - soma de senóides

Gaborets

Phase unwrapping

Efeitos

Filtros em tempo-frequência

Dispersão

Robotização

Murmurização

Compressão/expansão temporal

Alteração de altura

Separação de componentes estáveis e transitórios

Mutação entre dois sons

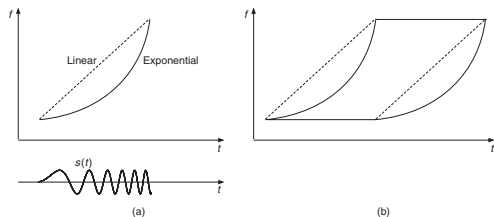
Denosing

Panorâmica espectral

Conclusões

Dispersão

Ideia: Atrasar componentes do sinal através de um filtro cuja resposta em frequência impõe atrasos que dependem da frequência.



```
1 n = 300;
2 x = (1:n)/n;
3 f0 = 50;
4 f1 = 4000;
5 %----- linear chirp -----
6 freq = 2*pi * (f0+(f1-f0)*x) / 44100;
7 %----- exponential chirp -----
8 rap = f1/f0;
9 freq = (2*pi*f0/44100) * (rap.^x);
10 %----- filter coefficients -----
11 fir = (sin(cumsum(freq)))';
```

Computando um trinado.

Estrutura da apresentação

Introdução

Modelos básicos de Phase Vocoder

Soma de banco de filtros

Análise/síntese bloco-a-bloco

Implementações

Banco de filtros - soma de senóides

FFT - IFFT

FFT - soma de senóides

Gaborets

Phase unwrapping

Efeitos

Filtros em tempo-frequência

Dispersão

Robotização

Murmurização

Compressão/expansão temporal

Alteração de altura

Separação de componentes estáveis e transitórios

Mutação entre dois sons

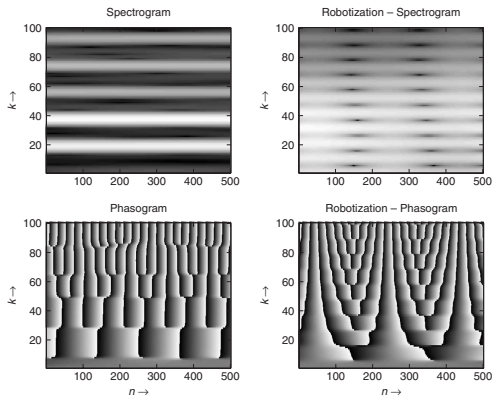
Denoising

Panorâmica espectral

Conclusões

Robotização

Ideia: aplicar uma altura fixa através do anulamento das fases.



```
1 f = fft(grain);  
2 r = abs(f);  
3 grain = fftshift(real(ifft(r))).*w2;
```

VX_filter.m

Estrutura da apresentação

Introdução

Modelos básicos de Phase Vocoder

Soma de banco de filtros

Análise/síntese bloco-a-bloco

Implementações

Banco de filtros - soma de senóides

FFT - IFFT

FFT - soma de senóides

Gaborets

Phase unwrapping

Efeitos

Filtros em tempo-frequência

Dispersão

Robotização

Murmurização

Compressão/expansão temporal

Alteração de altura

Separação de componentes estáveis e transitórios

Mutação entre dois sons

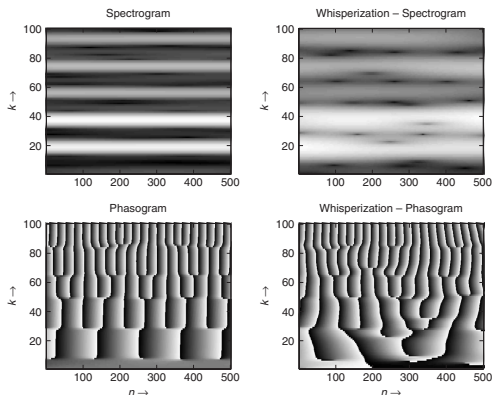
Denoising

Panorâmica espectral

Conclusões

Murmurização

Ideia: Adicionar uma fase aleatória a cada coeficiente obtido através da FFT.



```
1 f = fft(fftshift(grain));  
2 r = abs(f);  
3 phi = 2*pi*rand(s_win,1);  
4 ft = (r.*exp(i*phi));  
5 grain = fftshift(real(ifft(ft))).*w2;
```

Estrutura da apresentação

Introdução

Modelos básicos de Phase Vocoder

Soma de banco de filtros

Análise/síntese bloco-a-bloco

Implementações

Banco de filtros - soma de senóides

FFT - IFFT

FFT - soma de senóides

Gaborets

Phase unwrapping

Efeitos

Filtros em tempo-frequência

Dispersão

Robotização

Murmurização

Compressão/expansão temporal

Alteração de altura

Separação de componentes estáveis e transitórios

Mutação entre dois sons

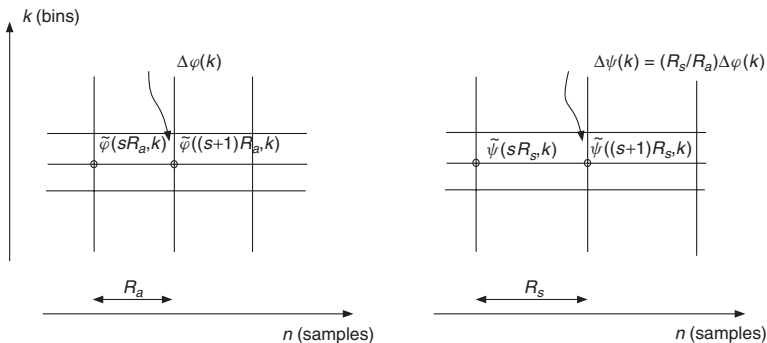
Denoising

Panorâmica espectral

Conclusões

Compressão/expansão temporal

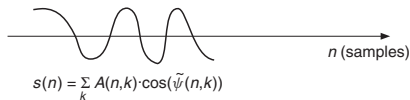
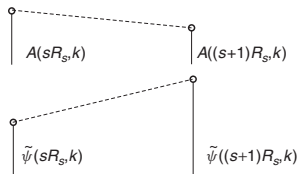
Ideia: mudar a escala temporal sem alterar o conteúdo espectral.



Compressão/expansão temporal

```
1 %=====
2 fc = fft(fftshift(grain));
3 f = fc(1:11);
4 r = abs(f);
5 phi = angle(f);
6 %—— calculate phase increment per block ——
7 delta_phi = omega + princarg(phi-phi0-omega);
8 %—— calculate phase & mag increments per sample ——
9 delta_r = (r-r0) / n2; % for synthesis
10 delta_psi = delta_phi / n1; % derived from analysis
11 %—— computing output samples for current block ——
12 for k=1:n2
13     r0 = r0 + delta_r;
14     psi = psi + delta_psi;
15     res(k) = r0'*cos(psi);
16 end
17 %—— values for processing next block ——
18 phi0 = phi;
19 r0 = r;
20 psi = princarg(psi);
21 %=====
```

VX_tstretch_bank.m



Estrutura da apresentação

Introdução

Modelos básicos de Phase Vocoder

Soma de banco de filtros

Análise/síntese bloco-a-bloco

Implementações

Banco de filtros - soma de senóides

FFT - IFFT

FFT - soma de senóides

Gaborets

Phase unwrapping

Efeitos

Filtros em tempo-frequência

Dispersão

Robotização

Murmurização

Compressão/expansão temporal

Alteração de altura

Separação de componentes estáveis e transitórios

Mutação entre dois sons

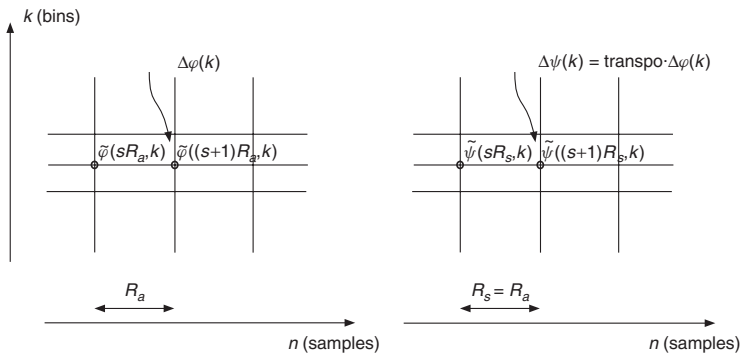
Denoising

Panorâmica espectral

Conclusões

Alteração de altura

Ideia: mudar a altura musical do sinal sem alterar a escala temporal.



Estrutura da apresentação

Introdução

Modelos básicos de Phase Vocoder

Soma de banco de filtros

Análise/síntese bloco-a-bloco

Implementações

Banco de filtros - soma de senóides

FFT - IFFT

FFT - soma de senóides

Gaborets

Phase unwrapping

Efeitos

Filtros em tempo-frequência

Dispersão

Robotização

Murmurização

Compressão/expansão temporal

Alteração de altura

Separação de componentes estáveis e transitórios

Mutação entre dois sons

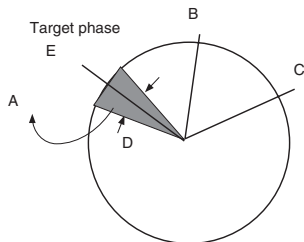
Denoising

Panorâmica espectral

Conclusões

Separação de componentes estáveis e transitórios

Ideia: Identificação de componentes “estáveis em frequência” e eliminação de componentes “transitórios”.



```
1 % =====  
2 f      = fft(fftshift(grain));  
3 theta  = angle(f);  
4 dev    = princarg(theta - 2*theta1 + theta2);  
5 % plot(dev);drawnow;  
6 %----- set to 0 magnitude values below 'test' threshold  
7 ft     = f.*(abs(dev) < vtest);  
8 grain  = fftshift(real(ifft(ft))).*w2;  
9 theta2 = theta1;  
10 theta1 = theta;  
11 % =====
```

Estrutura da apresentação

Introdução

Modelos básicos de Phase Vocoder

Soma de banco de filtros

Análise/síntese bloco-a-bloco

Implementações

Banco de filtros - soma de senóides

FFT - IFFT

FFT - soma de senóides

Gaborets

Phase unwrapping

Efeitos

Filtros em tempo-frequência

Dispersão

Robotização

Murmurização

Compressão/expansão temporal

Alteração de altura

Separação de componentes estáveis e transitórios

Mutação entre dois sons

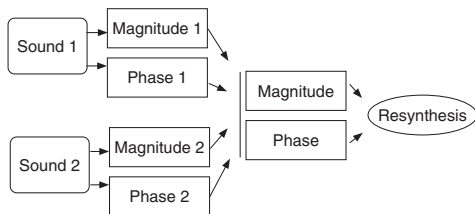
Denoising

Panorâmica espectral

Conclusões

Mutação entre dois sons

Ideia: gerar uma representação arbitrária em tempo frequência a partir de dois sinais originais e construir um sinal a partir dela.



```
1 % =====  
2 f1 = fft(fftshift(grain1));  
3 r1 = abs(f1);  
4 theta1 = angle(f1);  
5 f2 = fft(fftshift(grain2));  
6 r2 = abs(f2);  
7 theta2 = angle(f2);  
8 %----- the next two lines can be changed according to the effect  
9 r = r1;  
10 theta = theta2;  
11 ft = (r.* exp(i*theta));  
12 grain = fftshift(real(ifft(ft))).*w2;  
13 % =====
```

Estrutura da apresentação

Introdução

Modelos básicos de Phase Vocoder

Soma de banco de filtros

Análise/síntese bloco-a-bloco

Implementações

Banco de filtros - soma de senóides

FFT - IFFT

FFT - soma de senóides

Gaborets

Phase unwrapping

Efeitos

Filtros em tempo-frequência

Dispersão

Robotização

Murmurização

Compressão/expansão temporal

Alteração de altura

Separação de componentes estáveis e transitórios

Mutação entre dois sons

Denoising

Panorâmica espectral

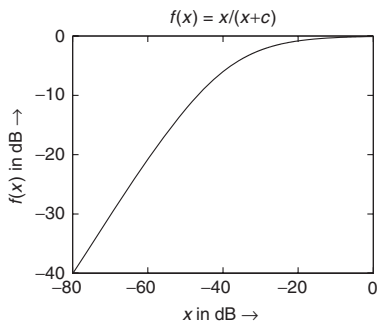
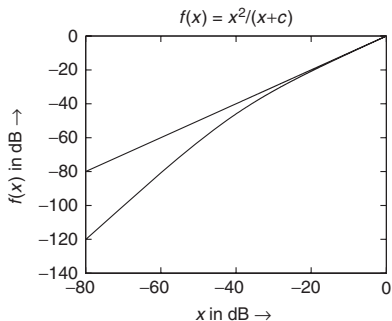
Conclusões

Denoising

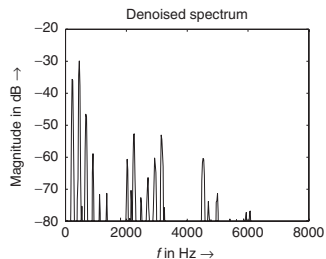
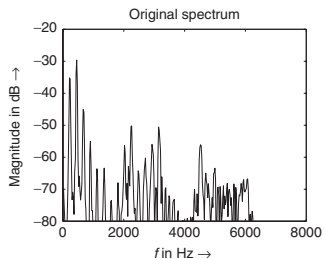
Ideia: atenuar “ruído” no domínio da frequência.

$$X_d(n, k) = f(X(n, k))$$

$$f(x) = \frac{x^2}{x+c}$$



Denosing



```
1 while pin<pend
2   grain = DAFx_in(pin+1:pin+s_win) .* w1;
3   % =====
4   f     = fft(grain);
5   r     = abs(f)/hs_win;
6   ft    = f.*r./ (r+coef);
7   grain = (real(ifft(ft))) .* w2;
8   % =====
9   DAFx_out(pout+1:pout+s_win) = ...
10    DAFx_out(pout+1:pout+s_win) + grain;
11   pin = pin + n1;
12   pout = pout + n2;
13 end
```

Estrutura da apresentação

Introdução

Modelos básicos de Phase Vocoder

Soma de banco de filtros

Análise/síntese bloco-a-bloco

Implementações

Banco de filtros - soma de senóides

FFT - IFFT

FFT - soma de senóides

Gaborets

Phase unwrapping

Efeitos

Filtros em tempo-frequência

Dispersão

Robotização

Murmurização

Compressão/expansão temporal

Alteração de altura

Separação de componentes estáveis e transitórios

Mutação entre dois sons

Denosing

Panorâmica espectral

Conclusões

Panorâmica espectral

Ideia: separação de componentes espectrais em canais diferentes.

A lei de Blumlein:

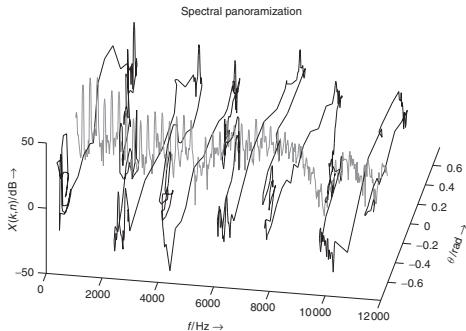
$$\sin \Theta(n, k) = \frac{g_L(n, k) - g_R(n, k)}{g_L(n, k) + g_R(n, k)} \sin \Theta_t$$

Se $\Theta = 45^\circ$, o espectro de saída de cada canal é dado por:

$$X_L(n, k) = \frac{\sqrt{2}}{2} (\cos \Theta(n, k) + \sin \Theta(n, k)) \cdot X(n, k)$$

$$X_R(n, k) = \frac{\sqrt{2}}{2} (\cos \Theta(n, k) - \sin \Theta(n, k)) \cdot X(n, k)$$

Panorâmica espectral



```
1  grain = DAFx_in(pin+1:pin+s_win) .* w1;
2  % =====
3  f      = fft(grain);
4  %----- compute left and right spectrum with Blumlein law at 45 degrees
5  ftL    = coef * f .* (cos(theta) + sin(theta));
6  ftR    = coef * f .* (cos(theta) - sin(theta));
7  grainL = (real(iff_t(ftL))).*w2;
8  grainR = (real(iff_t(ftR))).*w2;
9  % =====
10 DAFx_out(pout+1:pout+s_win,1) = DAFx_out(pout+1:pout+s_win,1) + grainL;
11 DAFx_out(pout+1:pout+s_win,2) = DAFx_out(pout+1:pout+s_win,2) + grainR;
12 pin = pin + n1;
13 pout = pout + n2;
```

Fim! o_O

Bibliografia:

 ZÖLZER, U. (Ed.). *DAFX - Digital Audio Effects*. [S.l.]: John Wiley & Sons, 2002.

Dados de contato:

- ▶ Meu email: ajb@ime.usp.br
- ▶ Esta apresentação: <http://www.ime.usp.br/~ajb/>
- ▶ CM no IME: <http://compmus.ime.usp.br/>