



# Estimação de iluminação para diferentes horários através de *light probes* temporalmente variáveis

CAIO DE FREITAS VALENTE – 6552442

THIAGO DE GOUVEIA NUNES – 6797289

# Motivação

- ▶ Renderizar cenas em períodos diferentes do dia utilizando uma iluminação convincente e realista
  - ▶ *Image Based Lighting!*
- ▶ IBL consiste na obtenção de *Light Probes*, imagens HDR omnidirecionais de uma cena, que são posteriormente usados na fase de renderização, de modo a substituir as fontes de luz da cena
- ▶ Mas essa técnica apresenta uma limitação, para cada período do dia a ser renderizado é necessário obter um novo *Light Probe*



# Image Based Lighting



Cena usada como base para geração do Light Probe



# Objetivos

- ▶ Gostaríamos de aliviar a restrição de que é necessário obter um *Light Probe* para cada momento a ser renderizado
- ▶ Para tal pretendemos utilizar interpolação para estimar os *Light Probes* para horários em que não temos dados



# Resultados esperados

- ▶ Esperamos obter *light probes* que aproximem de maneira razoável os horários para os quais não possuímos dados. Para tal faremos duas comparações para validar nossa aproximação:
  - ▶ Comparação do *Light Probe* obtido através de interpolação com um *Light Probe* obtido de maneira convencional
  - ▶ Renderização de um objeto simples utilizando o *Light Probe* interpolado como fonte de iluminação e com uma imagem do mesmo objeto inserido numa cena real
    - ▶ O objeto em questão seria um cubo branco

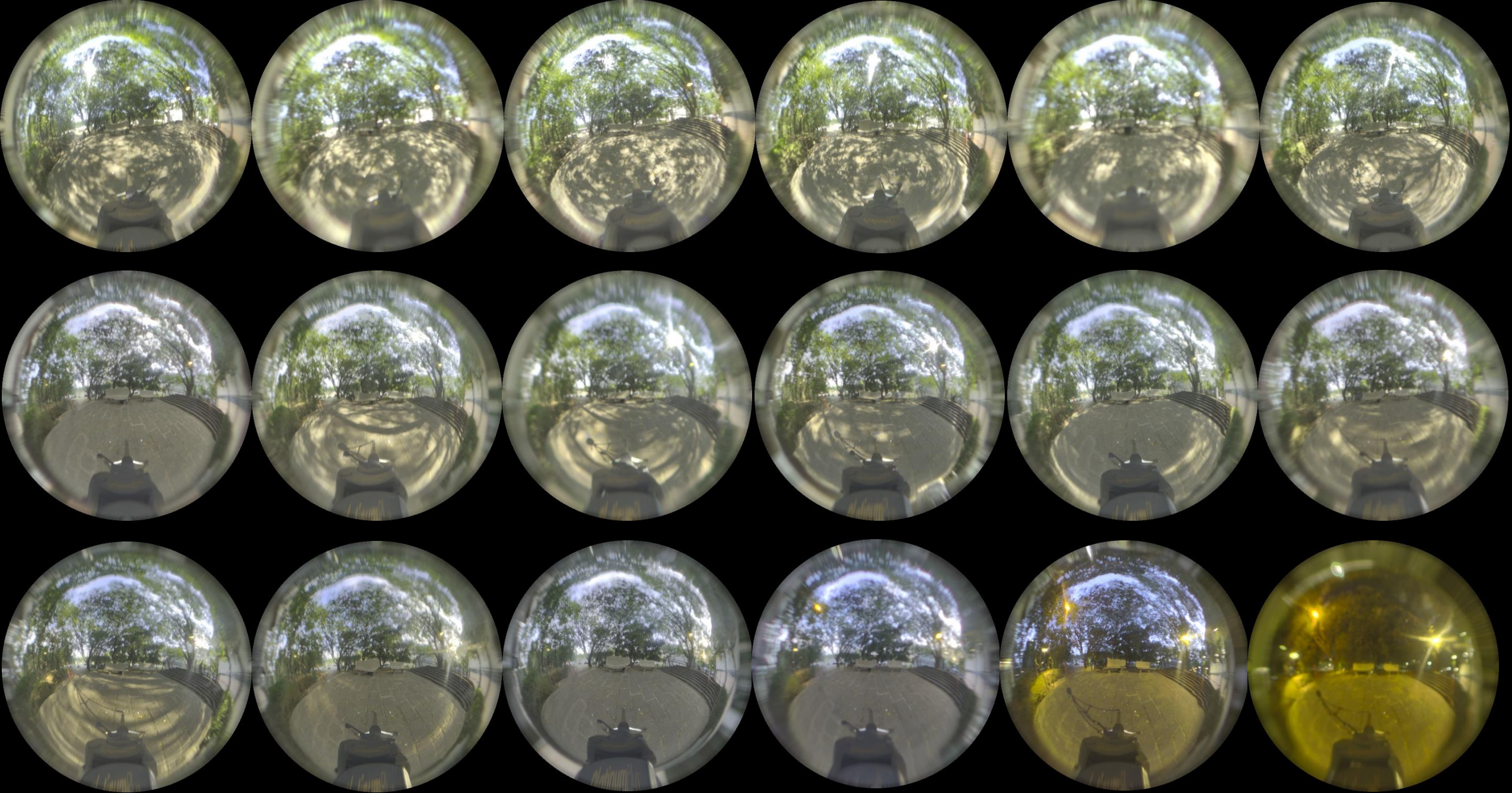
# Aquisição dos light probes

- ▶ Dados adquiridos!
- ▶ Cena interna e externa
- ▶ 18 tomadas para cena externa
- ▶ 17 tomadas para cena interna
- ▶ 533 imagens ao todo
- ▶ Diferença de 30-35 minutos entre cada aquisição
- ▶ Entre as 10:45 até as 20:00 (pausa para almoço apenas)
- ▶ Uso de um cubo branco como referencia para comparações
- ▶ Problemas com foco em algumas das tomadas

# Light Probes - externos



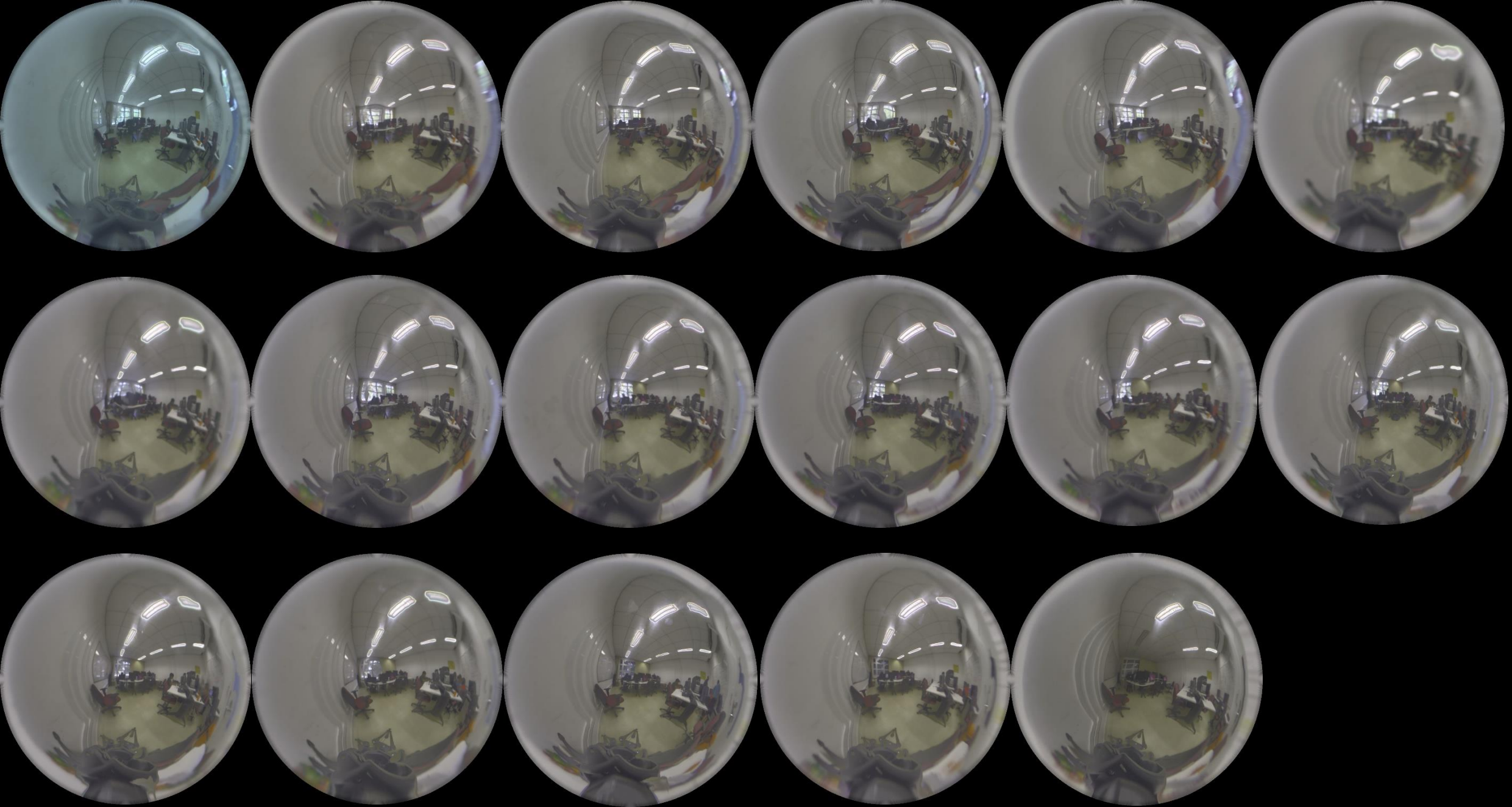






# Light Probes - internos







# “Cubo branco”



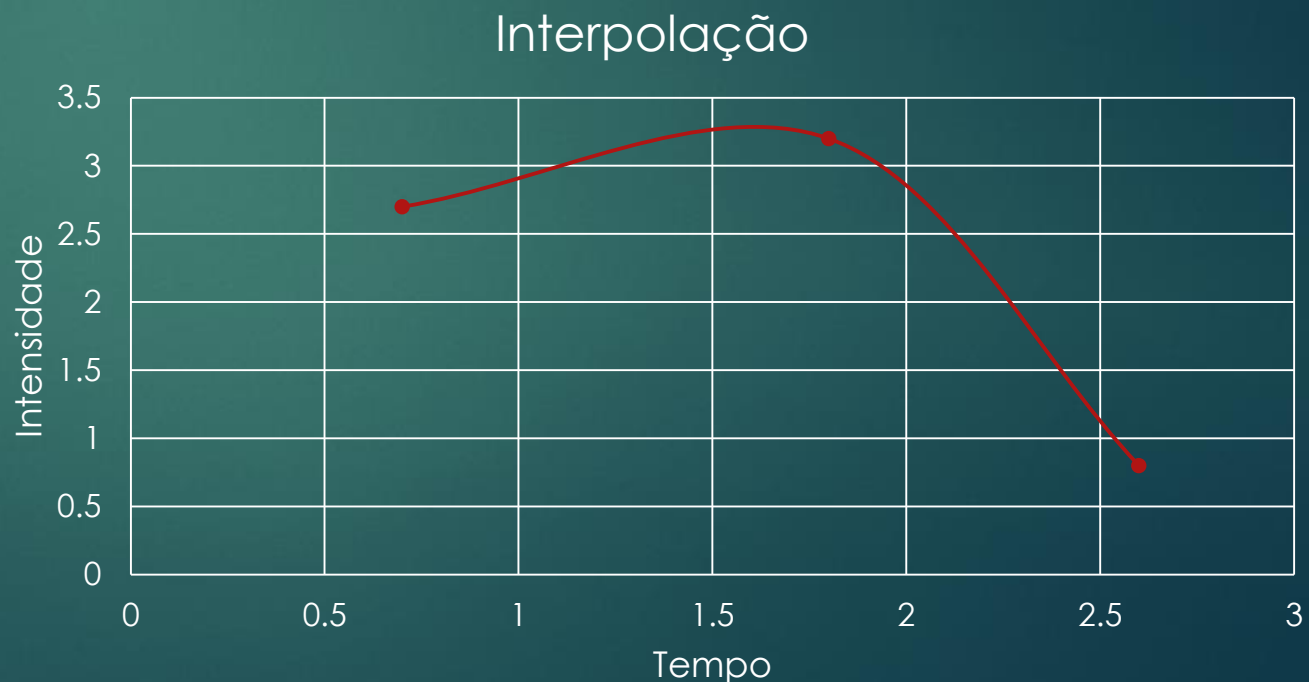
# Interpolações

- ▶ Linear
- ▶ Gauss forward central difference
- ▶ Gauss backward central difference
- ▶ Stirling
- ▶ Lagrange

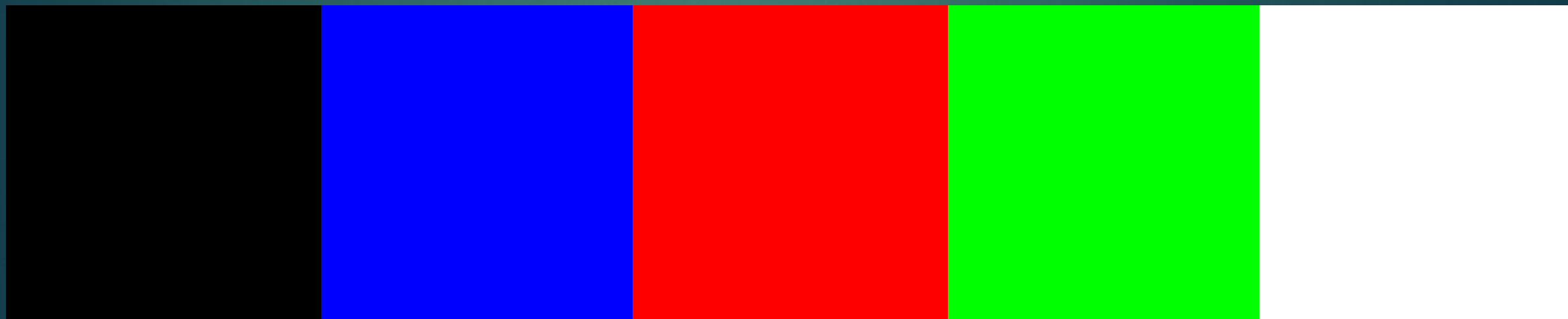


# Interpolações

- ▶ Interpolamos as intensidades de cada pixel em relação ao tempo
- ▶  $\dots, x_{-3}, x_{-2}, x_{-1}, x_0, x_1, x_2, x_3, \dots$  -- Conjunto de observações
- ▶  $\dots, y_{-3}, y_{-2}, y_{-1}, y_0, y_1, y_2, y_3, \dots$  -- Intensidade corresponde a cada observação

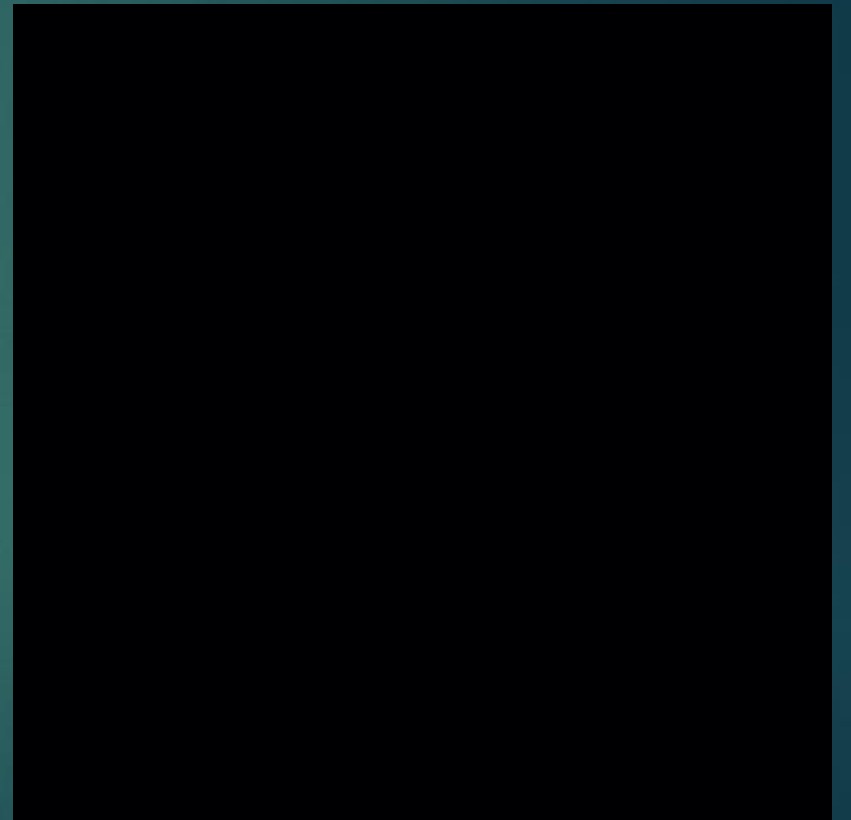
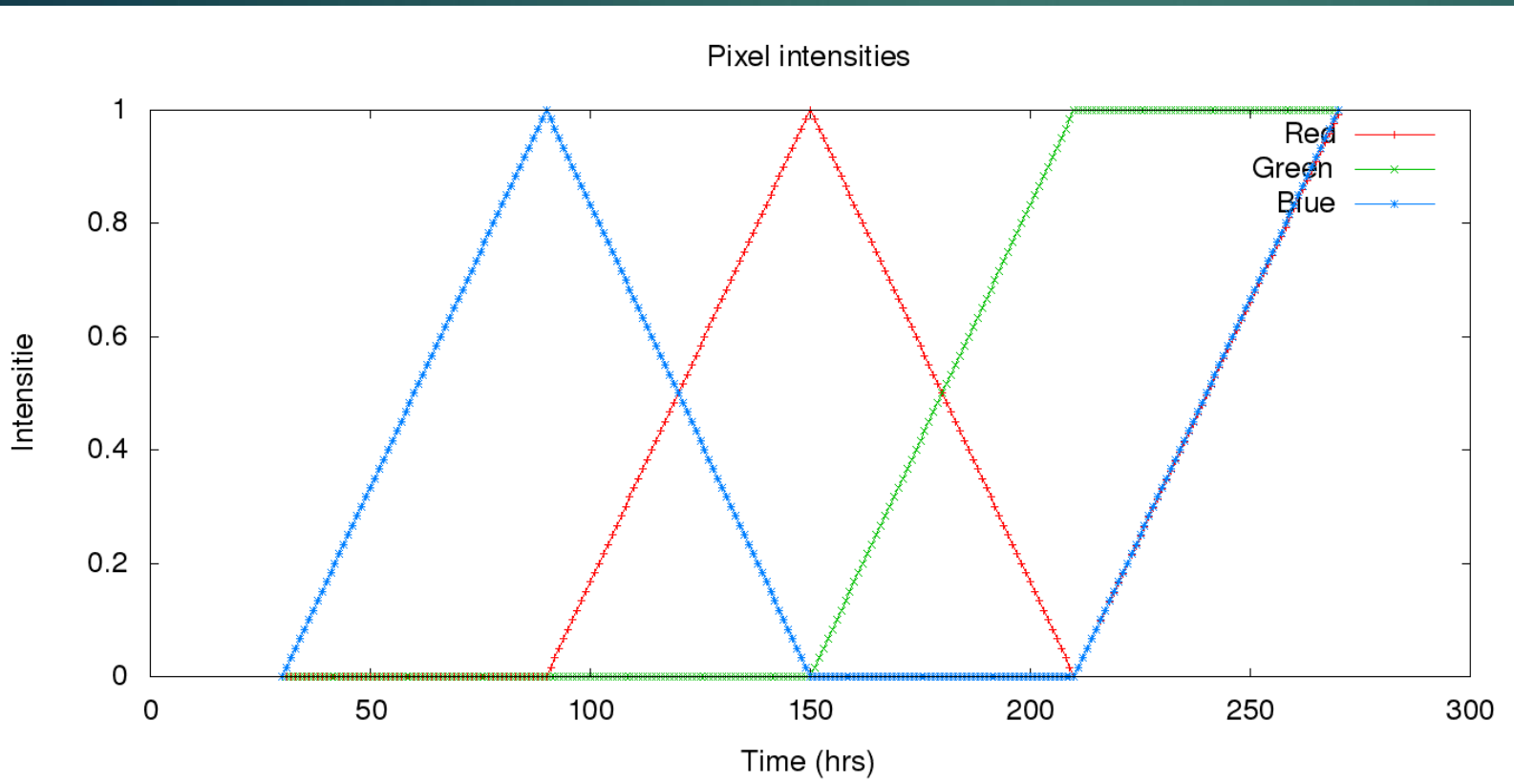


# Testes para as interpolações





# Linear



# Lagrange

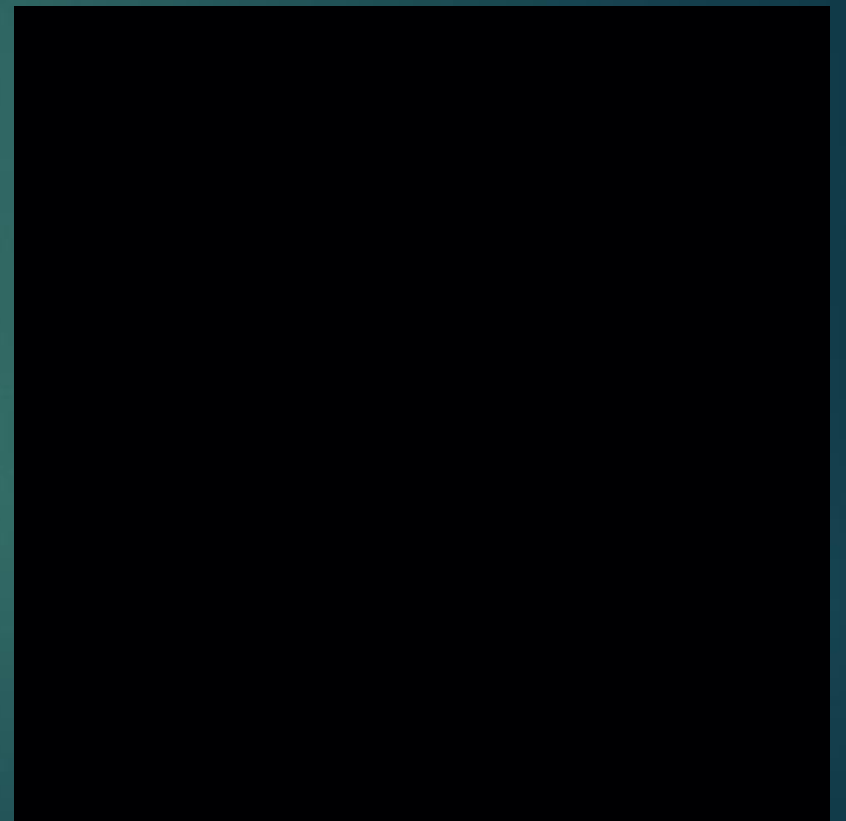
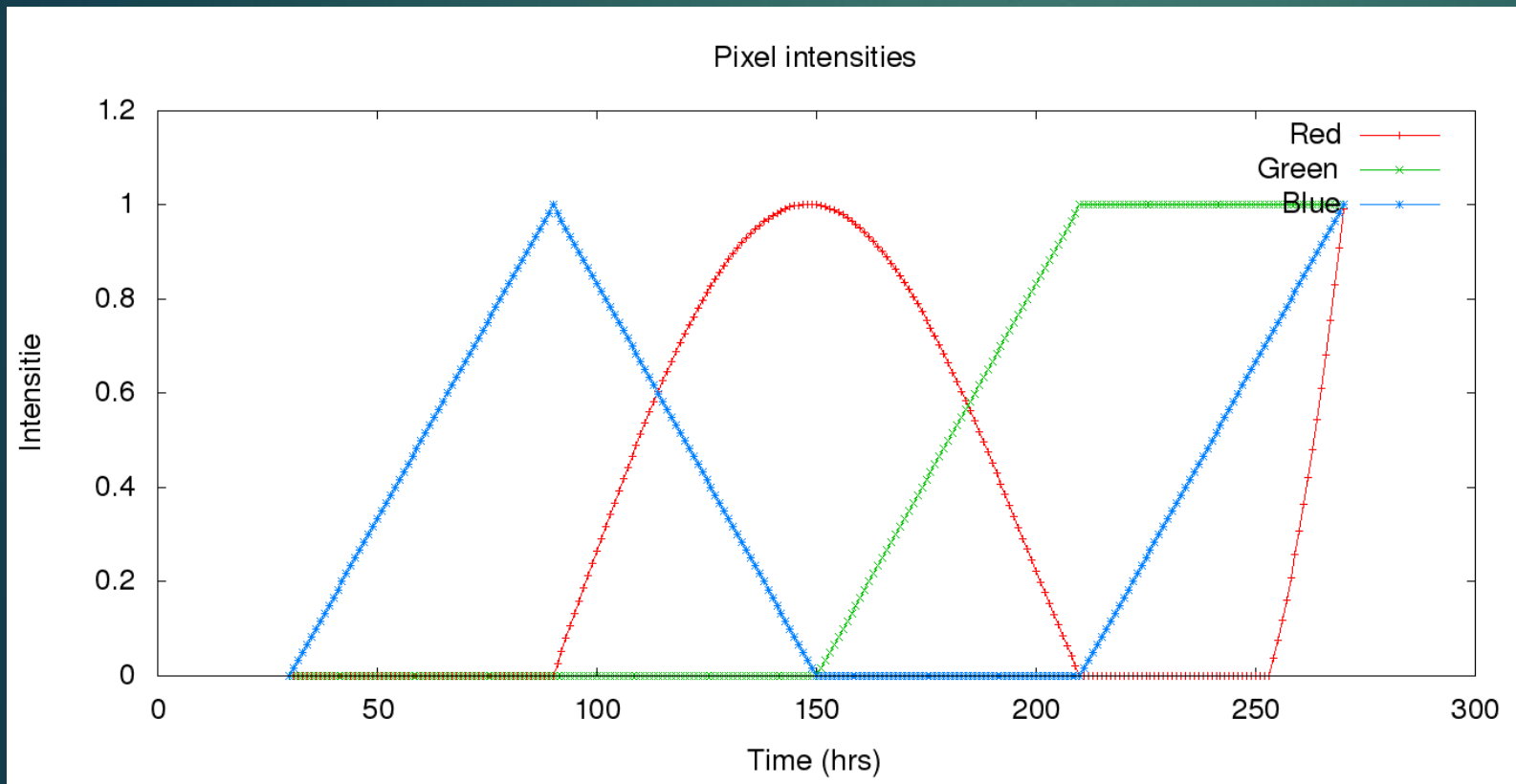
- Usa o polinômio de Lagrange para a interpolação:

$$P(X) = \sum_{j=1}^k P_j(X), \text{ onde:}$$

$$P_j(X) = \prod_{k=1 \text{ e } k \neq j}^n y_j \frac{x - x_k}{x_j - x_k}$$



# Lagrange



# Gauss forward central difference

- ▶ Supõe que as observações são equidistantes
- ▶ Assume uma equação polinomial utilizando as diferenças centrais

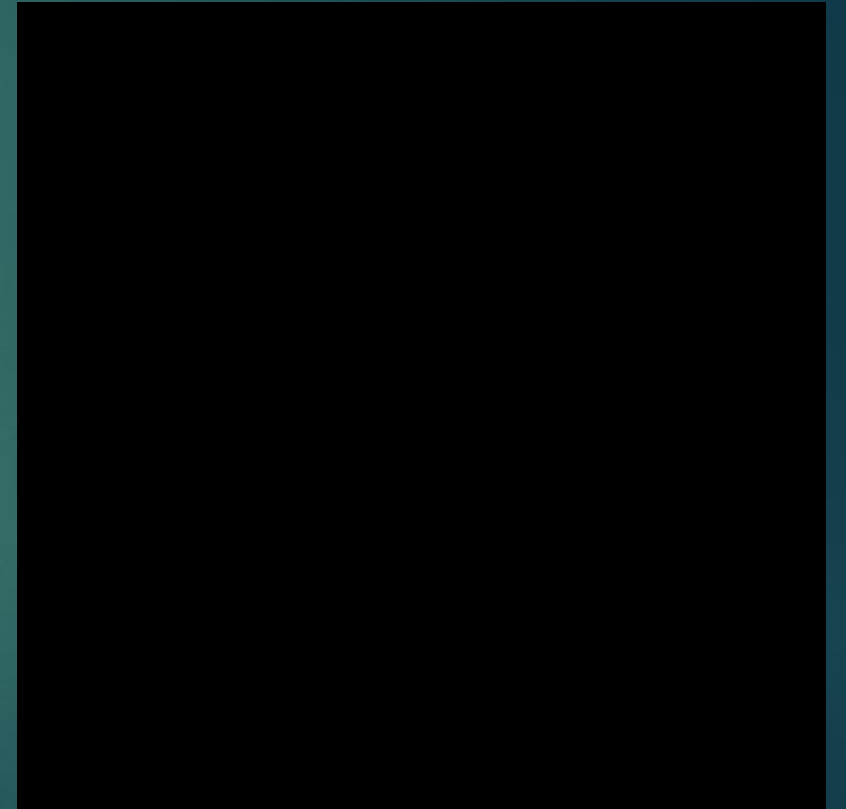
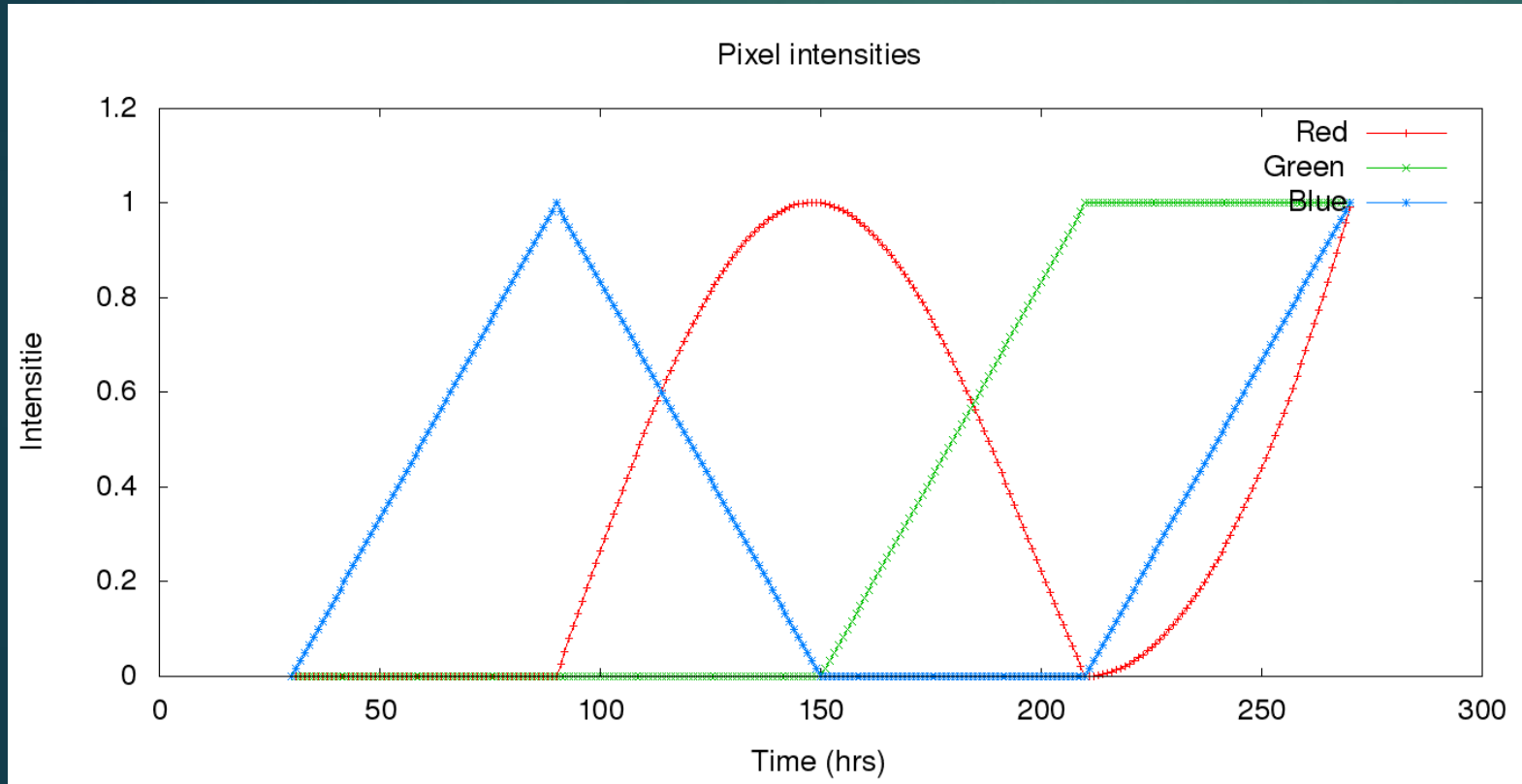
$$p = \frac{x - x_0}{h}, \text{ onde}$$

$x_0$  é a observação mais próxima do ponto que queremos interpolar e  $h$  é o período

$$y_p = y_0 + p\Delta y_0 + \frac{p(p-1)}{2!}\Delta^2 y_{-1} + \frac{p(p-1)(p+1)}{3!}\Delta^3 y_{-1} + \frac{p(p-1)(p+1)(p-2)}{4!}\Delta^4 y_{-2} + \dots$$

$x$	$y$	$\Delta y$	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
$\vdots$	$\vdots$				
$x_{-2}$	$y_{-2}$				
$x_{-1}$	$y_{-1}$	$\Delta y_{-2}$			
$x_0$	$y_0$	$\Delta y_{-1}$	$\Delta^2 y_{-2}$		
$x_1$	$y_1$	$\Delta y_0$	$\Delta^2 y_{-1}$	$\Delta^3 y_{-2}$	
$x_2$	$y_2$	$\Delta y_{-1}$	$\Delta^2 y_0$	$\Delta^3 y_{-1}$	$\Delta^4 y_{-2}$
$\vdots$	$\vdots$				

# Gauss Forward





# Gauss backward central difference

- Supõe que as observações são equidistantes
- Assume uma equação polinomial utilizando as diferenças centrais

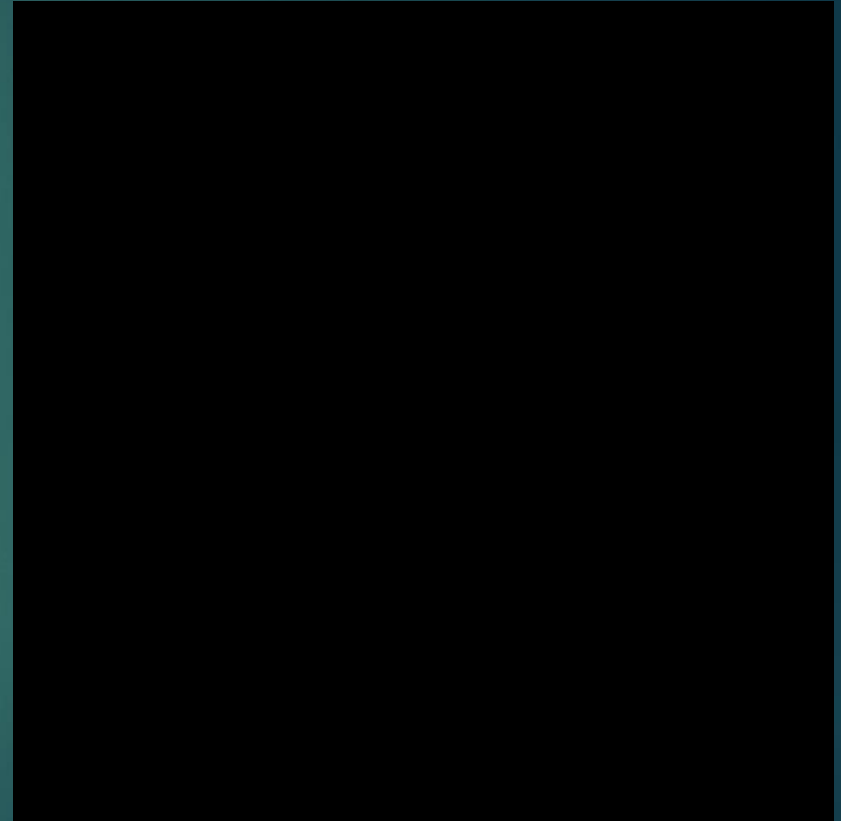
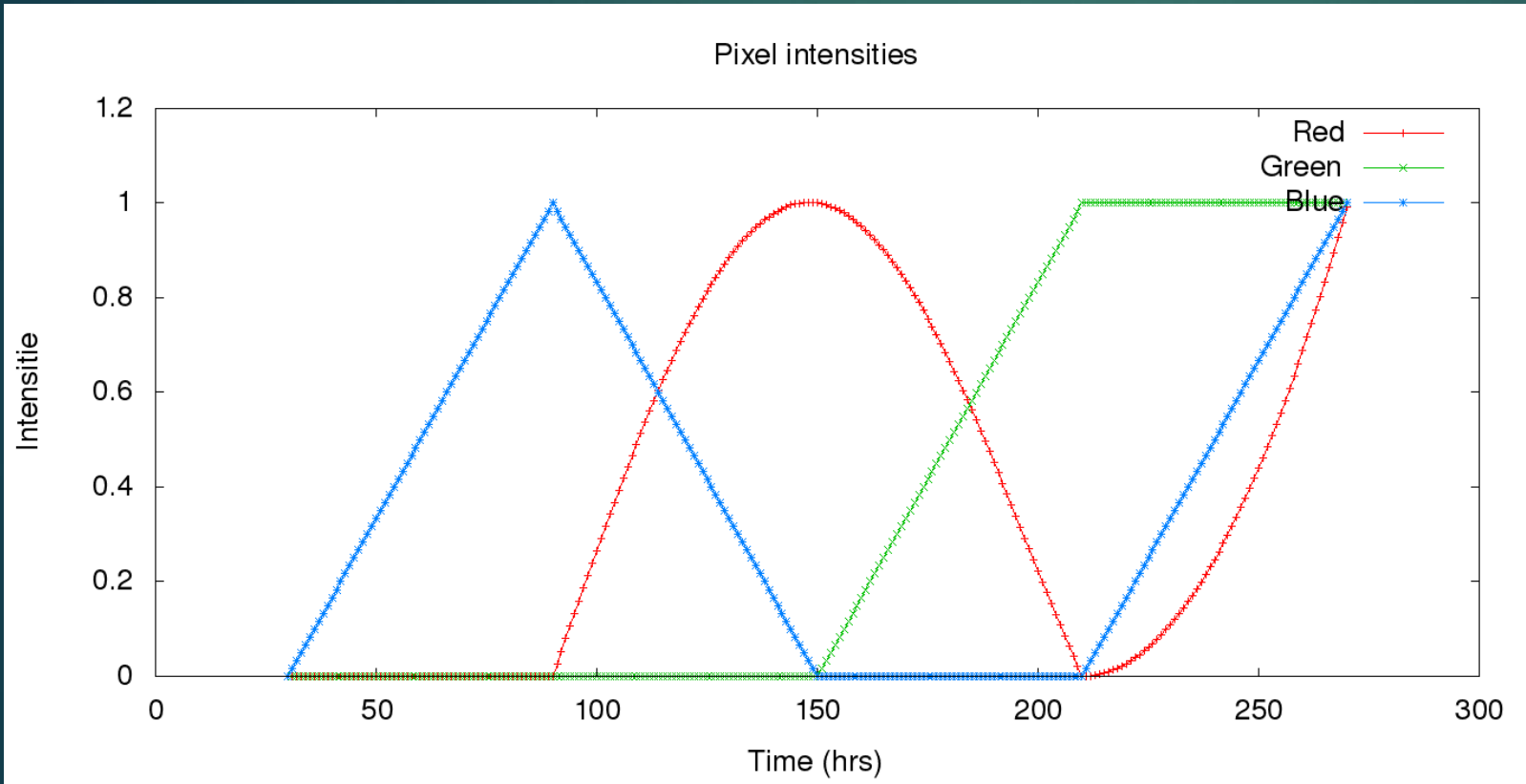
$$p = \frac{x - x_0}{h}, \text{ onde}$$

$x_0$  é a observação mais próxima do ponto que queremos interpolar e  $h$  é o período

$$y_p = y_0 + p\Delta y_{-1} + \frac{p(p+1)}{2!}\Delta^2 y_{-1} + \frac{p(p+1)(p-1)}{3!}\Delta^3 y_{-2} + \frac{p(p-1)(p+1)(p+2)}{4!}\Delta^4 y_{-2} + \dots$$

$x$	$y$	$\Delta y$	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
$\vdots$	$\vdots$				
$x_{-2}$	$y_{-2}$				
$x_{-1}$	$y_{-1}$	$\Delta y_{-2}$			
$x_0$	$y_0$	$\Delta y_{-1}$	$\Delta^2 y_{-2}$		
$x_1$	$y_1$	$\Delta y_0$	$\Delta^2 y_{-1}$	$\Delta^3 y_{-2}$	
$x_2$	$y_2$	$\Delta y_{-1}$	$\Delta^2 y_0$	$\Delta^3 y_{-1}$	$\Delta^4 y_{-2}$
$\vdots$	$\vdots$				

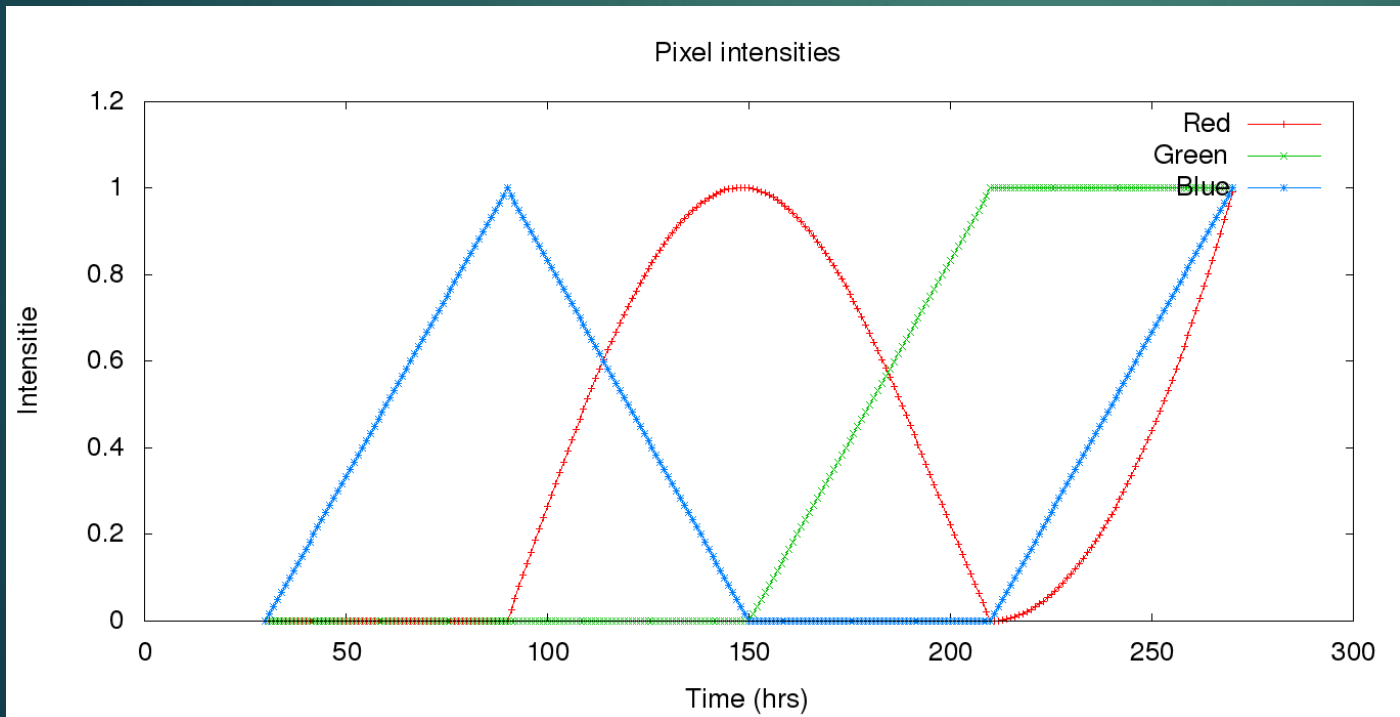
# Gauss Backward



# Stirling

- Supõe que as observações são equidistantes

$$\text{Stirling} = \frac{(\text{Gauss forward central difference} + \text{Gauss backward central difference})}{2}$$





# Código

- ▶ O código está em C++
- ▶ Utilizamos a biblioteca FreeImage para leitura e escrita do formato HDR
- ▶ Nosso código é responsável por ler os Light Probes, criar a interpolação para o momento escolhido e gerar o Light Probe artificial
- ▶ Com isso podemos gerar uma cena no Maya

# Light Probes Interpolados

- ▶ Gauss Forward/Backward e por consequência Stirling apresentam os mesmos resultados
- ▶ Note que as intensidades numa imagem HDR não são limitadas entre 0.0 e 1.0, logo suas diferenças também não são limitadas
- ▶ Definimos um critério simples de distancia entre imagens como:

$$\text{Distância}(Im1, Im2) = \sum_i^n \sum_j^m \sum_c^{red|green|blue} |Im1[i][j].c - Im2[i][j].c|$$

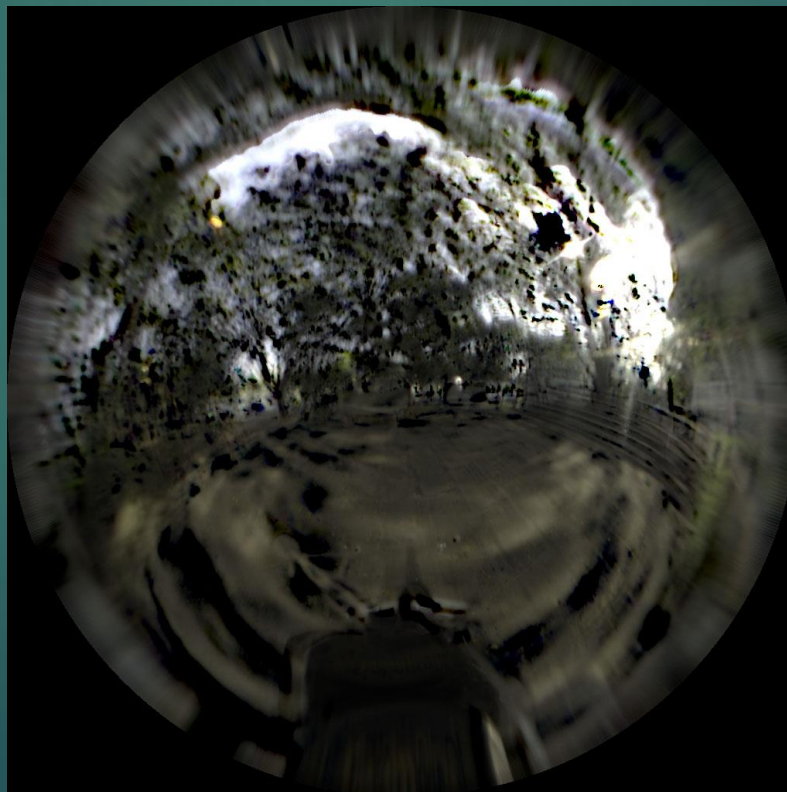
# Light Probes Interpolados

- ▶ As interpolações polinomiais podem gerar pixels com valores negativos, nesses casos travamos as intensidades em 0.0
  - ▶ Áreas pretas

Original



Lagrange



Linear





# Interpolação – Cena externa 1

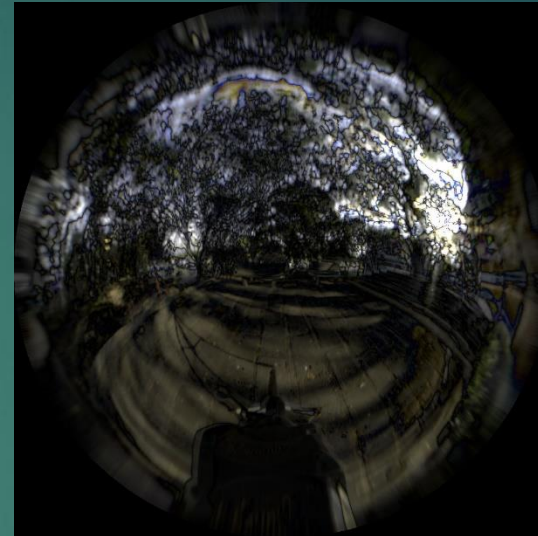
Gauss Forward



Original



Interpolada

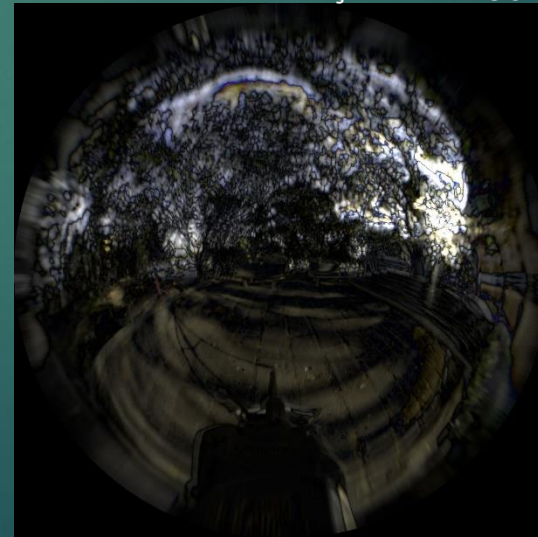


Diferença

Linear

Distancia

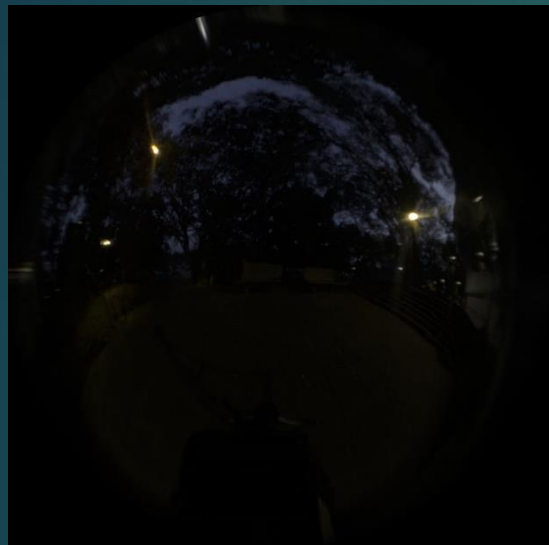
160554



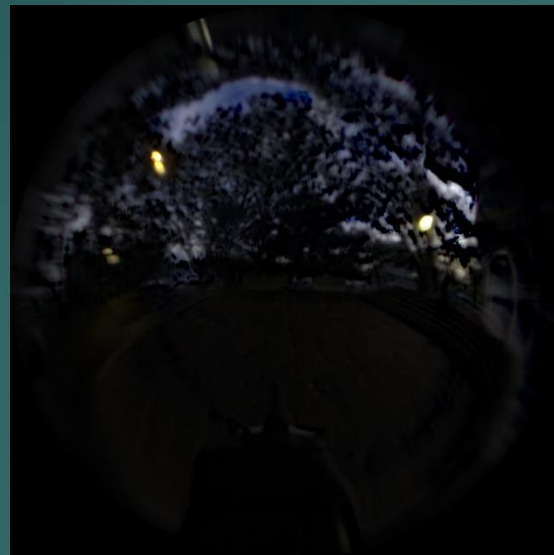
133904

# Interpolação – Cena externa 3

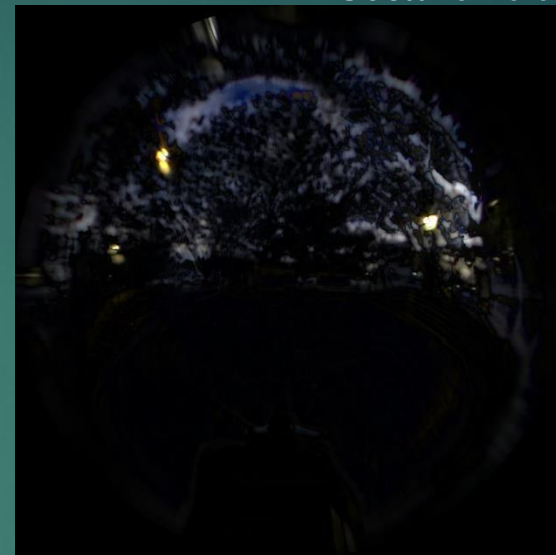
Gauss Forward



Original



Interpolada

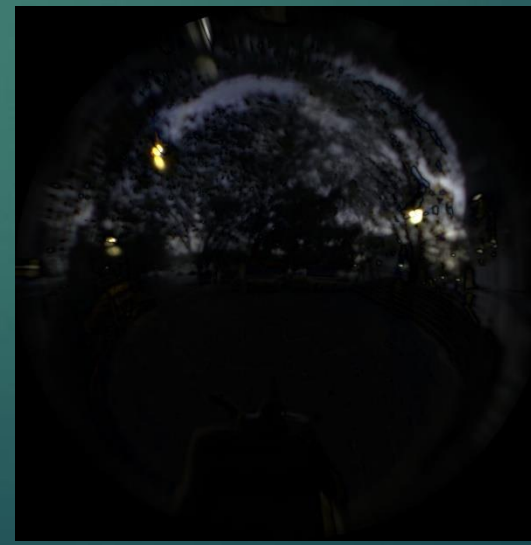
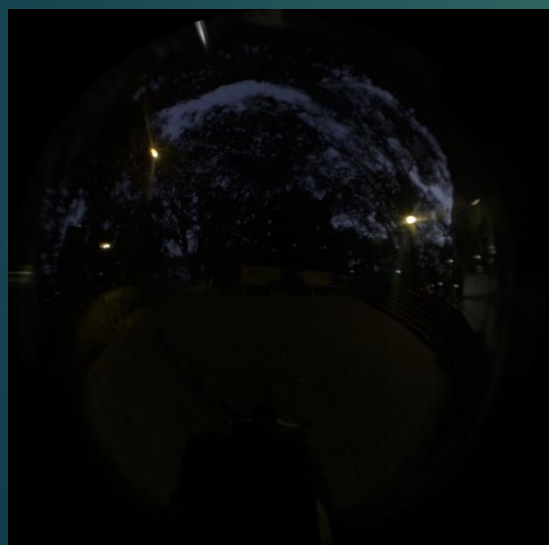


Diferença

Linear

Distancia

31024.3



32425.1

# Interpolação – Cena interna 1

Gauss Forward



Original



Interpolada



Diferença

Linear

Distancia

130939



127874

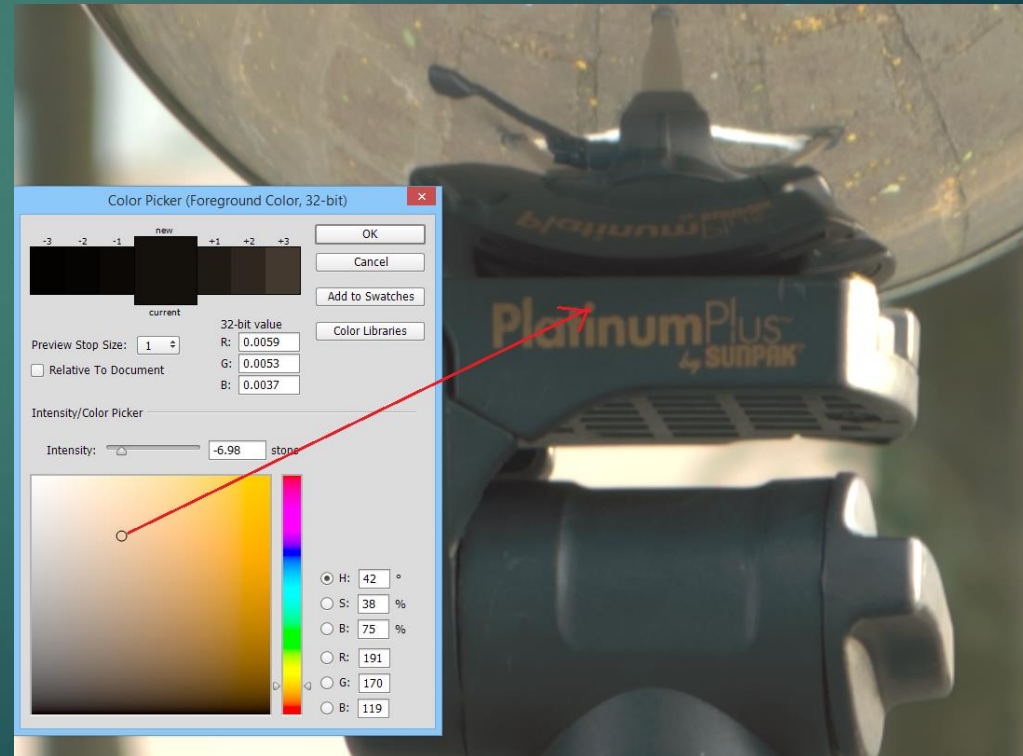
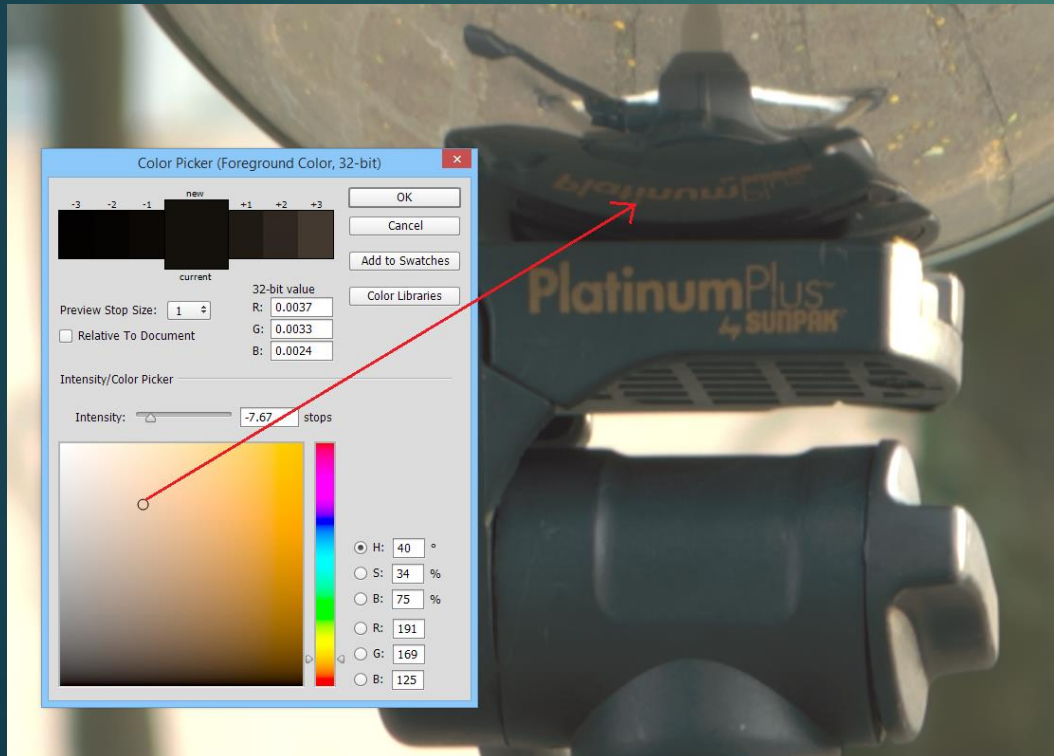


# Cenas Renderizadas

- ▶ Para cada cena de teste utilizamos as interpolações para gerarmos 5 light probes: linear, lagrange, gauss forward, backward, stirling
- ▶ Para cada cena de testes escolhemos as duas interpolações que apresentavam distancias mínimas, em relação ao light probe base, para prosseguir para a fase de renderização com o cubo branco

# Coeficiente de refletância

- ▶ O coeficiente de refletância da esfera foi determinada
  - ▶ Coeficiente = 1.594594...
- ▶ Mas não foi usado, o coeficiente teve que ser ajustado manualmente para cada cena, variando de 1.8 a 4.6



# Renderização – Cena externa 1



Imagem original

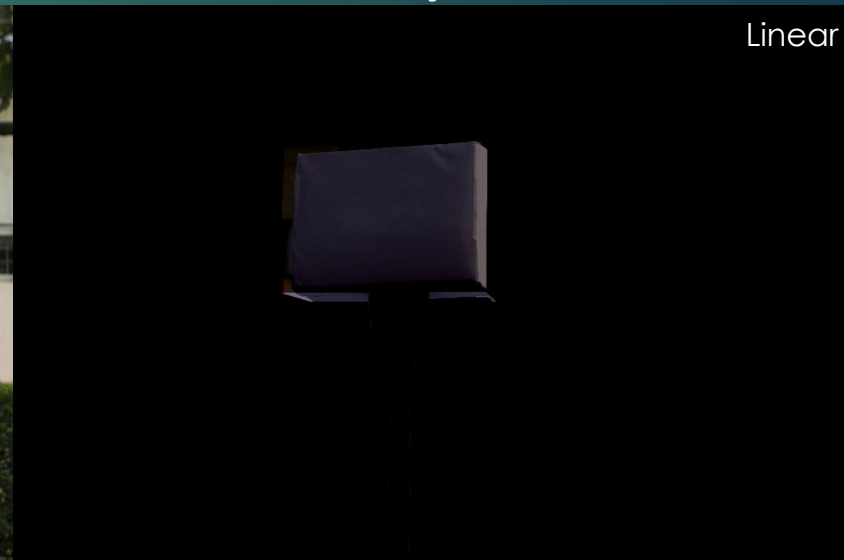


Imagem renderizada



Gauss Forward

Diferença



Linear



# Renderização – Cena externa 3

Gauss Forward



Imagem original

Imagem renderizada

Diferença

Linear





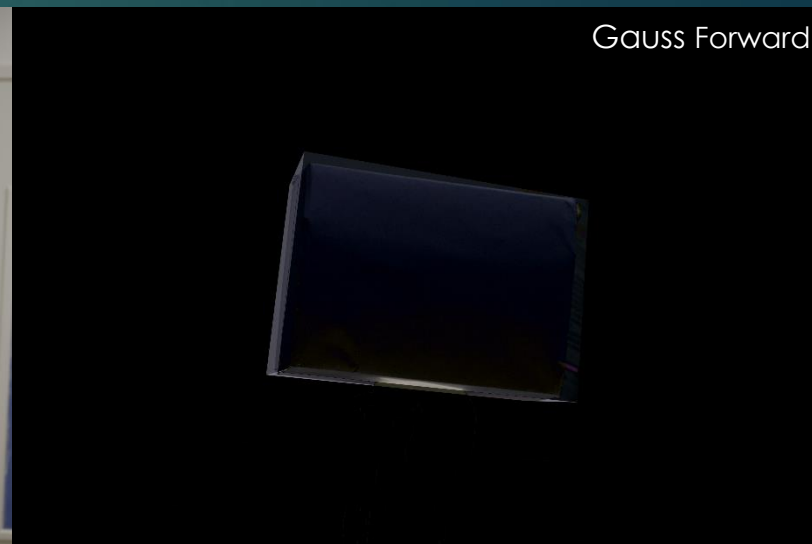
# Renderização – Cena interna 1



Imagem original



Imagem renderizada



Gauss Forward



Linear

# Conclusões

- ▶ Interpolação linear em todos os testes teve a menor ou a segunda menor distância em relação ao light probe original
- ▶ Gauss foi a segunda melhor interpolação, perdendo pra lagrange em apenas um caso
- ▶ Interpolações polinomiais de grau elevado como lagrange ou gauss podem gerar um erro muito grande
  - ▶ Áreas pretas nas imagens interpoladas
- ▶ A comparação do cubo não foi satisfatória por ter muitos elementos que não conseguimos capturar de maneira precisa
  - ▶ Coeficiente de refletância da esfera não foi usável no Maya
  - ▶ Modelagem e posicionamento do cubo

# Próximos passos

- ▶ Explorar outros tipos de interpolação locais de grau baixo, como interpolação quadrática ou cubica, de modo a reduzir o erro e tentar evitar áreas pretas nas imagens resultantes

# Referências

- ▶ Paul E. Debevec, J. M. (1997). *Recovering high dynamic range radiance maps from photographs*. Paper presented at the SIGGRAPH.
- ▶ Debevec, P. (2002). Tutorial: Image-Based Lighting. *IEEE Computer Graphics and Applications*, 26 – 34.
- ▶ Pozrikidis, C. (1998). *Numerical Computation in Science and Engineering* (O. U. Press Ed. 1 ed.).
- ▶ Milton Abramowitz, I. S. (1972). *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*: Courier Dover Publications.