Metamorfose de Sons e Aplicações

Siome Klein Goldenstein

25 de agosto de 1997

Sumário

Lista de Figuras

Introdução

O objetivo deste trabalho é estudar transformações de um sinal de áudio. Esse estudo é feito no contexto de metamorfose de objetos gráficos, utilizando o paradigma dos quatro universos de abstração.

Diversas aplicações de áudio existentes no nosso cotidiano são enquadradas nesta nomenclatura. Um primeiro exemplo são os *compressores* de voz, que, ao contrário do que o nome diz, não buscam comprimir nada, mas são apenas dispositivos que buscam "equalizar" as amplitudes de diferentes vozes.

Compressores são usados em geral em estações de rádio que transmitem simultaneamente as vozes de diferentes locutores. Cada um deles fala de uma forma distinta, um fala mais alto, outro fala longe do microfone, etc. Todos estes fatores tornam o efeito devastador para o ouvinte, tornando necessário o uso de um equipamento mais caro (pois deve funcionar igualmente bem para uma faixa de valores de entrada muito grande). O objetivo do compressor é fazer com que a diferença entre as vozes baixas e altas seja drasticamente diminuida. Para isso utiliza-se uma curva de *compansão* (talvez o nome correto do aparelho deveria ser compansor).

Um outro exemplo interessante é a alteração da duração de um sinal de áudio sem alteração do seu conteúdo. Em documentários, por exemplo, a trilha sonora é feita de forma independente das imagens. Existe um locutor que grava o texto e depois as imagens devem ser encaixadas de forma a criar um certo sincronismo entre o áudio e o vídeo.

Muitas vezes é necessário alterar ligeiramente o comprimento (a duração) da voz, de forma que os eventos do vídeo entrem em um sincronismo perfeito. Esta tarefa não é trivial e, caso seja feita da maneira ingênua, acarretará em uma séria distorção: o efeito de um disco de vinil tocando em uma rotação errada.

Especial atenção será dada ao problema de metamorfose de sons. O processo de metamorfose consiste em se obter uma transição contínua entre dois sinais de áudio. O desenvolvimento de técnicas robustas de especificação e cálculo de metamorfose é de grande importância, devido à grande variedade de aplicações. Uma aplicação interessante consiste em se criar novas vozes a partir de duas vozes dadas.

Essa aplicação vem sendo utilizada de forma bastante freqüente na indústria cinematográfica. Dois exemplos significativos são os filmes "Farinelli - Il Castratio", onde o canto do protagonista foi gerado artesanalmente a partir de da voz de um contra-tenor e uma soprano, e o filme "O Parque dos Dinossauros", onde os ruídos dos dinossauros foram obtidos através da combinação de sons de diversos animais. A combinação de vozes no filme "Farinelli" levou cerca de dois anos para ser realizada devido à falta de ferramentas adequadas.

Outras aplicações em música computacional também são motivantes. É possível, a partir da gravação da interpretação de uma mesma peça por dois músicos famosos, criar

uma peça com um "estilo", interpretação, intermediária. Desta forma seríamos capazes de mesclar estilos de interpretação, assim como criar músicos imaginários, todos tocando simultaneamente.

Ainda em música, seria possível, a partir da mesma peça tocada em tempos diferentes (aqui o termo "tempo" se refere a duração do compasso, a unidade básica temporal em música) criar versões em tempos intermediários ou mesmo com o tempo variável ao longo da peça.

E, finalmente, também seria possível a criação de novos intrumentos, baseado nos sons gravados de instrumentos reais originais. É possível até mesmo combinar todas estas técnicas simultaneamente.

Para a realização deste trabalho utilizamos algumas técnicas já estabelecidas em Computação Gráfica, bem como o ferramental matemático de distribuições Tempo × Freqüência lineares (Transformada de Fourier com Janela, Transformada Wavelet, etc).

Estrutura da Tese

Capítulo 1. É feita a conceituação básica a respeito de objetos gráficos e suas transformações, e como som pode ser enquadrado neste contexto.

Capítulo 2. Seu foco é sobre transformações de áudio. Estudamos também algumas técnicas simples para obter transformações.

Capítulo 3. Estudamos a primeira técnica para a obtenção do morph. É fruto direto das idéias e técnicas simples introduzidas no capítulo 2. Por ser um método ingênuo, por assim dizer, possui diversas limitações, que são discutidas e inspiram uma série de possíveis soluções.

Capítulo 4. Aproveitando a motivação do final do capítulo 3, desenvolve uma técnica baseada na anterior, porém utilizando o conceito de *segmentações temporais*. O método resolve alguns problemas, porém o resultado ainda é insatisfatório. Estes são novamente discutidos e criam a motivação para a terceira e última técnica.

Capítulo 5. Após uma curta introdução a respeito de representações tempo × freqüência, é descrita a técnica final, que busca resolver a maior parte das dificuldades existentes nos métodos anteriores. Sua principal distinção é operar sobre representações tempo × freqüência ao invés da representação temporal pura.

Capítulo 6. Damos alguns exemplos de aplicações. Diversas conclusões são tiradas, dando origem a uma série de possíveis trabalhos futuros e aplicações úteis e interessantes.

Apêndice A. screve como é a arquitetura desenvolvida para a implementação responsável por parte dos exemplos gerados.

Capítulo 1

Conceitos Básicos de Som

Ao longo desta dissertação o leitor irá se familiarizar com o problema de metamorfose ("morph") de sons e com algumas técnicas aqui desenvolvidas para sua solução.

Os diferentes métodos para solução do problema apresentam características distintas e, conforme forem sendo descritos, suas ações e intenções serão devidamente posicionadas nos quatro universos de abstração.

É necessário no entanto fazer uma breve preleção sobre a caracterização dos sons como objetos gráficos, suas modelagens e abstrações.

1.1 Som e os Quatro Universos de Abstração

Utilizamos para caracterização dos sons a formulação dos quatro universos, ilustrado na Figura 1.1, (Gomes & Velho, 1995a), onde o universo físico contém os fenômenos propriamente ditos, o universo matemático abrange a modelagem matemática, o universo de representação se encarrega de discretizar o modelo de forma que ele seja computacionalmente tratável e, finalmente, o universo de codificação, que se concentra na parte de organização dos dados propriamente dita no computador, tal como as estruturas de dados.

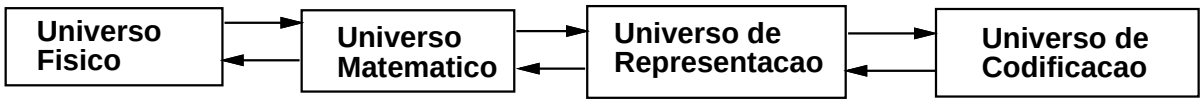


Figura 1.1: Os quatro universos de abstração.

1.1.1 O Fenômeno Físico

O som é uma forma de energia caracterizada pela mudança de pressão, de natureza oscilatória, que se propaga em um meio físico, podendo ele ser gasoso, líquido, ou sólido.

Os seres vivos possuem diferentes formas de geração e percepção deste fenômeno. No caso da produção de sons, por exemplo, os grilos e os louva-a-deus esfregam os extremos de suas asas, fazendo-as vibrar, enquanto os anfíbios possuem uma laringe, dotada de

cordas vocais sibilantes. A utilização dos sons pode ter toda uma gama de aplicações, inclusive como parte importante de mecanismos de reprodução.

Os seres humanos (assim como algumas outras espécies de mamíferos) possuem um mecanismo mais completo, já que utiliza os sons por eles gerados como principal método de comunicação e interação com outros indivíduos. Mais detalhes sobre este assunto podem ser encontrados em (O'Shaughnessy, 1987) e (Rowden, 1991).

O processo de percepção e interpretação de sons como estímulos externos possui um paralelo também válido: diferentes animais percebem de forma diferente os sons. Um exemplo clássico são os apitos para cachorros, inaudíveis para o ser humano.

Já mais próximo do problema de modelagem, pode-se verificar dois tipos de características básicas em um som: a "intensidade" e o o "timbre". Pense em um instrumento musical que pode emitir a mesma nota musical levemente ou com mais força. Pode emitir também duas notas diferentes com a mesma "força".

1.1.2 O Modelo Matemático do Som

Buscamos agora um modelo matemático que seja capaz de descrever o fenômeno já descrito. Uma forma natural de modelá-lo seria através do mapeamento das intensidades da pressão do meio físico de transporte do som ao longo do tempo.

Sendo assim o som pode ser analisado como uma função, obtendo todo o ferramental matemático conhecido como *análise funcional* para este modelo.

Um som é então representado como uma função $f: U \subset \mathbb{R} \to \mathbb{R}$, onde U é o *suporte*, a região do tempo onde ele está definido. Para cada $t \in U$, o número $f(t)$ representa uma medida da intesidade de pressão no meio físico de propagação.

Como já foi comentado, este fenômeno tem uma característica oscilatória, e é justamente esta característica que diferencia o também já mencionado "timbre" de duas notas musicais distintas. O "timbre" está diretamente ligado ao número de vezes que a onda se repete a cada segundo, e recebe o nome de *freqüência*.

Uma vez caracterizado o som como uma função, temos à nossa disposição todo o ferramedntal matemático de análise funcional. Do ponto de vista de engenharia estamos caracterizando o som como um "sinal unidimensional". Todas as técnicas matemáticas de Análise Funcional podem portanto ser utilizadas.

Relação entre os Universos Matemático e Físico

Do mesmo modo que cor, o som tem uma natureza perceptual. Isso significa que a relação entre o universo físico e o universo matemático deve levar em consideração propriedades de percepção do som por uma fonte receptora. O caso mais importante é a percepção humana do som. Esses estudos fazem parte de Psico-Acústica. Uma boa referência introdutória sobre a relação entre os aspectos físicos e psicofísicos é (O'Shaughnessy, 1987).

Em particular, a relação entre o universo físico e matemático abre uma área fértil de pesquisa sobre a modelagem matemática do sistema auditivo humano (Souza & Caloba, 1995), (Souza & Caloba, n.d.b), (Souza & Caloba, n.d.a).

Existe ainda uma certa distinção entre o que é, e não é, percebido por diferentes indivíduos. Além disto um mesmo indivíduo vai perdendo acuidade sonora com o passar dos anos.

Som e Imagem. Do ponto de vista do universo matemático existe uma grande semelhança entre som e imagem: o som é um sinal unidimensional enquanto a imagem é um sinal bidimensional. O problema de representação de som é semelhante ao problema de representação de imagem: representação de um sinal. No entanto, do ponto de vista do universo físico, som e imagem são de natureza distinta, tanto do ponto de vista físico como de percepção. Deste modo, deve-se ter bastante cuidado quando se busca analogias entre som e imagem.

O leitor pode desejar fazer um paralelo entre sons e trechos de filme, animações, mas ainda assim é difícil, uma vez que um pedaço estático do filme é uma imagem, que possui significado e interpretação; o som, por sua vez, quando estático no tempo não traz informação nenhuma.

1.1.3 Representação Discreta do Som

Uma função do tipo $f\colon \mathbb{R} \to \mathbb{R}$ é naturalmente não-discreta, uma vez que seu domínio pode ser interpretado como todos os pontos que compõe uma reta. Como o contra-domínio também é $\mathbb{R}$, esta interpretação também é válida para ele. Há então uma dificuldade inerente em representar f de forma finita, isto é, através da discretização tanto do domínio como também do contra-domínio da função.

Um dos processos mais comuns de discretização no domínio da função é conhecido pelo nome de *amostragem*, vastamente estudado na literatura (veja (Oppenhein & Willsky, 1983), (Lathi, 1974), (Antoniou, 1993)). O processo de amostragem pode ser realizado de diversas formas, porém, geralmente é feito de maneira uniforme. É chamado de *amostragem pontual*, quando sua n-ésima amostra f_n é obtida por

$$f_n = f(nT), \quad T > 0$$

onde T é o *período de amostragem* (intervalo entre duas amostras seguidas). Na amostragem uniforme, o inverso do período, $F_s = \frac{1}{T}$, é chamado de *freqüência ou taxa de amostragem*.

A freqüência de amostragem é então o número de vezes que o processo de amostragem é realizado em cada unidade de tempo. A unidade mais comum de medida de freqüência em engenharia e ciências aplicadas é o Hertz (Hz), que mede o número de ocorrências, oscilações, de um fenômeno, por segundo. Portanto, dizer que a função discreta possui uma taxa de amostragem com 10.000 Hz, por exemplo, significa dizer que para cada segundo foram realizadas 10.000 operações de amostragem.

Um exemplo concreto disto pode ser observado nos "compact disk players" (CDs de música). Lá o som é armazenado sob a forma discreta, e a taxa de amostragem é de 44.200 Hz, ou seja, para cada segundo de música há 44.200 amostras distintas.

Segundo o teorema de Shanon, sabe-se que uma função pode ser bem representada apenas pelas suas amostras se estas forem feitas suficientemente próximas. É de certa forma claro que este termo "suficientemente próximas" dependerá de cada sinal, uma vez que é importante que não haja grandes variações da função entre duas amostras consecutivas. Mais exatamente, a freqüencia de amostragem deve ser maior ou igual do que duas vezes a freqüência da maior oscilação do sinal a ser representado (isto se chama *Taxa de Nyquist*).

Continuando com o mesmo exemplo, o ouvido humano é capaz de escutar sons da faixa de 20Hz a 20.000Hz, isto então explica o porquê da taxa de amostragem do CD-Player ser da ordem de 40.000Hz.

Infelizmente a reconstrução ideal descrita por Shannon é muito custosa, portanto a reconstrução de uma amostragem pontual pode perder detalhes de altas freqüências do sinal, isto é, variações muito bruscas entre amostras consecutivas, mesmo se respeitado o a taxa de Nyquist. Desta forma, para evitar problemas de alias e de reconstrução é comum utilizar algum tipo de filtragem de passa baixa antes de proceder com o processo de amostragem. Uma técnica símples de realizar esta filtragem é através de uma *amostragem por área*. Neste caso a amostra f_n é obtida através de

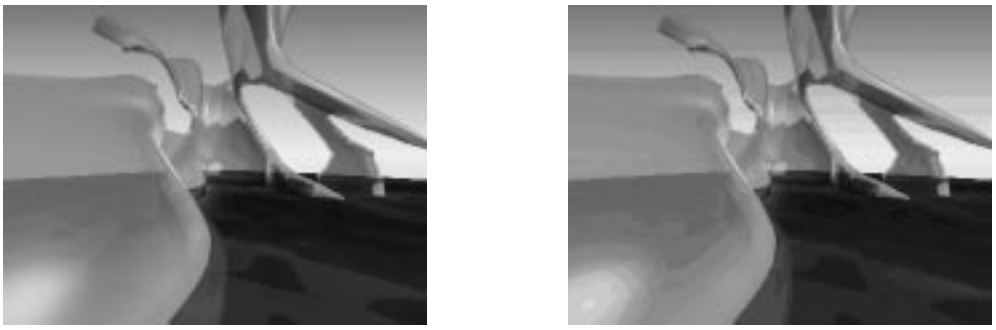
$$f_n = \frac{1}{T}\int_{nT}^{(n+1)T} f(t)dt.$$

Desta forma procura-se fazer com que o valor dependa do sinal durante todo o período, e não apenas de um momento único. Note que neste caso, que também é uniforme, ainda existem T e F_s

Quantização

O processo de amostragem discretiza o domínio; porém os valores da função também precisam de um processo similar. A passagem da imagem para o domínio discreto recebe o nome de *quantização*.

Em imagens o processo de quantização introduz franjas, fronteiras, chamadas de "countouring" (veja Figura 1.2). Em sons surge então a questão de qual seria o efeito resultante deste processo.



(a) (b)

Figura 1.2: Imagem original (a) e quantizada (b).

Perceptualmente, o efeito sonoro é de que um "chiado" é introduzido, como o som que se escuta quando uma televisão não está sintonizada em nenhum canal (ruído branco).

Em (Antoniou, 1993), estuda-se com bastante atenção o efeito da quantização sobre o comportamento do sinal. Para isso, utiliza-se a abordagem, comum em engenharia, de interpretar estas alterações como ruídos, empregando, para isso, um certo ferramental estatístico.

Existem diversas técnicas distintas de quantização, indo desde o método uniforme mais simples e chegando até mesmo a processos adaptativos. Quase todos, no entanto, partem de um princípio comum: a utilização uma segmentação do intervalo de interesse, associando a cada segmento (também chamado de célula de quantização) um valor quantizado.

Um sinal que possui tanto seu domínio, como sua imagens discretizados é chamado de *sinal digital*. Um bom exemplo de sinal digital é o sinal de um CD, que é amostrado com uma freqüência de 44.200 Hz e possui uma quantização uniforme de 16 bits.

Relação entre os Universos de Representação e Matemático

Para diversos tipos de aplicações é necessário o modelo matemático do sinal, definido sobre um intervalo de $\mathbb{R}$, portanto com infinitos pontos. Se o sinal está armazenado de forma digital, será então necessário "retornar" a sua configuração original, antes do processo de amostragem. A este processo denominamos *reconstrução*.

Chamamos de *reconstrução ideal* quando a reconstrução obtida no universo matemático é igual ao sinal original anterior a amostragem. Desta forma a "passagem" pelo universo de discretização não alterou o sinal desejado.

Vamos nos ater ao caso da amostragem pontual onde sabe-se que é teoricamente possivel realizar uma reconstrução ideal a partir das amostras. Para isso a amostragem deve ter sido feita com uma freqüência ao menos duas vezes maior que a maior freqüência existente no sinal em questão. Este resultado é o Teorema de Shannon. Para detalhes o leitor deve consultar (Oppenhein & Willsky, 1983).

1.1.4 Implementação de Som

A forma de implementar o armazenamento e processamento em hardware e software é bem variada. Diferentes padrões e conceitos são utilizados em cada plataforma, tornando qualquer tentativa de padronização uma tarefa complicada.

Um sinal de som digital é representado perfeitamente por um *vetor*, esta sendo então a estrutura de dados mais simples e imediata para sua implementação. No processo de codificação de vetores é necessário levar em conta diversos fatores, como, por exemplo, o espaço a ser ocupado (*compressão*), a alteração que o processo de codificação irá causar perceptualmente (*ruído de quantização*), etc.

Diversos tipos de dados podem ser utilizados para codificar os valores de cada um dos elementos do vetor, cada tipo de dado geralmente reflete um método distinto de quantização, possuindo assim aplicações mais e menos adequadas. Alguns deste tipos são:

- Inteiro positivo de n bits. Os valores representados variam então entre 0 e $2^n - 1$.
- Complemento a um com n bits. Os valores representados variam como no caso anterior com $n - 1$ bits, e o bit adicional indica o sinal. Desta forma uma amostra pode possuir um valor entre $-(2^{n-1} - 1)$ a $2^{n-1} - 1$.
- Complemento a dois com n bits. Os valores variam entre -2^{n-1} a $2^{n-1} - 1$.
- A-Law e μ-Law. Ao contrário dos exemplos anteriores, este não provém de uma quantização uniforme. Primeiro é aplicada uma função de "compansão" no sinal

para só então quantizá-lo. Esta função possui um comportamento exponencial. A idéia é utilizar um número menor de bits do que a quantização uniforme precisaria para obter uma dada relação sinal-ruído causada pela quantização. Valores pequenos possuem mais subdivisões do que os grandes.

- Ponto Flutuante. Utiliza o sistema de mantissa e expoente para representar o valor. Tenta resolver de uma forma diferente o mesmo problema do item anterior. Não necessita teoricamente de limites inferior e superior como nos outros casos.

O vetor pode também ser representado de outras maneiras, geralmente na busca por uma codificação mais compacta. Para isso podem ser utilizadas quantizações adaptativas, com compressão, etc. Um típico exemplo é o processo de codificação ADPCM ("Adaptative Diference Pulse Code Modulation"), onde,f para cada amostra, armazena-se apenas a diferença entre o ser valor real e a previsão do que "deveria" ser baseado nas amostras anteriores.

Outro aspecto interessante, com abordagem distinta, é a codificação de sinais baseado em propriedades de produção e percepção físicas do som. Dois exemplos deste tipo de codificação são o MIDI e o método de codificação por VOCODER.

Codificações MIDI representam bem apenas seqüências musicais, pois guardam informações a respeito de notas musicais, associando a cada uma delas um instante de início, duração, intensidade, instrumento e outros atributos próprios.

Codificadores derivados da idéia básica do VOCODER funcionam apenas para sinais de voz, pois procuram parametrizar diversas as características do aparelho vocal humano para cada instante de tempo, transmitindo-as. O receptor deve então a partir destes parâmetros sintetizar novamente o som, fazendo uma espécie de simulação.

Como vimos então, estes dois exemplos de codificações acima citados, são adequados apenas a sinais que possuam certas propriedades específicas. Desta forma possuem um alcance e utilização um pouco mais restrito.

Para mais detalhes a respeito de métodos e princípios de codificação de sinais de audio veja em (Rabiner & Schafer, 1978), (O'Shaughnessy, 1987) e (Roads, 1996).

1.2 Objetos Gráficos

O conceito de objeto gráfico busca unificar de forma genérica as diversas entidades existentes em computação gráfica. É necessário que o "modelo" utilizado seja suficientemente flexível de forma que seja válido e facilmente adaptável para toda a gama de aplicações que existem.

Definição 1. Um *objeto gráfico* $\mathcal{O}$ de um espaço euclidiano $\mathbb{R}^n$ consiste em um subconjunto $U \subset \mathbb{R}^n$ e uma função $f\colon U \to \mathbb{R}^p$. (Veja (Gomes *et al.* , 1996))

O conjunto U define a *forma*, ou mais precisamente o *suporte geométrico* do objeto. A função f define os seus atributos, portanto é chamada de *função de atributos*. Utilizaremos a notação $\mathcal{O}(U, f)$ sempre que for necessário enfatizar tanto o suporte geométrico U quanto a função de atributos f do objeto gráfico $\mathcal{O}$.

Na Figura 1.3 possuimos três exemplos ilustrativos. Em (a) temos uma imagem, caso clássico onde o suporte geométrico é simples, porém que possui uma função de atributo bastante complexa. Em (b) possuimos um modelo geométrico puro, a função de atributo

é simples (cor constante) e o suporte geométrico possui uma topologia mais complexa. Finalmente em (c) podemos observar um objeto que possui tanto suporte geométrico como função de atributos complexos, uma superfície com textura.

(a) (b) (c)

Figura 1.3: Três objetos gráficos distintos.

1.2.1 Som como objeto gráfico

Observando o modelo matemático introduzido na seção 1.1.2 para descrever um som,

$$S = f : U \subseteq \mathbb{R} \to \mathbb{R},$$

podemos claramente identificar U como sendo o suporte geométrico, e a amplitude como o atributo, logo a função f é a função de atributo. O sinal de áudio é representado então por um objeto gráfico cuja forma é simples (intervalo de reta) e cuja função de atributo é em geral extremamente complexa.

1.3 Transformações de Objetos Gráficos

Durante a manipulação de objetos gráficos, pode ser interessante executar determinadas transformações nestes objetos, modificá-los, deformá-los, etc. Transformar um objeto gráfico significa transformar seu suporte geométrico e transformar seus atributos. Duas importantes classes de transformações de objetos gráficos são quando há, de forma independente:

1- Transformações de suporte geométrico;

2- Transformações de atributos.

Utilizamos os seguintes termos:

Deformação. É um sinônimo para uma transformação qualquer;

Warp. Seqüência contínua de transformações de um objeto;

Morph. É um warp entre um objeto $\mathcal{O}_1$ e um objeto $\mathcal{O}_2$ previamente dados.

Matematicamente, um warp pode ser caraterizado usando famílias de transformações. Diz-se que T_v define uma família de transformações com parâmetro $v \in \mathbb{R}^k$, onde $\mathbb{R}^k$ é o espaço de parâmetros, se para cada v distinto fica bem definido $\mathcal{O}'_v = T_v(\mathcal{O})$. Desta forma, podemos imaginar a transformação T como sendo

$$T\colon \Omega \times \mathbb{R}^k \to \Omega,$$

onde Ω representa o espaço de objetos gráficos

$$\Omega = \{f\colon U \subseteq \mathbb{R}^n \to \mathbb{R}^p\}.$$

e $\mathbb{R}^k$ o espaço de parâmetros.

Se agora criarmos uma curva c no espaço de parâmetros do tipo

$$c\colon [0,1] \to \mathbb{R}^k,$$

então podemos reduzir nosso espaço de parâmetros. Desta forma podemos descrever uma família de transformações com espaço de parâmetros $\mathbb{R}^k$ através de um novo parâmetro $\alpha \in [0,1]$, usando a curva c. Diferentes curvas dão origem a diferentes famílias de transformações.

1.3.1 Transformação de Suporte Geométrico

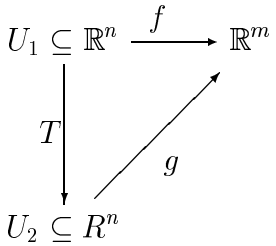
Transformações de Suporte Geométrico atuam somente sob o domínio. Dado $\mathcal{O}_1 = (U_1, f)$, uma transformação T de suporte geométrico transforma $\mathcal{O}_1$ no objeto gráfico $\mathcal{O}_2 = (T(U_1), g)$, onde a função de atributos g é definida do seguinte modo:

Dado um ponto $q \in U_1$, a transformação fará com que ele seja representado como $p \in U_2$, onde $p = W(q)$. Desta forma podemos dizer que

$$g(T(q)) = f(q), \text{isto é,} g \circ T = f.$$

Se T for invertível, então $g = f \circ T^{-1}$.

Graficamente podemos observar:



Na Figura 1.4 podemos observar um exemplo de uma transformação de suporte geométrico atuando sob uma imagem.

1.3.2 Transformação de Atributos

São chamadas de *Transformações de Atributos* aquelas que atuam sobre os valores da função de atributo. Sendo dois objetos gráficos $\mathcal{O}_1 \in \Omega$ e $\mathcal{O}_2 \in \Omega'$, onde

$$\Omega' = \{f\colon U \subseteq \mathbb{R}^n \to \mathbb{R}^p\}.$$

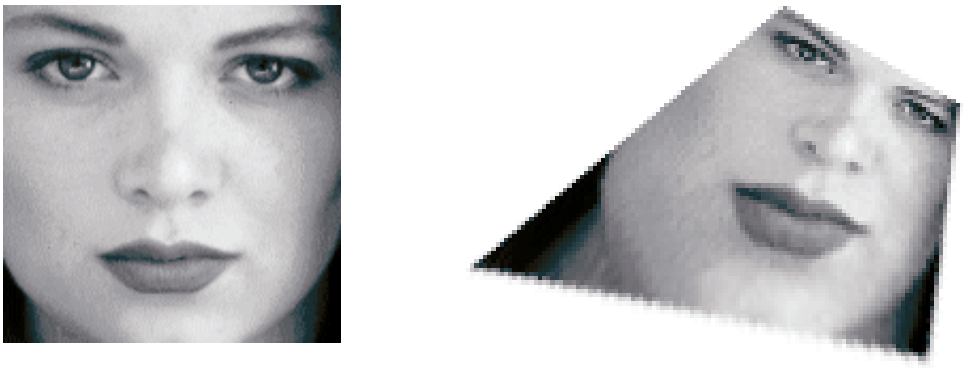


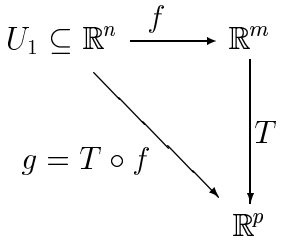
Figura 1.4: Transformação projetiva de uma imagem: transformação de suporte geométrico. Figura extraída de (Gomes *et al.* , 1997a)

Dizemos que T é uma transformação de atributo se

$$\mathcal{O}_2 = T(\mathcal{O}_1),$$

onde $\mathcal{O}_1 = f\colon U_1 \subset \mathbb{R}^n \to \mathbb{R}^m$ e $\mathcal{O}_2 = g\colon U_1 \subset \mathbb{R}^n \to \mathbb{R}^p$.

A transformação de atributos é então descrita por uma função $T\colon \mathbb{R}^m \to \mathbb{R}^p$. Desta forma $g = T(f) = T \circ f$ fica corretamente ilustrada no diagrama abaixo:



Exemplo 1 ("Cross-Dissolve"). Este é o exemplo mais simples de uma transformação de atributos. Uma transformação de *Cross-Dissolve* age geralmente sobre dois objetos gráficos, $\mathcal{O}_1 = (U, f)$ e $\mathcal{O}_2 = (U, g)$, com o mesmo suporte geométrico U, ou seja, $f, g\colon U \to \mathbb{R}^m$. Ela define uma família de novos objeto, também definido sobre U, parametrizados por $\lambda \in [0, 1]$. Definimos então a transformação de *Cross-Dissolve* C como

$$C\colon U \times [0, 1] \to \mathbb{R}^n.$$

Geralmente o termo "cross-dissolve" é naturalmente associado ao caso de *cross-dissolve linear*, quando

$$C(p, \lambda) = (1 - \lambda)f(p) + \lambda g(p), \quad p \in U, \lambda \in [0, 1].$$

A transformação de "cross-dissolve" genérica não precisa ser obtida através de uma função afim, mas mesmo assim busca realizar uma transição contínua em λ entre f e g. Desta forma, é necessário cumprir a certas condições fundamentais:

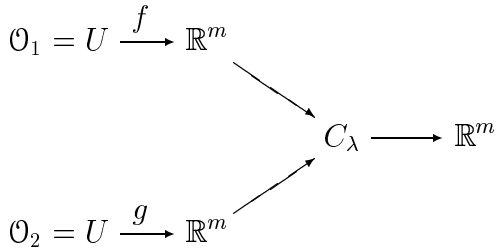
$$C_\lambda(p) = C(p, \lambda) = f(p)h_0(p, \lambda) + g(p), h_1(p, \lambda)$$

onde

$$\begin{aligned} &h_0(p,0) = 1; \\ &h_1(p,0) = 0; \\ &h_0(p,1) = 0; \\ &h_1(p,1) = 1 \quad \text{e} \\ &h_0(p,\lambda) + h_1(p,\lambda) = 1, \end{aligned} \tag{1.1}$$

ou seja, os atributos de cada objeto de seqüência são obtidos pela média ponderada de cada um dos atributos.

Podemos ilustrar a operação pelo diagrama abaixo:



Como a soma de h_0 e h_1 é sempre 1, basta termos um definido para a obtenção do outro.

Quando o cross-dissolve não é linear ele é chamado de *cross-dissolve adaptativo*.

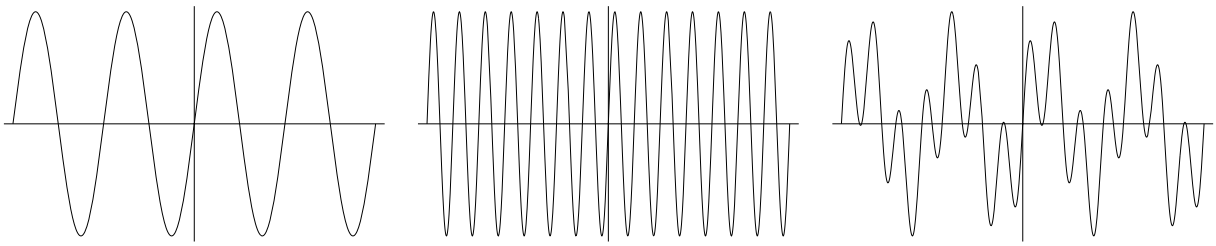
Na Figura 1.5 (c) observamos o efeito do cross-dissove linear entre duas imagens, em (a) e (b), para $\lambda = 0.5$.

Figura 1.5: "Cross-Dissolve" de duas imagens.

No caso de áudio, o cross-dissolve de dois sons é a operação conhecida como mixagem. A Figura 1.6 (c) mostra um cross-dissolve linear entre dois sons simples, duas senóides puras com freqüências diferentes, mostradas em (a) e (b), utilizando $\lambda = 0.5$.

1.3.3 "Morph"

Conforme discutimos anteriormente, a operação de metarmorfose φ entre dois objetos gráficos $\mathcal{O}_1, \mathcal{O}_2 \in \Omega$ deve efetuar uma transição contínua entre eles. Matematicamente esta transição deve ser parametrizada com um espaço de parâmetros $\mathbb{R}^k$. Para poder utilizar apenas um parâmetro escalar $\alpha \in [0,1]$, deve-se então, como já foi visto, escolher uma curva c no espaço $\mathbb{R}^k$.



(a) (b) (c)

Figura 1.6: "Cross-Dissolve" de dois sons.

Um modo intuitivo de realizar então o morph é fazer simultaneamente uma transformação do suporte geométrico e uma sobre os atributos.

A transformação do suporte geométrico busca "alinhar" as características geométricas principais de cada objeto. A transformação de atributo busca "interpolar" seus atributos. Geralmente para isso utiliza-se um "cross-dissolve".

Um exemplo desta "divisão" do processo em duas transformações distintas e simultâneas pode ser visto na Figura 1.7

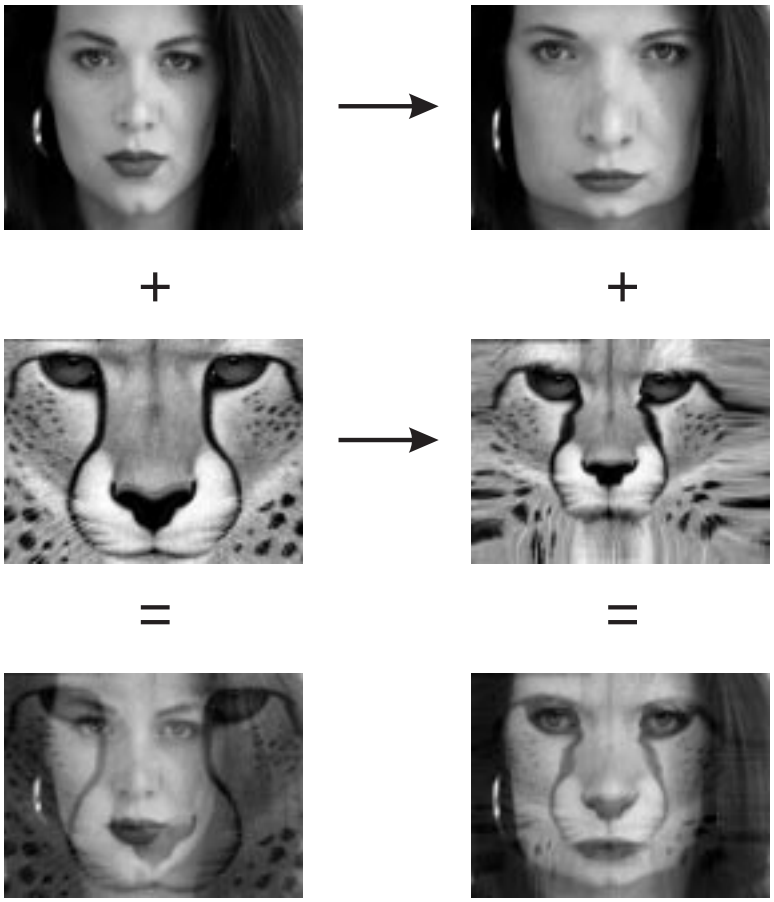


Figura 1.7: À esquerda o resultado de apenas um "Cross-Dissolve", a direita um processo completo de "Morph". Esta figura foi gentilmente cedida pelos autores de (Gomes *et al.* , 1997a)

Capítulo 2

Transformações de Áudio

No capítulo anterior falamos sobre transformações em objetos gráficos gerais, e foram dados até mesmo alguns exemplos com imagens. Ao longo das próximas seções iremos estudar como isso afeta e se enquadra a sons.

Primeiro será avaliado o significado das transformações de atributo, em seguida veremos o caso da transformação do domínio.

2.1 Transformação de Atributo

Como foi observado no capítulo anterior, o modelo matemático de um som é uma função do tipo $f\colon [a,b] \to \mathbb{R}$. Esta modelagem se adapta perfeitamente à teoria de objetos gráficos, sendo o intervalo $[a,b]$ o suporte geométrico e a própria função f, que determina a intensidade de pressão para cada tempo $t \in [a,b]$, a função de atributos.

No caso de sons, o atributo é apenas sua intensidade de pressão a cada instante, e é modelada matematicamente como um número real; logo o resultado da transformação de atributo também precisa ser real. Uma transformação de atributo T neste caso se resume a uma função

$$T\colon \mathbb{R} \to \mathbb{R}.$$

Perceptualmente isto equivale a alterar a amplitude do sinal.

Exemplo 1 (Transformação Linear). O caso linear onde $T(f(t)) = \alpha f(t)$ é utilizado por todos no dia a dia, é a função que todo amplificador de som busca realizar, onde o controle do volume é dado variando o parâmetro α.

Exemplo 2 (Transformação de Compansão). Outro exemplo bastante aplicado de transformação de atributo de sons é o processo de "compansão" (as vezes também chamado de compressão, termo que pode levar a interpretações enganosas). Em uma estação de rádio diversos locutores podem interagir durante uma mesma transmissão, no entanto, cada um possui naturalmente uma intensidade de voz, volume, diferente. Isto pode ser um problema muito sério: eventualmente um deles não será ouvido direito ou o outro ficará "gritando".

Para resolver isto automaticamente, sem a necessidade de um operador para manualmente regular o volume toda vez que houver uma troca de locutor, utiliza-se uma transformação de atributo onde T é uma função de *compansão*. Esta transformação T busca encolher, comprimir, a intensidade do áudio (atributo) e geralmete é uma função

logaritmica que, quando normalizamos tanto o domínio como a imagem para o intervalo $[0,1]$, é geralmente do tipo

$$g(x) = \frac{\log(1+\mu x)}{\log(1+\mu)},$$

onde μ controla o quanto será a deformação, observe que quando levamos, por limite, μ a zero recaimos no caso linear. Um exemplo típico com $\mu = 10$, 100 e 1000 é visto na Figura 2.1

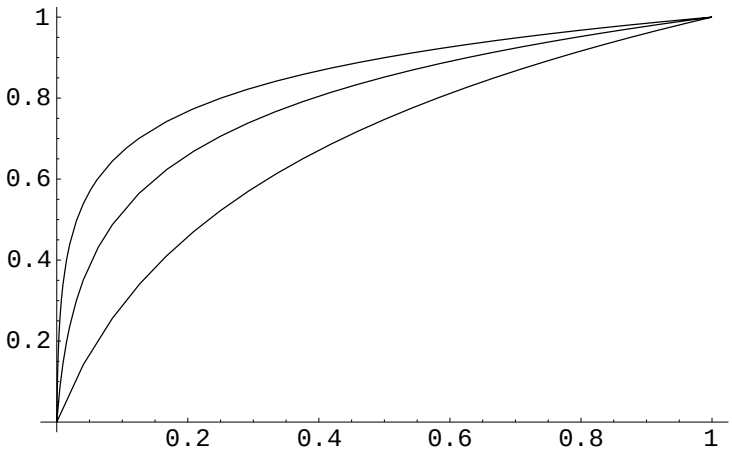


Figura 2.1: Típica função de compansão.

Existem aparelhos eletrônicos comercialmente disponíveis para executar tal objetivo e são chamados de "compressores". Em geral, dois métodos distintos são utilizados para implementar um *filtro não-linear* descrito por uma função como as da Figura 2.1. O primeiro é através da aproximação através de uma função linear por partes; o outro é através do uso da curva de resposta de um dispositivo eletrônico não linear, como uma junção de diodo, por exemplo.

2.1.1 Cross-Dissolve

Como vimos no capítulo anterior, a transformação de cross-dissolve define toda uma família de objetos a partir de dois objetos $\mathcal{O}_1$ e $\mathcal{O}_2$ originais e um parâmetro λ real. Vimos também que geralmente o cross-dissolve é realizado de forma linear, interpolando com o auxílio de λ (geralmente bem definido no intervalo $[0,1]$).

Dados dois sinais de áudio

$$f_1 \colon U \to \mathbb{R} \quad \text{e}$$
$$f_2 \colon U \to \mathbb{R},$$

para cada $\lambda \in [0,1]$ definimos o sinal $f_\lambda \colon U \to \mathbb{R}$ como

$$f_\lambda(t) = (1-\lambda) f_1(t) + \lambda f_2(t). \tag{2.1}$$

O processo assim definido é conhecido entre os profissionais de som e música como *mixagem*.

Cross-Dissolve Adaptativo

Diz-se que o processo de cross-dissolve é *adaptativo* quando a "ponderação" entre os atributos dos dois objetos gráficos depende não só de λ, mas também da localização espacial, que no caso do áudio é a variável t.

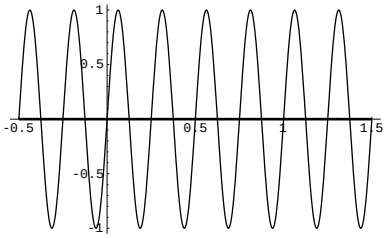
Desta forma a equação 2.1 é reescrita como

$$f_\lambda(t) = (1 - h(\lambda, t))f_1 + h(\lambda, t)f_2,$$

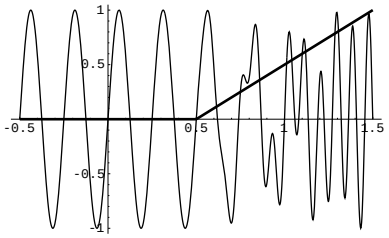
onde

$$h(0, t) = 0 \quad \text{e} \quad h(1, t) = 1.$$

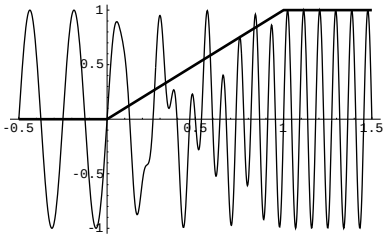
Um exemplo de um processo de cross-dissolve adaptativo pode ser visto na Figura 2.2. Em (a) e (e) vemos os objetos originais $\mathcal{O}_1$ e $\mathcal{O}_2$, assim como também a função $h(\lambda, t)$ para λ igual a zero e um. Em (b) (c) e (d) observamos o resultado do crossdissolve e a função $h(\lambda, t)$ para três diferentes valores intermediários de λ.



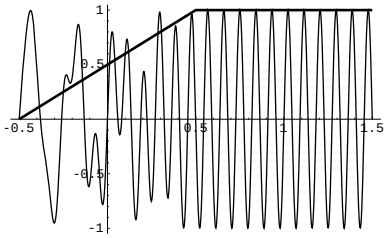
(a)



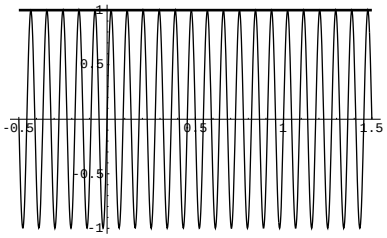
(b)



(c)



(d)



(e)

Figura 2.2: Cross-Dissolve adaptativo.

Uma interpretação física de um dos objetos gerados por esta família de transformações é o processo de transição feito entre músicas em uma discoteca através de uma simples mixagem (sem alterar as velocidades de rotação dos discos ou CDs).

2.2 Transformação de Suporte Geométrico

Em sinais sonoros o suporte geométrico é bastante simples: apenas um intervalo. Qualquer tipo de deformação deve então levar um intervalo original $[a, b]$ em um outro intervalo $[c, d]$. Sob uma ótica simples, então tudo o que se pode fazer é uma dilatação (ou contração) seguida de uma translação. Isto é ilustrado na Figura 2.3.

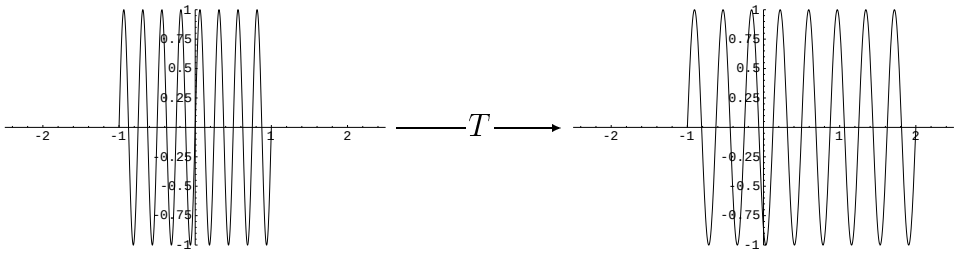


Figura 2.3: Transformação de suporte geométrico de um som.

Isto é simples de compreender no gráfico da função, mas é importante se ter em mente qual o resultado perceptual ao realizarmos este procedimento. Quando aumentamos o tamanho do suporte geométrico estamos na verdade expandindo a função no tempo. Ela, portanto, realiza as mesmas "oscilações" que antes, porém em um período maior. Por outro lado, quando diminuirmos o suporte, o número de oscilações permanece o mesmo, porém ocorrem em menos tempo.

Seguindo este raciocínio, podemos notar que há uma alteração nas freqüências do sinal, o que é muito claro se olharmos a propriedade de "escalamento do tempo" da transformada de Fourier:

$$x(at) \xleftrightarrow{\mathcal{F}} \frac{1}{a} X(\frac{\omega}{a}), \quad a \neq 0,$$

onde ω é a variável da freqüência.

O resultado perceptual é muito distinto do que ocorre com a extensão do suporte geométrico, o domínio de uma imagem, embora ambos objetos sejam sinais, um unidimensional (o som) e o outro bidimensional (a imagem).

Caracterizamos a transformação por *expansão* quando

$$|f(x_1) - f(x_2)| > \lambda |x_1 - x_2|, \quad \lambda \geq 1,$$

e por *contração* quando

$$|f(x_1) - f(x_2)| < \lambda |x_1 - x_2|, \quad \lambda \leq 1.$$

No caso geral podemos ter uma transformação não linear de domínio. Essas transformações, conforme o exemplo ilustrado na Figura 2.4, fazem uma expansão em algumas regiões e uma contração em outras.

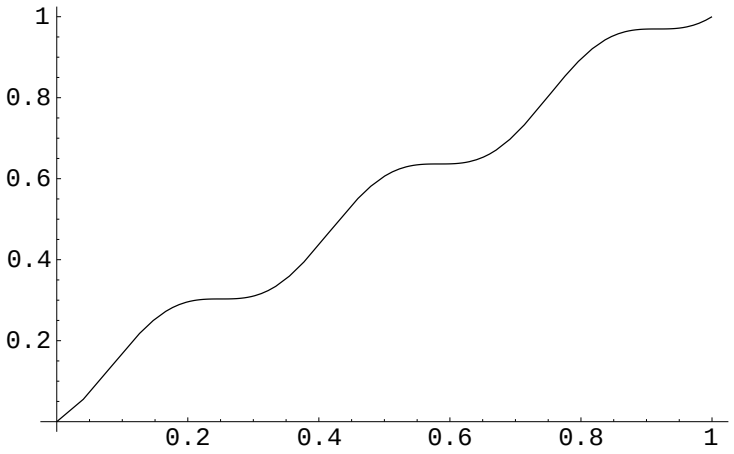


Figura 2.4: Função não linear para transformação de domínio.

Nas regiões onde a derivada é maior que 1 temos uma expansão local, onde a derivada é menor que 1 temos uma compressão. Este fato segue do teorema do valor médio do cálculo.

Em uma imagem, a expansão de domínio possui a característica de "zoom in", como se estivessemos nos aproximando, e a compressão dele o efeito de "zoom out", como se estivessemos nos afastando.

Já no caso de sons, este processo de distorção do suporte geométrico acarretará em um efeito similar a alteração da rotação dos discos de vinil antigo. A expansão do domínio diminui as freqüências e o som tem uma maior duração temporal. Perceptualmente causa a sensação que o disco esta andando mais devagar, com todos os sons surgindo mais grave e lentamente. Já compressão do domínio aumenta a freqüência e diminui a duração temporal do som, Perceptualmente isto aumenta a velocidade dos acontecimentos como também torna o som mais agudo.

Esta diferença tão grande vem do fato de o domínio no caso de sons ser o tempo, e da sua percepção ser de carater oscilatório.

Capítulo 3

Metamorfose por Warp do Domínio

Dados dois sons $\mathcal{O}_1([a,b], f_1)$ e $\mathcal{O}_2([c,d], f_2)$, uma transformação de morph entre eles define uma família de sons que descreve uma "transição" contínua de um ao outro. Podemos supor que essa família é parametrizada no intervalo $[0,1]$, tomando uma curva no espaço de parâmetros.

Se a operação de morph é definida por

$$T\colon \Omega \times [0,1] \to \Omega,$$

então $T(\mathcal{O}_1, 0) = \mathcal{O}_1$ e $T(\mathcal{O}_1, 1) = \mathcal{O}_2$.

Operações de morph entre objetos gráficos são descritas de diversas formas na literatura ((Wolberg, 1990), (Gomes *et al.* , 1995), (Gomes & Velho, 1997) e (Gomes *et al.* , 1997a)). Neste trabalho seguimos toda a conceituação de (Gomes *et al.* , 1997a), onde o *morph* é um caso particular de *warp*, que continuamente leva de um som a outro.

Como foi dito no capítulo anterior, o morph pode ser realizado através da combinação de uma família de tranformações de suporte geométrico, de forma a "alinhar" as características de cada domínio, combinada com um cross-dissolve.

A técnica de morphing descrita neste capítulo cria duas famílias de transformações de suporte geométrico, uma para cada objeto, e depois combina as intensidades utilizando o processo de cross-dissolve.

3.1 Alinhamento do Suporte Geométrico

Desejamos criar duas famílias de transformações de suporte geométrico T_1 e T_2, parametrizadas segundo o mesmo $\lambda \in [0,1]$. É imprescindível que estas duas famílias de transformações sejam capazes de manter os domínios alinhados para todo e qualquer valor de λ, já que é este o seu objetivo. Desta forma, para qualquer $\lambda \in [0,1]$, temos

$$T_{1,\lambda}([a,b]) = T_{2,\lambda}([c,d]).$$

Temos ainda que

$$T_{1,0}([a,b]) = [a,b] \qquad T_{2,0}([c,d]) = [a,b]$$
$$T_{1,1}([a,b]) = [c,d] \qquad T_{2,1}([c,d]) = [c,d].$$

Um modo simples de obter as famílias de transformações T_1 e T_2 é usar transformações afins. Considere dois sons $f_1 : [a, b] \to \mathbb{R}$ e $f_2 : [c, d] \to \mathbb{R}$. Definindo

$$e(\lambda) = (1 - \lambda)a + \lambda c \quad \text{e} \tag{3.1}$$
$$f(\lambda) = (1 - \lambda)b + \lambda d, \tag{3.2}$$

obtemos uma família de intervalos $I_\lambda = [e(\lambda), f(\lambda)]$ que se inicia em $[a, b]$ ($\lambda = 0$) e termina em $[c, d]$ ($\lambda = 1$), ilustrada na Figura 3.1. Para cada intervalo I_λ, temos uma transformação afim $T_{1,\lambda} : [a, b] \to I_\lambda$ definida por

$$T_{1,\lambda}(t) = e(\lambda) + \frac{t - a}{b - a}(f(\lambda) - e(\lambda)), \tag{3.3}$$

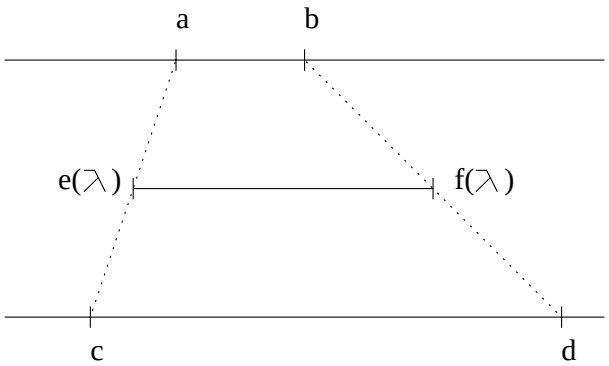


Figura 3.1: Família de intervalos I_λ.

Analogamente, definimos a tranfomação $T_{2,\lambda} : [c, d] \to I_\lambda$. Na realidade temos que

$$T_{2,\lambda} = T_{1,1-\lambda}.$$

Como foi visto no Capítulo 2, este é um caso muito particular, podendo esta operação ser realizada de forma não linear, contraindo o domínio em certas regiões e expandindo-o em outras, como foi descrito na Seção 2.2.

3.2 Cross-Dissolve e Morph

Após f_1 e f_2 sofrerem as transformações de suporte geométrico, os domínios dos novos objetos estarão alinhados, daí o morph é realizado aplicando a transformação de cross-dissolve linear descrita nas seções 1 e 2.1.1. Vejamos um exemplo.

Vamos supor dois sons originais $\mathcal{O}_1 = ([-2, 4], f_1)$ e $\mathcal{O}_2 = ([-3, 1], f_2)$ onde

$$f_1(t) = \text{sen}(2\pi t) \quad \text{e} \quad f_2(t) = \text{sen}(\pi(t + 1)).$$

Com a combinação das duas etapas de transformações simples aqui descritas, podemos descrever $\mathcal{O}_{3,\lambda}(U, g)$ como a família de sons resultante do processo de morph, onde $U = [e, f]$ conforme 3.1, e

$$g(t) = (1 - \lambda)f_1(T_{1,\lambda}^{-1}(t)) + \lambda f_2(T_{2,\lambda}^{-1}(t)) \tag{3.4}$$

Na Figura 3.2 podemos ver as diversas etapas. Em (a) e (b) vemos os sons originais $\mathcal{O}_1$ e $\mathcal{O}_2$, em (c) e (d) vemos os dois após a transformação de suporte geométrico para o valor de $\lambda = 0.5$. Finalmente em (e) encontramos o resultado final, após o cross-dissolve entre (c) e (d), também utilizando o mesmo valor de λ.

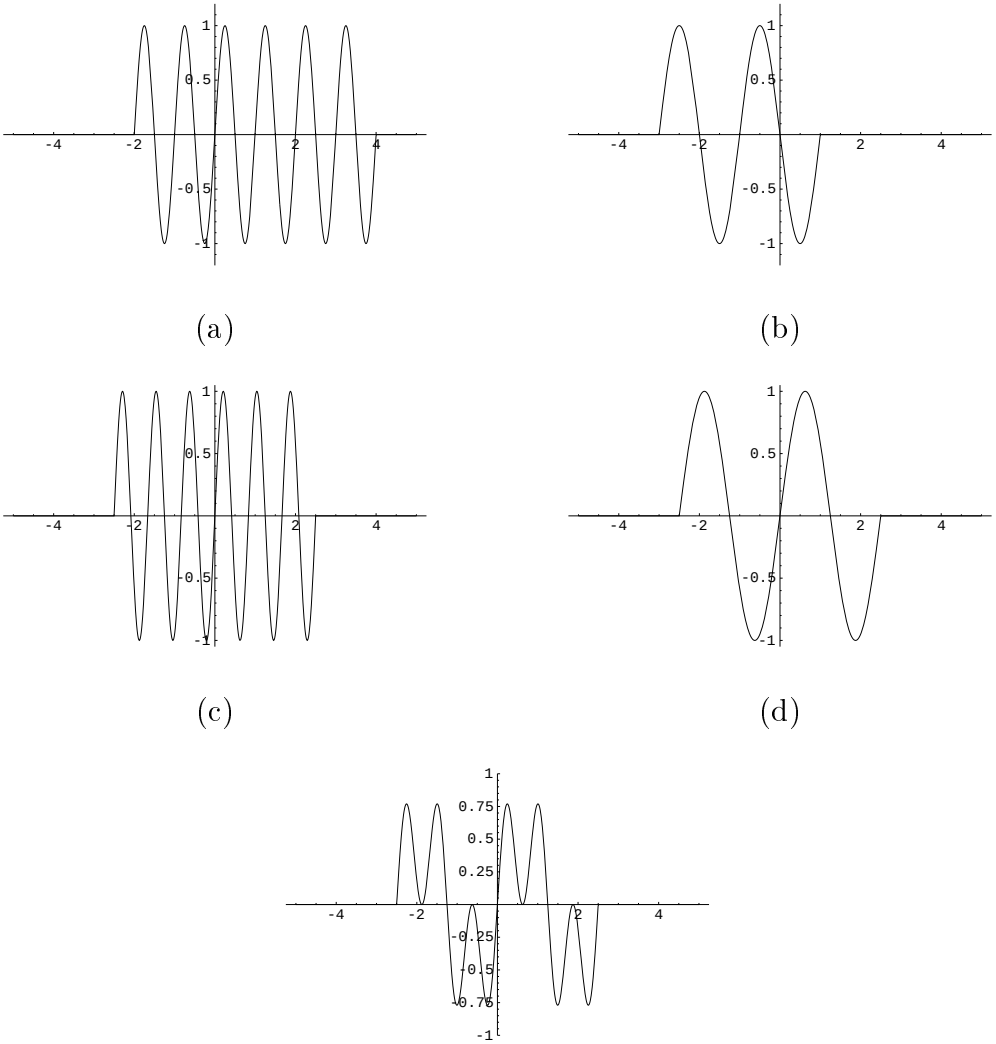


Figura 3.2: Exemplo simples da operação de morph no universo matemático.

Para ilustrar completamente o processo, discretizamos uniformemente o parâmetro λ em nove etapas, os sons resultantes estão ilustrados na Figura 3.3.

Note que,neste método, o parâmetro é sempre o mesmo para a transformação de suporte geométrico e cross-dissolve. Uma abordagem mais genérica não faria isto, deixando o espaço de parâmetros como um subconjunto $V \subseteq \mathbb{R}^2$. Seria necessário então criar uma curva $c\colon [0,1] \to V$, como foi anteriormente descrito.

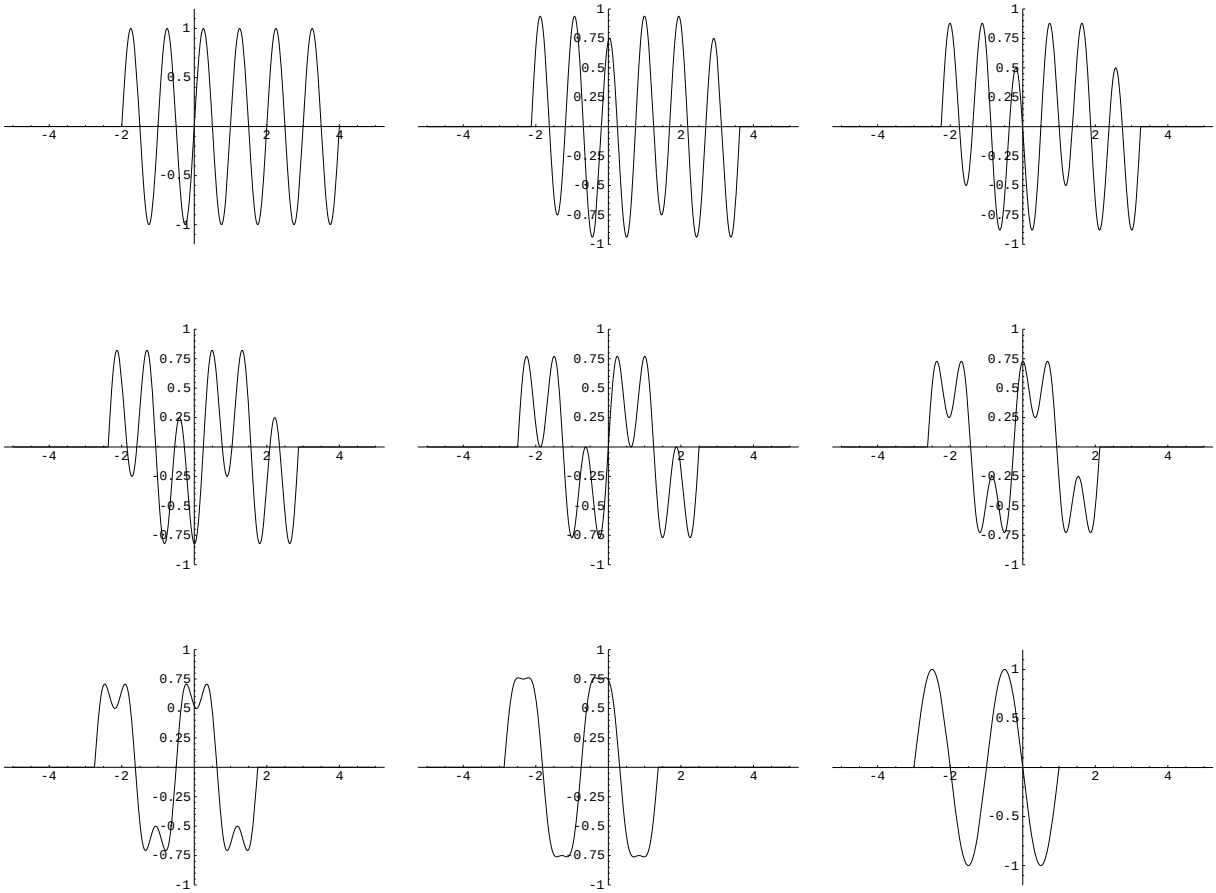


Figura 3.3: Família de transformações de morph através da discretização do parâmtetro k.

3.3 Operação no Universo de Representação

Uma vez que determinamos de forma clara e concisa a transformação de morph no universo matemático, onde tanto o suporte geométrico dos sons quanto a imagem das suas funções de atributos são naturalmente definidos ao longo de uma infinidade de valores, é necessário agora e atender um pouco o processo, possibilitando operar sobre versões discretas de objetos gráficos, posicionados no universo de representação.

Nada impede que a imagem da função de atributos, que neste caso particular de som é apenas um escalar, continue sendo representada sob uma forma "contínua"; basta para isso representar o valor de cada "amostra" através de um número em ponto flutuante (mas isto é uma questão de implementação, sendo realmente relevante para questões de codificação).

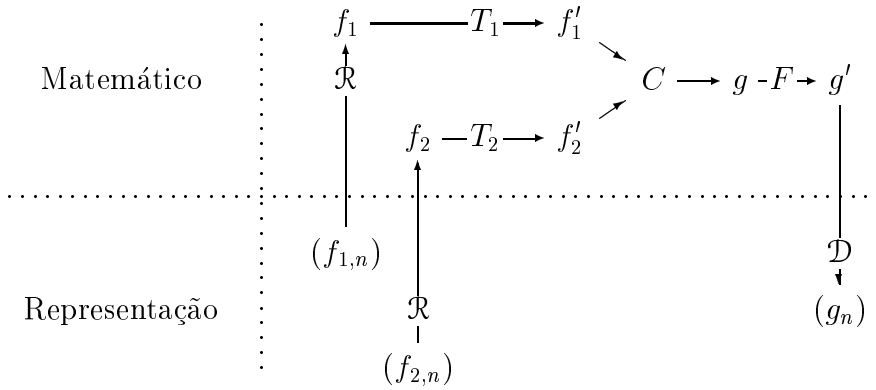
O suporte geométrico no entanto necessita ser discretizado, e, para isso, geralmente é realizada uma amostragem pontual com período de amostragem T.

A função passa a ser representada por uma seqüência; portanto (no caso em que a amostragem é pontual) $f_{1,n} = f_1(nT)$ e $f_{2,n} = f_2(nT)$. Um intervalo $[a, b]$ passa a ser então representado por um conjunto finito, cujo tamanho dependerá unicamente do período de

amostragem T.

Como o processo de morph está definido no universo matemático, será preciso então reconstruir o som, como descrito na Seção 1.1.3. Lembramos que há muita coisa por trás deste processo, cuja teoria e meandros é bastante investigada em (Oppenhein & Willsky, 1983), (Gomes & Velho, 1995b) e (Gomes *et al.* , 1995).

Se o processo de reconstrução for representado pelo operador $\mathcal{R}$ e o de discretização pelo operador $\mathcal{D}$, então o diagrama abaixo ilustra a operação de morphing através dos universos de abstração:



Este esquema de "recontrução, filtragem e discretização" é definido em (Gomes *et al.* , 1997a) como o processo de *resampling*.

O diagrama acima reflete o caso geral, em que as de transformações do suporte geométrico são feitas no universo matemático. Isto é necessário porque durante este processo as freqüências do sinal também são alteradas. Em regiões de contração surgem freqüências mais altas que antes não necessariamente existiam. Já em regiões de expansão surgem freqüências mais baixas. Para poder realizar novamente o processo de discretização é necessário realizar uma filtragem após a transformação, para que não haja problema durante o processo de amostragem.

Por ser um método geral, certas aplicações necessitam alterar o esquema de forma específica. Se a amostragem é feita por área, por exemplo, não há necessidade de filtragem, uma vez que esta filtragem já está no processo de amostragem utilizado. Em outras situações pode até mesmo ser conveniente alterar a ordem das operações.

No nosso caso, o processo pode ser implementado determinando primeiramente qual será domínio contínuo do resultado das operações de warping e morph. Devemos então discretizá-lo, e só então calcular o valor resultante nos pontos de tempo que darão origem no processo de discretização às amostras desejadas. Utilizando simplesmente no processo de discretização uma amostragem pontual com período τ, a fórmula de cálculo fica

$$g_{\lambda,n} = (1-\lambda)\mathcal{R}(f_1)(T_{1,\lambda}^{-1}(n\tau)) + \lambda\mathcal{R}(f_2)(T_{2,\lambda}^{-1}(n\tau)). \tag{3.5}$$

Quando ocorre uma contração do suporte geométrico, ocorrerá uma expansão no conteúdo espectral da função de atributo, resultado imediato das propriedades da transformada de Fourier. Eventualmente esta distorção será tal que a nova taxa de amostragem τ não será suficientemente elevada para representar a função de forma adequada. Uma forma de minimizar este problema seria através de uma filtragem antes de efetuar a amostragem pontual. Isto também equivaleria a fazer uma amostragem por área.

O processo de reconstrução pode ser realizado de diversas formas. Um método bastante simples, e que foi utilizado na implementação, é a interpolação linear entre amostras, ilustrado na Figura 3.4. Esta é uma etapa sensível, em imagens este tipo de reconstrução não é perceptualmente suficiente para obter bons resultados, em som não foram feitos experimentos suficientes para afirmar nada a respeito.

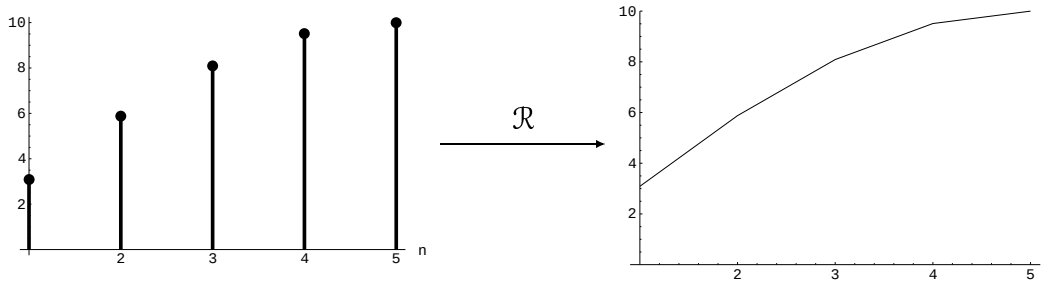


Figura 3.4: Reconstrução através da interpolação linear.

3.4 Morph Adaptativo

No processo descrito até agora, o resultado do morph era automaticamente bem definido através do parâmetro λ, utilizado simultaneamente tanto para a transformação de suporte geométrico, como para a operação de cross-dissolve linear.

Uma forma de flexibilizar o processo seria utilizar transformações de suporte geométrico não-lineares, outra seria utilizar um cross-dissolve adaptativo. É claro que nada impediria a combinação das duas abordagens em uma mesma transformação. Vamos agora criar um exemplo simples, onde utilizamos apenas o cross-dissolve adaptativo.

Considerando dois os sons originais $\mathcal{O}_1([a,b], f_1)$ e $\mathcal{O}_2([c,d], f_2)$, desejamos criar um *morph adaptativo* de $\mathcal{O}_1$ para $\mathcal{O}_2$ que possua nos sons por ele gerados uma "similaridade" diferente ao longo do tempo. Um elemento da família em $\lambda = 0.5$, o meio da transformação, deveria se parecer com o $\mathcal{O}_1$ no início, e com $\mathcal{O}_2$ no final, ao longo do tempo esta similariade deveria "continuamente" variar de um para outro.

Para isso, utilizaremos, como transformação do suporte geométrico, uma simples transformação afim, sendo o cross-dissolve será definido através de uma função $h(t, \alpha)$. Um caso simples seria quando h_0 efetua uma transição linear, ou seja

$$h_0(t,\lambda) = \begin{cases} 0 & \text{caso } t < -2\lambda + 1, \\ t - (2\lambda + 1) & \text{caso } -2\lambda + 1 \leq t \leq -2\lambda + 2, \\ 1 & \text{caso } t > -2\lambda + 2. \end{cases} \tag{3.6}$$

Podemos observar uma discretização da função de cross-dissolve adaptativo na Figura 3.5.

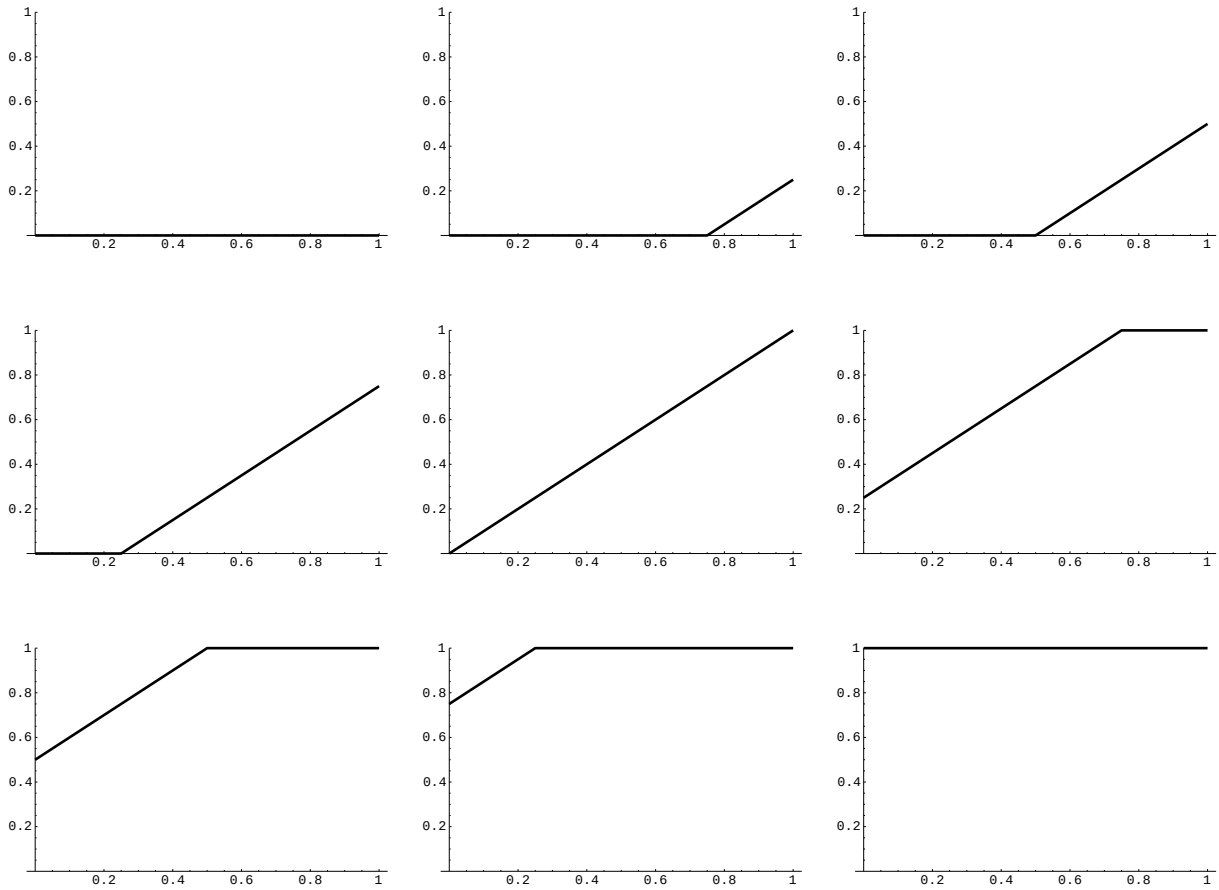


Figura 3.5: Cross-dissolve adaptativo onde h_0 possui um comportamento "linear".

Um caso mais interessante é quando $h(t, \lambda)$ varia baseada em uma cossenoide. Matematicamente isto pode ser escrito como

$$\lambda(t) = \begin{cases} 0 & \text{caso } t < 2\lambda + 1; \\ \frac{1-\cos(\pi(t-2\lambda-1))}{2} & \text{caso } -2\lambda + 1 \leq t \leq -2\lambda + 2; \\ 1 & \text{caso } t > -2\lambda + 2. \end{cases} \tag{3.7}$$

Estas duas famílias funções são ilustradas graficamente, lado a lado, na Figura 3.6.

Exemplo 1. A partir de dois sinais $\mathcal{O}_1 = ([-5, 3], f_1)$ e $\mathcal{O}_2 = ([-3, 5], f_2)$, onde $f_1(t) = \text{sen}(2\pi t)$ e $f_2 = \text{sen}(6\pi t)$, utilizando uma transformação de suporte geométrico afim e o cross-dissolve descrito na Equação 3.7, obtemos a família ilustrada através de uma discretização uniforme na Figura 3.7.

3.5 Deficiências do Método

Embora o método aqui descrito realize toda uma família contínua de transformações, o resultado perceptual não é o desejado.

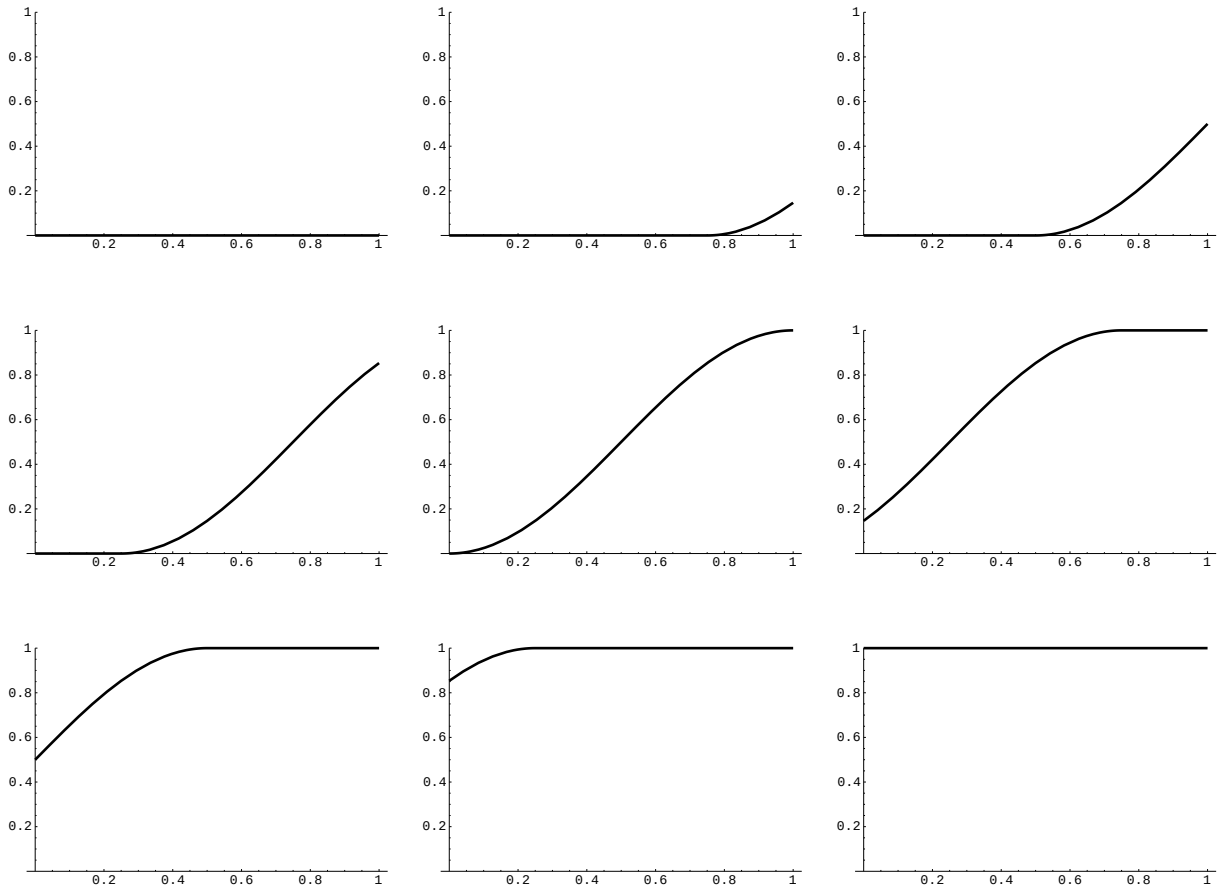


Figura 3.6: Cross-dissolve adaptativo onde h_0 possui um comportamento não-linear.

Um dos problemas mais fáceis de serem identificados pode ser visto ao aplicar este processo a dois trechos de voz, falando exatamente a mesma coisa, porém gravadas por locutores distintos. Será possível claramente identificar que no resultado existem duas pessoas distintas diferentes falando ao mesmo tempo, quando o desejado seria que estivesse presente apenas uma pessoa com uma voz diferente das vozes iniciais.

Isto ocorre porque cada um dos locutores pronuncia de forma única as diferentes "unidades fonéticas" (fonemas, etc.), prolongando certos trechos para dar ênfase, e passando mais rápido por outros. Desta forma, ao realizar o warp, estes trechos importantes não estarão "temporalmente alinhados", levando o ouvinte a perceber "dois" diferentes inícios, fins, etc. Para mais detalhes a respeito de o que são estas "unidades fonéticas" e quais as suas características e propriedades veja em (O'Shaughnessy, 1987) e (Rowden, 1991).

Vamos ilustrar este problema com mais um exemplo artificial, onde os sons originais possuem cada um dois trechos claramente distintos.

Exemplo 2. Partindo de dois sons originais $\mathcal{O}_1([-2,4], f_1)$ e $\mathcal{O}_2([-3,1], f_2)$ ilustrados na Figura 3.8. Em (a) observamos $\mathcal{O}_1$, que possui um trecho com um comportamento no intervalo $[-2,0)$ e outro comportamento distinto no intervalo $[0,4]$. Neste caso fabricado são apenas senoides com freqüências diferentes. Já $\mathcal{O}_2$, em (b), possui o mesmo tipo de comportamento, porém os intervalos são $[-3,-\frac{1}{3})$ e $[-\frac{1}{3},1]$.

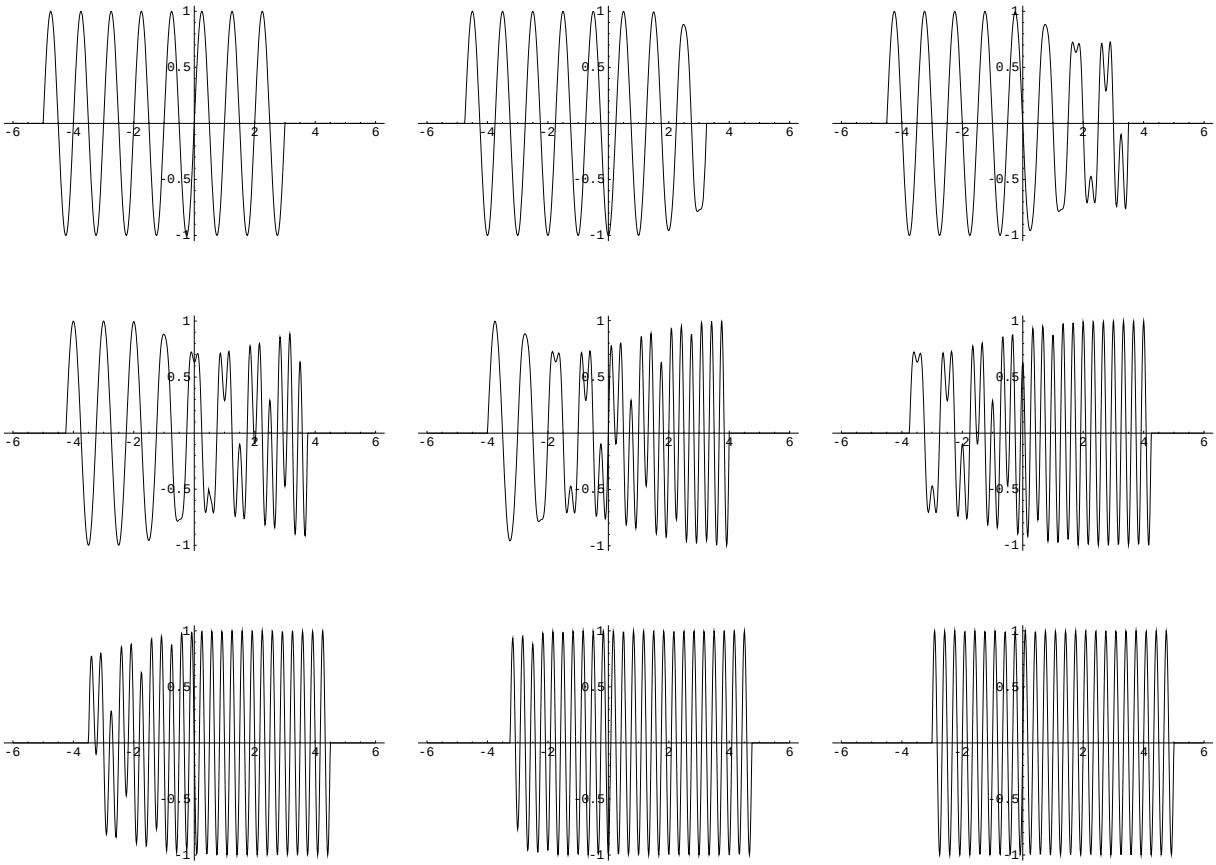
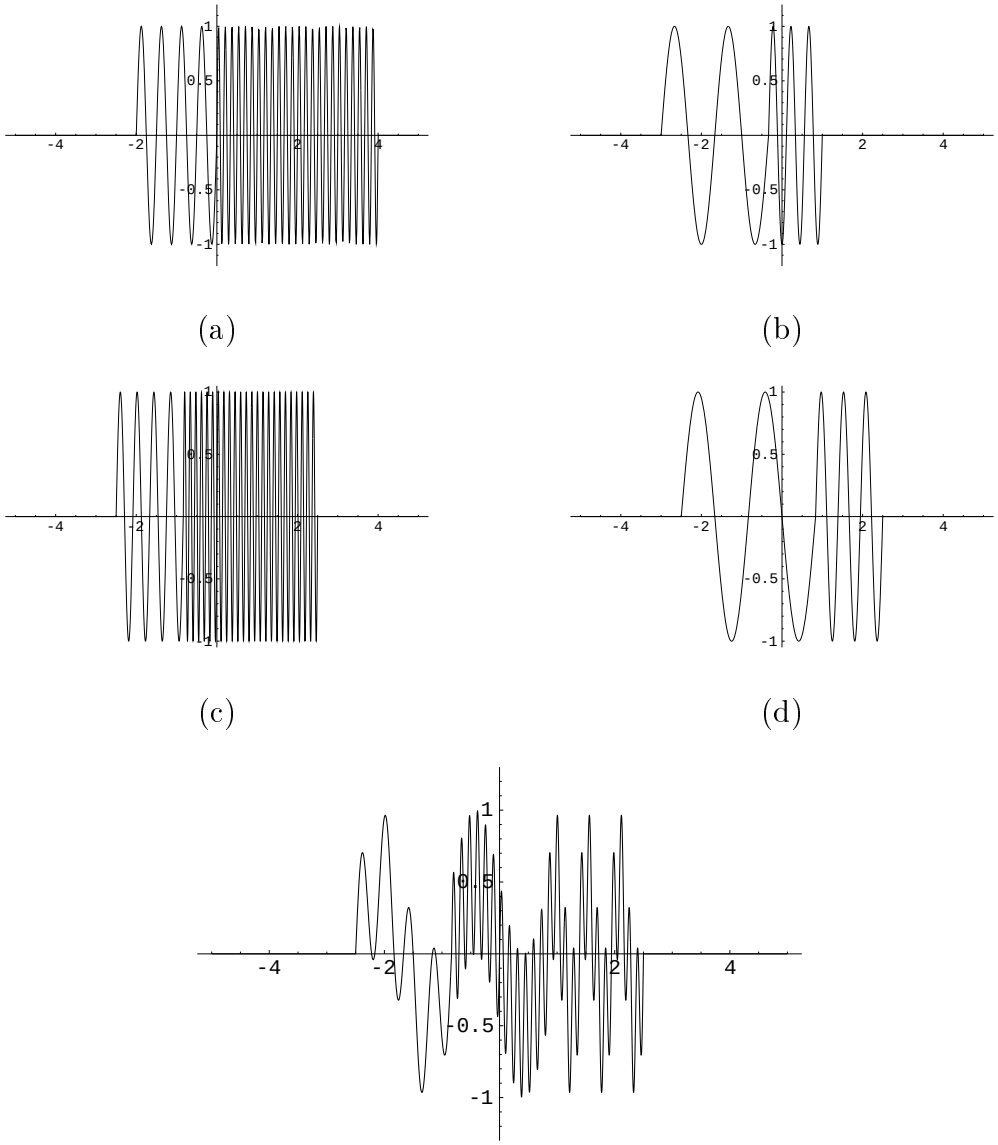


Figura 3.7: Morph adaptativo utilizando cross-dissolve com h da Equação 3.7.

Ao aplicar a metamorfose descrita neste capítulo obteremos o resultado mostrado na Figura 3.8. Em (c) e (d) estão o warp dos sons originais, e em (e) o resultado após a etapa de cross-dissolve.

Podemos observar que o resultado final possui três regiões distintas ao invés de apenas duas, como seria esperado. O primeiro intervalo é causado pela interação entre o primeiro intervalo de U_1 com o primeiro de U_2, o segundo pelo segundo intervalo de U_1 com o primeiro de U_2 e, finalmente, o terceiro, resultado da interação entre o segundo intervalo de U_1 com o segundo intervalo de U_2. A Figura 3.9 mostra apenas o relacionamento entre os intervalos.

Para resolver este problema o método que será introduzido no próximo capítulo busca "alinhar" as características importantes de cada som.



(a) (b)

(c) (d)

(e)

Figura 3.8: Morph em duas etapas utilizando o método descrito neste capítulo.

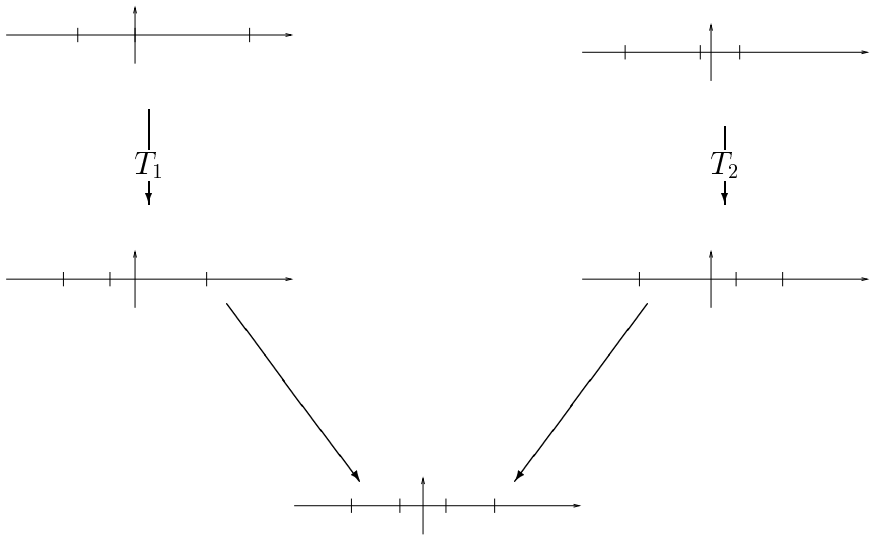


Figura 3.9: Comportamento dos intervalos do processo ilustrado na Figura 3.8.

Capítulo 4

Metamorfose por Segmentação Temporal

Neste capítulo descrevemos uma segunda técnica de morph, onde o método imediato do capítulo anterior é na verdade apenas um caso particular.

Para resolver problema apresentado, o método agora introduzido busca "alinhar" as características importantes de cada som, para só então trabalhar o morph. Para isso, é necessário introduzir o conceito de *segmentação*.

4.1 Segmentação de Sons

A segmentação de sons é um caso típico de uma *partição de um conjunto*. Uma partição busca separar e classificar um conjunto maior, segundo alguma função de classificação.

Mais precisamente, considere uma função $f\colon U \to V$, e uma família $\{V_\lambda\}$ de subconjuntos de V. Para cada subconjunto $V_\lambda \subset V$, definimos $U_\lambda = \{u \in U; f(u) \in V_\lambda\}$. Se V_λ são disjuntos dois a dois, é claro que:

a) $\bigcup U_\lambda = U$;

b) $U_\lambda \cap U_\mu = \varnothing$ se $\lambda \neq \mu$.

Diz-se então que os U_λ formam uma partição, ou segmentação, do conjunto U. O problema de segmenar um conjunto pode ser descrito em três etapas:

1. Determinar a função de segmentação f;
2. Determinar a família $\{V_\lambda\}$;
3. Calcular a partição $\{U_\lambda\}$.

A segmentação de um objeto gráfico $\mathcal{O} = (U, f)$ consiste em segmentar o seu suporte geométrico U. Em geral, utiliza-se a própria função de atributos f para obter diferentes segmentações de U, mas isso não é uma regra.

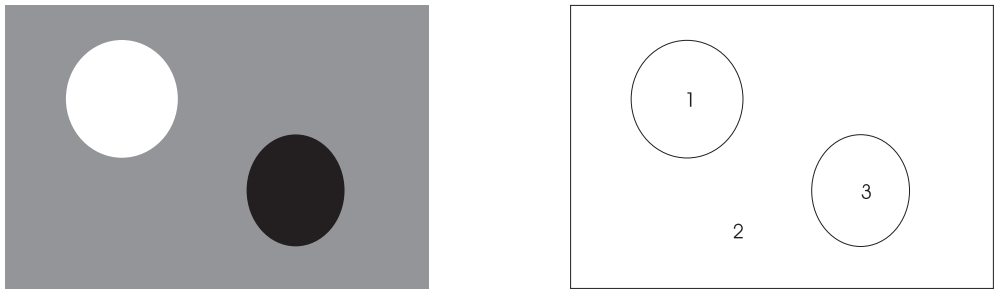
Exemplo 1 (Segmentação por Cor). Considere a imagem da Figura 4.1 (a).

Na Figura 4.1 (b) temos uma segementação dessa imagem em dois conjuntos $\{p; f(p) < 128\}$ e $\{p; f(p) \geq 128\}$. Os pontos pertencentes ao primeiro conjunto são pintados com preto, os do segundo com branco.

(a) (b)

Figura 4.1: Exemplo de segmentação por cor.

Exemplo 2 (Segmentação por Bordo). Neste caso procura-se segmentar a imagem baseado nas bordas das regiões formadas pelos objetos que constituem a imagem, como pode ser observado na Figura 4.2.



(a) (b)

Figura 4.2: Exemplo de segmentação por bordo.

Exemplo 3 (Segmentação por Freqüência). Consiste em se fazer uma segmentação do suporte do objeto gráfico baseado nas freqüências de distribuição espectral da função de atributos.

Podemos obter segmentações por bordo (veja o exemplo 2) baseado numa segmentação por freqüência. Para isso devemos caracterizar os bordos em termos de freqüência. Um exemplo disto é ilustrado na Figura 4.3.

4.1.1 Ω como intervalo de $\mathbb{R}$

Estamos particularmente interessados na segmentação de um sinal de áudio, $f : [a, b] \to \mathbb{R}$. Como foi visto, segmentar um sinal de áudio significa subdividir o domínio temporal em subintervalos. Nesse caso basta achar uma sequencia de números reais

$$a = t_0 < t_1 < t_2 < \cdots < t_n = b.$$

Figura 4.3: Exemplo de segmentação por freqüência utilizando transformada wavelet. Figura extraída de (Gomes *et al.* , 1997b).

Cada intervalo $[t_i, t_{i+1}]$ define um conjunto da partição.

Existe uma única transformação afim T que leva o intervalo $[0, 1]$ no intervalo $[a, b]$ no qual está definida a segmenação do som. Através dessa transformação definimos uma partição

$$0 = \alpha_0 < \alpha_1 < \cdots < \alpha_n = 1$$

do intervalo $[0, 1]$, de modo que $T(\alpha_i) = t_i$. Esta transformação é ilustrada na Figura 4.4.

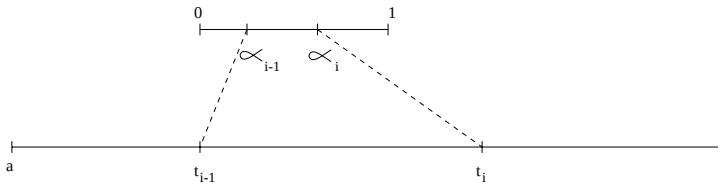


Figura 4.4: Transformação de $[\alpha_{i-1}, \alpha_i]$ para $[t_{i-1}, t_i]$.

O gráfico de T é dado pela Figura 4.5, ou seja

$$T(x) = (b - a)x + a = a(1 - x) + bx.$$

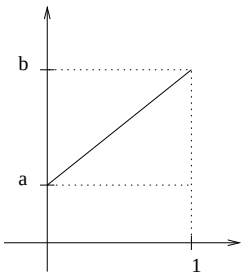


Figura 4.5: Gráfico de T.

Portanto, cada intervalo $U_i = [t_{i-1}, t_i]$ da partição de $[a, b]$ é dado por

$$U_i = [t_{i-1}, t_i] = T[\alpha_{i-1}, \alpha_i] = [a(1 - \alpha_{i-1}) + b\alpha_{i-1}, a(1 - \alpha_i) + b\alpha_i].$$

O método acima faz portanto uma parametrização da partição do intervalo $[a, b]$ por uma partição do intervalo $[0, 1]$.

Esta parametrização torna as segmentações em objetos independentes no domínio do qual o som está definido, podendo então ser armazenados, trocados e manipulados com uma maior robustez, facilitando o processo de implementação.

4.2 Warp e Morph Utilizando as Segmentações

Nesta seção vamos descrever um método de metamorfose de áudio usando segmentação. Para isso considere dois sinais de áudio $f\colon [a, b] \to \mathbb{R}$ e $g\colon [c, d] \to \mathbb{R}$.

Tomemos uma segmentação

$$a = t_0 < t_1 < \cdots < t_{n-1} < t_n = b$$

do intervalo $[a, b]$, e uma segmentação

$$c = s_0 < s_1 < \cdots < s_{n-1} < s_n = d$$

do intervalo $[c, d]$. Note que estamos supondo que as duas segmentações possuem o mesmo número n de intervalos.

Cada intervalo $[t_{i-1}, t_i]$ da partição de $[a, b]$ define o sinal de áudio $f_i = f|[t_{i-1}, t_i]$ dado pela restrição de f ao intervalo. Analogamente, a restrição $g|[s_{i-1}, s_i]$ define o sinal de áudio g_i. A metamorfose entre os sons f e g é obtida fazendo, separadamente, a metamorfose entre os sinais f_i e g_i.

A metamorfose entre f_i e g_i é realizada utilizando o método do capítulo anterior. É claro que o som resultante possui naturalmente uma segmentação com n intervalos.

Desta forma podemos criar n transformações de suporte geométrico, $T_{1,i,\lambda}$ e $T_{2,i,\lambda}$, uma para cada semento de cada um dos sinais. Adaptando então a Equação 3.3 à realidade de cada segmento obtemos:

$$T_{1,i,\lambda} = u_{i-1} + \frac{t - t_{i-1}}{t_i - t_{i-1}}, \tag{4.1}$$

onde

$$e(\lambda) = u_{0,\lambda} < u_{1,\lambda} < \cdots < u_{n,\lambda} = f(\lambda)$$

é a segmentação do sinal resultante em λ obtida por

$$u_{i,\lambda} = (1 - \lambda)t_i + \lambda s_i.$$

Através deste processo, obtemos a característica desejada, realizando o "alinhamento temporal".

Exemplo 4. Vamos agora com o auxílio dos mesmos sons originais do exemplo 2 do Capítulo 3, utilizar a transformação de suporte geométrico afim por partes, descrita na Equação 4.1, para definir uma família de objetos de morph.

Na Figura 4.6 vemos uma discretização da família resultante do processo de morph. É interessante reparar no comportamento dos segmentos ao longo da transformação.

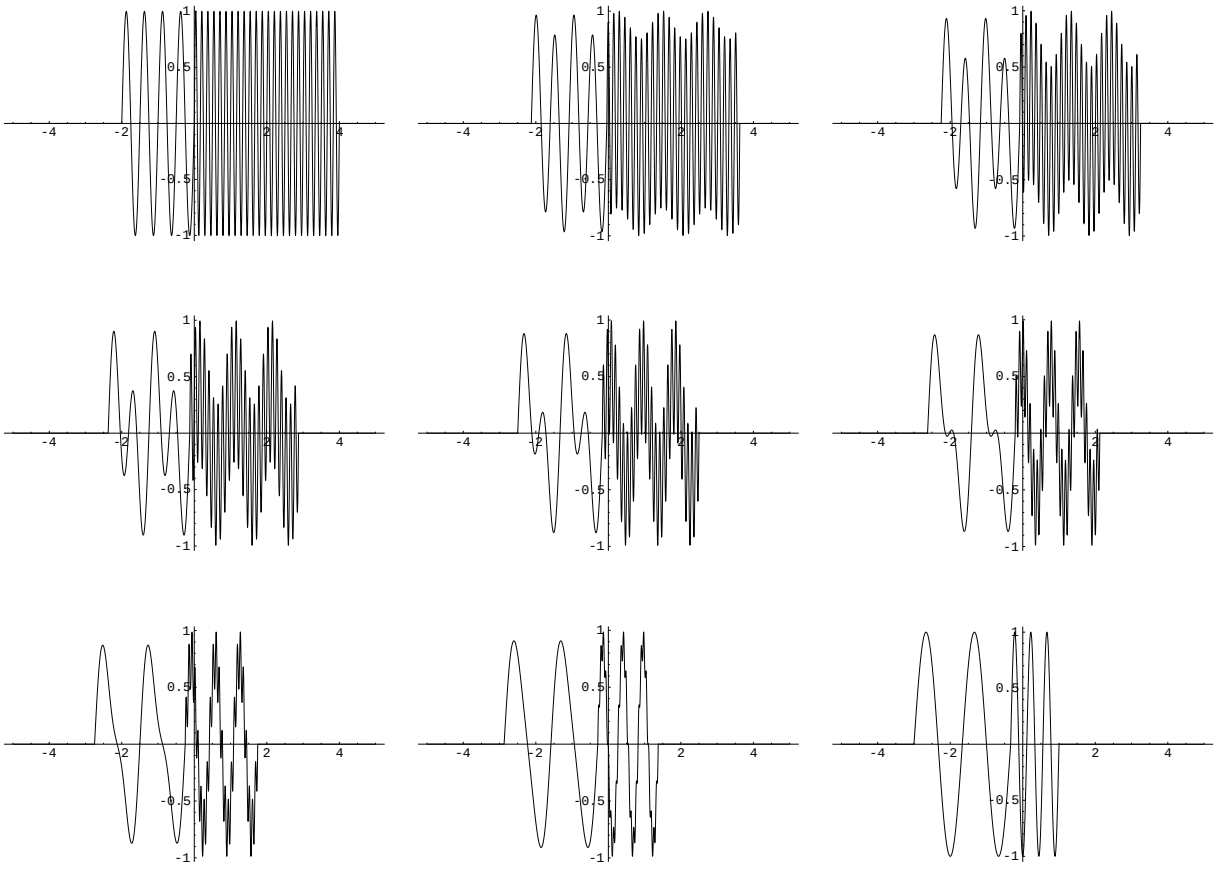


Figura 4.6: As diversas etapas de uma transição de morph utilizando segmentação temporal entre os dois sons representados na Figura 3.8 (a) e (b).

4.3 Deficiências do Método

As técnicas de warp e morph, utilizando a segmentação, resolvem de forma eficiente o problema de alinhamento temporal, como foi visto através da comparação entre os exemplos 2 e 4.

O método descrito no Capítulo 3 não era capaz de preservar no resultado a divisão natural que havia nos sons originais. No exemplo 2, artificialmente produzido para realçar esta característica, partia-se de sons para os quais era claramente possível identificar dois trechos com comportamentos distintos, dando origem a sons que possuiam até mesmo três trechos, sendo que um deles era causado por uma espécie de interferência entre trechos não correlacionados.

Com o uso da técnica da segmentação este problema é resolvido, uma vez que a transformação do suporte geométrico passa a ser realizado por partes, subjugado ao warp da segmentação propriamente dita. Desta forma, é possível mapear trechos correspondentes dos sons originais no mesmo intervalo do objeto final.

Se observarmos com bastante atenção o exemplo 4, notamos que a família contínua definida pela operação de morph é realmente suave e preserva a característica estrutural, mantendo sempre apenas dois trechos distintos.

Surge, no entanto, uma questão fundamental: todo o processo é baseado na utilização de uma segmentação adequada nos dois sons originais. Será que é fácil obtê-la? É necessário intervenção humana neste processo?

A área de segmentação de sinais unidimensionais é utilizada em uma enorme gama de aplicações, completamente distintas entre si, e talvez por isto ainda seja cheia de problemas e questões em aberto. Até o momento, diversas técnicas foram desenvolvidas utilizando como ferramentas básicas certas características específicas de determinados tipos de sinais.

Por exemplo, a segmentação é uma etapa fundamental nos processos de reconhecimento e interpretação de voz, onde procura-se segmentar o sinal em células de informações mais básicas, que são então comparadas e analisadas.

Ainda para o processamento de voz, utiliza-se processos de segmentação para criar automaticamente dicionários de fones e fonemas, partindo de frases longas, que serão depois utilizados em aplicações de síntese por concatenação destas unidades básicas (veja (van Hemert, 1991)).

Geralmente, estas aplicações buscam identificar mudanças bruscas no comportamento espectral do sinal ao longo do tempo. Assumem também que o sinal original possui uma estrutura muito específica de formação (para o caso de voz, veja (Rabiner & Schafer, 1978), (O'Shaughnessy, 1987) e (Rowden, 1991)).

Alguns trabalhos, no entanto, ou não assumem pré-condições ou podem ser ajustados para uso em diversas classes de sinais ((Wesfreid & Wickerhauser, 1993), (Li & Gibson, 1996), (Li, 1996), (Svendsen & Soong, 1987)).

Geralmente, deve-se utilizar a intervenção humana, mesmo se apenas para verificar os resultados obtidos em uma etapa automática.

Sem ainda entrar no mérito perceptual, notamos que tanto o primeiro, como também o segundo som do último exemplo, é composto por senoides puras, com uma única freqüência em cada segmento, e que os resultados da operação de morph apresentam a superposição de duas senoides com freqüências distintas, resultado da composição direta através da soma no processo de cross-dissolve.

Seria mais natural que a família de objetos gerado pelo morph fornecesse algo como o representado na Figura 4.7, onde além dos segmentos serem preservados, as freqüências das senoides puras também são continuamente "interpoladas".

Um outro exemplo interessante que também nos motiva à criação de um método mais elaborado, seria através do uso de sons compostos novamente por dois segmentos cada, e também usando em cada um deles senoides puras. Desta vez, porém, as senoides definidas são iguais nos dois sons, sendo eles diferenciados apenas pelas durações de cada intervalo.

Na Figura 4.8 podemos ver na primeira e última linha os dois sons originais, na segunda linha observamos três etapas intermediárias igualmente espaçadas ($\lambda = 0,25$, $\lambda = 0,5$ e $\lambda = 0,75$) do processo de morph utilizando segmentação. Já na terceira linha, observamos as mesmas três do que seria realmente desejado:

Perceptualmente,f o morph com segmentações passa a criar sons "únicos", ao invés da "mistura" de sons distintos. Podemos ilustrar isto com um exemplo de dois trechos de fala: com o método do Capítulo 3, o resultado seria duas vozes diferentes, falando simultaneamente (nosso ouvido percebe claramente que são duas pessoas distintas). Já com o método utilizando segmentações a voz gerada é única, não é possível identificar "duas" pessoas. No entanto, o som obtido não é de uma voz intermediária: ele fica com uma aparência "não-natural", artificial. Isto é motivo mais do que suficiente para buscar técnicas mais elaboradas.

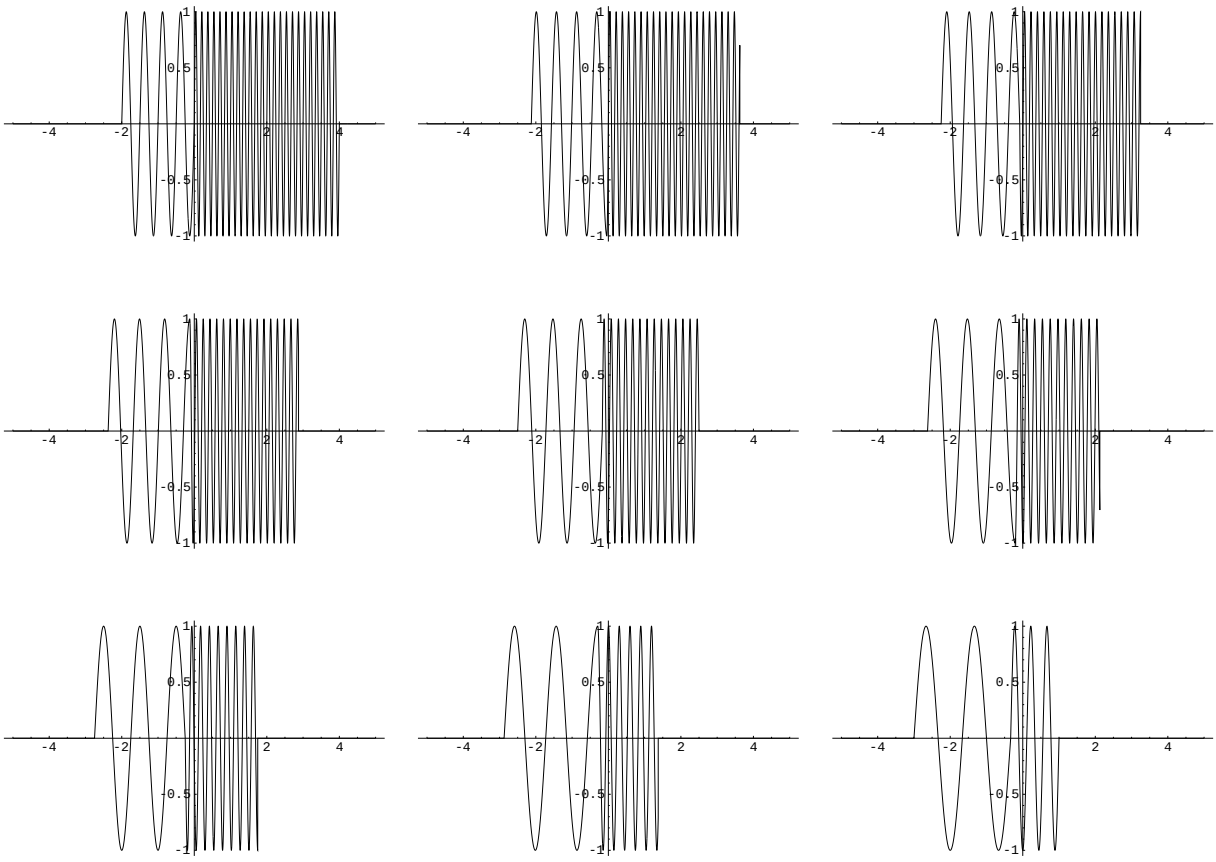


Figura 4.7: Uma família de morph mais intuitiva.

Para obter esse resultado devemos sair do domínio do espaço e definir morph no domínio tempo × freqüência . Esse será o assunto dos próximos capítulos.

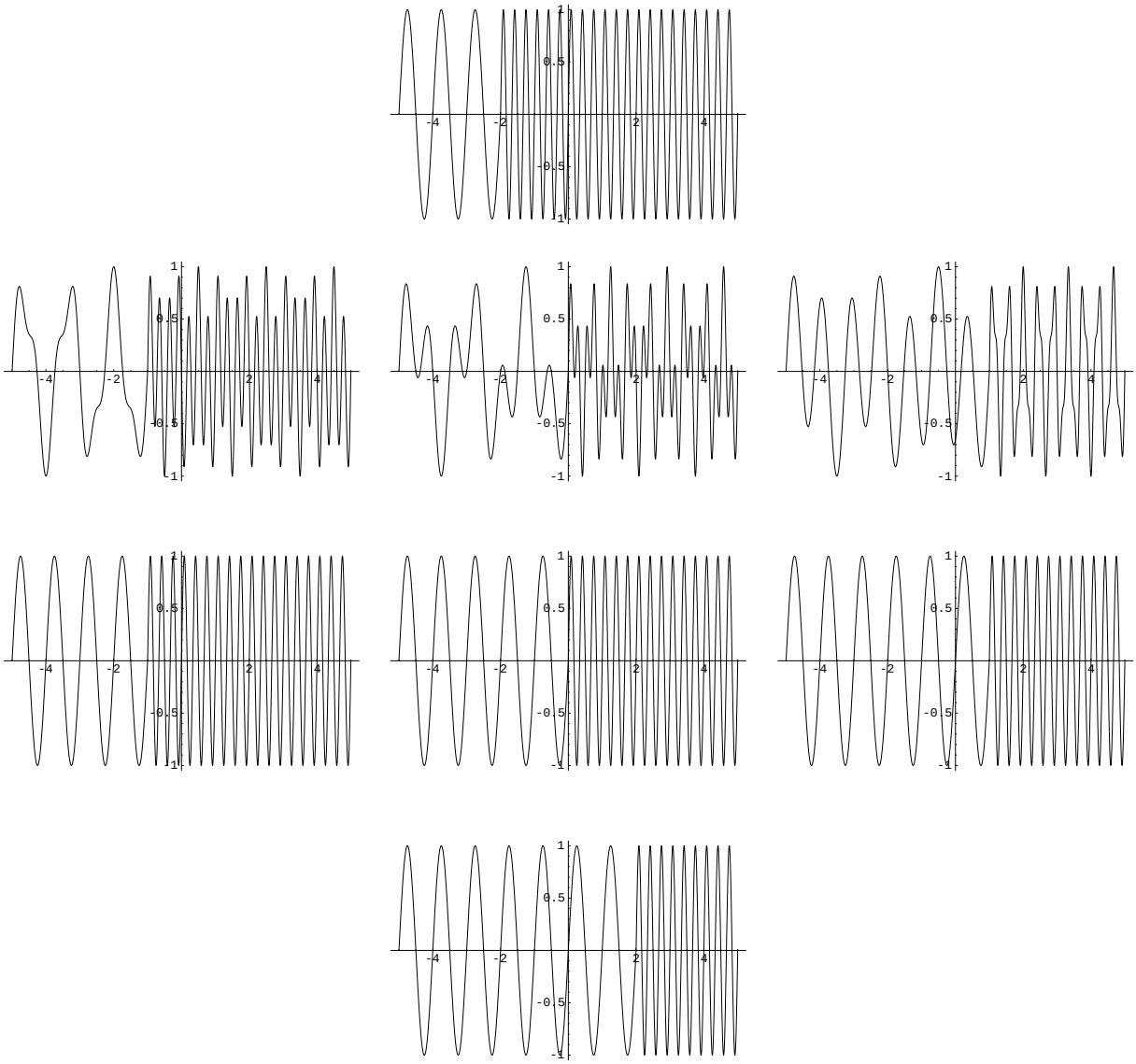


Figura 4.8: Uma família de morph mais intuitiva.

Capítulo 5

Transformações em Tempo × Freqüência

Vimos no último capítulo que o cross-dissolve não é uma boa operação para obtermos metamorfose de áudio. O cross-dissolve é motivado pela operação análoga com imagens. Ocorre que, ao observar as imagens resultantes, o olho humano não faz uma decomposição das componentes de freqüência das imagens originais. O mesmo, porém, não ocorre com o aparelho auditivo: ao somarmos dois sinais, o ouvido escuta o resultado decompondo as componentes originais de freqüência.

Concluimos daí que a mistura correta a ser utilizada em uma metamorfose de áudio deve ser bem mais elaborada, levando-se em conta fatores tempo × freqüência.

Na realidade, essa mistura deve se basear nas propriedades de filtragem utilizadas pelo sistema auditivo de percepção de som. Essa filtragem pode ser corretamente modelada por uma transformada wavelet (Souza & Caloba, n.d.a), sendo esta uma motivação para a utilização de wavelets como uma das bases para o processo de metamorfose de sons.

Neste capítulo descreveremos a terceira e última técnica para a obtenção da metamorfose de dois sons, que utiliza representações tempo × freqüência.

Como já foi comentado, o ouvido humano é sensível a variações de pressão do ar, e diferencia os sons captados tanto pela velocidade com que ocorrem estas variações (que recebe o nome de freqüência, i.e. número de oscilações por segundo), como também sua intensidade.

5.1 Representações Tempo × Freqüência

Quando o som é descrito simplesmente através de uma função de atributos f definida sobre um intervalo $[a, b]$ descrevendo a "pressão do ar" a cada instante de tempo (isto é chamado de *representação temporal*), não estamos fornecendo, pelo menos de forma explícita, nenhuma informação de como são estas variações. Apesar disto, é geralmente fácil determinar o início e fim de certas ocorrências significativas.

Uma interpretação interessante para utilização de *representações tempo × freqüência* pode ser emprestada da álgebra linear: *mudança de base*. Realizamos uma transformação que modificará a "aparência" da informação, a forma com que ela está representada, mas não o seu conteúdo, uma vez que é possível retornar para a forma original através da transformação inversa.

Representações tempo × freqüência são atualmente muito utilizadas e pesquisadas em diversas áreas da Matemática Aplicada (Computação Gráfica, Processamento de Sinais, etc.). O termo também é muito amplo abrangendo uma enormidade de assuntos, técnicas e problemas ainda em aberto.

Existem diversos livros e artigos que abordam este assunto e seria impossível tentar em uma seção desta tese, ou mesmo em um capítulo inteiro, querer incluir todos os detalhes e meandros. Ao leitor interessado em mais detalhes sugerimos a leitura de (Gomes *et al.* , 1997b).

Considere um espaço de funções

$$\mathcal{F} = \{f : U \subset \mathbb{R} \to \mathbb{R}\}.$$

Uma transformação tempo × freqüência é um operador linear invertível

$$T : \mathcal{F} \to \widetilde{\mathcal{F}},$$

onde

$$\widetilde{\mathcal{F}} = \{g : U \times W \to \mathbb{R}\},$$

e $W \subset \mathbb{R}$. Temos portanto que $\widetilde{f} = T(f) : U \times W \to \mathbb{R}$, onde o par $(t, \omega) \in U \times W$ representa a freqüência ω [1] da função f no tempo t.

Existem diversas transformadas que realizam uma transformação tempo × freqüência, dentre elas podemos citar a transformada de Fourier com janela e a transformada de wavelets (veja (Gomes *et al.* , 1997b), (Daubechies, 1992), (Vaidyanathan, 1993), (Daubechies, 1990), (Cohen, 1989), (Hlawatsch & Boudreaux-Bartels, 1992), (Vetterli & Kovacevic, 1995), (Strang & Nguyen, 1996)).

5.1.1 Transformada de Fourier com Janela

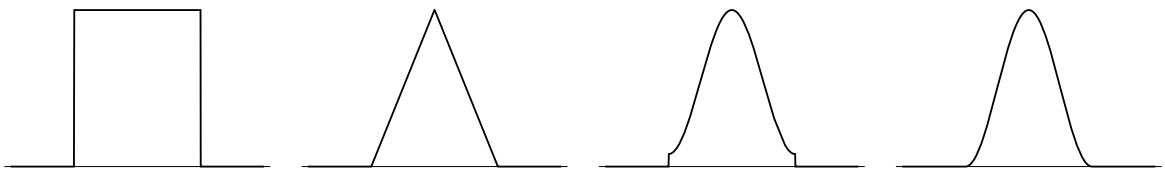
Uma forma de tentar se obter a localização temporal de certos eventos espectrais é através do uso de uma função de janela. Uma *função de janela* tem como objetivo manter apenas uma região do domínio válida e eliminar tudo o que fica fora dela. Diversas funções podem ser usadas. Alguns exemplos clássicos usado em processamento de sinais para o projeto de filtros com resposta finita são as janelas retangular, hanning, hamming, Blackmann, Kaiser. Veja na Figura 5.1.1.

Após a multiplicação da função pela janela transladada, preserva-se apenas a região desejada e é então possível calcular a transformada de Fourier. Este tipo de análise se chama *Transformada de Fourier com Janela.*

Fica claro que todas as características irão depender da função $\gamma(t)$ utilizada como janela. A expressão matemática é dada por:

$$\widetilde{f}(t, \omega) = \int_{-\infty}^{+\infty} f(\tau)\gamma(\tau - t)e^{-2\pi i\omega\tau}d\tau. \tag{5.1}$$

[1] Aqui utilizamos Hertz, e não radianos por segundo como o que a variável ω induziria o pessoal de Processamento de Sinais a interpretar.



(a) (b) (c) (d)

Figura 5.1: Algumas funções de janela com suporte finito. (a) Retangular, (b) Bartlet, (c) Hamming e (d) Hanning

Podemos interpretar a expressão da transformada inversa como um método de síntese, onde compomos o nosso sinal pela "soma" das contribuições das diferentes funções de base.

$$f(t) = \int \int_{-\infty}^{+\infty} \widetilde{f}(\tau, \omega) g(t - \tau) e^{2\pi\omega t} d\tau d\omega. \tag{5.2}$$

É muito importante observar que as funções moduladoras na etapa de análise e síntese, γ e g, são diferentes. Para uma dada função de análise γ, existe uma infinidade de possíveis g, basta que elas atendam entre si a relação

$$\int_{-\infty}^{+\infty} g(t)\gamma^*(t)dt = 1. \tag{5.3}$$

A *Transformada Discreta de Fourier com Janela* geralmente é implementada através de um banco de filtros, como o que é ilustrado na Figura 5.2.

Cada saída do banco de filtros (implementado com o uso de um bloco de FFT) é o resultado de um "filtro passa-banda", dando então a informação da presença de freqüências de uma região do espectro no conjunto de amostras da entrada.

5.1.2 Transformada de Wavelets

Enquanto a transformada de Fourier com janela busca similaridades do sinal em questão com translações de exponenciais complexas moduladas pela função de janela, a transformada wavelet busca similaridades com versões dilatadas e transladadas de uma mesma função ψ especial, que possui características oscilatórias. Esta função ψ é chamada de *função wavelet*.

Esta condição de oscilação pode ser descrita através de

$$\int_{-\infty}^{+\infty} \psi(t)dt = 0. \tag{5.4}$$

Através então de suas dilatações e translações criaremos toda uma família de funções $\psi_{a,b}$, que são obtidas segundo

$$\psi_{a,b}(t) = |a|^{-\frac{1}{2}} \psi\left(\frac{t-b}{a}\right). \tag{5.5}$$

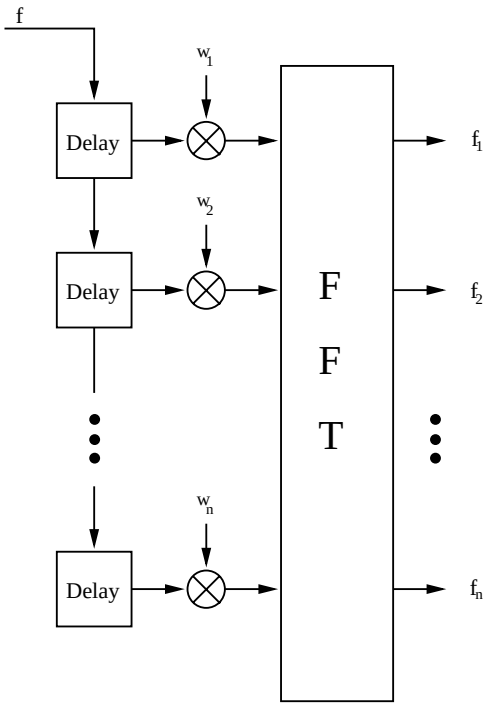


Figura 5.2: Implementação com banco de filtros da Transformada de Fourier com Janela.

A decomposição de wavelets fornece uma representação descrita por a e b. É preciso notar, no entanto, que a representa escala, e não freqüência como no caso da transformada de Fourier com janela. Como é possível então relacionar escala com freqüência?

Em (Hlawatsch & Boudreaux-Bartels, 1992) observa-se que a função ψ possui características espectrais de passa banda, ou seja, pode-se dizer que no domínio espectral suas características estão concentradas em torno da freqüência f_0. Desta forma, fazendo $a = \frac{f}{f_0}$, a Equação 5.5 fica

$$\psi_{f,b}(t) = \left(\frac{f}{f_0}\right)^{-\frac{1}{2}} \psi\left(\frac{f}{f_0}(t-b)\right). \tag{5.6}$$

A transformada de wavelet é então definida como

$$\widetilde{f}(a,b) = |a|^{-\frac{1}{2}} \int_{-\infty}^{+\infty} f(t)\psi(\frac{t-b}{a}).$$

A *Transformada Discreta de Wavelets* geralmente é implementada através de um banco de filtros, aproveitando a estrutura piramidal que surge naturalmente da análise em multiresolução (Gomes *et al.* , 1997b). Isto pode ser observado na Figura 5.3.

Note que são sempre utilizados os mesmos dois filtros $\overline{H}$ e $\overline{G}$. Esses filtros possuem as características de serem passa-baixa e passa-alta, respectivamente, particionando o espectro em duas bandas.

O filtro $\overline{H}$ é responsável por gerar versões cada vez mais simplificadas, enquanto o filtro $\overline{G}$ separa os "detalhes" entre estes diferentes níveis de representação, ou seja, os coeficientes da transformada propriamente dita.

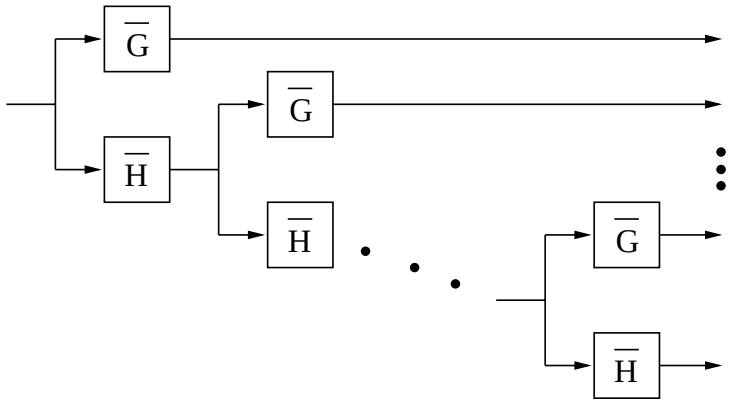


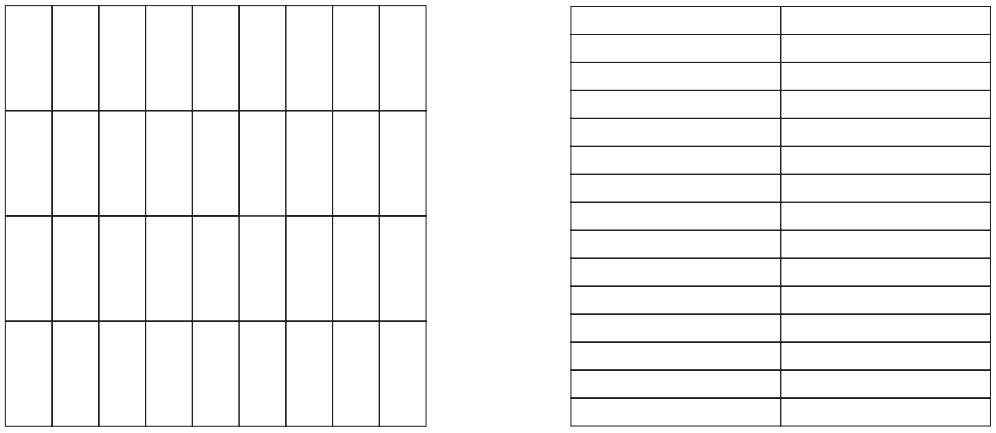
Figura 5.3: Implementação com banco de filtros da Transformada Wavelet.

5.1.3 Decomposição Atômica

O uso de uma destas transformadas decompõe o sinal em átomos de representação dispostos no plano tempo freqüência. Átomos estes que variam em forma no plano tempo × freqüência , dependendo da transformada.

Na transformada discreta de Fourier com janela, cada n amostras se transformam em n informações a respeito de diferentes "bandas" de freqüência. Todas elas duram o período de tempo das n amostras, gerando átomos de representação retangulares, como está ilustrado na Figura 5.4.

O uso de janelas menores implica em átomos estreitos no eixo do tempo e compridos no eixo das freqüências (a), já o uso de janelas maiores implica em átomos compridos no eixo do tempo e estreitos no eixo das freqüências (b). Esta dualidade vem do princípio da incerteza (o mesmo utilizado em mecânica quântica) que diz que as resoluções entre tempo e freqüência são inversamente proporcionais.



(a) (b)

Figura 5.4: Decomposição atômica da transformada de Fourier com janela com dois tamanhos de janela diferentes.

A transformada discreta de Wavelet por outro lado não possui átomos uniformes. Sua estrutura piramidal faz com que os coeficientes de freqüências mais altas sejam mais concentrados temporalmente e, devido ao princípio da incerteza, mais esparsos espectralmente. Cada nível de freqüências possuirá uma geometria de átomos distinta, como ilustrado na Figura 5.5.

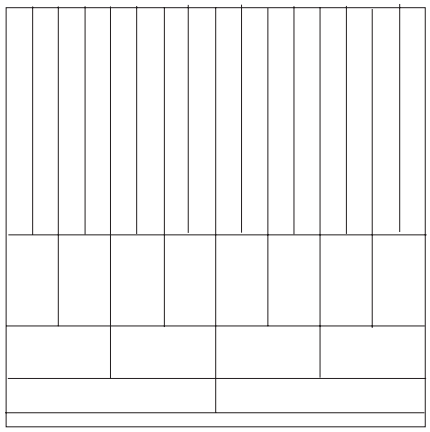


Figura 5.5: Decomposição atômica da transformada de Wavelet.

A geometria do átomo fornece diversas informações a respeito da componente que ele representa na decomposição do sinal original. Átomos compridos representam funções na decomposição com grande suporte temporal; átomos localizados em freqüências altas implicam em funções com grandes variações durante seu suporte.

Na Figura 5.6 vemos dois diferentes átomos de Fourier e suas respectivas funções de decomposição. Na Figura 5.7 vemos o mesmo, porém aplicado à transformada de wavelet.

É importante ressaltar que, ao ilustrar uma representação atômica por uma imagem, a intensidade de cada átomo é dada pelo seu tom de cinza.

5.2 Transformações básicas em Tempo × Freqüência

As transformações do domínio tempo × freqüência se aplicam à transformada $\widetilde{f}(t,\omega)$. Como já foi visto antes, podemos ter transformações de atributos (neste caso os valores de $\widetilde{f}$) ou de suporte geométrico (que aqui são um subconjunto do plano tempo × freqüência).

5.2.1 Transformação de Atributo

Um caso particular e importante de *transformações de atributo* ocorre quando fazemos uma modulação da transformada $\widetilde{f}$ por uma função que depende apenas da freqüência. Mais precisamente:

$$T(\widetilde{f}(t,\omega)) = T(\omega)\widetilde{f}(t,\omega).$$

Em Processamento de Sinais isto é chamado de processamento em sub-bandas, ou seja, para cada região do espectro é dado um tratamento distinto.

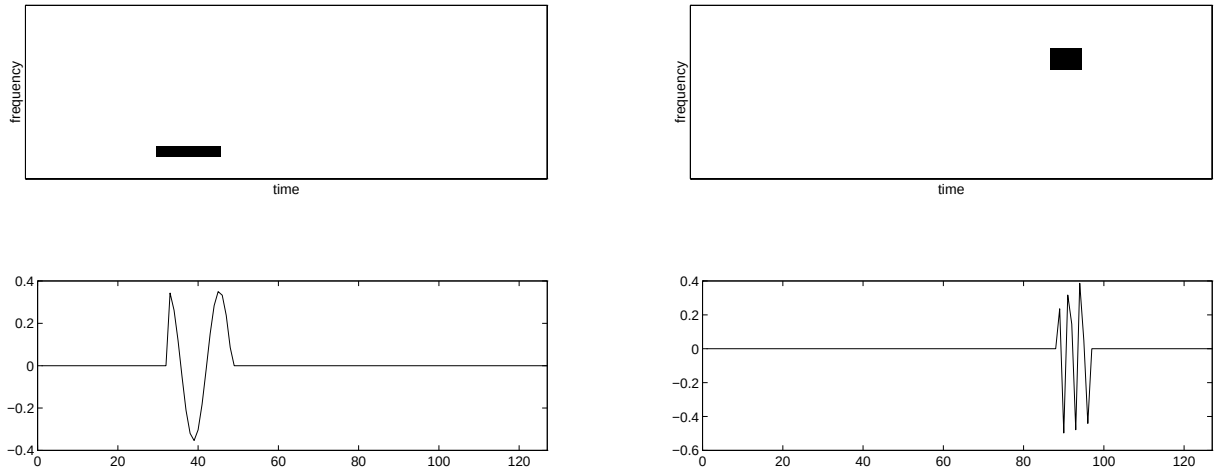


Figura 5.6: Átomos de Fourier e suas respectivas funções.

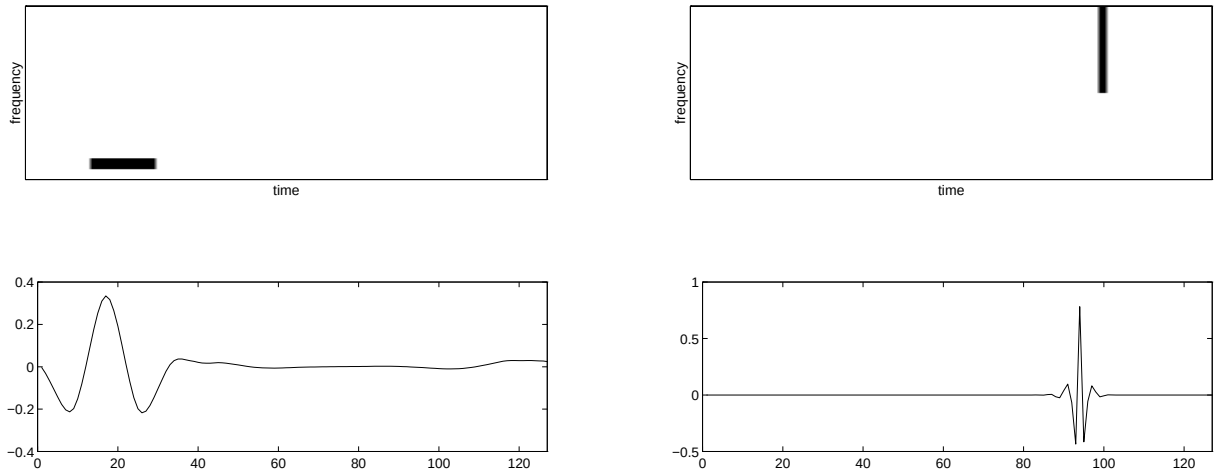


Figura 5.7: Átomos de Wavelet e suas respectivas funções.

Em áudio este tipo de operação é geralmente chamada de *equalização* sendo largamente utilizada. Existem aparelhos para a sua realização com diversos níveis de complexidade, estando presentes tanto em equipamentos de som simples como também nos grandes estúdios de masterização de gravadoras.

Se a função de modulação depende também do tempo, isto é,

$$T(\widetilde{f}(t,\omega)) = T(t,\omega)\widetilde{f}(t,\omega),$$

dizemos que temos uma *transformação de atributos adaptativa.*

Existe na literatura diversas técnicas já estabelecidas, que geralmente utilizam uma série de critérios estatísticos para estabelecer e controlar o comportamento da transformação ao longo do tempo. Estas transformações podem ser utilizadas para tarefas

bastante complexas, como remoção de ruído ou equalização dinâmica do som em uma boate.

5.2.2 Transformação de Suporte Geométrico

Transformações de suporte geométrico operam em subconjuntos do domínio tempo × freqüência e portanto possuem inúmeras aplicações. Como neste caso o domínio é composto de duas dimensões, tempo e freqüência, costuma-se classificá-las em dois tipos:

1- Deformações sobre o eixo do tempo;

2- Deformações sobre o eixo da freqüência.

Agora ilustraremos dois tipos de transformações de suporte geométrico que se enquadram nas classes agora descritas.

Expansão e Compressão Temporal

O objetivo de uma operação de expansão ou compressão temporal é alterar a duração de um som sem distorcer sua percepção pelo ouvido humano. Ilustrando de forma concreta, seria por exemplo fazer com que uma gravação de uma pessoa falando fosse mais rápida ou devagar, sem no entanto alterar o timbre da voz.

Se o processo for ingenuamente realizado no tempo com uma transformação de expansão ou contração temporal (como as descritas no Capítulo 2), o efeito resultante disto seria o mesmo que o de um disco de vinil tocando na rotação errada. Embora a duração total do som pudesse realmente ser alterada, suas características espectrais, e consequentemente a percepção do ouvido humano ao fenômeno, também seriam completamente distorcidas.

Diversas aplicações para esta transformação em especial podem ser mencionadas, inclusive uma delas sendo de extrema importância na indústria cinematográfica e de televisão. Vamos supor um documentário, ou filme, onde uma série de imagens e eventos surgem visualmente enquanto um narrador lê um texto. A realização do sincronismo entre áudio e vídeo é uma tarefa complicada. Muitas vezes, a gravação do texto é feita de forma completamente independente, até mesmo em estúdios separados, e cabe ao editor a tarefa de adequar entre si os diferentes trechos de video à voz.

Seria interessante, e extremamente útil, poder fazer com que o locutor falasse ligeiramente mais rápido ou devagar em certos trechos, de forma que a interação entre som e imagem ficasse mais adequadamente sincronizada.

Deseja-se então expandir ou comprimir temporalmente características sonoras perceptuais. Mas nossa percepção é ligada à forma com que ocorrem as variações de pressão, logo desejamos expandir e comprimir a ocorrência de tipos de variações.

Vamos imaginar então uma senóide pura com um segundo de duração, desejamos que através de uma transformação T ela dure dois segundos sem alteração perceptual. Para realizar isto T deve fazer com que ela oscile da mesma forma durante mais 1s, como é visto na Figura 5.8.

Seria possível tentar realizar algoritmos complexos para determinar como é o seu comportamento periódico, e, após isto, replicar seus períodos até "completar" o tempo

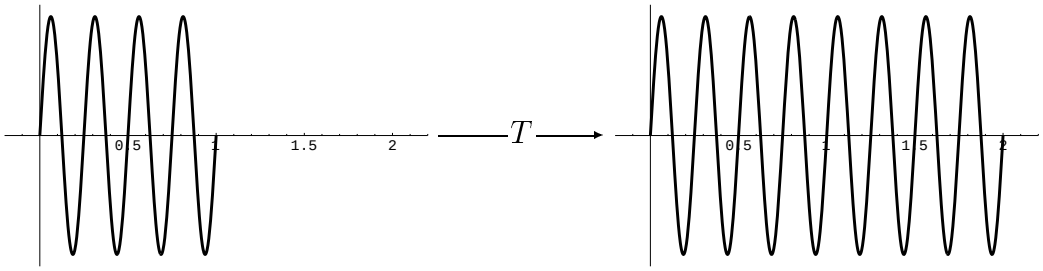


Figura 5.8: Expansão no tempo de uma senóide.

desejado. No caso de áudio, isso requer técnicas extremamente robustas de análise, reconstrução e síntese, o que não é uma tarefa fácil. Buscamos, portanto, um método mais imediato de realizar isto.

Sabemos que a representação tempo × freqüência nos informa o comportamento oscilatório da função analisada em uma região temporal. Exatamente que tipo de oscilações são essas vai depender da representação específica, onde transformadas de fourier com janela usam exponenciais complexas (senóides e cossenóides) e a transformada wavelet utiliza a própria função wavelet geradora.

O leitor deve observar que a expansão temporal de um átomo no domínio tempo × freqüência é responsável por uma replicação das freqüências deste átomo, conforme ilustrado na Figura 5.9.

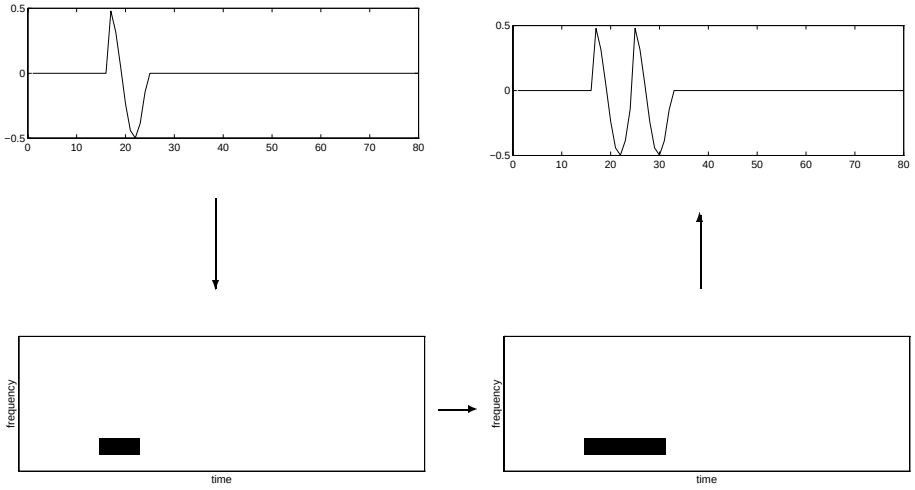


Figura 5.9: Processo de expansão no tempo.

A idéia então é replicar, ou estender, este comportamento oscilatório através então da

expansão do alcance de cada uma destas oscilações. Desta forma, podemos escrever T como

$$T(\widetilde{f}(t,\omega)) = \widetilde{f}(T(t), f),$$

onde T pode ser realizada de uma infinidade de formas, tanto lineares como não lineares, como foi visto no Capítulo 2. Uma forma simples seria levar o intervalo de tempo original de $[a,b]$ para $[c,d]$ através de uma transformação afim onde $T(a) = c$ e $T(b) = d$. Esta transformação é descrita pela expressão

$$T(t) = \frac{t-a}{b-a}(d-c) + c.$$

No universo de representação é necessário um pouco mais de cuidado, uma vez que a representação tempo × freqüência é composta naturalmente pelos coeficientes dos átomos utilizados na decomposição. A expansão ou compressão destes átomos implicará na realização de um processo de "resampling", descrito na Seção 3.3.

Exemplo 1. Apresentamos na Figura 5.10 um exemplo real de expansão e compressão temporal. De cima para baixo, observamos primeiramente a representação temporal do sinal original, em seguida sua representação no domínio tempo × freqüência (os valores de cinza representam a intensidade).

A transformação tempo × freqüência utilizada foi uma derivação da "Lapped Orthogonal Transform" (Malvar, 1992) descrita em (Wickerhauser, 1994) como "Local Trigonometric Transform". O tamanho das janelas foi especificado em 64 amostras com uma superposição de 32 e a função de "folding" foi do tipo senóide. Por último, o processo de reconstrução utilizado após o warp da imagem tempo × freqüência foi linear.

No terceiro gráfico observamos a representação tempo × freqüência estendida de 50% e no quarto a representação temporal do som resultante. Nos últimos dois gráficos observamos o processo semelhante, porém com uma compressão temporal de 50%.

O leitor pode observar que há realmente uma replicação de freqüências no processo. Se dando ao trabalho de contar as oscilações, vê-se que a parte harmônica do sinal transformado possui 50% a mais de oscilações no primeiro caso, e 50% a menos no segundo.

5.2.3 Deslocamento de Freqüências

Este tipo de transformação age sobre o eixo das freqüências de forma similar que a transformação da seção anterior sobre o eixo do tempo:

$$T(\widetilde{f}(t,\omega)) = \widetilde{f}(t, T(\omega)).$$

Em música um tipo de transformação que se enquadra nesta classificação é a chamada de *transposição*. A transposição muda o tom onde uma música está sendo executada (na Figura 5.11 observamos em (a) um trecho de uma peça simples na notação músical convencional, e em (b) o mesmo trecho, da mesma peça transposta de uma oitava para cima) e pode ser então examinada como uma "mudança" das freqüências.

Sabe-se que a razão entre as freqüencias de uma mesma nota musical em oitavas adjacentes é exatamente 2, e também que a razão entre as freqüências de notas separadas por um semi-tom pode ser, sob certas condições, considerada constante e igual a $\sqrt[12]{2}$.

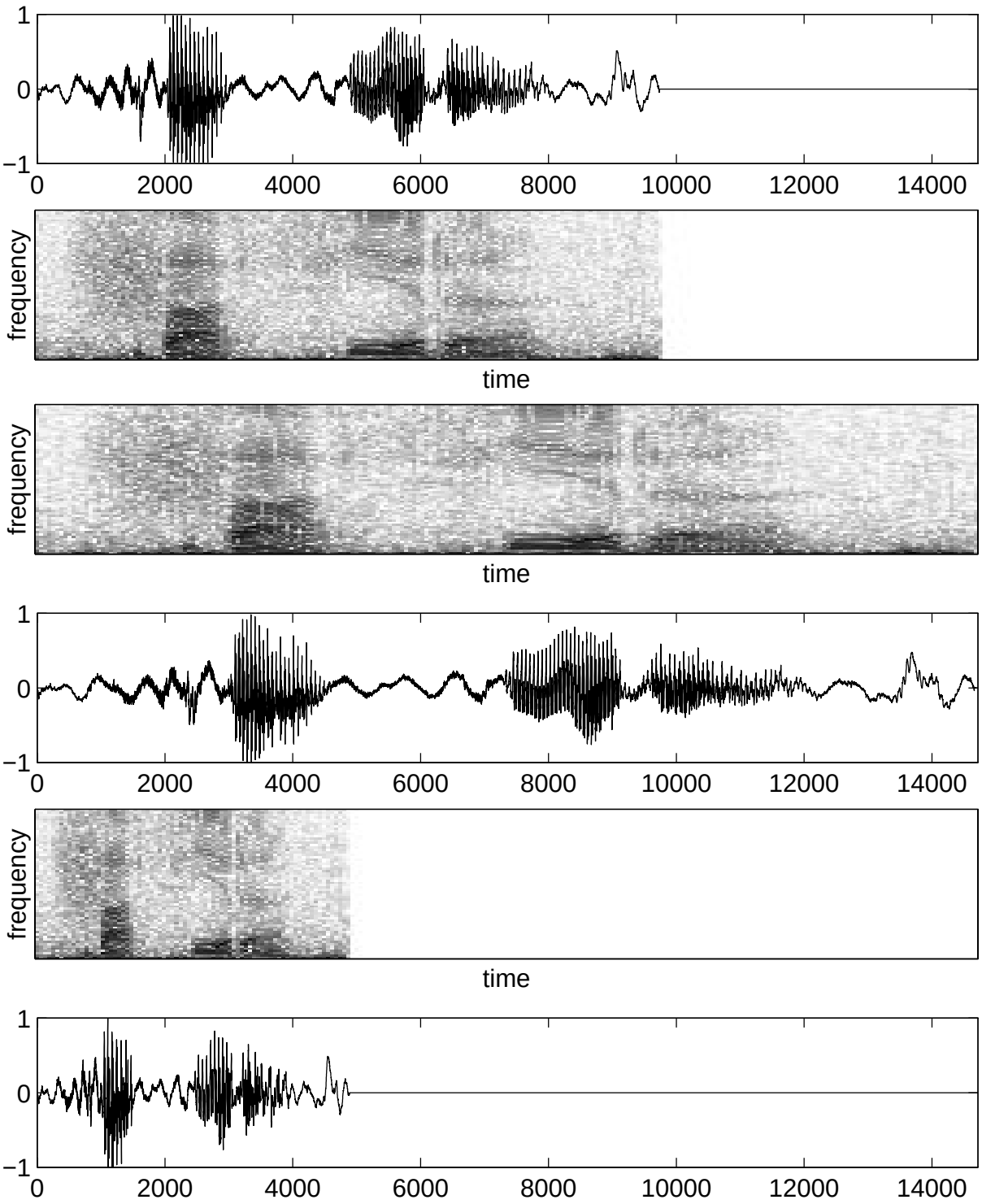
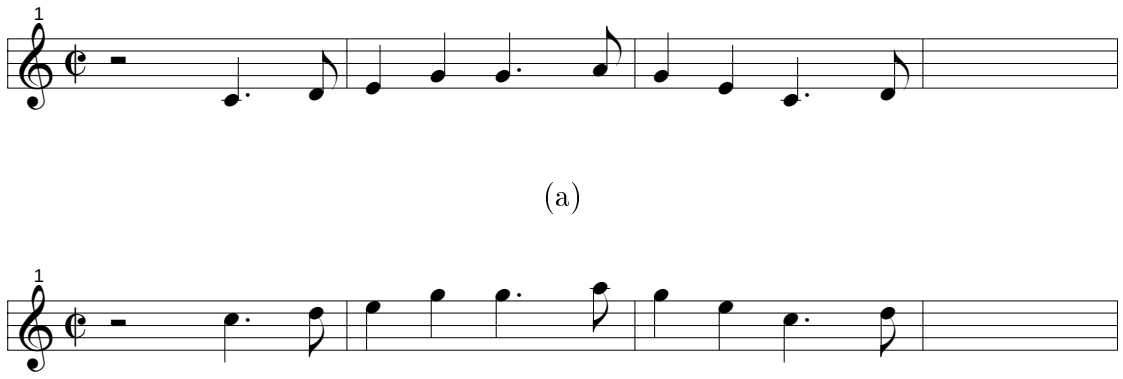


Figura 5.10: Expansão e compressão no tempo da palavra "safira".

Desta forma é possível determinar a razão r, que é constante, entre as freqüencias das notas de uma mesma peça executada em tons diferentes. Desta forma, a transformação pode ser escrita como:

$$\widetilde{f}(t, T(\omega)) = \widetilde{f}(t, r\omega).$$



(a)

(b)

Figura 5.11: Transposição de tom em música.

Novamente, é necessário cuidado no processo de resampling quando trabalhando no universo de representação. Além disto, mesmo quando considerando apenas o universo matemático, certas técnicas específicas exigem uma correção alteração de fase extra como é descrito em (Kronland-Martinet, 1988), (Arfib & Delprat, 1993), (A. Grossmann & Kronland-Martinet, 1987), (Kronland-Martinet & Grossmann, 1991) e (Arfib, 1991).

5.3 Metamorfose

Como nos métodos anteriores buscamos uma família de transformações que realize uma transformação contínua entre dois sons $\mathcal{O}_1$ e $\mathcal{O}_2$. Aqui, no entanto, serão utilizadas as representações tempo × freqüência na busca de resultados perceptuais mais próximos aos objetivos.

A metamorfose será decomposta em três etapas sucessivas:

1. Alinhamento temporal;
2. Alinhamento espectral;
3. Cross-dissolve.

5.3.1 Alinhamento Temporal

Na transformação utilizando segmentação, descrita no Capítulo 4, um grande feito foi realizado: o correto alinhamento entre características e eventos locais de cada um dos sons. Infelizmente o processo de deformação do suporte geométrico também criava localmente distorções espectrais indesejadas, obtendo-se assim um resultando perceptual não natural.

No entanto, vimos na Seção 5.2.2 que, com o auxílio de representações tempo × freqüência, é possível fazer ajustes nas durações dos sons com um mínimo de distorção espectral.

Considerando dois sons $\mathcal{O}_1([a,b], f_1)$ e $\mathcal{O}_2([c,d], f_2)$, suas respectivas segmentações temporais $a = t_0 < t_1 < \cdots < t_n = b$ e $c = s_0 < s_1 < \cdots < c_n = d$ e também suas representações tempo × freqüência $\widetilde{f}_1(t,\omega)$ e $\widetilde{f}_2(t,\omega)$, criamos como na Seção 4.2 duas famílias de warp T_1 e T_2, uma para cada som, subjugadas as segmentações que darão origem aos objetos $\widetilde{f}'_{1,\lambda}$ e $\widetilde{f}'_{1,\lambda}$ alinhados temporalmente:

$$\widetilde{f}'_{1,\lambda} = T_1(\widetilde{f}_1(t,\omega)) = \widetilde{f}_1(T_{1,i,\lambda}(t),\omega) \quad \text{e} \quad \widetilde{f}'_{2,\lambda} = T_2(\widetilde{f}_2(t,\omega)) = \widetilde{f}_2(T_{2,i,\lambda}(t),\omega),$$

onde T_1 e T_2 foram determinados na Equação 4.1.

Podemos observar na Figura 5.12 o que esta etapa busca realizar.

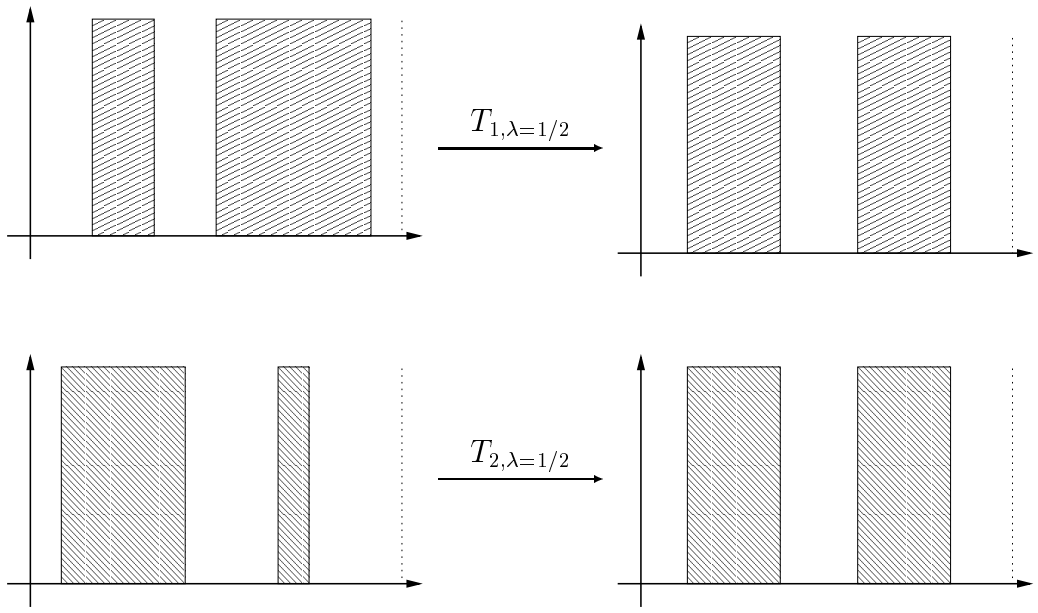


Figura 5.12: Alinhamento temporal.

5.3.2 Alinhamento espectral

É razoável esperar que se a segmentação temporal foi corretamente realizada, então dentro de cada "faixa vertical" não há grandes variações ao longo do tempo, em cada "região espectral". Isto é natural, uma vez que uma mudança brusca no comportamento espectral implicaria em uma mudaça também no comportamento da representação temporal. Logo, este ponto seria um sério candidato a dividir dois segmentos.

Assim como buscava-se um alinhamento no domínio do tempo, esta etapa busca relacionar entre si regiões distintas do espectro de freqüência de cada um dos pares de "faixas verticais" definidos na segmentação temporal.

Cria-se então, para cada um dos pares de faixas verticais, um par de segmentações na freqüência $0 = f_0 < f_1 < \cdots < f_n = f_{\max}$ e $0 = g_0 < g_1 < \cdots < g_n = g_{\max}$. A

interpretação e manipulação é semelhante ao caso temporal, sendo definidas as famílias $Q_{1,\lambda}$ e $Q_{2,\lambda}$:

$$Q_1(\widetilde{f}_1(t,\omega)) = \widetilde{f}_1(t, Q_{1,\lambda}(\omega)) \quad \text{e} \quad Q_2(\widetilde{f}_2(t,\omega)) = \widetilde{f}_2(t, Q_{2,\lambda}(\omega)).$$

Na Figura 5.13 observamos este comportamento.

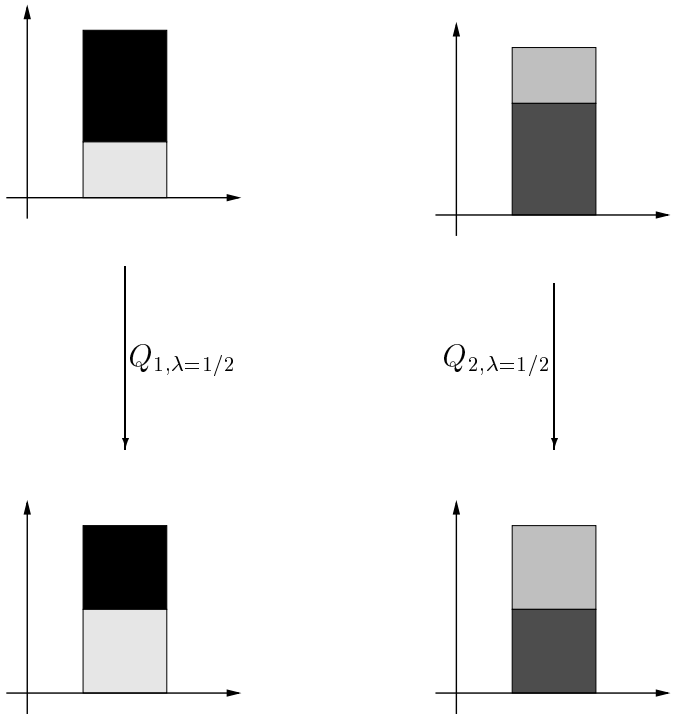


Figura 5.13: Alinhamento espectral.

5.4 Cross-Dissolve

Após a concatenação das transformações de alinhamento temporal e espectral obtemos duas famílias de objetos $\mathcal{O}_{1,\lambda}$ e $\mathcal{O}_{2,\lambda}$ que possuem a mesma região do plano tempo × freqüência , como suporte geométrico e uma mesma partição do plano, realizada através de retângulos com lados paralelos aos eixos. Isto está ilustrado na Figura 5.14. Em (a) e (c) estão as representações tempo × freqüência dos sinais originais, que sofrem então as transformações de alinhamento temporal $T_{1,\lambda}$ e $T_{2,\lambda}$ e de alinhamento espectral $Q_{1,\lambda}$ e $Q_{2,\lambda}$, resultando então nas representações de (b) e (d), cujas características discriminadas pelas segmentações temporal e espectral estão claramente "alinhadas".

Para a obtenção da representação tempo × freqüência do objeto final,x realizamos então um simples cross-dissolve entre $\mathcal{O}_{1,\lambda}$ e $\mathcal{O}_{1,\lambda}$ através de

$$\widetilde{f}_{3,\lambda}(t,f) = (1-\lambda)\widetilde{f}_{1,\lambda}(t,f) + \lambda\widetilde{f}_{2,\lambda}(t,f).$$

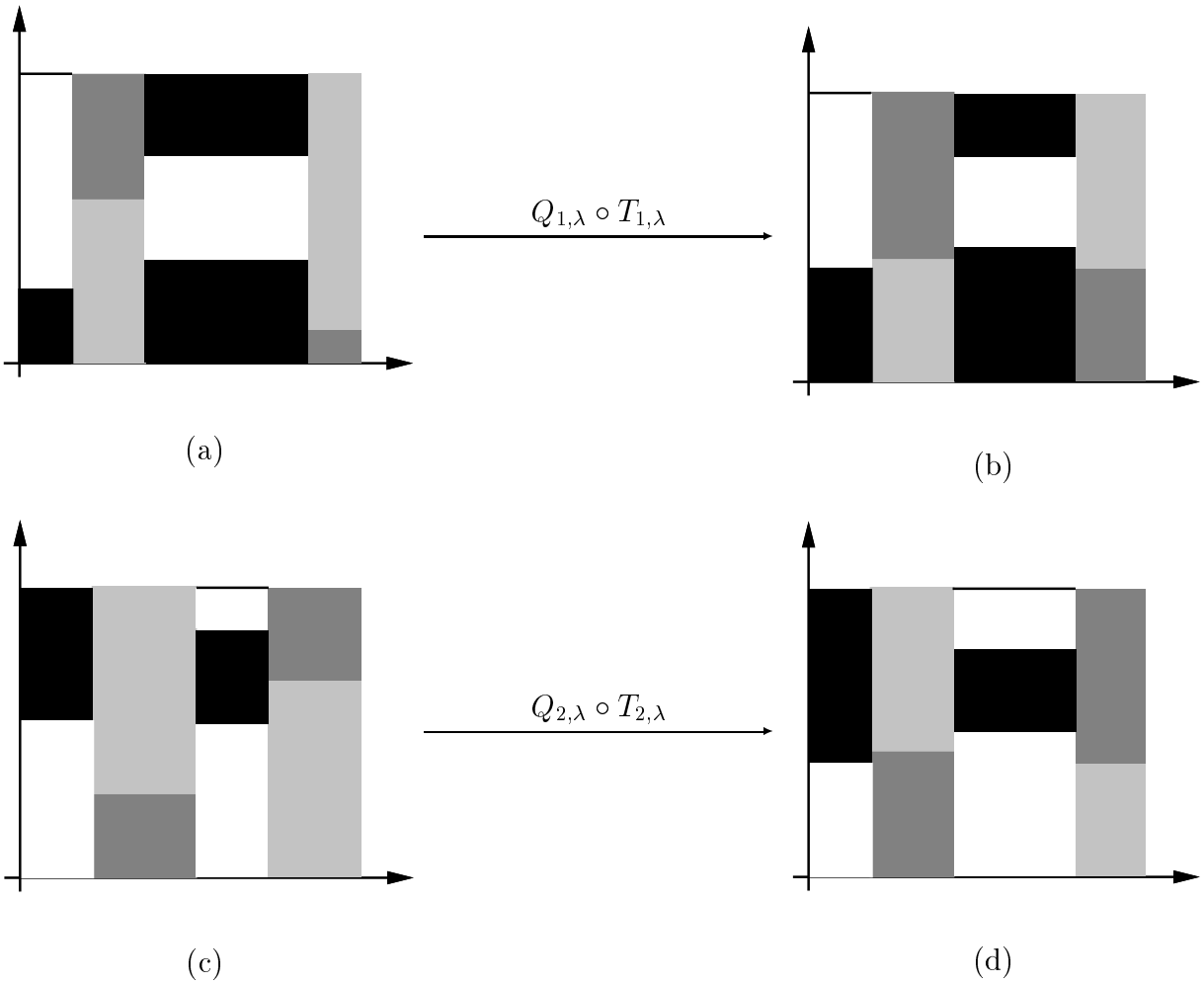


Figura 5.14: Efeito das transformações de alinhamento sobre as partições das representações tempo × freqüência.

5.5 Considerações Finais

Os resultados obtidos na prática são altamente dependentes tanto da transformação utilizada para representação tempo × freqüência como também das segmentações utilizadas no plano tempo × freqüência.

Mais experimentos são necessários para determinar a melhor transformada para cada tipo distinto de aplicação.

Como no caso de morphing de outros objetos gráficos, métodos automáticos e semi-automáticos para a segmentação dos sons originais devem ser desenvolvidos para proporcionar uma melhor manipulação dos objetos.

Capítulo 6

Conclusões

Ao longo desta dissertação caracterizamos sons como objetos gráficos, adquirindo assim uma série de ferramentas e paradigmas já estabelecidos em Computação Gráfica.

Três métodos para obtenção de metamorfose de sons foram estudados, sendo que o último necessita de um ferramental matemático mais complexo, as representações tempo × freqüência. Vimos também que uma série de outras transformações e aplicações são possíveis utilizando estas representações.

Compressão e Expansão no Tempo

A alteração da duração de um sinal de áudio sem introdução de distorções perceptuais, ou seja, a alteração das suas informações espectrais, é uma aplicação muito útil, existindo inclusive diversos softwares comerciais caríssimos que executam tal tarefa. Um exemplo típico de sua utilização é na alteração do comprimento de trechos de uma trilha sonora para o sincronismo com imagens, durante uma composição de um vídeo ou filme.

Observou-se no Capítulo 3 que embora uma simples deformação do domínio fosse capaz de alterar a duração, ela também distorcia as freqüências. Ao introduzir técnicas de manipulação de representações tempo × freqüência tornou-se então possível realizar este tipo de transformação conforme desejado.

Realização Uniforme

A primeira forma de realizar este tipo de operação é através de uma abordagem livre de contexto. Simplesmente efetua-se um warp uniforme sobre o eixo do tempo da representação tempo × freqüência do sinal de áudio original, como foi feito no exemplo 1 do Capítulo 5. Esta técnica já apresenta resultados bastante satisfatórios.

Realização Segmentada

Em certas circunstâncias esta abordagem não é suficiente. No caso de sinais de música, é necessário preservar certas propriedades do sinal responsáveis por diversas características perceptuais.

Uma das abordagens para descrição de uma nota gerada por um instrumento é utilizar, além da freqüência principal (que determina a nota propriamente dita) e das informações espectrais próprias que caracterizam o instrumento, uma função de envoltória.

A *função de envoltória* (chamada em inglês de "amplitude envelope") controla amplitude do sinal ao longo da duração da nota, e é classificada em três regiões: "Attack", "Decay", "Sustain" e "Release" , ilustradas de forma genérica na Figura 6.1. Estas regiões são resultado da construção física do instrumento, no caso de sons naturais, ou da sua modelagem matemática, no caso de sons sintetizados.

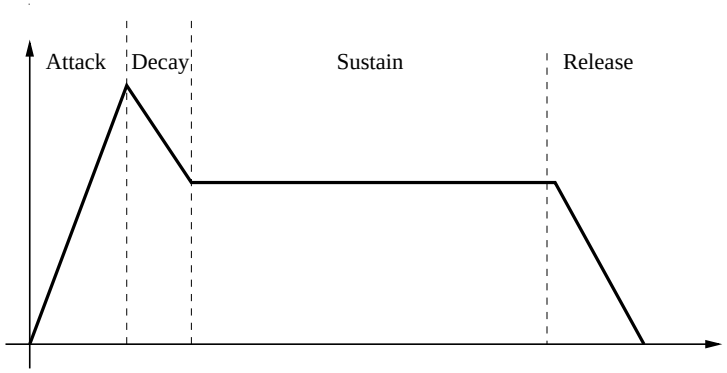


Figura 6.1: Regiões genéricas de uma função de envoltória.

O fato é que a duração das regiões de "attack", "decay" e "release" independem da duração que a nota assume. Portanto, isto deve ser levado em conta no processo de expansão temporal.

A simples expansão uniforme alteraria estas regiões. É necessário então utilizar uma expansão localizada apenas nas regiões relativas ao "sustain". Isto pode ser realizado com a técnica descrita na Seção 5.3.1.

Mesmo no caso de voz humana o correto seria usar este tipo de abordagem, já que certas consoantes com efeito "explosivo" ('b' e 't' por exemplo) não podem ter suas durações simplesmente ampliadas.

Metamorfose

A utilização de metamorfose de sons possui uma série de diferentes aplicações.

Através deste processo, a partir de vozes humanas, é possivel "criar" novas vozes que não seria possível de serem obtidas de outro modo. Um exemplo disto, como já foi mencionado, é o caso da voz do protagonista do filme "Farinelli - Il Castratum".

É possível também obter efeitos de transição interessantes, que tornam possível a integração da noção de metamorfose do som a uma seqüência de imagens. Seria possível criar um vídeo da metamorfose entre duas pessoas falando um mesmo texto. Conforme o rosto de um dos protagonistas fosse se transformando no rosto do outro, sua voz também iria se modificando gradativamente.

Em música, seria possível através de uma técnica de segmentação robusta, a partir de uma mesma peça tocada por dois músicos famosos, com diferentes estilos de interpretação, obter uma peça interpretada com um estilo intermediário.

Seria também possível a criação de "instrumentos" intermediários, embora já existam diversas técnicas com abordagens completamente distintas com bons resultados para esta questão.

Muito trabalho ainda existe nesta área, e diversas técnicas para metamorfose ainda virão a surgir. A maior dificuldade da técnica utilizando representações tempo × freqüência é a determinação das segmentações, tanto temporal como espectrais. A qualidade e os resultados que serão obtidos dependem muito de como as segmentações são posicionadas.

Uma questão que foi pouco estudada nesta tese é a adequabilidade de cada tipo de representação tempo × freqüência para cada uma das diferentes aplicações (voz, música, efeitos especiais, etc.). Caberia ainda um intenso estudo perceptual, analisando os resultados obtidos a partir de transformações utilizando as mesmas segmentações, porém realizadas em distintas representações tempo × freqüência.

Segmentações e Outras Considerações

Até o momento, existe muito pouca coisa na área de segmentação automática de sons; podemos citar (van Hemert, 1991), (Wesfreid & Wickerhauser, 1993) (Svendsen & Soong, 1987) e (Li & Gibson, 1996) .

Geralmente, as técnicas desenvolvidas não são genéricas o suficiente para qualquer tipo de aplicação, utilizando características e propriedades muito específicas do tipo de sinal em questão, como. por exemplo, a identificação da mudança dos formantes em um sinal de voz.

Sistemas automáticos, ou semi-automáticos, podem então ser desenvolvidos para simplificar ou até mesmo eliminar a intervenção humana no processo genérico de identificação de "features".

Uma dificuldade maior surge quando cada som original é composto por diferentes componentes. Por exemplo, uma orquestra, ou não tão complicado assim, mas ainda com complexidade bem mais ampla: o caso de um piano, onde inúmeras notas coexistem a cada instante de tempo. O processo de segmentação deve, de certa forma, mapear eventos distintos em átomos distintos na sua decomposição. Para isso, o uso de representações tempo × freqüência adaptadas pode ser útil ("wavelet packet" ou "cossine packet" por exemplo).

A abordagem tomada nesta dissertação, de primeiro segmentar e transformar temporalmente a imagem tempo × freqüência, para só então deformar espectralmente cada um destes segmentos, pode não ser sempre suficiente. Poderia ser necessário sincronizar características espectrais distintas em instantes de tempo diferentes. O caso da metamorfose entre os estilo de interpretação de dois pianistas ilustra bem este esquema, quando o intervalo entre distintas notas que ocorrem simultaneamente varia de forma diferente.

Essencialmente, o que foi feito neste trabalho foi a transformação do som em uma imagem, onde as suas duas dimensões são relacionadas com fenômenos característicos de sons: tempo e freqüência. Desta forma, procuramos utilizar de processos de transformação de imagens já estabelecidos em computação gráfica, como o morph, para a obtenção de transformações no som.

Seria interessante investigar os efeitos sobre sons de outros tipos de transformações nas suas representações tempo × freqüência.

Apêndice A

Implementação

A estrutura criada para implementação se baseia no princípio de independêcia e modularidade.

Diversas bibliotecas básicas, de suporte, foram criadas para a manipulação das estruturas de dados mais centrais do problema, como sons, segmentações e representações tempo $\times$ freqüência.

O problema de morph é solucionado pelo uso em conjunto de diversos pequenos programas que funcionam independentemente e resolvem apenas pequenas etapas de cada vez. Desta forma, é possível substituir determinados módulos por outros similares sem a necessidade de recompilar todo o código. Experimentos com diferentes combinações de módulos também podem ser realizados, dando uma maior flexibilidade para o experimentação.

O projeto foi organizado em uma estrutura de diretórios mais complexa do que a necessária até o momento, porém que permitirá facilmente a sua expansão no futuro. Podemos observá-la na Figura A.1.

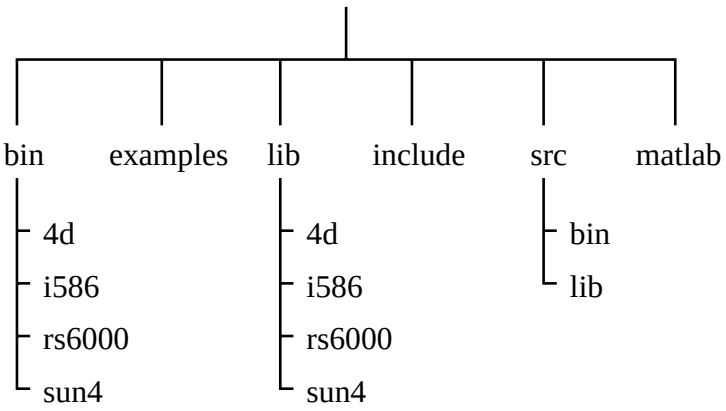


Figura A.1: Estrutura de diretórios.

No diretorio `src` ficam localizados todos os arquivos do código fonte, divididos ainda em dois subdiretórios: `bin` e `lib`. No primeiro ficam localizados os arquivos responsáveis pelos programas em si e, no segundo, os arquivos necessários para a criação das bibliotecas básicas de suporte.

Sob `include` ficam localizadas os "headers", tanto dos programas como também das bibliotecas.

Após a compilação, as bibliotecas são movidas para o diretório `lib` e os programas executáveis para o diretório `bin`. No entanto, é necessário lembrar que há uma série de subdiretórios embaixo deles, um para cada arquitetura (o nome é obtido através do comando `arch`).

Resta ainda o diretório `examples`, onde exemplos importantes para demonstração são armazenados, e o diretório `matlab`, onde ficam programas e experimentos feitos em matlab antes da implementação em `C`.

A.1 Bibliotecas Básicas de Suporte

Foram criadas 6 diferentes bibliotecas básicas:

libgen É a biblioteca de funções gerais. Possui rotinas de alocação de memória já com teste de sucesso e de uso comum a algumas outras, como procura de um arquivo pelo `path` do usuário. Todas as rotinas desta biblioteca possuem o prefixo identificador "`Gen_`".

libsnd Manipulação de arquivos e estruturas de som. Suas principais rotinas são as responsáveis pela leitura e escrita de arquivos de som dos tipos "`wav`" e "`au`". Possuem o prefixo "`Snd_`", com exceção das rotinas de conversão para os formatos de quantização μ-*law* e *a-law*, que foram pegos sem alteração do arquivo `g711.c`, disponibilizado ao público pela *Sun Microsystems, Inc.*

libseg Leitura e gravação de arquivos que descrevem segmentações tempo $\times$ freqüência de um som. Suas rotinas possuem no nome o prefixo "`Seg_`".

libdct Rotinas para o cálculo rápido da transformada cosseno direta e inversa. Estas rotinas são de autoria de Henrique S. Malvar, e estão descritas em (citar o livro do malvar). Seus nomes originais não foram alterados em respeito aos desejos do autor.

libtf Rotinas responsáveis pela gravação e leitura de arquivos que descrevam uma representação tempo $\times$ freqüência. É importante ressaltar que o formato de arquivo desenvolvido foi em texto puro, tornando mais símples a sua manipulação e implementação (não há desta forma o problema de diferentes ordems dos bytes para diferentes arquiteturas de computadores). O preço pago por isso é alto: surgem arquivos *muito* grandes, e o tempo necessário para sua gravação e leitura pode ser alto. Os nomes das rotinas possuem o prefixo "`TF_`".

libwt Rotinas para o cálculo da transformada rápida de wavelet. Esta parte do código ainda não está congelada, e ainda vem recebendo diversas alterações. Seu prefixo é "`WT_`".

A.1.1 Estruturas de Dados

Os programs e bibliotecas necessitam basicamente de apenas três estruturas de dados distintas: uma para a representação de sons, uma para a representação de segmentações e uma para representações tempo × freqüência.

Uma possível opção de projeto seria unificar tudo em uma única estrutura, uma vez que a segmentação, mesmo que apenas em última instância, será relacionada a um som, e a imagem tempo × freqüência pode ser considerada uma espécie de representação alternativa. Foi no entanto dada preferência a independência entre elas, desta forma cada programa modular só necessita manipular dados que a ele interessam.

Som

A seguir vemos o pedaço de código extraído do arquivo header `snd.h` responsável pela definição da estrutura de dados do som.

```
typedef  struct glob_head  Sound;
typedef  float             Data;
typedef  struct local_head Piece;

struct  glob_head
{
  char   *name;         /* the name of the file      */
  char   *comments;     /* some comments about it    */
  float  length;        /* total time lenght in sec. */
  int    number;        /* number of pieces          */
  char   same_sr;       /* all pieces of same sr     */
  Piece  **chunk;       /* pointer to vector of pnts */
};


struct  local_head
{
  long int   sr;        /* local sampling rate       */
  int        nchannel;  /* local number of channels  */
  long int   size;      /* local numb. of samples    */
  float      llenght;   /* local lenght in sec.      */
  Data       **data;    /* Data. May have mult. chan */
};
```

Esta estrutura foi concebida de forma a permitir uma segmentação acoplada, onde cada segmento pode inclusive possuir uma taxa de amostragem distinta.

Cada "segmento" seria representado por uma estrutura do tipo "Piece". O som pode possuir múltiplos canais, cada canal é armazenado em um vetor distinto. Cada "Piece" pode possuir um número diferente deles.

Até o momento a estrutura de dados vem sendo "sub-utilizada", pois não tem sido utilizada a possibilidade de múltiplos segmentos, mesmo quando trabalhando com segmentações.

Segmentação

A estrutura utilizada para representar segmentações é definida no arquivo `seg.h`, abaixo extraimos dele apenas o trecho que a define.

```
#ifndef __MY_REAL__
#define __MY_REAL__ 1
typedef float       real;
#endif

typedef struct segm  Segm;

struct segm
{
  int   ntseg;       /* number of segments (marks - 1) */
  real  *tlim;       /* Vector with time boundaries */
  int   *nfseg;      /* Vector with number of freq. sub segm */
  real  **flim;      /* Vector of vectors of freq. boundaries */
};
```

A segmentação temporal é tratada apenas como um vetor de números reais, ordenados e inclusos no intervalo entre zero e um.

A estrutura guarda o número de segmentos (`ntseg`), portanto o vetor de números possuirá `ntseg+1` elementos.

A segmentação no domínio da freqüência é subjugada a cada "segmento temporal", e possui analogia semelhante. No vetor `nfseg` encontra-se o número de segmentos na freqüência para cada segmento temporal.

Representação Tempo × Freqüência

A estrutura usada para implementação de representações tempo × freqüência é definida no seguinte trecho de código removido do arquivo `tf.h`.

```
typedef struct tf    TF;

struct tf
{
  int   nt;                    /* number of time slices */
  int   nf;                    /* number of freq. slices */
  int   begt;                  /* time slice where sound begins */
  int   sizt;                  /* real length of sound */
  int   st;                    /* factor of scale in t */
  int   sf;                    /* factor of scale in f */
  real  max;                   /* max value of TF */
  real  min;                   /* min value of TF */

  real  *tf;                   /* The TF itself */
};
```

É importante ressaltar que o tipo `real` é antes definido como `float`, da mesma forma "condicional" que foi mostrado na seção anterior.

A estrutura acima descrita armazena uma imagem retangular, uniformemente espaçada. Certos cuidados são necessários para sua utilização com a transformada wavelet por exemplo, onde os átomos de decomposição não são todos iguais no plano tempo × freqüência, caso onde não há um mapeamento direto entre cada átomo da decomposição com cada elemento da estrutura.

As variáveis `nt` e `nf` indicam as dimensões da matriz armazenada, e `st` e `sf` os fatores de escala de cada um dos eixos. Desta forma é possível representar regiões que se sabe de antemão que são constantes por apenas um valor, como no caso da transformada de fourier com janela.

Em diversas situações será necessário estender o comprimento do som original para que ele possua uma dimensão temporal compatível com a representação tempo × freqüência em questão. Por exemplo, em uma transformada de Fourier com janela é necessário que o som original tenha um número de amostras múltiplo do tamanho da janela, já no caso da transformada wavelet, este tamanho deve ser uma potência de dois. Esta extensão por sua vez pode ser realizada de diferentes maneiras, antes do início do sinal, após seu término, etc. Por causa de tudo isso existem dois campos especiais: `begt` e `sizt`, que servem para explicitar que parcela da imagem tempo × freqüência em questão representa realmente o sinal original. Todas as duas são dadas em termos de amostras originais, portanto é preciso utilizá-las em conjunção com `nt` e `nf` escaladas de `st` e `sf`.

Para facilitar e acelerar uma série de cálculos são também guardados explicitamente os valores máximo e mínimo dos coeficientes.

A.2 Programas Clientes

O processo de metamorfose de dois sons utilizando as suas representações tempo × freqüência pode ser decomposto em diferentes etapas, que são de certa forma independentes.

Optou-se então pela realização de um projeto modular, onde cada etapa é realizada por um programa independente, que pega as informações e dados necessários de arquivos, e grava seus resultados em arquivos.

Na Figura A.2 encontra-se o diagrama de interconecção entre os módulos.

Devido a modulariadade é muito simples utilizar tanto a transformada de fourier com janela como também a transformada wavelet, para isto basta utilizar diferentes módulos de conversão entre a representação temporal usual e a representação tempo × freqüência.

As setas pontilhadas apontando para o módulo de segmentação servem para indicar a "disponibilidade", o que não implica em nenhum instante que o módulo é obrigado a utilizar os dados disponíveis.

Alguns programas foram criados para ocupar estas lacunas:

- `au2dct`;
- `dct2au`;
- `au2wt`;
- `wt2au`;

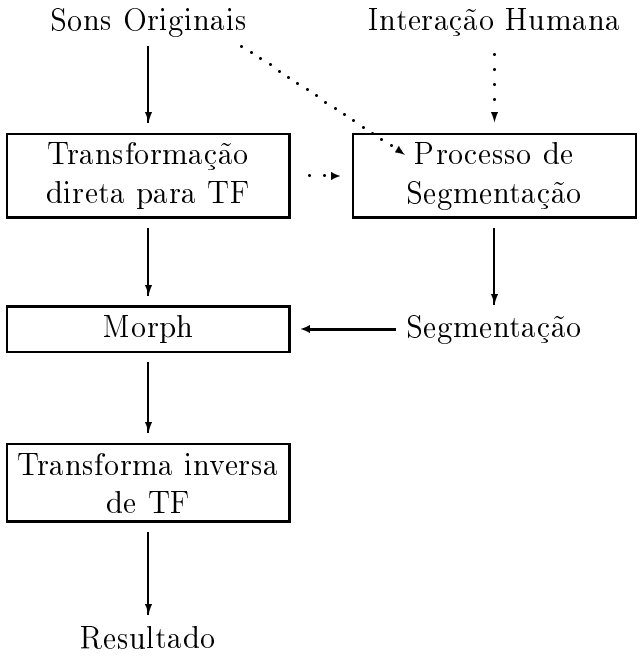


Figura A.2: Estrutura modular do sistema.

- `edit-seg`;
- `mor-tf`;
- `v-tf`.

Conversão. Os programas `au2dct`, `dct2au`, `au2wt` e `wt2au` são responsáveis pelas transformações direta e inversa para as representações tempo × freqüênciada transformada de fourier com janela e transformada wavelet.

Metamorfose. O programa `mor-tf` é responsável pelas realização do morph, combinando as operações de warp e cross-dissolve das representações tempo × freqüência. Note que o resultado pode não ser uma "representação" válida de imediato, uma vez que podem passar a existir variações dentro do que seriam os átomos, e portanto regiões constantes, da representação. O programa responsável pela transformada inversa *deve* saber lidar com isso.

Segmentação. É necessário alguma ferramenta para criação das segmentações a serem utilizadas. O programa `edit-seg` foi criado com esse intuito, e depende completamente da interação humana. O usuário é cria um par compatível de segmentações baseado nas representações tempo × freqüênciados dois sons desejados. Note que neste caso específico, o módulo não usa a representação temporal do som, da mesma forma que um dispositivo segmentação automática não usaria a intervenção do usuário.

Outros Existe mais um programa de auxílio, o `v-tf`, que serve unicamente para visualizar uma representação tempo × freqüência.

Referências Bibliográficas

A. Grossmann, J. Morlet & Kronland-Martinet, R. 1987. Analysis of sound patterns through wavelet transform. *International Journal on Pattern Analysis and Artificial Intelligence*, January.

Antoniou, Andreas. 1993. *Digital Filters*. 2^{nd} edn. McGraw-Hill.

Arfib, D. & Delprat, N. 1993. Musical transformations using the modification of time-frequency images. *Computer Music Journal*, **17**(2), 66–72. (Sound examples on soundsheet with 13(1) 1989).

Arfib, Daniel. 1991. Analysis, Transformation, and Resynthesis of Musical Sounds with the Help of Time-Frequency Representation. *Pages 87–118 of:* Poli, Giovanni De; Piccialli, Aldo & Roads, Curtis (eds), *Representations of Musical Signals*. MIT Press.

Cohen, Leon. 1989. Time-Frequency Distributions - A Review. *Proceedings of the IEEE*, **77**(7), 941–981.

Daubechies, Ingrid. 1990. The Wavelet Transform, Time-Frequency Localization and Signal Analysis. *IEEE Transactions on Information Theory*, **36**(5), 961–1005.

Daubechies, Ingrid. 1992. *Ten Lectures on Wavelets*. SIAM.

Gomes, Jonas & Velho, Luiz. 1995a. Abstraction paradigms for computer graphics. *The Visual Computer*, **11**(5), 227–239.

Gomes, Jonas & Velho, Luiz. 1995b. *Computação Gráfica: Imagem*. IMPA-SBM.

Gomes, Jonas & Velho, Luiz. 1997. *Image Processing for Computer Graphics*. New York: Springer Verlag.

Gomes, Jonas; Costa, Bruno; Darsa, Lúcia & Velho, Luiz. 1995. *Deformação e Metamorfose de Objetos Gráficos*. IMPA. 20 Colóquio Brasileiro de Matemática.

Gomes, Jonas; Costa, Bruno; Darsa, Lucia & Velho, Luiz. 1996. Graphical Objects. *The Visual Computer*, **12**, 269–282.

Gomes, Jonas; Costa, Bruno; Darsa, Lúcia; Velho, Luiz & Beier, Thaddeus. 1997a. *Warping and Morphing of Graphical Objects*. ACM Press. SIGGRAPH'97 Course Notes.

Gomes, Jonas; Velho, Luiz & Goldenstein, Siome. 1997b. *Wavelets: Teoria, Software e Aplicações*. IMPA. 21 Colóquio Brasileiro de Matemática.

Hlawatsch, F. & Boudreaux-Bartels, G. G. 1992. Linear and Quadratic Time-Frequency Signal Representation. *IEEE SP Magazine*, April, 21–67.

Holmes, J. N. 1993. *Speech Synthesis and Recognition.* Chapman & Hall.

Kronland-Martinet, Richard. 1988. The Use of the Wavelet Transform for the Analysis, Synthesis and Processing of Speech and Music Sounds. *Computer Music Journal*, **12**(4), 11–20. (Sound examples on soundsheet with 13(1) 1989).

Kronland-Martinet, Richard & Grossmann, Alex. 1991. Application of Time-Frequency and Time-Scale Methods (Wavelet Transforms) to the Analysis, Synthesis, and Transformation of Natural Sounds. *Pages 45–85 of:* Poli, Giovanni De; Piccialli, Aldo & Roads, Curtis (eds), *Representations of Musical Signals.* MIT Press.

Lathi, Bhagwandas P. 1974. *Signals, Systems, and Control.* Harper & Row.

Li, Ta-Hsin. 1996. Discrimination of Time Series by Parametric Filtering. *Journal of the American Statistical Society*, **91**(433), 284–293.

Li, Ta-Hsin & Gibson, Jerry D. 1996. Speech Analysis and Segmentation by Parametric Filtering. *IEEE Transactions on Speech and Audio Processing*, **4**(3), 203–213.

Malvar, Henrique S. 1992. *Signal Processing with Lapped Transforms.* Norwood, MA: Artech House.

Oppenhein, Alan V. & Willsky, Alan S. 1983. *Signal and Systems.* Prentice-Hall inc.

O'Shaughnessy, Douglas. 1987. *Speech Communication, Human and Machine.* Addison-Wesley.

Rabiner, Lawrence R. & Schafer, Ronald W. 1978. *Digital Processing of Speech Signals.* Prentice-Hall Inc.

Roads, Curtis. 1996. *The Computer Music Tutorial.* Mit Press.

Rowden, Chris. 1991. *Speech Processing.* McGraw-Hill.

Souza, M. N. & Caloba, L. P. An Analytical Auditory Wavelet. *Submited.*

Souza, M. N. & Caloba, L. P. A Comparison between Fourier and Biological Auditory Based Time-Frequency Distributions, Applied to the Speech Signals. *Submited to Speech Communications.*

Souza, M. N. & Caloba, L. P. 1995. A Study of a Cochlear Simulator. *Pages pp. 376–380 of:* 38^{th} *Midwest Syposium on Circuits and Systems.*

Strang, Gilbert & Nguyen, Truong. 1996. *Wavelets and Filter Banks.* Wellesley, MA: Wellesley-Cambridge Press.

Svendsen, Tobjørn & Soong, Frank K. 1987 (April). On the Automatic Sementation of Speech Signals. *Pages 77–80 of: Proc. of ICASSP.*

Vaidyanathan, P. P. 1993. *Multirate Systems and Filter Banks*. Englewood Cliffs, New Jersey: Prentice Hall PTR.

van Hemert, Jan P. 1991. Automatic Segmentation of Speech. *IEEE Transactions on Signal Processing*, **39**(4), 1008–1012.

Vetterli, Martin & Kovacevic, Jelena. 1995. *Wavelets and Subband Coding*. Englewood Cliffs, New Jersey: Prentice Hall PTR.

Wesfreid, Eva & Wickerhauser, Mladen Victor. 1993. Adapted Local Trigonometric Transforms and Speech Processing. *IEEE Transactions on Signal Processing*, **41**(12), 3596–3600.

Wickerhauser, Mladen Victor. 1994. *Adapted Wavelet Analysis from Theory to Software*. Wellesley, MA: A. K. Peters.

Wolberg, G. 1990. *Digital Image Warping*. IEEE Computer Society Press.