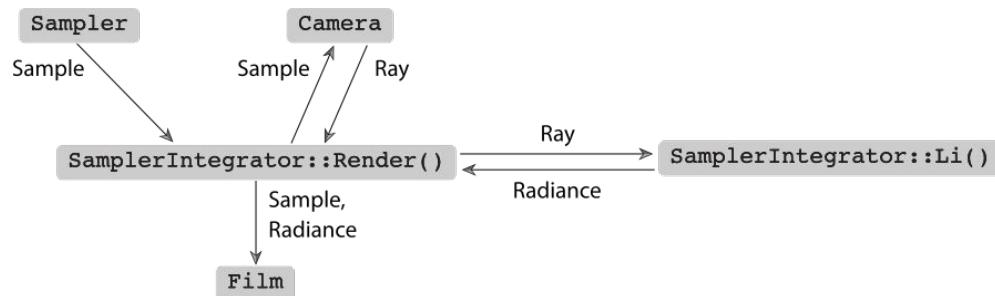


# Aceleración de RayTracer Random Parameter Filtering

- Rodrigo Gabriel Salazar Alva
- Alvaro Guerrero Jiménez

# PBRT

## RayTracing



# Fórmula de Radiancia

$$I(i, j) = \int_{i-\frac{1}{2}}^{i+\frac{1}{2}} \int_{j-\frac{1}{2}}^{j+\frac{1}{2}} \cdots \int_{-1}^1 \int_{-1}^1 \int_{t_0}^{t_1} f(x, y, \dots, u, v, t) dt dv du \cdots dy dx.$$

## Parámetros

- $x, y$  : posición en plano de proyección
- $u, v$  : posición del lente
- $t$  : tiempo de captura
- +parámetros aleatorios

## Aproximación:

- **Método:** Monte-Carlo
- **Variación:**  $O(\frac{1}{n})$

# Escena

**RenderSystem:** PBRT v2

**Escena:** Conference

**Samples Per Pixel:** 8096

**Tiempo de Ejecución:** 32m 33s



# Profiling



Profile	
Integrator::Render()	100.00% ( 2:24:28.66)
Camera::GenerateRay[Differential]()	1.35% ( 0:01:57.24)
Film::AddSample()	0.42% ( 0:00:36.25)
Film::MergeTile()	0.00% ( 0:00:00.05)
Sampler::GetSample[12]D()	0.47% ( 0:00:41.17)
Sampler::StartPixelSample()	0.56% ( 0:00:48.72)
<b>SamplerIntegrator::Li()</b>	<b>96.51% ( 2:19:26.20)</b>
Accelerator::Intersect()	40.84% ( 0:59:00.23)
Other Shape::Intersect()	0.02% ( 0:00:01.31)
Triangle::Intersect()	10.38% ( 0:14:59.87)
BSDF::Sample_f()	8.55% ( 0:12:20.85)
Direct lighting	39.39% ( 0:56:54.27)
Accelerator::Intersect()	0.03% ( 0:00:02.90)
Other Shape::Intersect()	0.01% ( 0:00:00.81)
Triangle::Intersect()	0.01% ( 0:00:00.48)
Accelerator::IntersectP()	11.15% ( 0:16:06.23)

8096 Samples Per Pixel (PBRT v3)

# Random Parameter Filtering

## On Filtering the Noise from the Random Parameters in Monte Carlo Rendering

*ACM Transactions on Graphics (TOG)*  
Vol. 31, No. 3, May 2012

Pradeep Sen

Soheil Darabi

UNM Advanced Graphics Lab



input Monte Carlo (8 samples/pixel)



after RPF (8 samples/pixel)

# Propuesta de RPF

**Hipótesis:** Es posible mejorar la aproximación de radiancia de un pixel incorporando data de píxeles cercanos

**Propuesta:**

*Filtro bilateral cruzado ajustado a las dependencias entre parámetros aleatorios y características de las muestras*

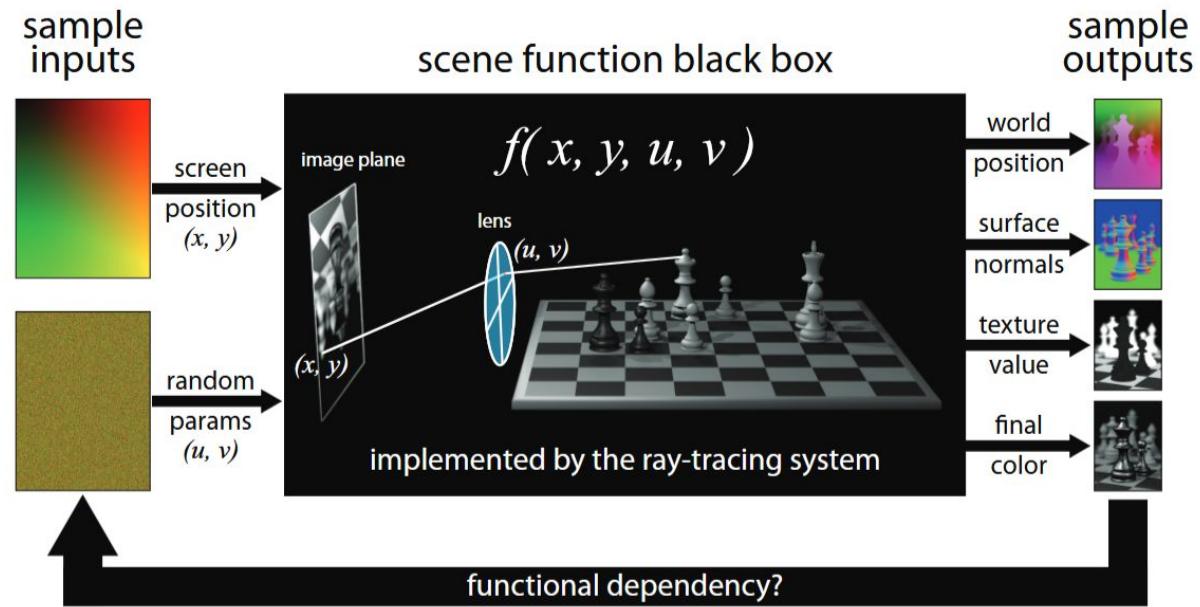
$$w_{ij} = \exp\left[-\frac{1}{2\sigma_{\mathbf{p}}^2} \sum_{1 \leq k \leq 2} (\bar{\mathbf{p}}_{i,k} - \bar{\mathbf{p}}_{j,k})^2\right] \times \\ \exp\left[-\frac{1}{2\sigma_{\mathbf{c}}^2} \sum_{1 \leq k \leq 3} \alpha_k (\bar{\mathbf{c}}_{i,k} - \bar{\mathbf{c}}_{j,k})^2\right] \times \\ \exp\left[-\frac{1}{2\sigma_{\mathbf{f}}^2} \sum_{1 < k < m} \beta_k (\bar{\mathbf{f}}_{i,k} - \bar{\mathbf{f}}_{j,k})^2\right],$$

- Posición semejante = Mayor peso
- Color semejante = Mayor peso
- Características semejantes = Mayor peso

# Renderizado

Creación del vector X para cada *sample* durante el renderizado:

$$X_i = \{p_{i,1}, p_{i,2}\} \parallel \{r_{i,1}, r_{i,2}, \dots\} \parallel \{f_{i,1}, f_{i,2}, \dots\} \parallel \{c_{i,1}, c_{i,2}, c_{i,3}\}$$



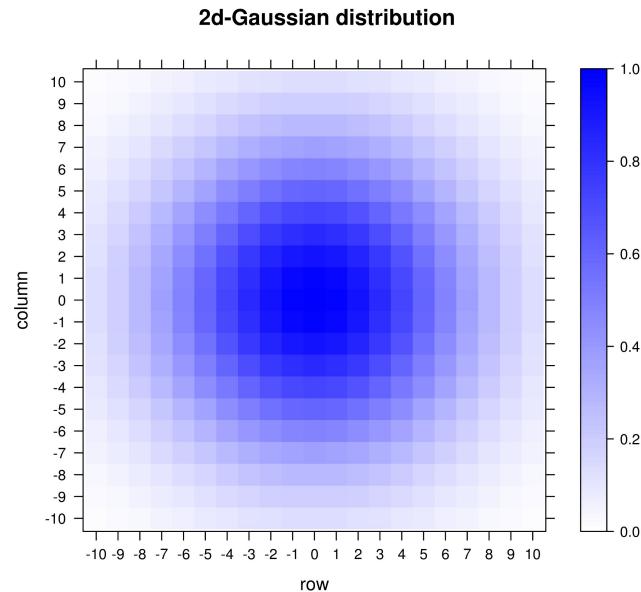
Post-procesamiento: RPF es independiente del método de integración empleado.

# Construcción del *neighbourhood*

## No mezclar relaciones de dependencia

- **Localidad:** Cuadrado de tamaño `box_size`
- **Relevancia:** Filtrado en base a semejanza de propiedades.

```
for scene feature  $k = 1$  to  $m$  do
    if  $|\mathbf{f}_{j,k} - \mathbf{m}_{\mathcal{P},k}^{\mathbf{f}}| > \{3|30\}\sigma_{\mathcal{P},k}^{\mathbf{f}}$  and
        $|\mathbf{f}_{j,k} - \mathbf{m}_{\mathcal{P},k}^{\mathbf{f}}| > 0.1$  or  $\sigma_{\mathcal{P},k}^{\mathbf{f}} > 0.1$  then
         $flag \leftarrow 0$ 
        break
    end if
end for
if  $flag$  equals to 1 then
     $\mathcal{N} \leftarrow$  sample  $j$ 
end if
```



**Optimización:** Selección aleatoria de píxeles en base a distribución gaussian

# Estimación de dependencias estadísticas

## Información Mutua

Cuantificación de la dependencia entre dos variables aleatorias.

$$I(X; Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

## Dependencia estadística

- 1D x 1D:  $D_{y,k}^{x,l} = \mu(X_{\mathcal{N},l}; X_{\mathcal{N},k})$
- 1D x MD:  $D_{y,k}^x \approx \sum_l D_{y,k}^{x,l} = \sum_l \mu(X_{\mathcal{N},l}; X_{\mathcal{N},k})$
- MD x MD:  $D_y^x \approx \sum_k D_{y,k}^x$

# Contribuciones fraccionales

$$\bar{W}_{\mathbf{f},k}^{\mathbf{r}} = \frac{D_{\mathbf{f},k}^{\mathbf{r}}}{D_{\mathbf{f},k}^{\mathbf{r}} + D_{\mathbf{f},k}^{\mathbf{p}} + \varepsilon}$$

Contribución de los parámetros aleatorios al feature k

$$W_{\mathbf{c},k}^{\mathbf{r}} = \frac{D_{\mathbf{c},k}^{\mathbf{r}}}{D_{\mathbf{c},k}^{\mathbf{r}} + D_{\mathbf{c},k}^{\mathbf{p}} + \varepsilon}$$

Contribución de los parámetros aleatorios al canal de color k

$$W_{\mathbf{c}}^{\mathbf{f},k} = \frac{D_{\mathbf{c}}^{\mathbf{f},k}}{D_{\mathbf{c}}^{\mathbf{r}} + D_{\mathbf{c}}^{\mathbf{p}} + D_{\mathbf{c}}^{\mathbf{f}}}.$$

Contribución del feature k al color final

$$W_{\mathbf{c}}^{\mathbf{r}} = \frac{1}{3}(W_{\mathbf{c},1}^{\mathbf{r}} + W_{\mathbf{c},2}^{\mathbf{r}} + W_{\mathbf{c},3}^{\mathbf{r}})$$

Contribución de los parámetros aleatorios al color final

# Factores alfa y beta

$$\alpha_k = 1 - W_{\mathbf{c},k}^{\mathbf{r}}$$

Alpha:

A medida que los parámetros aleatorios generan una menor contribución sobre el color, incrementa la importancia de su diferencia para el filtro

$$\beta_k = W_{\mathbf{c}}^{\mathbf{f},k} (1 - W_{\mathbf{f},k}^{\mathbf{r}})$$

Beta:

A medida de que un feature contribuye más a los colores y ha sido afectado menos por los parámetros aleatorios, incrementa la importancia de su diferencia para el filtro

Ajustes para múltiples pasadas del filtro

$$\alpha_k = \max(1 - 2(1 + 0.1t)W_{\mathbf{c},k}^{\mathbf{r}}, 0),$$

$$\beta_k = W_{\mathbf{c}}^{\mathbf{f},k} \cdot \max(1 - (1 + 0.1t)W_{\mathbf{f},k}^{\mathbf{r}}, 0),$$

# Influencia de Vecino j sobre Pixel i

- Penaliza distancia  
(posición, color y features)
- Alpha y Beta ajustan  
penalización
- Cuando i=j siempre es 1

$$w_{ij} = \exp\left[-\frac{1}{2\sigma_p^2} \sum_{1 \leq k \leq 2} (\bar{\mathbf{p}}_{i,k} - \bar{\mathbf{p}}_{j,k})^2\right] \times \\ \exp\left[-\frac{1}{2\sigma_c^2} \sum_{1 \leq k \leq 3} \alpha_k (\bar{\mathbf{c}}_{i,k} - \bar{\mathbf{c}}_{j,k})^2\right] \times \\ \exp\left[-\frac{1}{2\sigma_f^2} \sum_{1 < k < m} \beta_k (\bar{\mathbf{f}}_{i,k} - \bar{\mathbf{f}}_{j,k})^2\right],$$

# Mezclar Samples

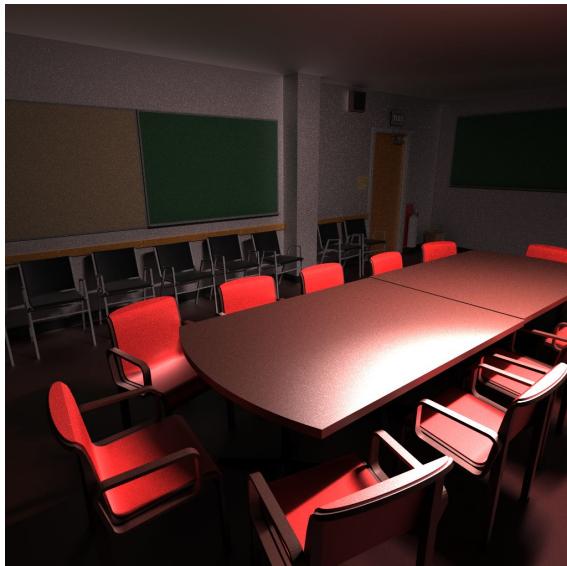
$$\mathbf{c}'_{i,k} = \frac{\sum_{j \in \mathcal{N}} w_{ij} \mathbf{c}_{j,k}}{\sum_{j \in \mathcal{N}} w_{ij}},$$

Promedio ponderado de los radiancia de los vecinos.

# Experimentación: Mismo SPP

PBRTv2 (Latest)

8096spp



32m 20.0s

PBRTv2 (Base)

8spp



1.8s

PBRTv2 (RPF)

8spp



3m 41.7s

# Experimentación: Mismo SPP

PBRTv2 (Latest)

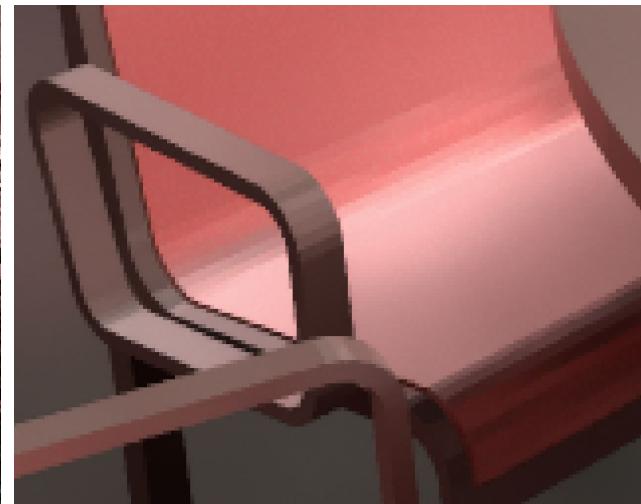
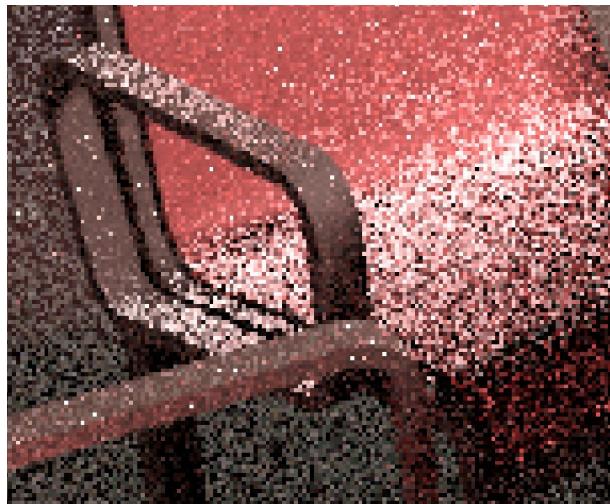
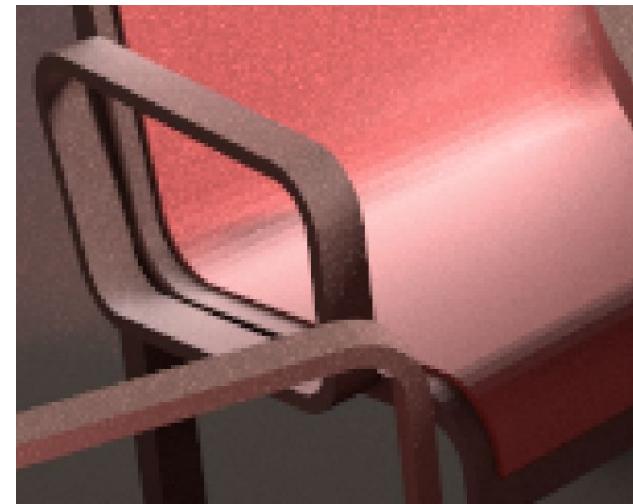
8096spp

PBRTv2 (Base)

8spp

PBRTv2 (RPF)

8spp



32m 20.0s

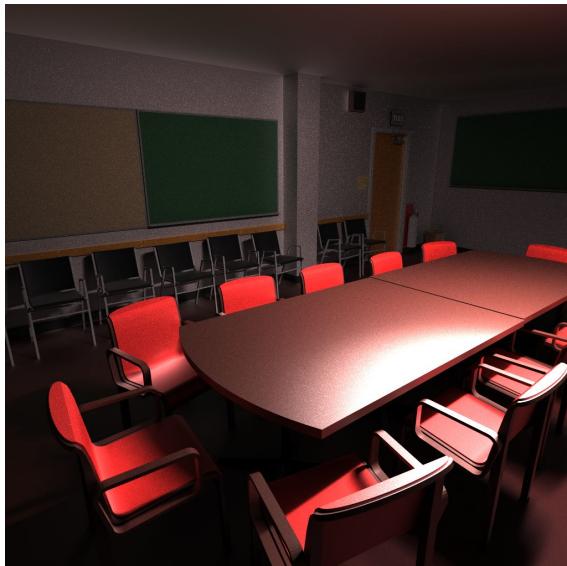
1.8s

3m 41.7s

# Experimentación: Tiempo comparable

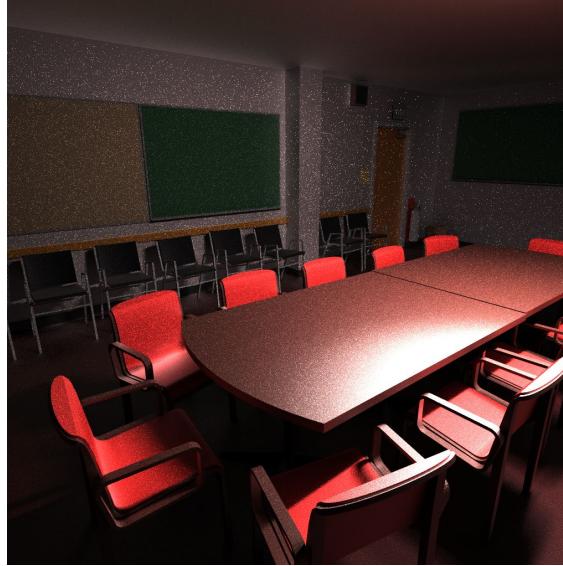
PBRTv2 (Latest)

8096spp



PBRTv2 (Latest)

1024spp



PBRTv2 (RPF)

8spp



32m 20.0s

3m 52.2s

3m 41.7s

# Experimentación: Tiempo comparable

PBRTv2 (Latest)

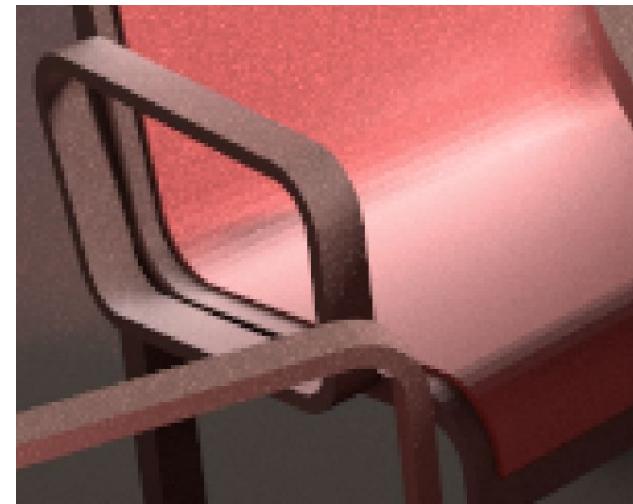
8096spp

PBRTv2 (Latest)

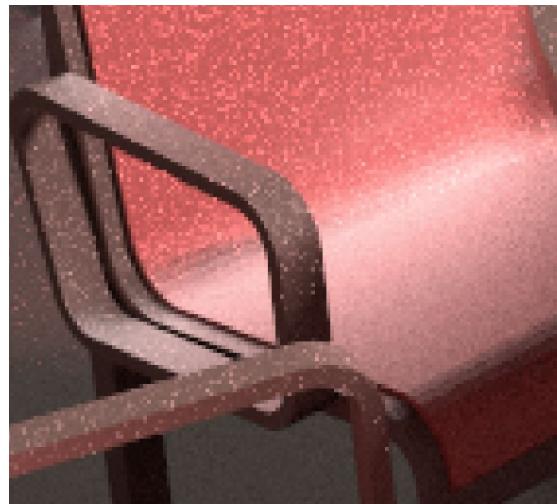
1024spp

PBRTv2 (RPF)

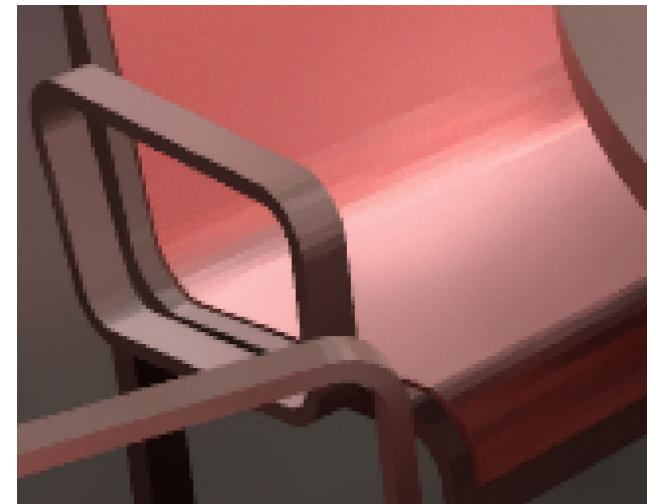
8spp



32m 20.0s



3m 52.2s

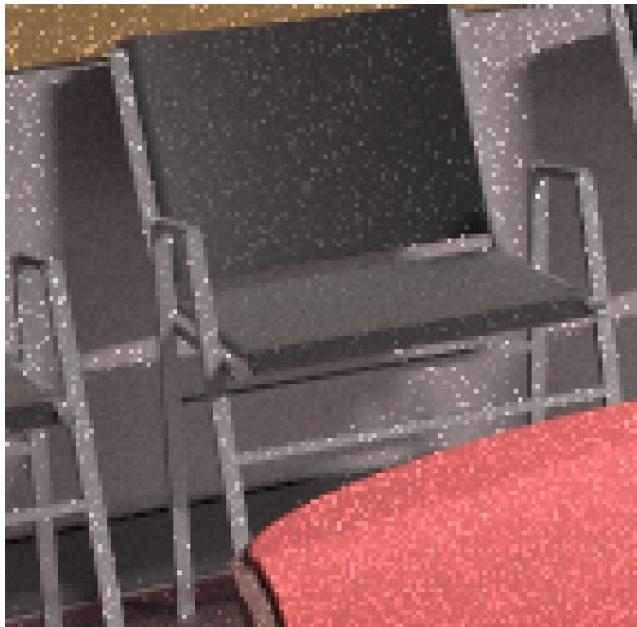


3m 41.7s

# Experimentación: Discrepancias

PBRTv2 (Latest)

8096spp



PBRTv2 (RPF)

8spp



# Experimentación: Discrepancias

PBRTv2 (Latest)

8096spp

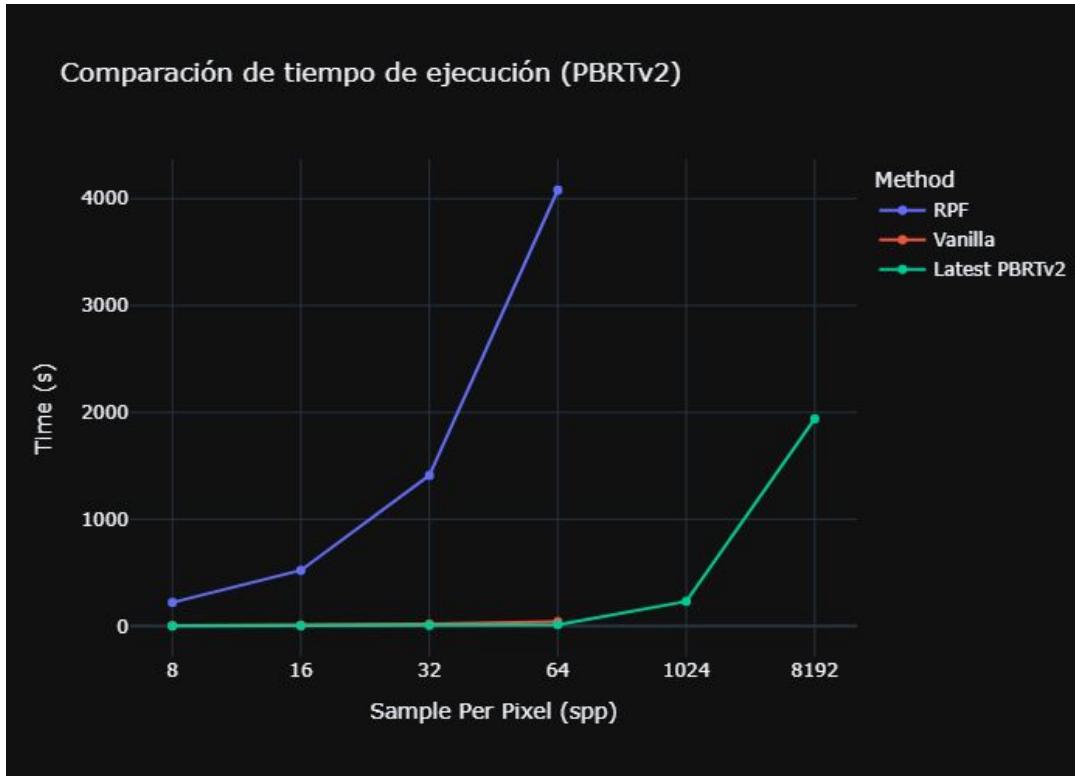


PBRTv2 (RPF)

8spp

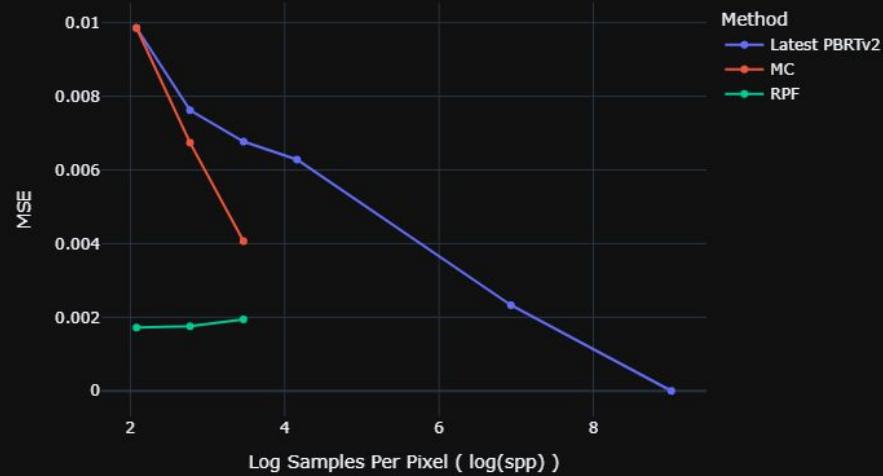


# Experimentación: Tiempo

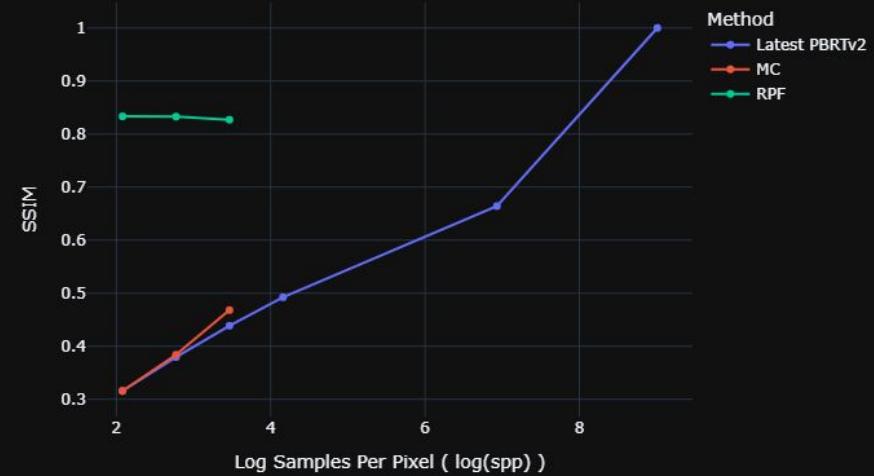


# Experimentación: Error / Similitud

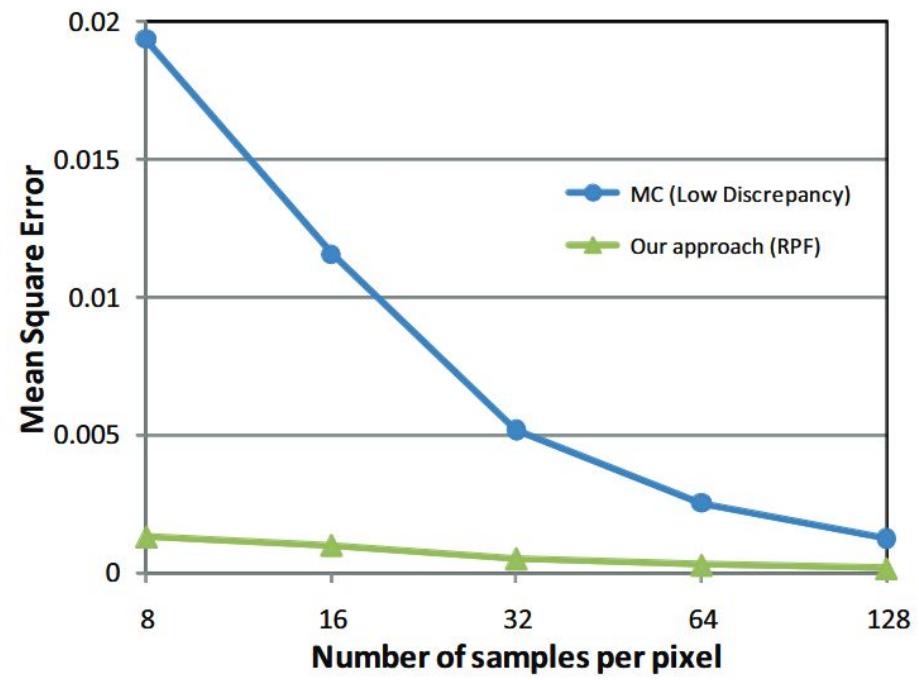
Comparación de MSE (PBRTv2)



Comparación de SSIM (PBRTv2)



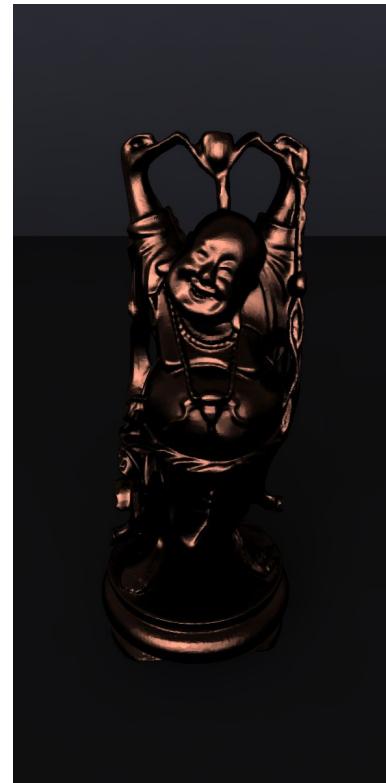
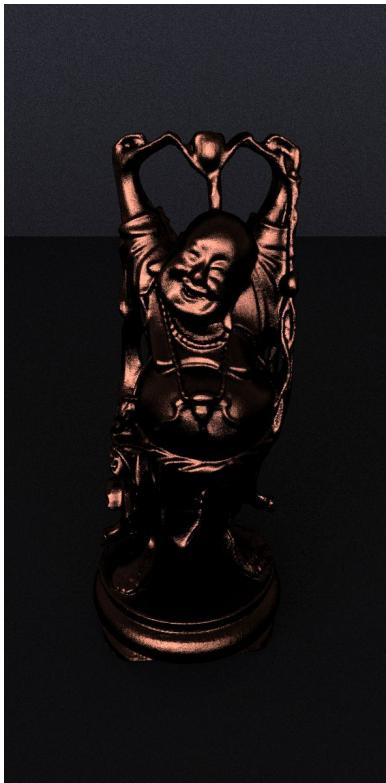
# Resultados del Paper



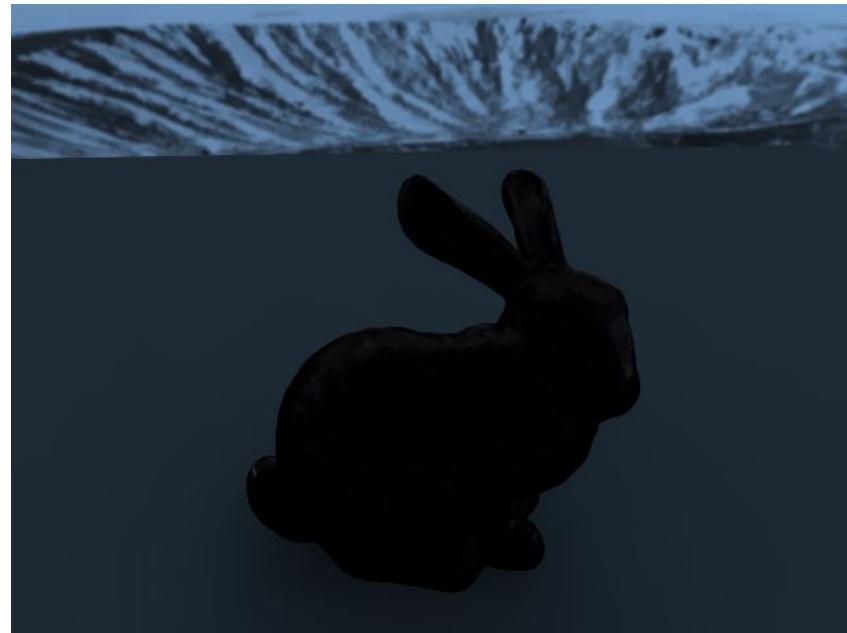
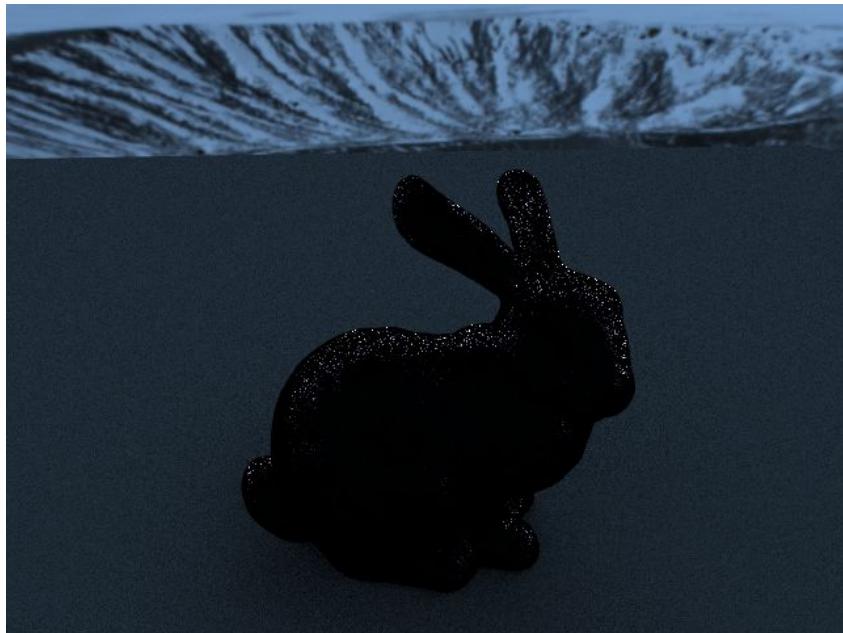
(e) Our approach (RPF)

(f) Reference Monte Carlo (8,192 spp)

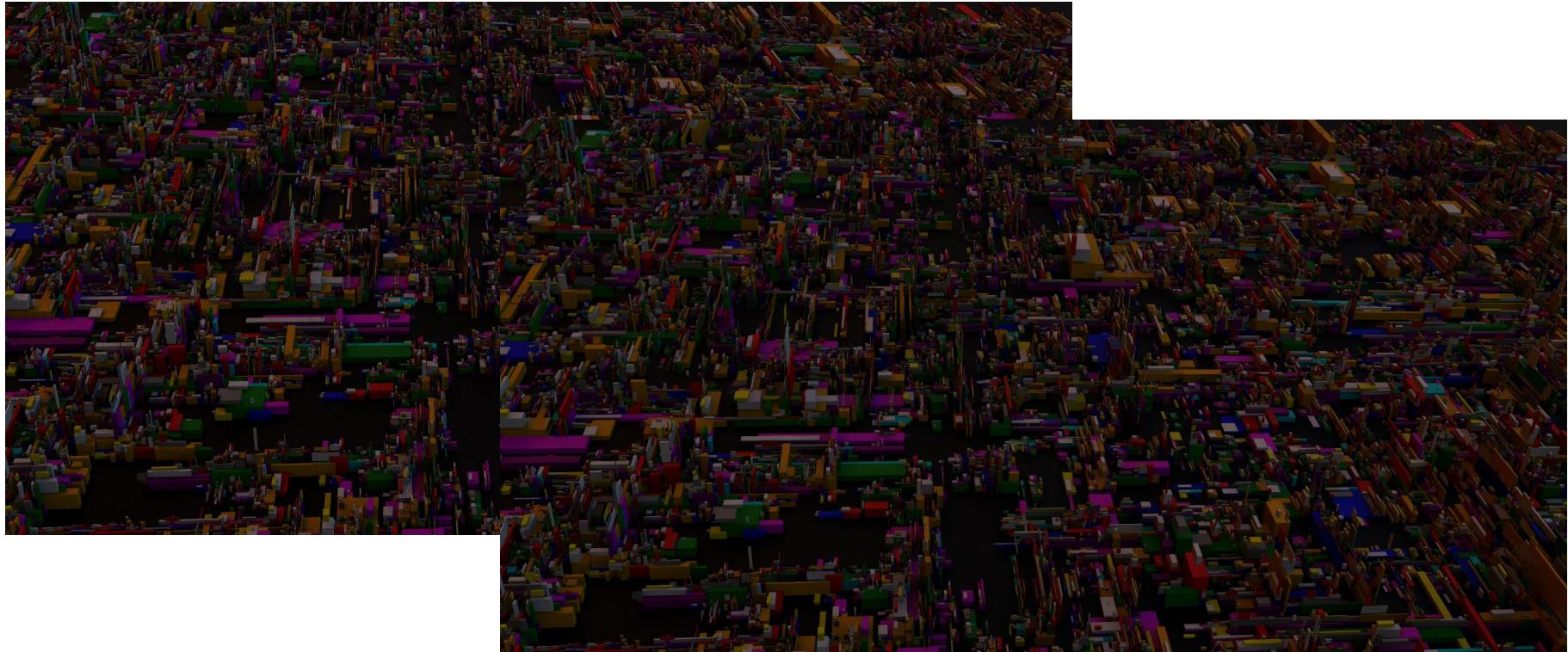
# Pruebas - Buddha



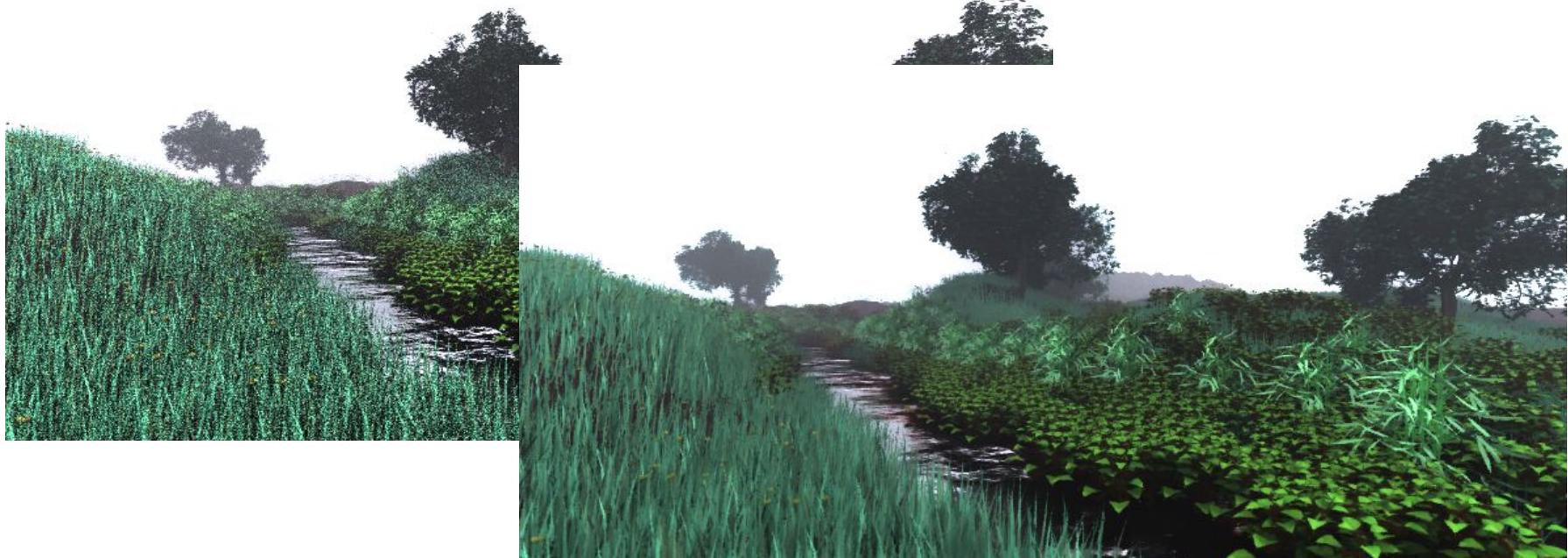
# Pruebas - Conejo



# Pruebas - Vista aérea de ciudad



# Pruebas - Vegetación



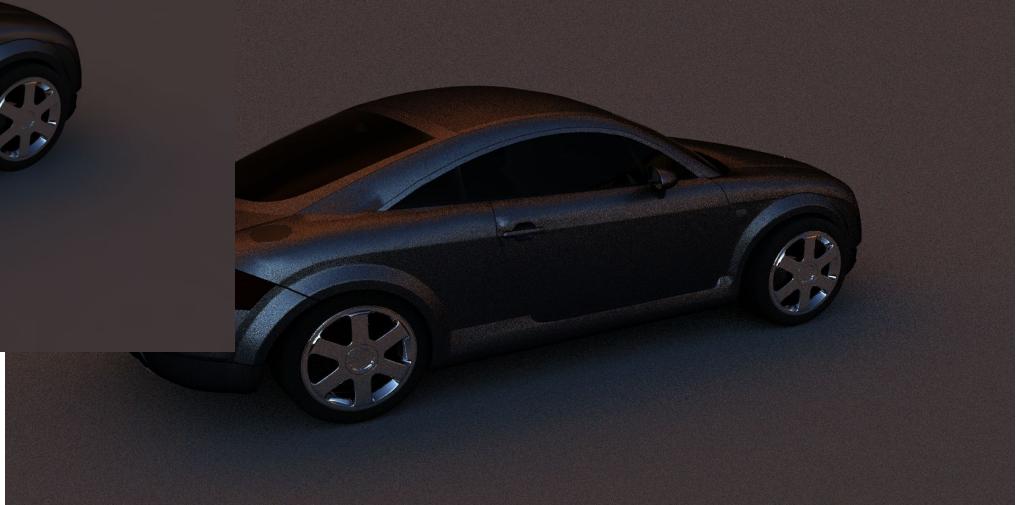
# Pruebas - Habitación



# Pruebas - Sponza



# Pruebas - Auto



# Pruebas - Villa



# Referencias

- Sen, P., & Darabi, S. (2011). *Implementation of Random Parameter Filtering*. University of New Mexico.
- Sen, P., & Darabi, S. (2012). On Filtering the Noise from the Random Parameters in Monte Carlo Rendering. *ACM Trans. Graph.*, 31(3), 18:1-18:15.