

## 7

### Resultados

#### 7.1

##### Harmônicos de Variedade

Comparamos o processo de implementação dos filtros harmônicos na CPU e na GPU. Obviamente, era esperado que o desempenho da implementação da GPU superasse o da CPU. Na tabela (7.1) mostramos o resultado da deformação de uma única malha na CPU, com 6 e 12 deformações.

Validamos que a implementação na GPU suporta processamento em tempo real para manter a sincronização com o som. Estas experiências permitem estimar o tamanho da galeria apropriada, dependendo da placa gráfica (ver tabela (7.1)). Concluimos que para os modelos com cerca de 50.000 vértices, um tamanho correto para a galeria seria entre 6 e 12 em uma GeForce 130 com 48 núcleos de 500 MHz.

Testamos a visualização da música em diferentes modelos em tempo real. Nós incluímos uma chamada que atualiza o filtro a cada 50 milissegundos, e a renderização é feita seguindo o ciclo de renderizações, tal que mesmo com um grande número de galerias, o que poderia prejudicar a renderização em tempo real da aplicação, o som não para de tocar.

Para finalizar, adicionamos uma variável  $m \in [0, 1]$  que controla a suavidade com que as frequências são passadas para a malha. Então as amplitudes associadas as frequências do som são sempre ponderadas:

$$\tilde{g}(\xi)_{new} = m \cdot \tilde{g}(\xi)_{old} + \tilde{g}(\xi).$$

Isto evita os efeitos trêmulos durante a animação com músicas, para música muito ritmada. Veja os exemplos ilustrados nas figuras 7.1 e 7.2.

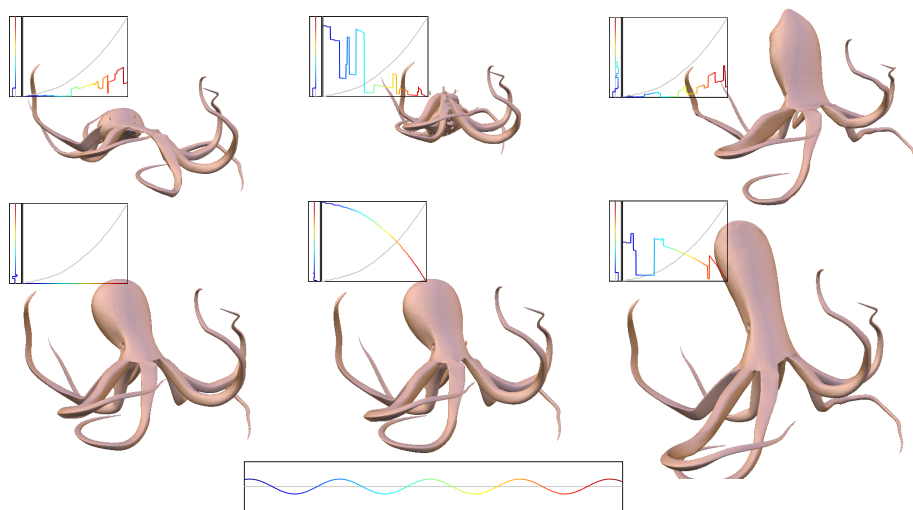


Figura 7.1: Galeria depois de uma reprodução dos itens 1, 4 e 5 da figura 6.3 nesta ordem de leitura.



Figura 7.2: Visualização de música rock no armadillo.

modelo	#verts	#freqs	pre-	CPU	GPU	GPU
	$n$	$\#k$	process	S=1	S=6	S=12
			<i>secs</i>	<i>fps</i>	<i>fps</i>	<i>fps</i>
pig	1 843	184	4	54.2	203.0	122.8
triceratops	2 832	256	7	46.9	153.9	63.6
neptune	9 392	276	32	12.7	58.6	31.2
multitorus	11 898	270	43	11.5	73.9	40.1
dinosaur	14 054	533	74	4.7	28.3	13.9
octa	15 136	529	25	4.1	26.8	13.2
octopus	20 351	546	129	3.0	21.4	7.1
alien	24 988	540	232	3.4	22.4	12.1
david	24 988	804	116	2.6	14.8	7.3
david head	30 058	1317	232	1.8	15.0	7.0
cat	30 059	271	379	9.6	45.7	23.0
gargoyle	30 059	1052	96	2.1	13.7	6.9
bunny	34 834	1070	478	1.7	12.1	5.8
buste	37 874	1075	329	1.5	13.4	3.4
blooby	42 432	1065	553	1.5	12.5	7.5
egea	63 739	275	718	0.9	7.2	3.2
head	65 002	1607	739	0.5	8.7	4.3
armadilo	86 488	2376	1 134	0.2	10.3	3.4

Tabela 7.1: Todos os modelos são normalizados em um box volumétrico  $[-1, 1]^3$ , e a galeria de  $S$  itens é renderizada em uma janela de resolução  $1024 \times 768$ . Todos os testes são realizados em um processador  $3.06GHz$  com uma GeForce GT 130 com  $512MB$  de RAM. A velocidade da deformação é medida em frame por segundos (*fps*), enquanto o pré-processamento do cálculo das bases harmônicas é expresso em segundos.

## 7.2

### Análise Harmônica combinada com Modal

Nossos primeiros testes foram sobre a eficácia do skinning. Devido a dificuldade em encontrar modelos articulados com rigging disponíveis na web, construímos nossos próprios modelos no software Blender 2.61 (1).

Foram realizados testes simples, como animação com um único modo e amplitudes baixas à movimentos mais complexos envolvendo combinações de mais de um modo normal e amplitudes mais altas. A possibilidade do skinning funcionar com deformações mais exageradas é menor. De todas as formas, tivemos resultados bastantes favoráveis com nosso skinning. (Veja as imagens 5.3 e 7.3).

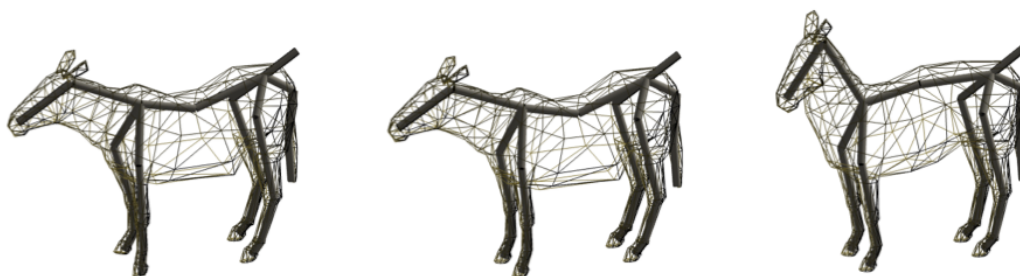


Figura 7.3: Zebra em um único modo de frequência 0.213596 Hz e amplitude 1.5.

Posteriormente, fizemos experiências com a visualização de música. Usamos diversos tipos de música para verificar a influência das frequências. O teste realizado com o modelo da mão com uma música de rock foi aquele em que ocorreram maiores deformações. No teste com o modelo do laço onde usamos uma música de forró (cujas frequências, no nosso caso, são tipicamente maiores que as da música de rock) a oscilação foi mais rápida. Veja figuras 5.4, 7.5, e finalmente 7.6.

Por fim, testamos a performance do uso da placa. Verificamos para os modelos citados que, graças ao uso da placa gráfica, o processamento se deu em tempo real. Fizemos uma comparação análoga à tabela 7.2, onde analisamos o tempo de processamento e renderização de um único modo na CPU, e na GPU.

Todos os testes foram executados em um computador com um processador de 3.06GHz com uma GeForce GT 130 com 512MB de RAM. Verificamos



Figura 7.4: Simulando a deformação do laço sob uma música de forró.

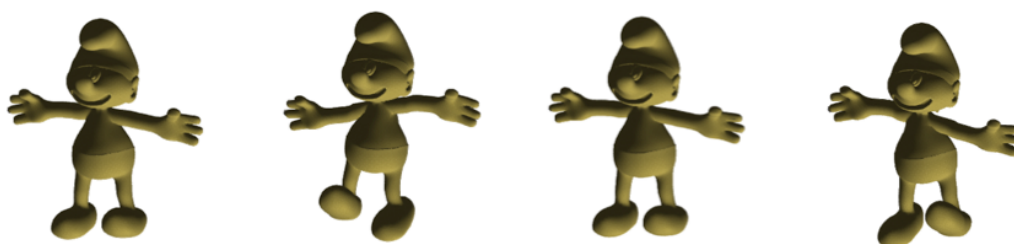


Figura 7.5: Modelo do Smurf ao som de MPB.



Figura 7.6: Zebra sob a influência de uma música francesa.

que para este caso um tamanho razoável para a galeria teria entre 9 a 15 modos normais.

modelo	#verts	#freqs	pre- process	CPU	GPU
	$n$	$\#\nu$		NM=1	NM = 1
			<i>secs</i>	<i>fps</i>	<i>fps</i>
zebra	572	23	0.021	3.718	13.727
laço	576	13	0.055	3.717	13.59
mão	762	19	0.024	3.707	14.1
smurf	32168	36	0.029	3.826	4.74

Tabela 7.2: A galeria de **NM** itens é renderizada em uma janela de resolução  $1024 \times 768$ . Todos os testes são realizados em um processador  $3.06GHz$  com uma GeForce GT 130 com  $512MB$  de RAM. A velocidade da deformação é medida em frame por segundos (*fps*), enquanto o pré-processamento do cálculo das bases harmônicas é expresso em segundos.