

Acompanhamento Computacional

Mariana Zaparoli Martins

Computação Musical – IME USP

Roteiro

■ Um Algoritmo On-Line para Acompanhamento em Tempo-Real

Roger B. Dannenberg (Carnegie Mellon University)

- Introdução
- O Modelo
- O Algoritmo
- Mais Heurísticas

Roteiro

- Acompanhamento
 - Implementação
 - Limitações
 - Conclusões
-
- Bibliografia

Um Algoritmo On-line para Acompanhamento em Tempo Real

Roger B. Dannenberg
Computer Science Department
Carnegie-Mellon University
Pittsburgh, USA

Introdução

- Acompanhamento Computacional em Tempo Real ?
- Divisão em três subproblemas
 - Detecção
 - Correlação
 - Produção do Acompanhamento
- Tolerância a Erros
- Foco no segundo subproblema: Correlação

Introdução

- Unix *diff*

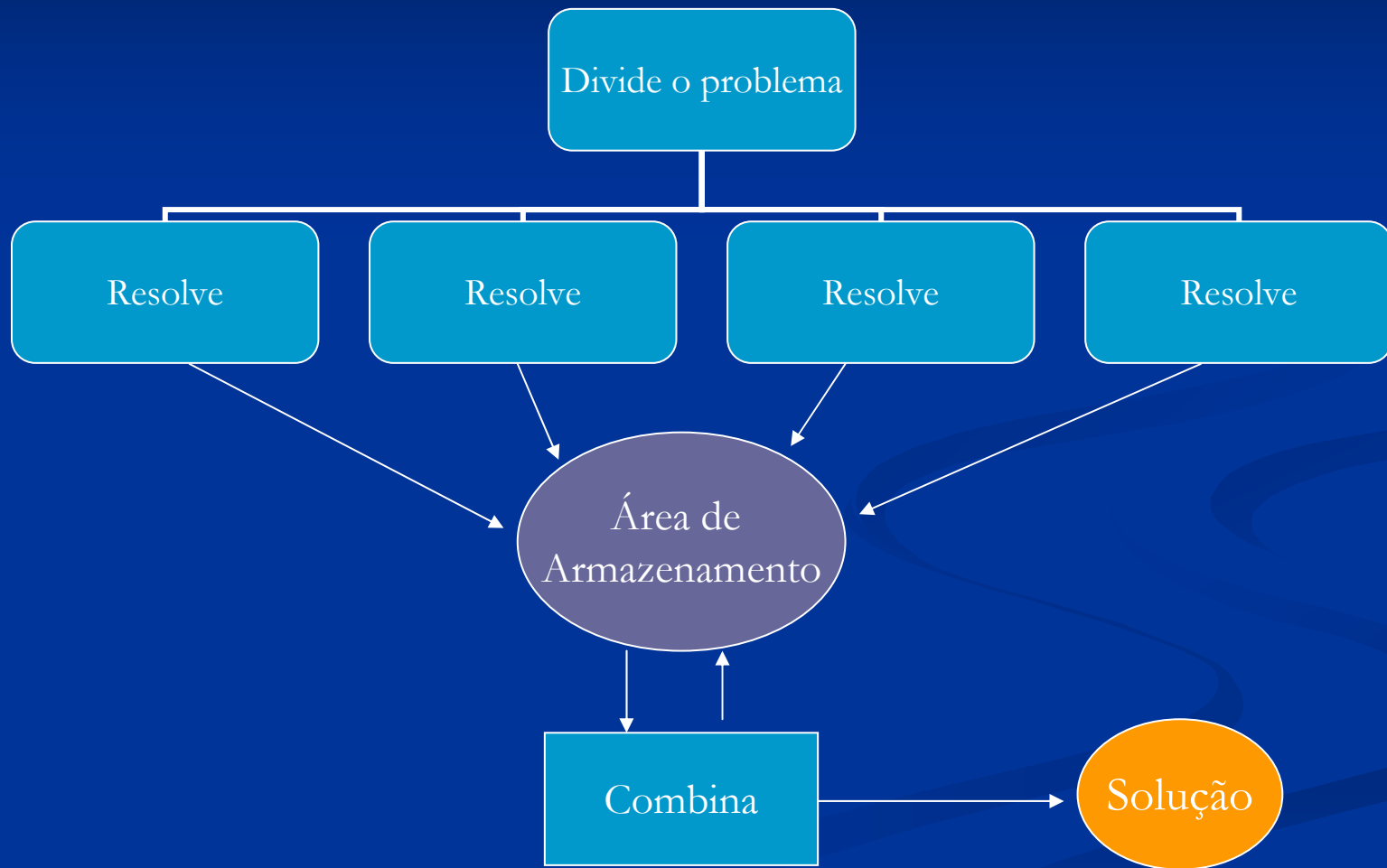
- Contexto musical: consumo de tempo proporcional ao tamanho da partitura

Inadequado!

- Algoritmos Dinâmicos

- Divide o problema em subproblemas
- Resolve cada subproblema apenas uma vez!
- Armazena resultados parciais

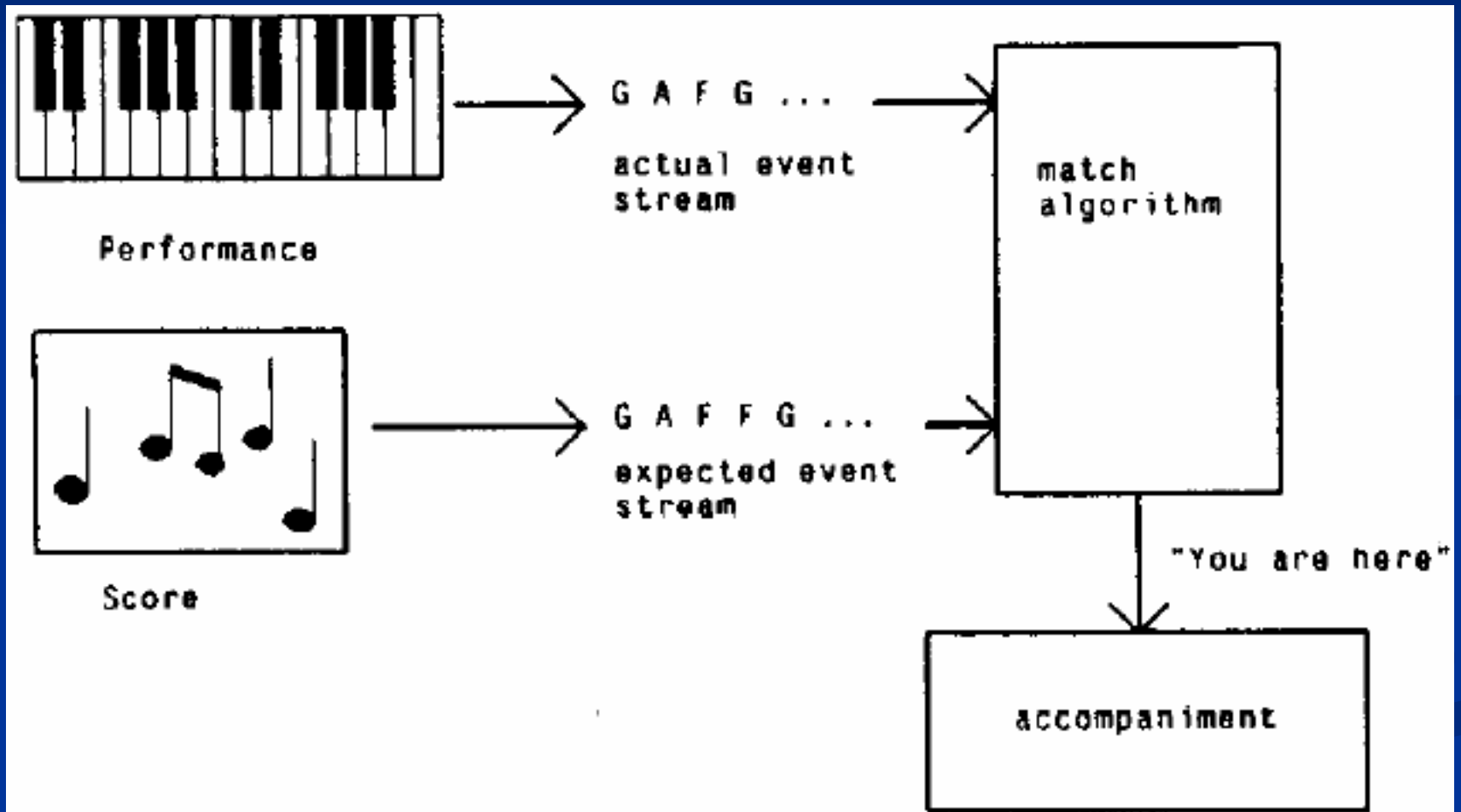
Introdução - Algoritmos Dinâmicos



O Modelo

- Classe limitada de eventos (Ex: teclas apertadas)
- Performance solo
 - fluxo contínuo de eventos
- Partitura
 - lista ordenada de eventos esperados
- Encontrar o melhor casamento entre os dois fluxos de informações (*streams*)

O Modelo



O Modelo

- O que é considerado o “melhor” casamento?
 - A maior subsequência comum entre o *stream* que chega do solista e o armazenado na partitura
- Eventos não esperados na performance: notas extras
- Eventos omitidos da partitura: notas omitidas

O Modelo

■ Solo:

A G E D G B C

■ Melhor casamento:

A G E G B C

■ Partitura:

A G E G A B C

O Algoritmo

- Matriz de inteiros (*bestlength*)
 - Cada linha corresponde a um evento na partitura
 - Cada coluna corresponde a um evento detectado na performance
- Na r -ésima linha e c -ésima coluna, qual seria o tamanho do melhor casamento no tempo corrente?
 - Resposta: verificar coluna e linha anteriores

O Algoritmo

para todo $i, bestlength[i, -1] \leftarrow 0;$

para todo $i, bestlength[-1, i] \leftarrow 0;$

para cada novo evento da performance $p[c]$ faça

para cada evento da partitura $s[r]$ faça

$$bestlength[r,c] \leftarrow \max(bestlength[r-1, c], \\ bestlength[r, c-1]);$$

if $p[c]$ casar com $s[r]$ então

$$bestlength[r,c] \leftarrow \max(bestlength[r, c], \\ 1 + bestlength[r-1, c-1]);$$


O Algoritmo

- Matriz *bestlength* para a partitura A G E G A B C depois dos eventos da performance A G E D
- P = parte tocada pelo músico
- S = parte armazenada da partitura (*score*)

O Algoritmo

P	S	A	G	E	D	G	B	C
A		1	1	1	1			
G		1	2	2	2			
E		1	2	3	3			
G		1	2	3	3			
A		1	2	3	3			
B		1	2	3	3			
C		1	2	3	3			

Algoritmo

O Algoritmo

- QUAIS eventos devem ser casados para se obter o melhor casamento?
- **Algoritmo On-line:** resultados incrementalmente assim que a entrada se torne disponível

O Algoritmo

P	S	A	G	E	D	G	B	C
A		1	1	1	1	1	1	1
G		1	2	2	2	2	2	2
E		1	2	3	3	3	3	3
G		1	2	3	3	4	4	4
A		1	2	3	3	4	4	4
B		1	2	3	3	4	5	5
C		1	2	3	3	4	5	6

O Algoritmo

- Maior eficiência
 - Espaço
 - Armazenamento das colunas anterior e a corrente
 - Tempo
 - Limite de adição/omissão de eventos (janela)
- Valores computados usando janela de tamanho três.

O Algoritmo

P	S	A	G	E	D	G	B	C
A		1	1					
G		1	2	2				
E		1	2	3	3			
G				3	3	4	4	
A					3	4	4	4
B						4	5	5
C								6

O Algoritmo

- O tamanho da janela
 - Pelo menos um a mais que o dobro do número de erros consecutivos tolerados
- Problema
 - Tendência a pular a frente na partitura quando um evento extra (errado) é detectado
 - Evento extra casa coincidentemente com evento futuro na partitura
 - O solista produziu um evento errado ou saltou evento(s)?

O Algoritmo

- Correlação “futura” → escolha errada

- Exemplo:

- partitura [B G E F C D]



- tocado [F B G]

- Tamanho do melhor casamento: 2

O Algoritmo

- partitura 2 [B G E F C B D G]

- tocado [F B G]

- Heurísticas são necessárias para desencorajar este salto na partitura

Mais Heurísticas

- Três heurísticas foram implementadas:
 - A primeira heurística limita a quantidade pelo qual o centro da janela pode mudar
 - A segunda heurística associa penalidades a casamentos que pulem eventos da partitura
 - + 1 para cada par de eventos casados
 - - 1 para cada evento pulado na partitura
 - Pequena mudança na implementação
 - Valor mais alto na matriz pode ocorrer em várias linhas de uma coluna

O Algoritmo

```
para todo  $i, bestlength[i, -1] \leftarrow 0;$   
para todo  $i, bestlength[-1, i] \leftarrow 0;$   
para cada no evento tocado  $p[c]$  faça  
  para cada evento da partituras  $s[r]$  faça  
     $bestlength[r,c] \leftarrow \max(bestlength[r-1, c] - 1,$   
                                 $bestlength[r, c-1]);$   
  if  $p[c]$  casar com  $s[r]$  então  
     $bestlength[r,c] \leftarrow \max(bestlength[r, c],$   
                                 $1 + bestlength[r-1, c-1]);$ 
```


Mais Heurísticas

- A terceira heurística faz com que quando houver possibilidade de escolha, se reporte o casamento ocorrido com o primeiro evento (ordem temporal)

Acompanhamento

- Acompanhamento composto e armazenado na memória
- Quando tocar cada evento do acompanhamento?
- Sistema convencional de tempo real
 - Tempo de início anexado a cada evento da partitura
 - Tempos comparados ao relógio de tempo real para decidir quando iniciar cada evento

Acompanhamento

- Propriedades requeridas para acompanhantes em tempo real
 - Variação da velocidade do relógio de tempo real
 - Reinicialização do relógio
- Assume-se que o relógio de tempo real (*hardware*) não pode mudar a velocidade ou ser reiniciado
- **Relógio Virtual:** implementado utilizando o relógio físico como referência

Acompanhamento

- $\text{Tempo Virtual} = (R - R_{\text{ref}}) * S + V_{\text{ref}}$
 - R é o tempo real
 - R_{ref} é o tempo real no qual o relógio virtual foi setado pela última vez
 - S é a velocidade do relógio virtual
 - V_{ref} é o tempo virtual no qual o relógio virtual foi setado pela última vez

Acompanhamento

- O relógio virtual é setado a cada vez que é reportado um casamento em resposta a um evento detectado na entrada
- O tempo virtual é determinado pelo evento correspondente na partitura solo
- O tempo real é determinado pelo relógio físico (*hardware*)

Implementação

- Pequeno microcomputador de 8 bits
- Teclado para entrada e um sintetizador digital para saída
- As partituras do solo e do acompanhamento lidas de um arquivo
- O usuário pode tocar em um teclado e ouvir ambos o solo e o acompanhamento

Implementação

- O sistema faz detecção de tom em tempo real do som de um trompete como solo na entrada
- Gera o acompanhamento de uma voz na saída
- Detecção de tom e síntese são auxiliadas pelo *hardware*

Limitações

- Não são feitos ajustes de andamento de uma maneira musical
- Supõe-se que a entrada vinda do solista é um conjunto totalmente ordenado de eventos. Isto não é válido em pelo menos dois casos:
 - Múltiplos sensores para captura da entrada
 - Múltiplos eventos tocados “simultaneamente”
- Sempre se considera a ordem temporal

Conclusões

- O algoritmo consegue acompanhar um instrumentista mesmo quando notas extras são tocadas ou quando notas são omitidas.
- Variações são possíveis
 - Dar prioridades a alguns eventos
- O sistema é simples e previsível
 - Pode causar algumas surpresas quando confrontado com material musical não tradicional

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.
- 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.