

## 4 Resultados

Neste Capítulo serão discutidos os resultados obtidos. Todas as imagens têm realçados os pontos no bordo do domínio e os da silhueta, e as variedades são iluminadas com uma única fonte de luz, e tolerância para a aproximação da raiz é de 0.001, e foram utilizadas 10 amostras de intervalos.

Os testes foram feitos em um computador Intel Core i5 2.5 GHz e com placa gráfica Intel HD Graphics 3000. A implementação foi feita em Visual Studio C++ 2010.

A seção 4.1 apresenta as análises de frames por segundo observando a variação de diferentes parâmetros do método. A seção 4.2 apresenta a visualização de variedades variando os ângulos de observação. A seção 4.3 apresenta alguns testes feitos em CPU para analisar a qualidade das imagens se fossem lançados muitos raios.

As imagens apresentadas neste Capítulo, correspondem as funções implícitas definidas na Tabela 4.1.

Função	$f(x, y, z, w) =$
1	Torus $w - (x^2 + y^2 + z^2 + R^2 - r^2)^2 + 4R^2(x^2 + y^2)$ com $R = 2$ , e $r = 1$
2	Cubo emaranhado $w - x^4 + 5x^2 - y^4 + 5y^2 - z^4 + 5z^2 - 11.5$
3	Gota $w - 0.5 * x^4(x + 1.0) + (y^2 + z^2)$
4	$y - 2zw$
5	$x - z^2 + w^2$
6	$x^2 - y^2 - z^2 + w^2 0.5$
7	$2xy - 2zw$
8	$(x^3 - 3xy^2 - z)$
9	$3x^2y - y^3 - w$
10	$2wz + y^3 + xy$
11	$x^2y^2 + zw$

Tabela 4.1: Funções implícitas de teste.

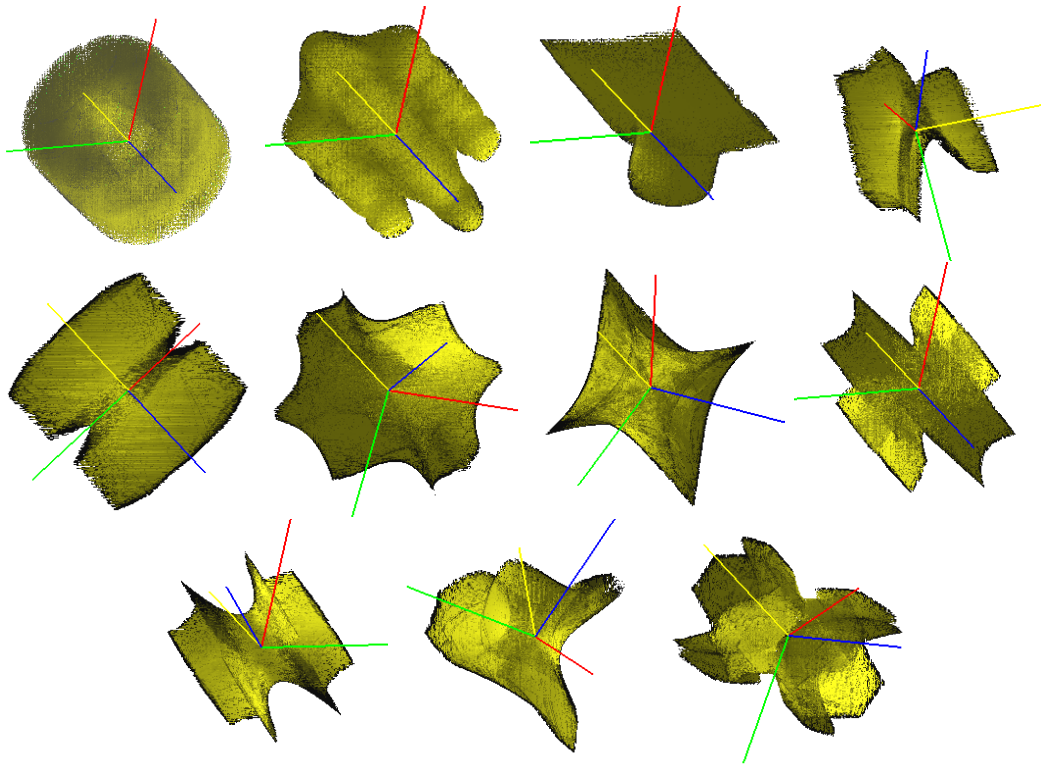


Figura 4.1: Visualização das variedades implícitas da Tabela 4.1, em diferentes ângulos, para as funções 1 até 11, na ordem da esquerda para a direita.

## 4.1 Avaliação

Nesta seção serão apresentadas figuras e tabelas resumindo os diferentes testes realizados. Primeiramente segue uma apresentação de figuras para diferentes funções 4.1.1. Depois segue uma avaliação de frames por segundo variando o número de raios 4.1.2, variando o número de intervalos 4.1.3, e variando o número de nível do árvore 4.1.4.

### 4.1.1 Diferentes variedades implícitas

A Figura 4.1 apresenta a visualização das variedades implícitas da Tabela 4.1, para as funções 4 até 11, feitas com 20 raios.

### 4.1.2 Variando o número de raios

A Tabela 4.2, apresenta os cálculos de frames por segundo (fps) em função do número de raios lançados para as funções definidas na Tabela 4.1.

Nas Figuras 4.2 e 4.3 podem ser observados os resultados.

À medida que aumenta-se os raios, melhor fica a qualidade mas o número de fps diminui. Para as funções 1 e 2 a melhor taxa qualidade e interatividade fica entre 10 e 15 raios, e para as outras funções fica entre 15 e 20 raios.

Função	Número de raios	fps	Imagens
1	10	10	Figura 4.2
1	15	8	
1	20	6	
1	30	5	
2	10	9	
2	15	7	
2	20	5	
2	30	4	
4	10	18	
4	15	12	
4	20	9	
4	30	6	
5	10	17	
5	15	12	
5	20	9	
5	30	6	
7	10	18	Figura 4.3
7	15	12	
7	20	9	
7	30	6	

Tabela 4.2: Frames por segundo obtidos variando a quantidade de raios lançados desde cada pixel.

### 4.1.3

#### Variando a quantidade de intervalos

A Tabela 4.3 apresenta os resultados obtidos dos cálculo de frames por segundo, variando o número de intervalos em (7, 10, 15 e 20) para 10 e 30 raios lançados por pixel. Nas Figuras 4.4 e 4.5 podem ser observados os resultados, ter 30 raios fica em melhor qualidade a imagem, mais não fica interativo, por que seus fps são baixos, a melhor taxa de qualidade e interativo fica com 15 intervalos com 10 raios, e 10 intervalos com 30 raios.

### 4.1.4

#### Variando o nível do 16-Tree

A Tabela 4.4 apresenta os resultados da análise feita variando o nível máximo da árvore, considerando os níveis (0, 1, e 2), para 10, 20 e 30 raios lançados por pixel. Nas Figuras 4.6 e 4.7 podem ser observados os resultados. Os frames por

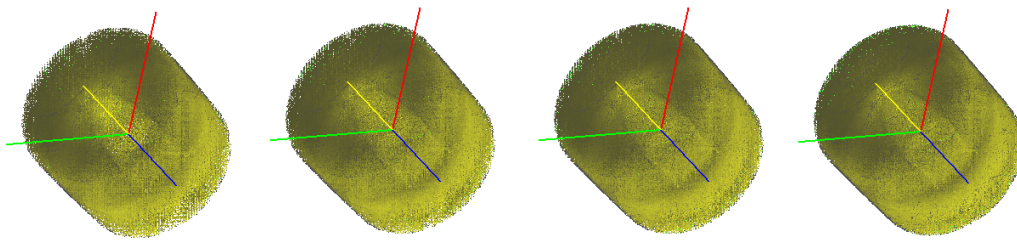


Figura 4.2: Qualidade da função 1, variando a quantidade de raios: 10, 15, 20 e 30 raios por cada pixel.

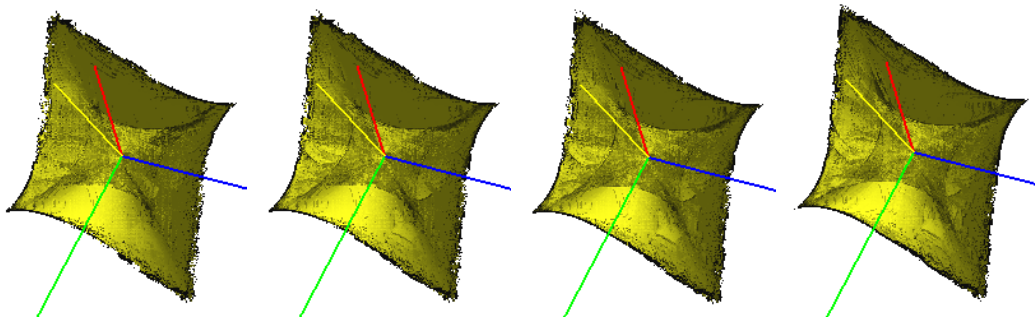


Figura 4.3: Qualidade da função 7, variando a quantidade de raios: 10, 15, 20 e 30 raios por cada pixel.

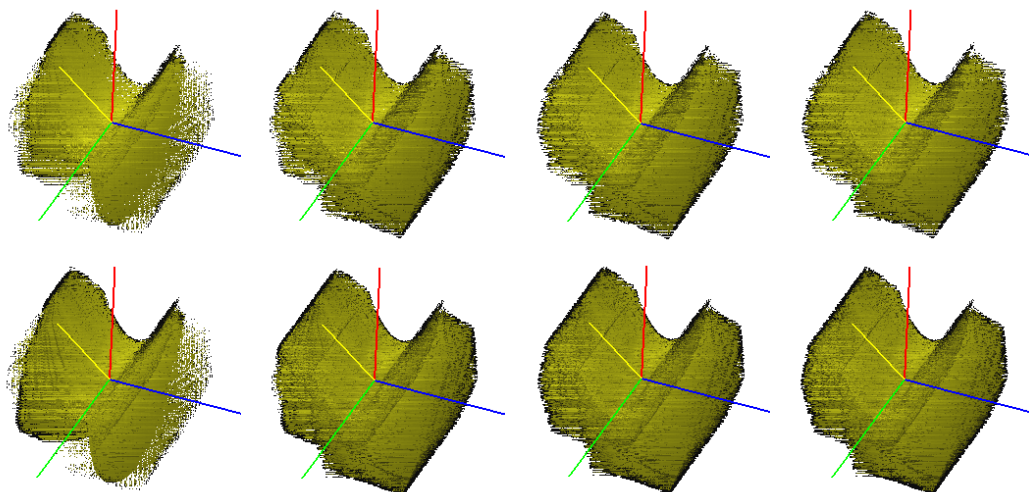


Figura 4.4: Qualidade da função 5, variando a quantidade de intervalos (7,10,15,20) e de raios por cada pixel, 10 raios para a primeira fila e 30 raios para a segunda fila.

Função	Número de intervalos	fps		Imagens
		10 raios	30 raios	
5	7	26	9	Figura 4.4
5	10	18	6	
5	15	12	4	
5	20	9	3	
6	7	24	8	Figura 4.5
6	10	17	6	
6	15	11	4	
6	20	3	1	
9	7	23	8	
9	10	17	6	
9	15	11	4	
9	20	8	3	

Tabela 4.3: Frames por segundo obtidos variando a quantidade de intervalos, para 10 e 30 raios lançados desde cada pixel.

segundo decrescem muito com o nível 2 do árvore, e com 30 raios. A melhor qualidade está com o nível 0 da árvore e com 20 raios.

Função	Nível	fps			Imagens
		10 raios	20 raios	30 raios	
8	0	20	16	15	
8	1	7	6	3	
8	2	1	1	1	
10	0	10	6	5	Figura 4.6
10	1	3	2	1	
10	2	1			
11	0	9	5	3	Figura 4.7
11	1	4	2	1	
11	2	1			

Tabela 4.4: Frames por segundo obtidos variando o nível do árvore, para 10, 20 e 30 raios lançados desde cada pixel.

#### 4.1.5

##### Variando o nível do 16-Tree e os intervalos

A Tabela 4.5 apresenta os resultados da análise feita variando o nível máximo da árvore (0, 1 e 2) e variando também a quantidade de intervalos, para 10 e 20 raios lançados por pixel. Na Figura 4.8 podem ser observados os resultados, pode-se utilizar mais níveis diminuindo o número de intervalos, e também mantendo o número de raios entre 10 e 15.

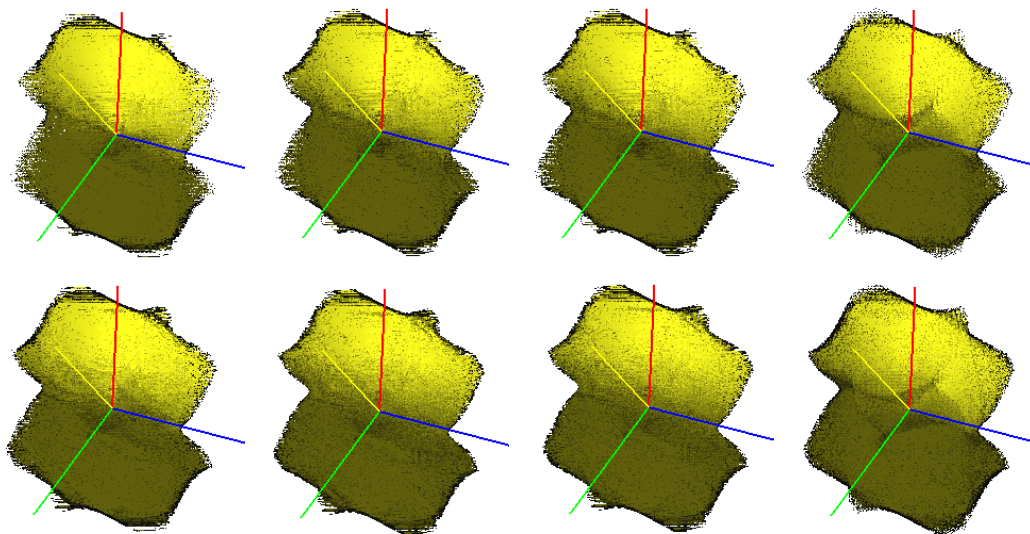


Figura 4.5: Qualidade da função 6, variando a quantidade de intervalos (7,10,15,20) e de raios por cada pixel, 10 raios para a primeira fila e 30 raios para a segunda fila.

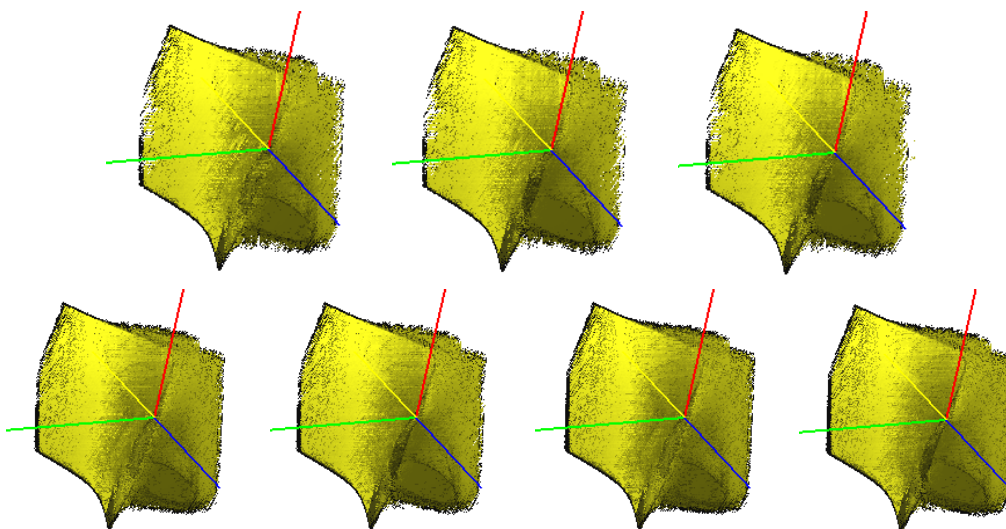


Figura 4.6: Qualidade da função 10, variando o nível máximo do árvore e de raios por cada pixel. A primeira fila: nível 0,1 e 2 com 10 raios. Segunda fila, começando da esquerda: nível 0 e 1 com 20 raios, nível 0 e 1 com 30 raios.

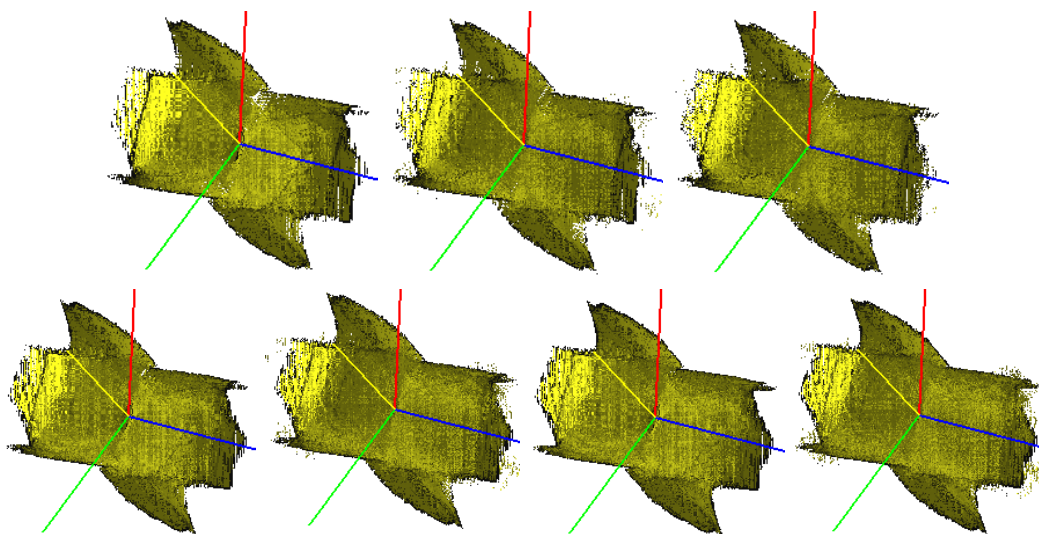


Figura 4.7: Qualidade da função 11, variando o nível do árvore e de raios por cada pixel. A primeira fila: nível 0,1 e 2 com 10 raios. Segunda fila, começando da esquerda: nível 0 e 1 com 20 raios, nível 0 e 1 com 30 raios.

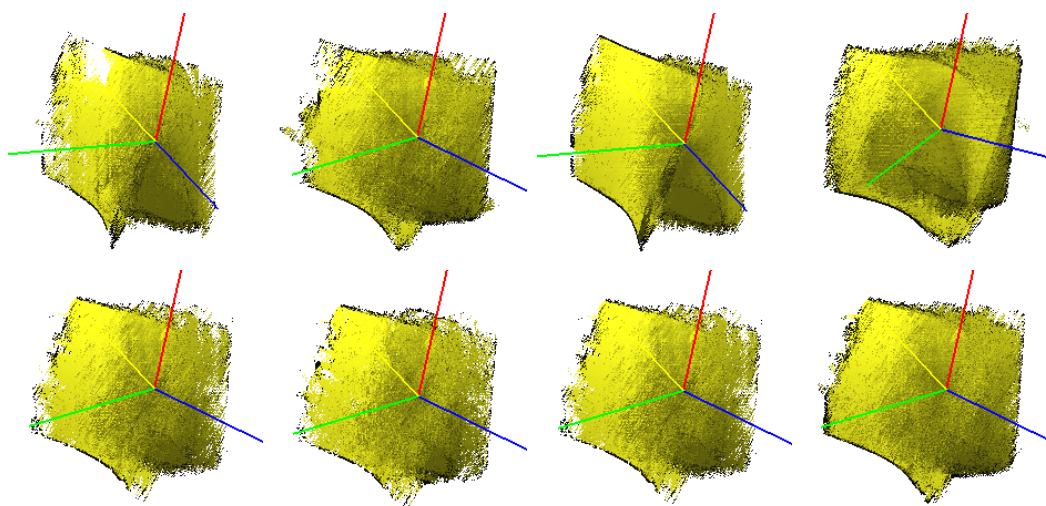


Figura 4.8: Qualidade da função 10, variando o nível da árvore, a quantidade de intervalos e o número de raios por cada pixel. A primeira fila: nível 1, da esquerda com 5 intervalos e as da direita com 7 intervalos para 10 e 20 raios. Segunda fila: nível 2, da esquerda com 2 intervalos e as da direita com 4 intervalos para 10 e 20 raios.

Função	Nível	Intervalos	fps			Imagens
			10 raios	15 raios	20 raios	
10	0	10	10	6	6	Figura 4.8
	1	5	5	4	2	
	1	7	4	4	2	
	2	2	4	3	2	
	2	4	2	1	1	

Tabela 4.5: Frames por segundo obtidos variando o nível do árvore e a quantidade de intervalos, para 10 e 20 raios lançados desde cada pixel.

#### 4.1.6 Resultados

Pelos testes realizados, podem ser observados quais são os parâmetros ótimos para a visualização da variedade, para obter uma boa relação de qualidade e interatividade. Tendo presente que a qualidade varia para cada função.

Olhando as imagens, se pode observar que o parâmetro que mais afeta a qualidade final é a quantidade de raios. Por isso para o nível 0 da árvore considerar intervalos entre 10 e 15, e lançar entre 20 e 30 raios. Para o nível 1 da árvore considerar intervalos entre 5 e 7, e lançar entre 10 e 15 raios. Para níveis maiores da árvore, diminuir o número de intervalos e lançar uns 10 raios.

#### 4.2 Variando os ângulos

Uma maneira para visualizar as variedades em  $\mathbb{R}^4$  é gerar uma sequência de imagens mudando os seis ângulos de Euler da matriz de rotação. Os ângulos são mudados um cada vez, a Tabela 4.6 apresenta os ângulos usados, um exemplo se pode ver na Figura 4.9.

#### 4.3 Testes em CPU

Com o intuito de conhecer a qualidade que poderia-se obter se fosse lançados mais raios, foi implementada uma versão em CPU, seguindo o mesmo padrão do pipeline de OpenGL para o *fragment shader*, assim que o código feito em GLSL pode ser facilmente colado para a versão em CPU. A continuação apresenta-se os resultado obtidos.

##### 4.3.1 Variando a quantidade de raios

Pode-se ver na Figura 4.10 as imagens geradas em CPU da função 7 com 50, 75 e 100 raios por pixel.



Ângulos	$\theta_1$	$\theta_2$	$\theta_3$	$\theta_4$	$\theta_5$	$\theta_6$
1	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.38	0.0	0.0	0.0	0.0
3	0.0	0.75	0.0	0.0	0.0	0.0
4	0.38	0.75	0.0	0.0	0.0	0.0
5	0.75	0.75	0.0	0.0	0.0	0.0
6	0.75	0.75	0.38	0.0	0.0	0.0
7	0.75	0.75	0.75	0.0	0.0	0.0
8	0.75	0.75	0.75	0.38	0.0	0.0
9	0.75	0.75	0.75	0.75	0.0	0.0
10	0.75	0.75	0.75	0.75	0.38	0.0
11	0.75	0.75	0.75	0.75	0.75	0.0
12	0.75	0.75	0.75	0.75	0.75	0.38
13	0.75	0.75	0.75	0.75	0.75	0.75
14	1.13	1.13	1.13	0.75	0.75	0.75
15	1.51	1.51	1.51	0.75	0.75	0.75

Tabela 4.6: Ângulos usados na seção variando os ângulos 4.2.

### 4.3.2

#### Tempo de processamento

A Tabela 4.7 apresenta os tempos de processamento em milissegundos do *Ray Casting* feito em CPU, considerando diferentes nível da árvore e várias quantidades de raios, considerando 10 intervalos.

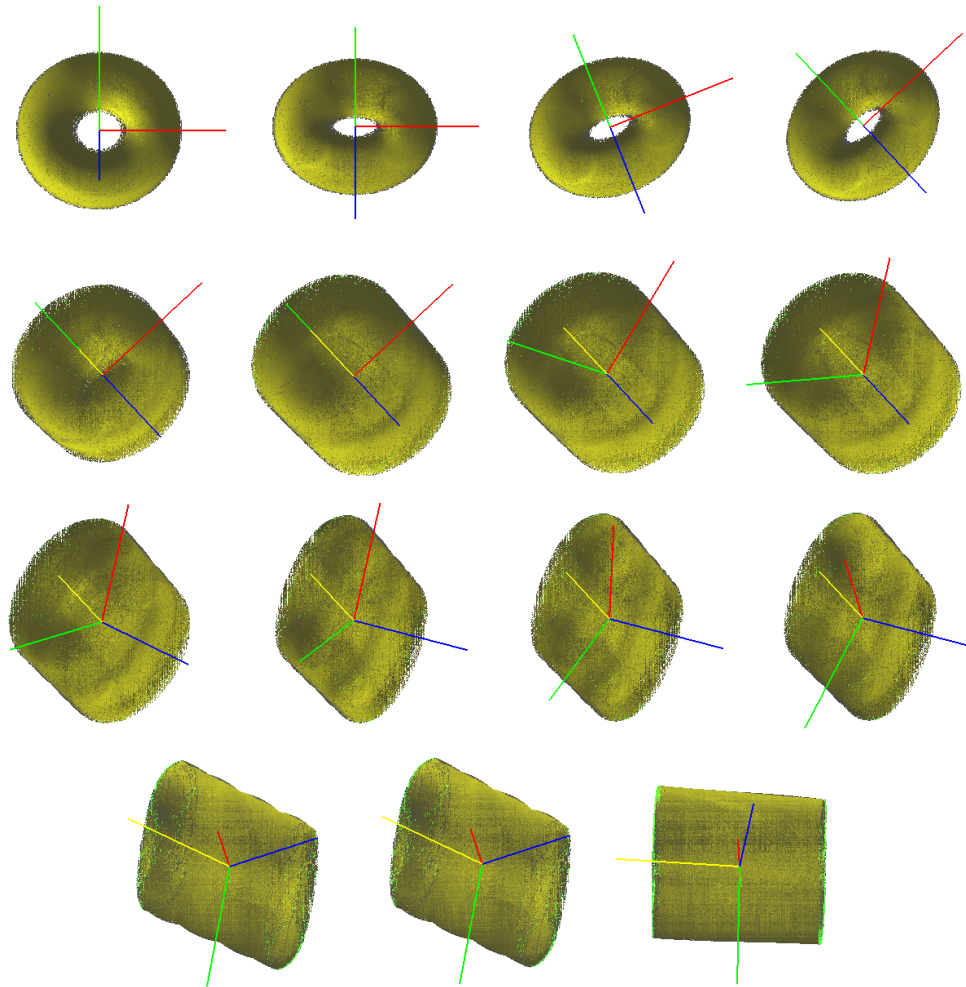


Figura 4.9: Sequencia de imagens geradas pela função 1, nos ângulos da Tabela 4.6, lançado com 10 raios por pixel.

Função	Nível	Raios	Tempo (milissegundos)	Imagens
7	0	50	18235	Figura 4.10
7	0	75	27120	
7	0	100	35528	
7	1	50	72346	
10	0	20	8082	
10	0	50	19082	
10	1	50	47075	
10	2	50	81506	
11	0	20	8025	Figura 4.11
11	0	50	18508	
11	0	100	36664	
11	1	50	33716	

Tabela 4.7: Tempo de processamento em milissegundos do *Ray Casting* em CPU, para diferentes funções com diferentes níveis da árvore e diferente quantidades de raios.

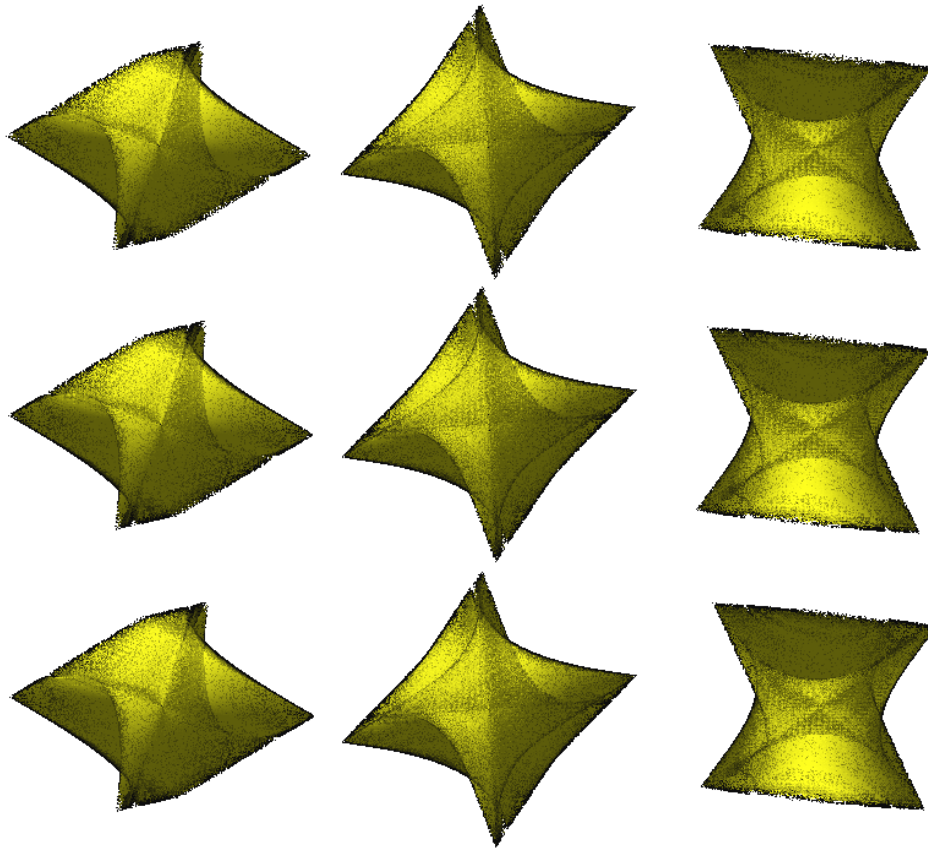


Figura 4.10: Imagens geradas em CPU pela função 7 da Tabela 4.1, nos ângulos 8, 10 e 14 da Tabela 4.6. A primeira fila com 50 raios, a segunda fila com 75 raios, a terceira fila 100 raios.

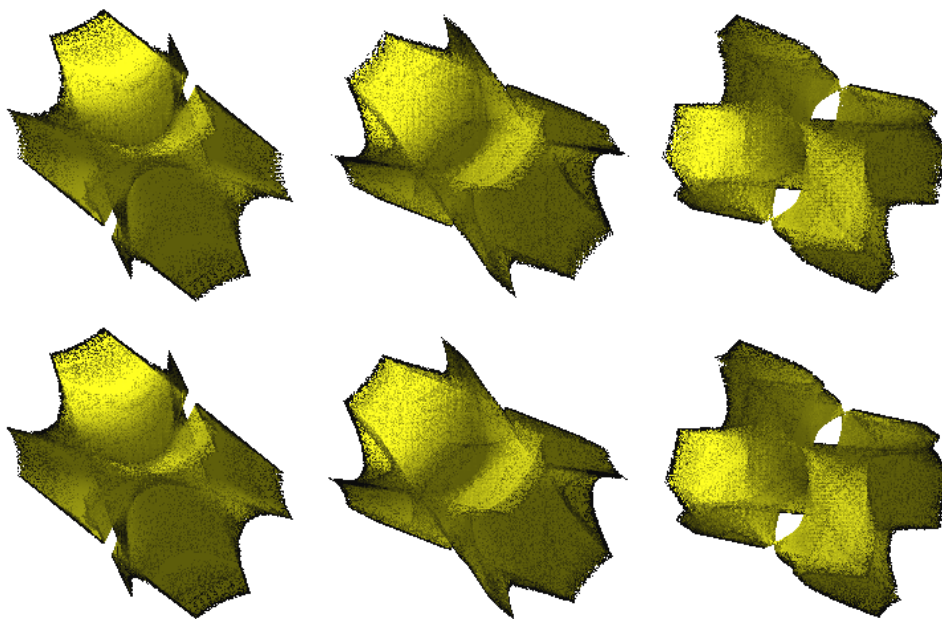


Figura 4.11: Qualidade da função 11 com *Ray Casting* em CPU. A primeira fila com 50 raios e a segunda fila com 100 raios.