

Acompanhamento Computacional Parte II

Mariana Zaparoli Martins
Computação Musical – IME USP

Roteiro

■ Novas Técnicas para Aumentar a Qualidade do Acompanhamento Computacional

Roger B. Dannenberg (Carnegie Mellon University) e Hirofumi Mukaino (Yamaha)

- Introdução
- Estrutura Básica
- Múltiplos Matchers
- Trilos e Glissandos
- Decisões Atrasadas

Roteiro

- Outras melhorias
- Conclusões

- Bibliografia

Novas Técnicas para Aumentar a Qualidade do Acompanhamento Computacional

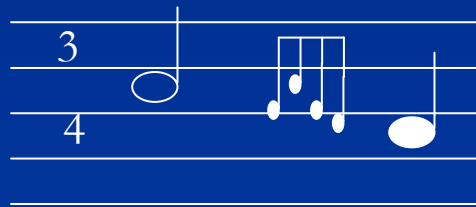
Roger B. Dannenberg e Hirofumi Mukaino
The Yamaha Corporation
Hamamatsu, Japão

Introdução

- **Objetivo:** sistema de acompanhamento computacional polifônico confiável
- **Melhorias**
 - Uso de uma quantidade limitada de paralelismo
↳ duas hipóteses
 - Atraso de decisão
 - Métodos para manipulação de ornamentos na partitura: trilos, glissandos e *grace notes*

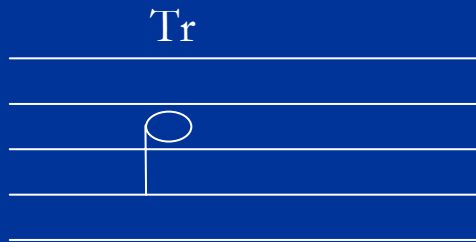
Introdução

■ Grace Notes

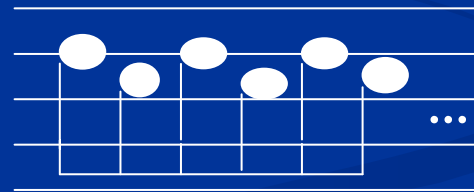


■ Trilo

■ Escreve



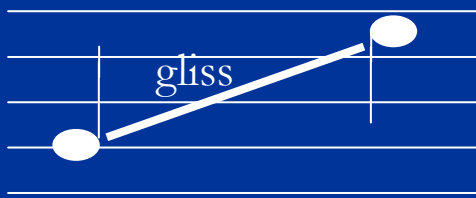
Soa



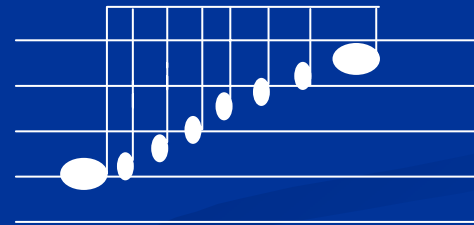
Introdução

- Glissando

- Escreve



- Soa



- Todas as notas são pré-definidas no sistema



Estrutura Básica

- Definição de termos
 - **Partitura:** conjunto de eventos no tempo
 - A partitura armazenada é composta em duas partes
 - **partitura solo:** tocada pelo performista
 - **partitura de acompanhamento:** tocada pelo sistema
 - **Tempo na partitura:** tempo virtual
 - Um **evento composto** é um grupo de eventos que são tocados quase ao mesmo tempo.
 - Acorde

Estrutura Básica

- O sistema tem três partes importantes:
 - Pré-processador
 - Processa a entrada do MIDI-IN
 - Monta eventos compostos usando informações de tempo
 - Detecta e processa trilos e glissandos
 - Matcher
 - Compara a performance com a partitura solo armazenada
 - Utiliza apenas um pedaço da partitura em um dado tempo, chamado de janela (*window*)
 - Reporta os casamentos a parte seguinte

Estrutura Básica

- Acompanhante
 - Toca a parte do acompanhamento da partitura
- Se o músico não está tocando a partitura que está na janela?
 - O Matcher não conseguirá acompanhar...
 - Aumenta o tamanho da janela?
 - **Não!!!**
 - Tempo de processamento proporcional ao tamanho da janela

Estrutura Básica

- Se o solista parar de tocar?
 - O acompanhamento continuará independentemente até o solista retornar
- Em que parte ele recomeçará a tocar?
 - Onde tinha parado??
 - Em sincronia com o acompanhamento???
- Múltiplos Matchers

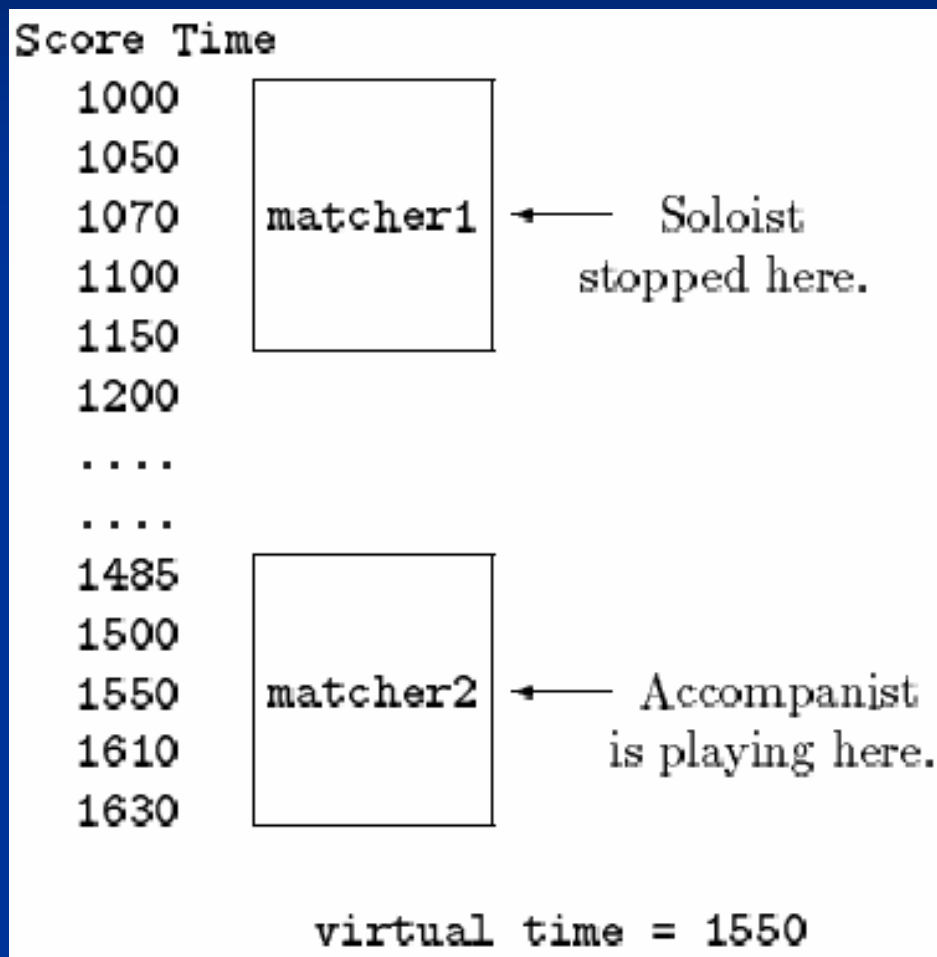
Múltiplos Matchers

- Considera apenas um subconjunto da partitura (janela) em um dado momento
- Matcher como objeto:
 - Primeiramente: algoritmo que encontra o casamento entre a performance e a partitura armazenada
 - Agora: objeto, se pode criar quantas instâncias se quiser
 - Pode-se criar quando necessário
 - Liberar quando se tornar desnecessário

Múltiplos Matchers

- Criando Matchers
 - Muitos Matchers executando ao mesmo tempo?
 - **Não!!!**
 - Invocar apenas na incerteza da posição do solista
 - Quando ele para de tocar
 - Tocar muitas notas extras
 - Checa-se o tempo virtual do acompanhamento e o da janela do Matcher
 - Se o tempo virtual corrente não está na janela
 - └─> novo Matcher

Múltiplos Matchers



Múltiplos Matchers

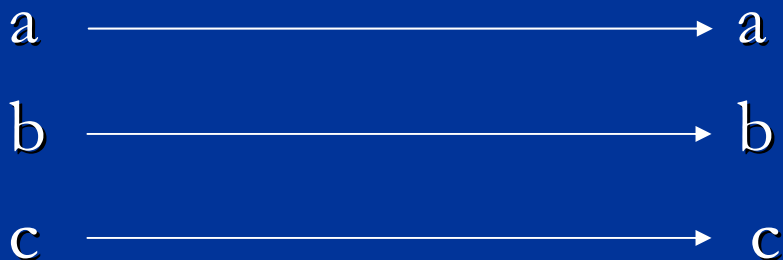
- Terminando um Matcher
 - Quando um dos matchers encontra um casamento, o outro é terminado
 - Mantém-se o objeto em um estado inativo até que ele seja necessário novamente

Trilos e Glissandos


- Considerando evento composto como um evento
 - Mapeamento um-para-um (performance/partitura)

- Solo

Partitura



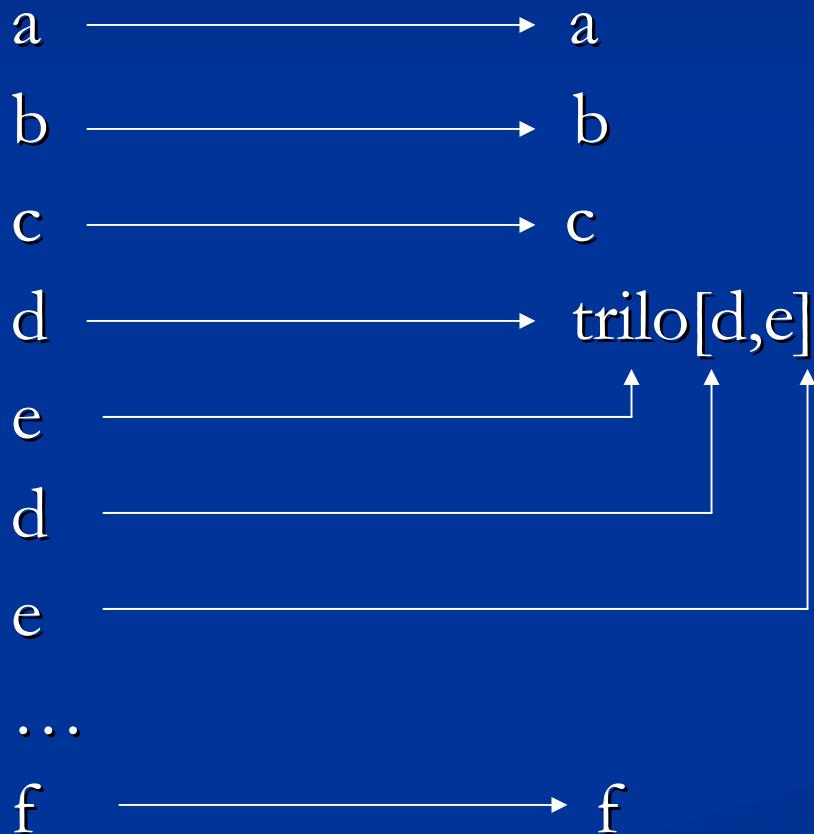
Trilos e Glissandos

- Mas isso não é verdade pra trilos e glissandos...
 - Dependem do instrumentista 
 - Não são especificados precisamente na partitura
 - Exemplo: $\text{trilo}[a,b] = [a,b,a,b,a,b,a,b,a,b]$?
 - **Não sei!**
 - Quantas iterações terá? Depende do instrumentista...
 - Mecanismo de mapeamento muitos-para-um

Trilos e Glissandos

■ Solo

Partitura

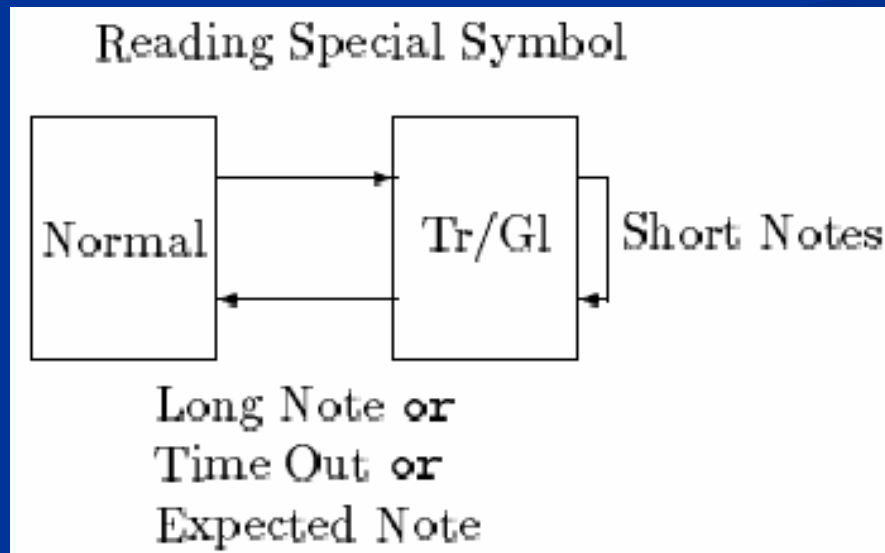


Trilos e Glissandos

- Para evitar:
 - Aumento de complexidade do Matcher
 - Diminuição da performance
- Usa-se o Pré-processador para fazer um evento composto especial para o trilo usando alguma ajuda da partitura
 - Pré-processador converte trilo em evento simples

Trilos e Glissandos

- Pré-processador
 - Máquina de estados finita
 - Dois estados: Normal e Trilo/Glissando
 - Tr/Gl partitura e tempo esperado do ornamento



Trilos e Glissandos

- Pré-processador no estado Trilo/Glissando
 - Manda um símbolo especial ao Matcher
 - E enquanto ficar neste estado, não manda mais informação nenhuma ao Matcher

performance	preprocessor	score
	->matcher	
a	a	a
b	b	b
c	c	c
d	Tr/GL	Tr/GL
e		
d		
e		
d		
f	f	f

Trilos e Glissandos

- Informando ao Pré-Processador
 - Toda vez que o Matcher faz um casamento, ele pode prever a próxima nota
 - Se for um símbolo especial
 - Matcher avisa o pré-processador
- **transição de estado**

Decisões Atrasadas

- Erro de manipulação na entrada pelo Pré-processador
 - Envio de notas extras ao Matcher
 - Possibilidade de se encontrar o casamento errado
 - Prevenção: descartar as mensagens do Matcher
 - O Acompanhante pode confiar no casamento da n -ésima nota se o casamento da $n-1$ -ésima foi reportada recentemente

Decisões Atrasadas

- Casamentos consecutivos
 - Aumentam a confiança do acerto do acompanhamento
 - Caso contrário, suspeita-se do casamento
 - Solista errou recentemente....
 - O Matcher atrasa a mensagem para o Acompanhante
 - Se o solista estiver tocando um ornamento, a próxima nota virá rapidamente

Decisões Atrasadas

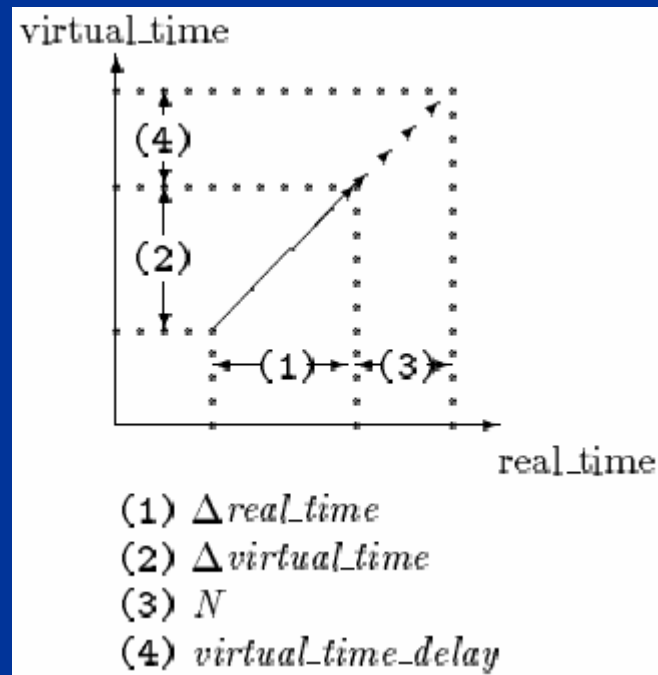
- Pré-processador irá pegar a nota com atraso
 - Perceber que tem algo errado (Matcher não casou a nota)
 - A mensagem atrasada é cancelada
 - Mas se nada aconteceu para a mensagem ser cancelada, ela é mandada ao Acompanhante
- Atrasa-se reportar um casamento apenas em caso de casamento duvidoso
 - E apenas por um período de cerca de 100ms

Decisões Atrasadas

- Caso suspenda-se um casamento N ms
- E a velocidade é definida como:
 - $\text{Velocidade} = \Delta \text{ tempo virtual} / \Delta \text{ tempo real}$
- O atraso no tempo virtual será:
 - $\text{Atraso_tempo_virtual} = N * \text{velocidade}$
- Se o casamento, cujo
 - $\text{Tempo_virtual} = \text{tempo_virtual_do_casamento} + \text{atraso_tempo_virtual}$

Decisões Atrasadas

- Ocorreria em:
 - $\text{Tempo_real_do_evento} + N$ no tempo real



Decisões Atrasadas

- Detalhes de Implementação
 - Buffer: guarda uma mensagem atrasada vinda do Matcher
 - Quando se quer atrasar uma mensagem
 - Salva-se o lugar do casamento (quando ele ocorre)
 - Seta-se um flag pra indicar que o buffer está cheio
 - Salva-se o **tempo limite do atraso**: tempo real + tempo de atraso

Decisões Atrasadas

- Exemplo: Supor que
casamento ocorreu em 1200ms +
tempo de suspensão é 100ms

tempo limite de atraso será 1300ms
- Para cancelar uma mensagem
 - Reseta o flag para indicar que o buffer está vazio
- Buffer é checado frequentemente

Decisões Atrasadas

- Se o tempo limite do buffer cheio passou sem pegar nenhuma nova nota
 - Manda mensagem ao Acompanhante

- Mas se pegou uma nova nota no tempo de atraso
 - Cancela mensagem

Outras Melhorias

- Equivalência de Oitava
 - Negligenciar diferenças de oitavas na nota

Outras Melhorias

■ Vetores de Bits

- Para lida com notas polifônicas, foram utilizados vetores de bits para representar conjuntos
- O vetor é um inteiro de 2 bytes
- Bit menos significativo corresponde ao C, próximo bit corresponde C# ou Db e assim por diante

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
				B	A#	A	G#	G	F#	F	E	D#	D	C#	C

Outras Melhorias

- Valores de notas limitados a [0..11]
- Se fosse utilizado todo o conjunto MIDI [0..127], seriam necessários 16 bytes
- Cálculo de casamento utiliza muito tempo
- Operações com bits são operações rápidas
 - Diferenças entre a partitura e os eventos tocados
 - notas extras e/ou notas não tocadas
 - Notas extras podem ser ornamentos, ignorando-as, temos
 - somente as notas não tocadas

Outras Melhorias

- Contando os bits do resultado 2 setados em 1, temos o número de notas não tocadas
- Se este número for menor que uma constante que é associada ao número de notas do evento
 - Trata como um casamento
- O sistema Acompanhante pode permitir o performista tocar um acorde difícil imperfeitamente

Outras Melhorias

■ Exemplo:

- Notas não tocadas = 1
- Notas na partitura = 4
- Sendo permitido, se tem o casamento



■ Sendo:

- X = número de notas no evento da partitura
- Y = número de notas não tocadas permitidas

Outras Melhorias

■	<u>X</u>	<u>Y</u>	
	1	0	→ deve ser tocada
	2	0	
	3	1	→ permite não tocar uma nota
	4	1	
	5	2	→ permite não tocar duas notas

Outras Melhorias

■ Vantagens

- Baixo consumo de tempo de processamento
- Eliminação de ornamentos
- Permite o instrumentista adicionar algumas notas extras sem nenhuma penalidade

Conclusões

- Sistema final muito robusto
- Técnicas de manipulação de ornamentos com bons resultados
- Problema pode surgir quando o solista errar justamente antes de começar um trilo ou um glissando

Conclusões

- Trabalhos futuros:
 - Utilizar modelos baseados em conhecimento intuitivo humano
 - Utilizar um detector automático de trilos e glissandos
 - Criar um ambiente integrado, onde se pudesse tocar, gravar e editar música facilmente

Conclusões

- Tratar repetições e saltos nas indicações das partituras utilizando Matchers extras.
- Implementar uma capacidade do sistema de procurar em toda partitura onde o mesmo se encontra caso se perca totalmente.
- Tratamento de improvisações maiores

Bibliografia

- Dannenberg, R. B. 1984. “*An On-Line Algorithm for Real-Time Accompaniment*”. Proceedings da *Internacional Computer Music Conference (ICMC)* 1984. San Francisco: Computer Music Association, pp 193-198.
- Dannenberg, R. B e Mukaino, H. 1988. “*New Techniques for Enhanced Quality of Computer Accompaniment*”. Proceedings da *Internacional Computer Music Conference (ICMC)*, 1988

Bibliografia

Computer Music Association, pp 243-249.

- Toscani, L. V. e Veloso, P. S. “Complexidade de Algoritmos – Análise, Projeto e Métodos”. Série de Livros Didáticos – número 13. Instituto de Informática da UFRGS. Editora Sagra Luzzato.