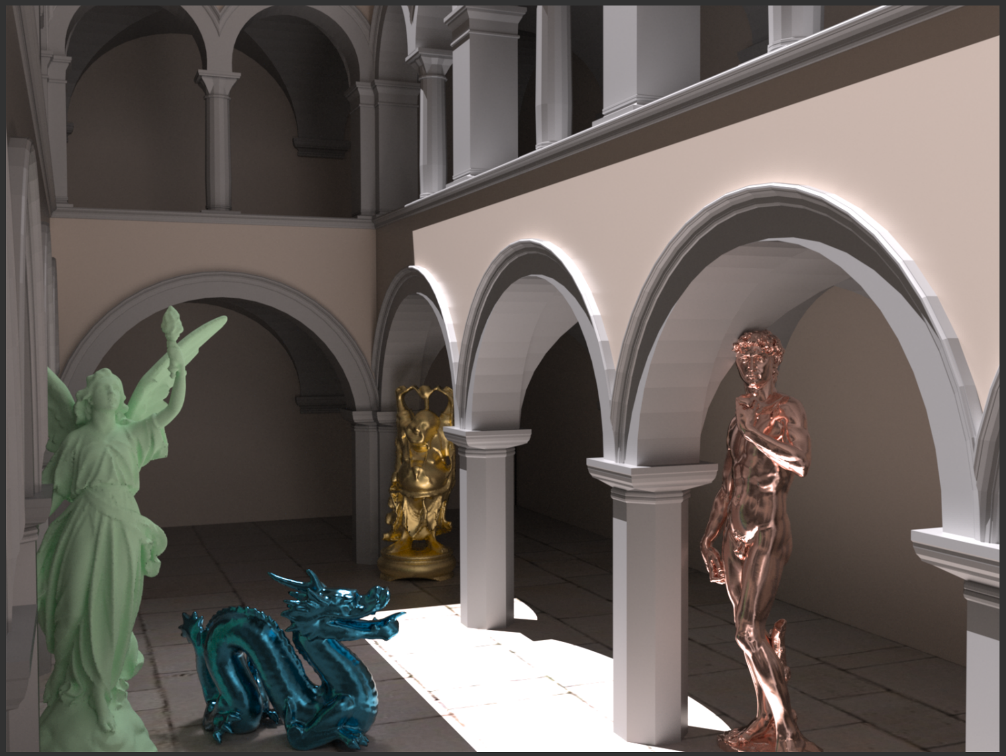
# Rational BRDF

R. Pacanowski, O. Salazar Celis,
C. Schlick, X. Granier, P. Poulin and A. Cuyt

LP2N - CNRS - INRIA - Bordeaux University
Université de Montréal - Universiteit Antwerpen

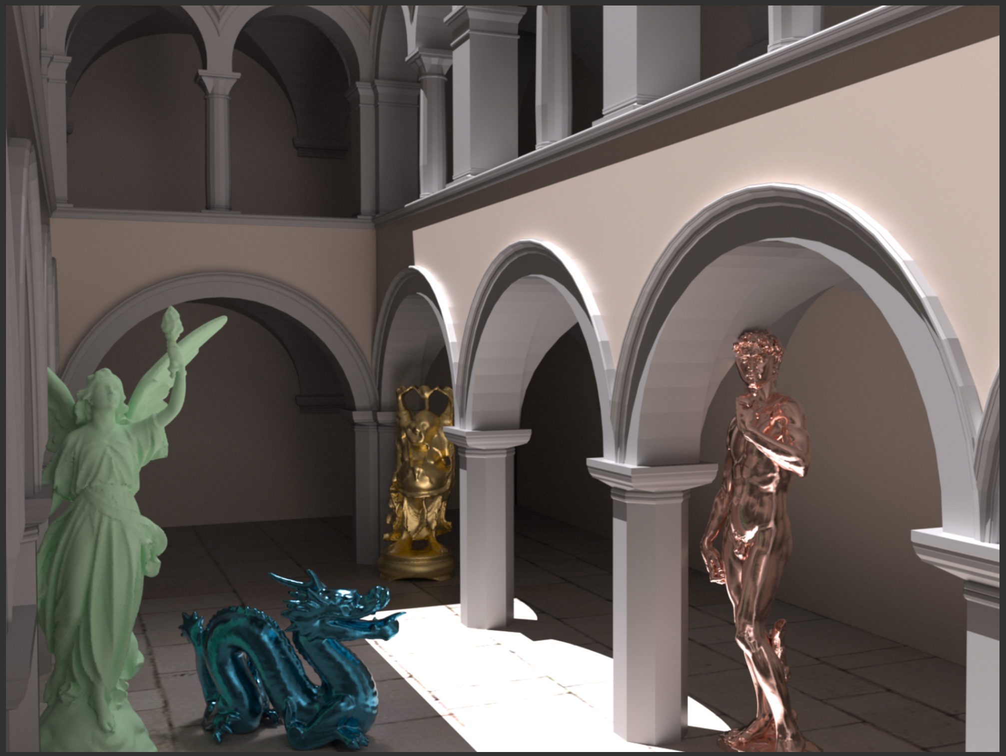
# Motivation

- BRDF
  - Central Role in CG
  - Material/Reflectance behavior
  - 4D Function



# Motivation

- BRDF
  - Central Role in CG
  - Material/Reflectance behavior
  - 4D Function
- BRDF Models
  - Analytical Models



$$\rho(\boldsymbol{v}, \boldsymbol{l}) = k_d + k_s \left(\boldsymbol{n} \cdot \boldsymbol{h}\right)^e$$

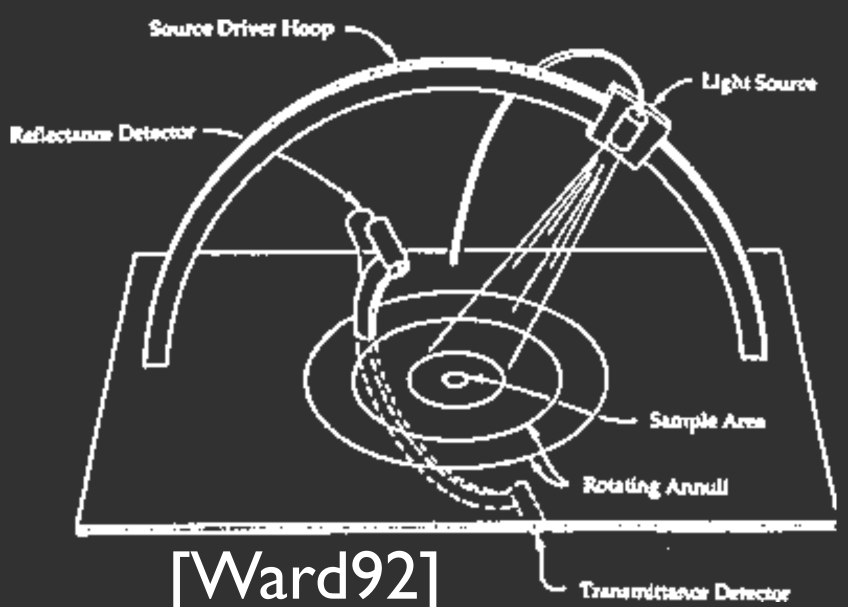
[Blinn]

$$\rho(\boldsymbol{v}, \boldsymbol{l}) = \frac{D(\boldsymbol{h}) F(\boldsymbol{l}) G(\boldsymbol{v}, \boldsymbol{l})}{4 \left(\boldsymbol{n} \cdot \boldsymbol{l}\right) \left(\boldsymbol{n} \cdot \boldsymbol{v}\right)}$$

[Torrance-Sparrow]

# Motivation

- BRDF
  - Central Role in CG
  - Material/Reflectance behavior
  - 4D Function




[Ward92]

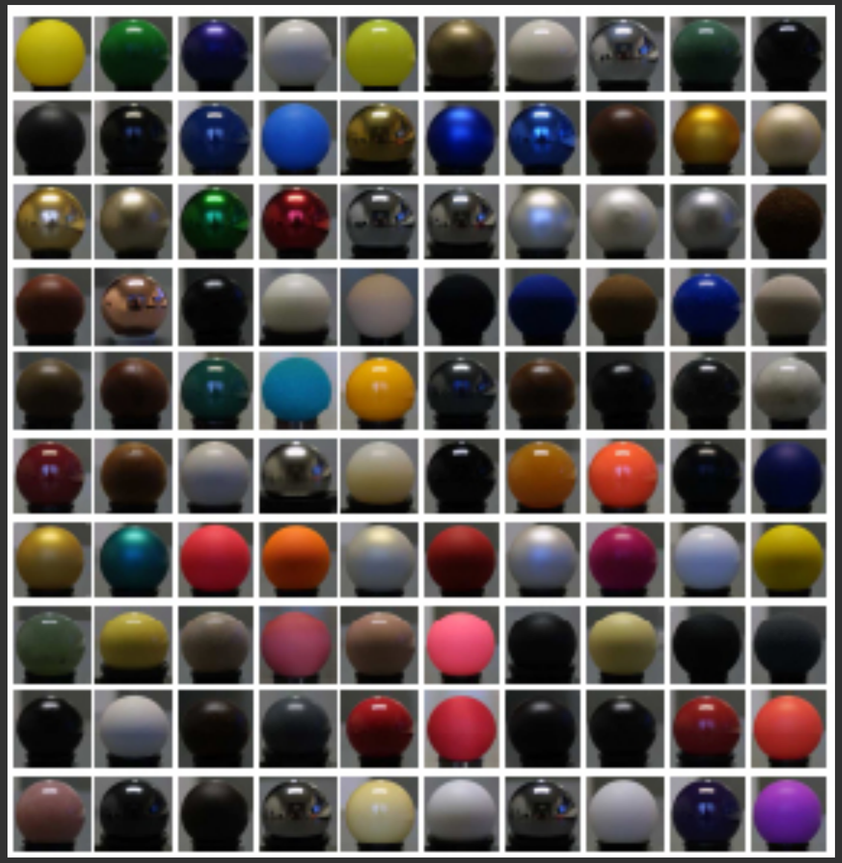
- BRDF Models
  - Analytical Models
  - Data driven
    - Gonioreflectometer
      - [Ward92][LFTW05]
    - CCD
      - [Matusik03] [Ngan05]

[Matusik03]

# Motivation : Handling measured BRDF

Challenges for BRDF Representations

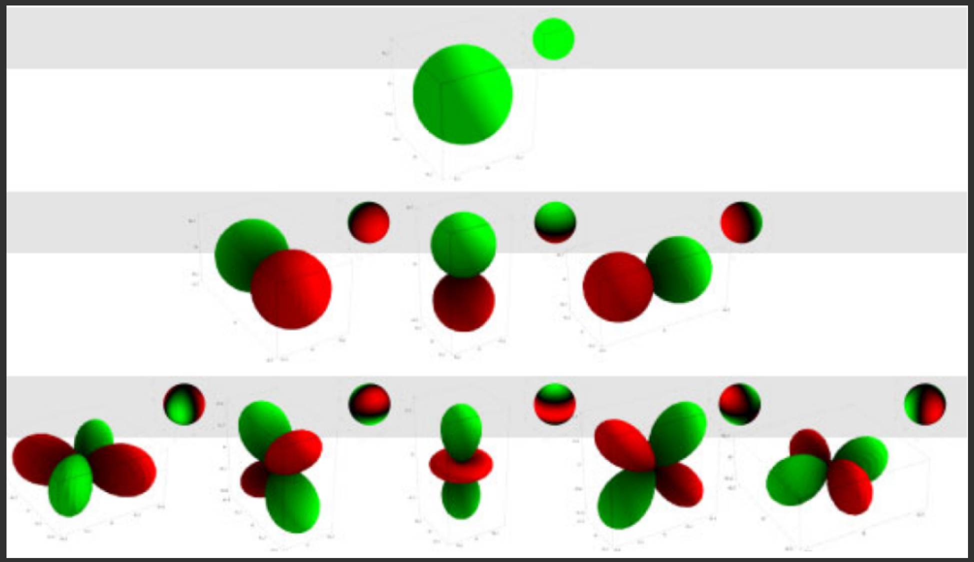
- Low Memory cost
  - BRDF in Merl-MIT: 33MB
- Handle all types of measured materials
  - From Lambertian to mirror
- Efficient evaluation for rendering
- Importance sampling friendly
  - Global Illumination. Monte-Carlo...



Merl-MIT database

# Previous Work: basis functions

- Spherical Harmonics [Cabral87,Westin92,...]
- Zernike Polynomials [Koenderink96]
- sRBF [Zickler05,...]
- Spherical Wavelets [Schröeder95]



- ⇔ Linear decomposition
  - ✓ Easy fitting (projection into the basis)
  - ✓ Fast evaluation
  - ✗ Memory cost increases with specularity
    - ⇔ Quadratic cost [Mahajan08]

# Previous Work: analytical models

- Empirical (ad-hoc): e.g, [Phong75]
- Physics-based Models: e.g., [Ward92]

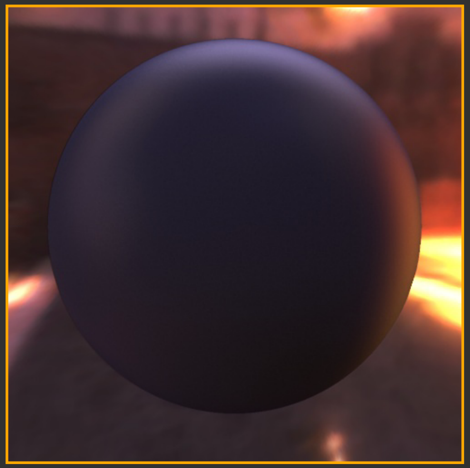
✓ Low number of coefficients

✓ High specularity well handled

✗ Limited representations [Ngan05]

✗ Non-linear fitting technique [*Levenberg-Marquardt,SQP*]

- Numerically unstable
  - With >= 3 lobes
- No guaranty on global convergence



Lafortune

# Previous Work: BRDF importance sampling

- Analytical BRDF Models [Phong, Blinn, Lafortune, Ashikhmin,..]
  - Closed-form importance sampling function
  - ✓ Low memory consumption
  - ✗ Efficiency decreases with grazing angle
    - cosine factor (except. [Kurt 2010])

- Tabulated Data Approaches (e.g., [Lawrence 2004])
  - ✓ Take into account BRDF and cosine factor
  - ✗ Memory cost for high specular materials

# Contributions

## Framework

- Rational Functions as representations for
  - BRDF
  - inverse CDF (for BRDF importance sampling)
  - Low memory cost
- Approximation technique
  - *A priori* error control
  - Global convergence guaranteed
- New Estimator for Monte-Carlo Importance Sampling
  - No probability density function (pdf) storage needed

# Rational Functions Framework

- Rational Functions as representations for
  - BRDF
  - Inverse CDF (for BRDF importance sampling)

- Approximation technique
  - *A priori* error control
  - Global convergence guaranteed
  - Correct Inverse CDF guaranteed

- New Estimator for Monte-Carlo Importance Sampling
  - No probability density function (pdf) storage needed

# Rational Functions

$$r_{n,m}(\boldsymbol{x}) = \frac{p_n(\boldsymbol{x})}{q_m(\boldsymbol{x})} = \frac{\sum_{j=0}^{n} p_j\, b_j(\boldsymbol{x})}{\sum_{k=0}^{m} q_k\, b_k(\boldsymbol{x})}$$

$n\,,\, m$ number of coefficients

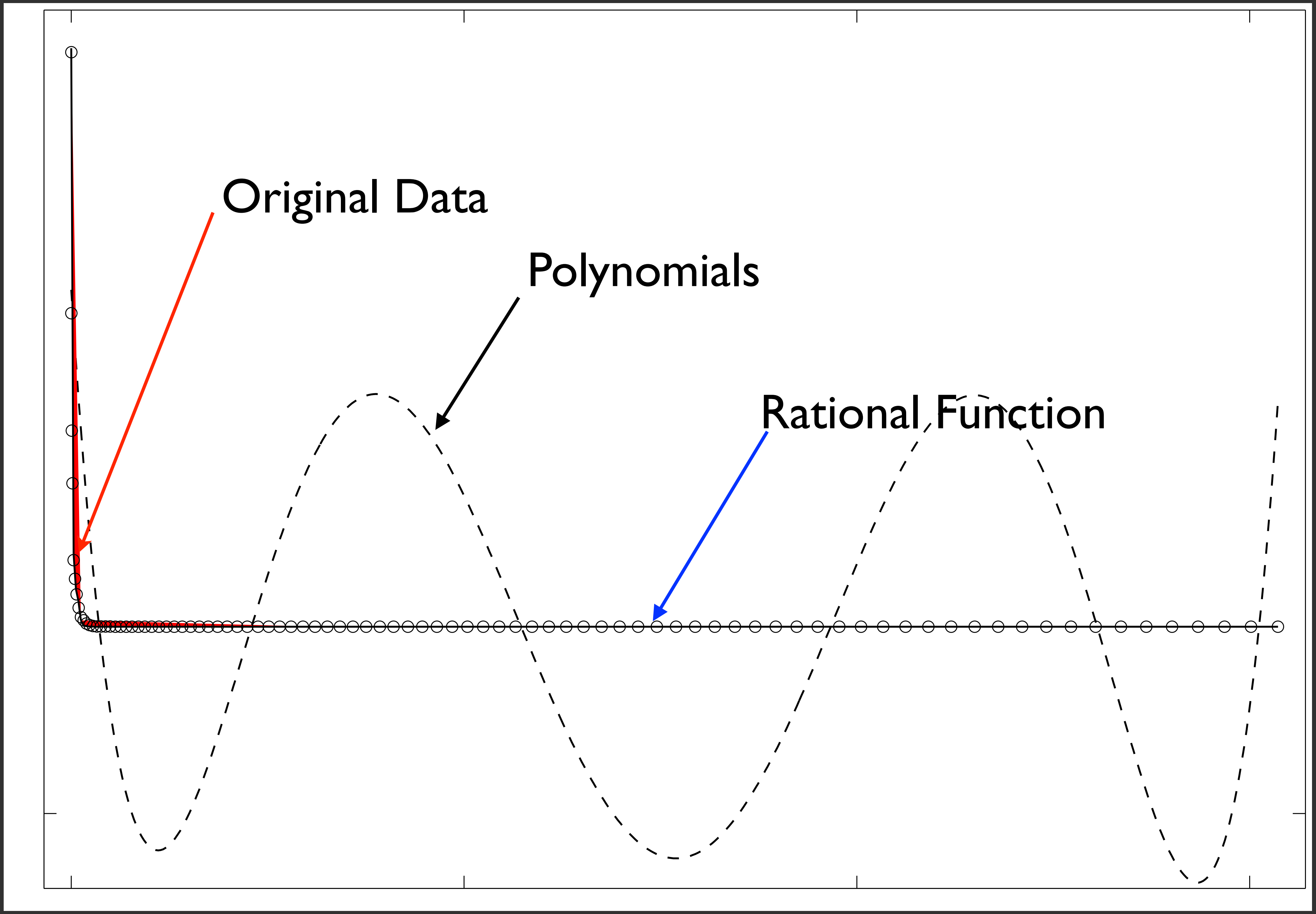
$p_j\,, p_k$ coefficients

$b_j\,,\, b_k$ basis functions

- Widely used in approximation theory
  - Related to Schlick BRDF Model [EG94]
- More powerful than polynomials
  - ideal for steep changes (e.g., BRDF specular lobes)

# Rational Functions

Approximation Comparisons with 7 coefficients



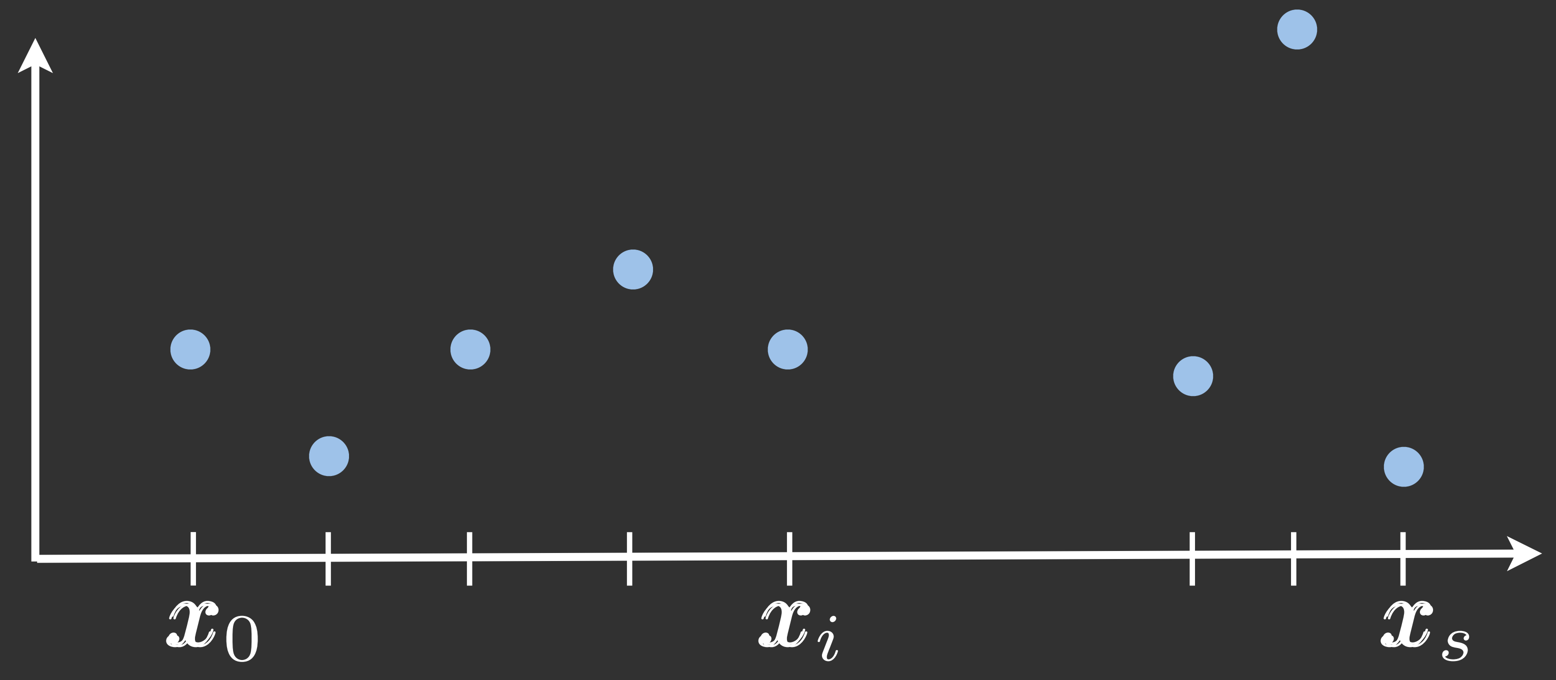
# Rational Functions Framework

- Rational Functions as representations for
  - BRDF
  - Inverse CDF (for BRDF importance sampling)

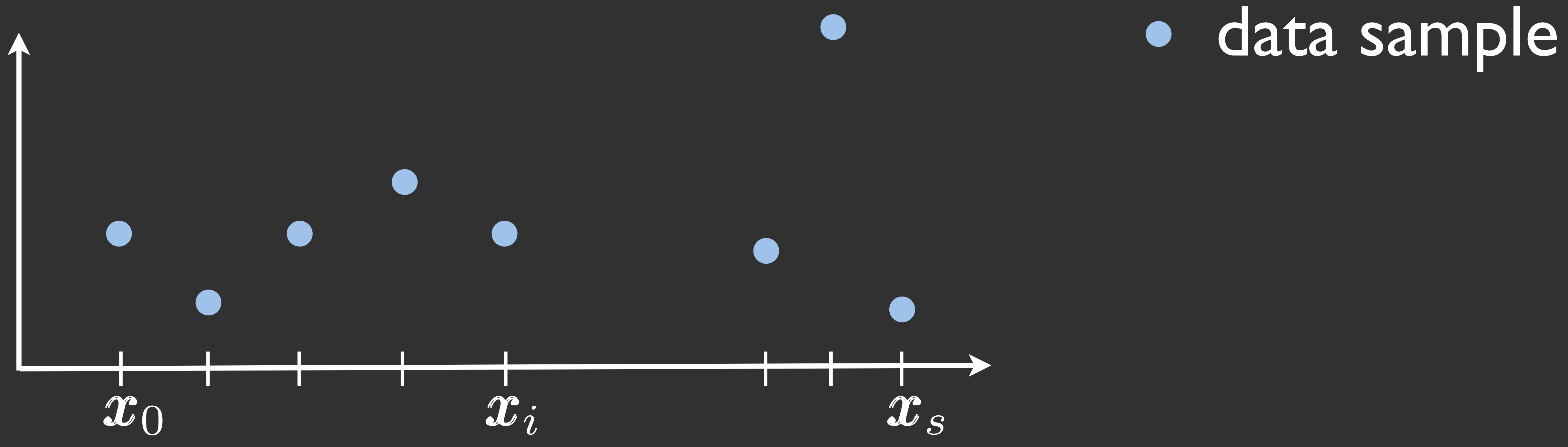
- Approximation technique
  - *A priori* error control
  - Global convergence guaranteed

- New Estimator for Monte-Carlo Importance Sampling
  - No probability density function (pdf) storage needed

# Fitting data: problem statement



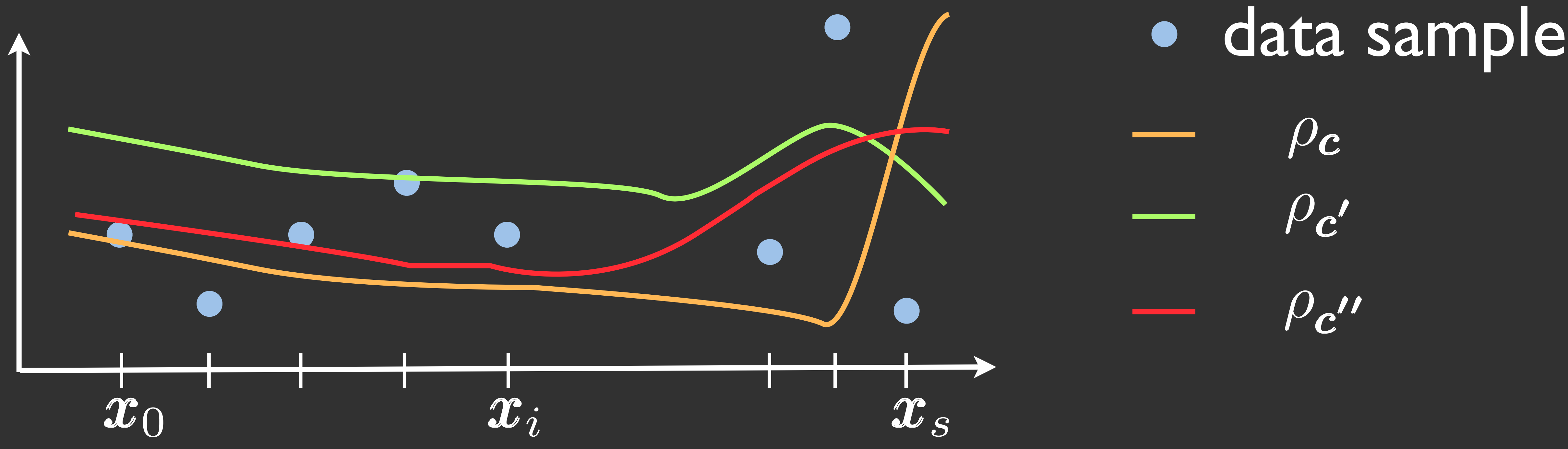
# Fitting data: problem statement




- Choose a BRDF model ρ(v,l) and optimize its parameters :

$$\min_{\boldsymbol{c}} ||\rho_{\boldsymbol{c}}(\boldsymbol{v},\boldsymbol{l}) - data(\boldsymbol{v},\boldsymbol{l})||^2 \qquad \rho_{\boldsymbol{c}} = c_0 + c_1(\boldsymbol{n}\cdot\boldsymbol{h})^{c_2}$$

# Fitting data: problem statement

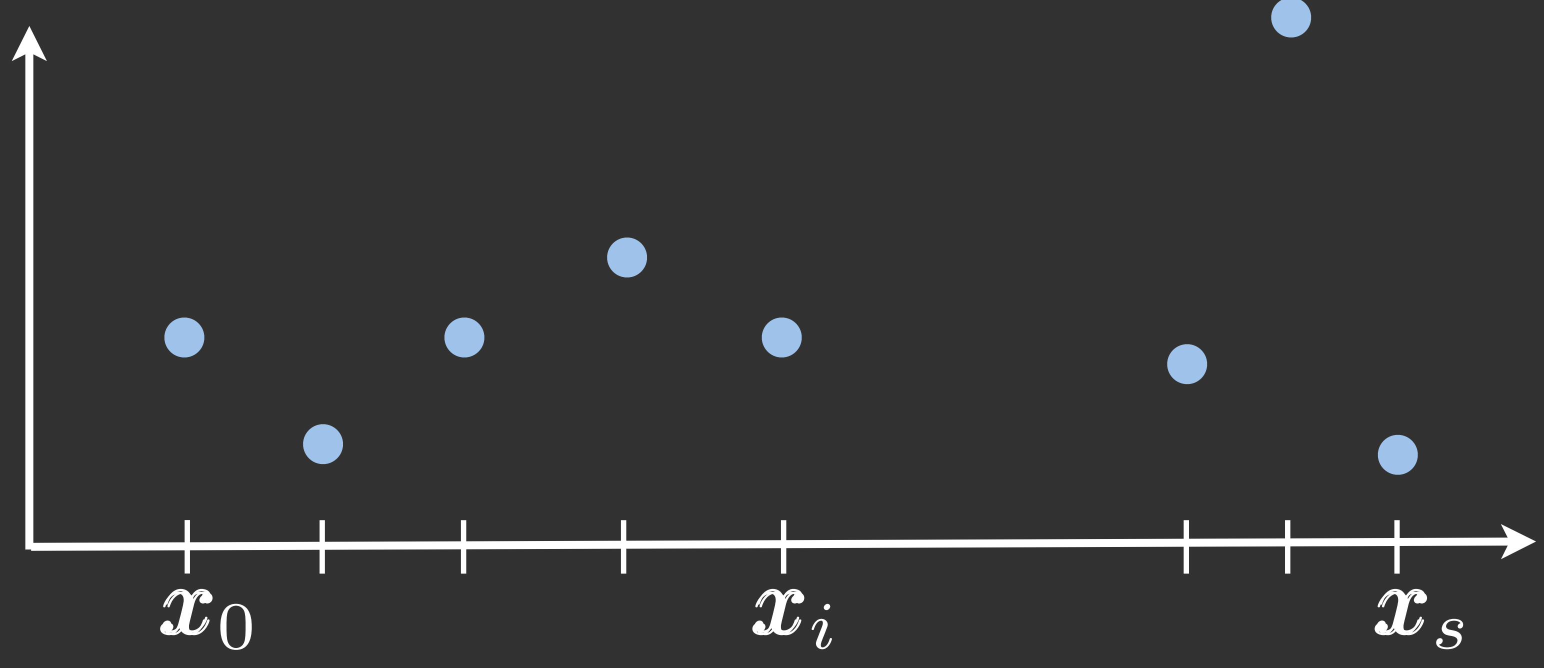



- Choose a BRDF model ρ(v,l) and optimize its parameters :

$$\min_{\boldsymbol{c}} ||\rho_{\boldsymbol{c}}(\boldsymbol{v},\boldsymbol{l}) - data(\boldsymbol{v},\boldsymbol{l})||^2 \qquad \rho_{\boldsymbol{c}} = c_0 + c_1(\boldsymbol{n}\cdot\boldsymbol{h})^{c_2}$$

- ✗ Local convergence
- ✗ No error control

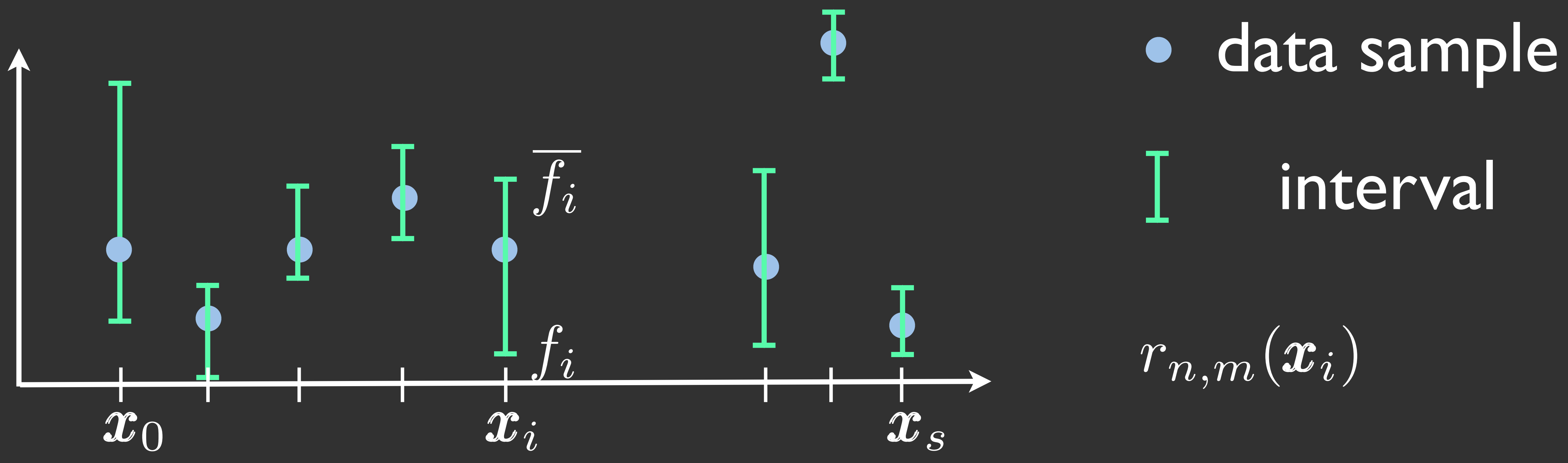
# Fitting data: Rational Function Approach



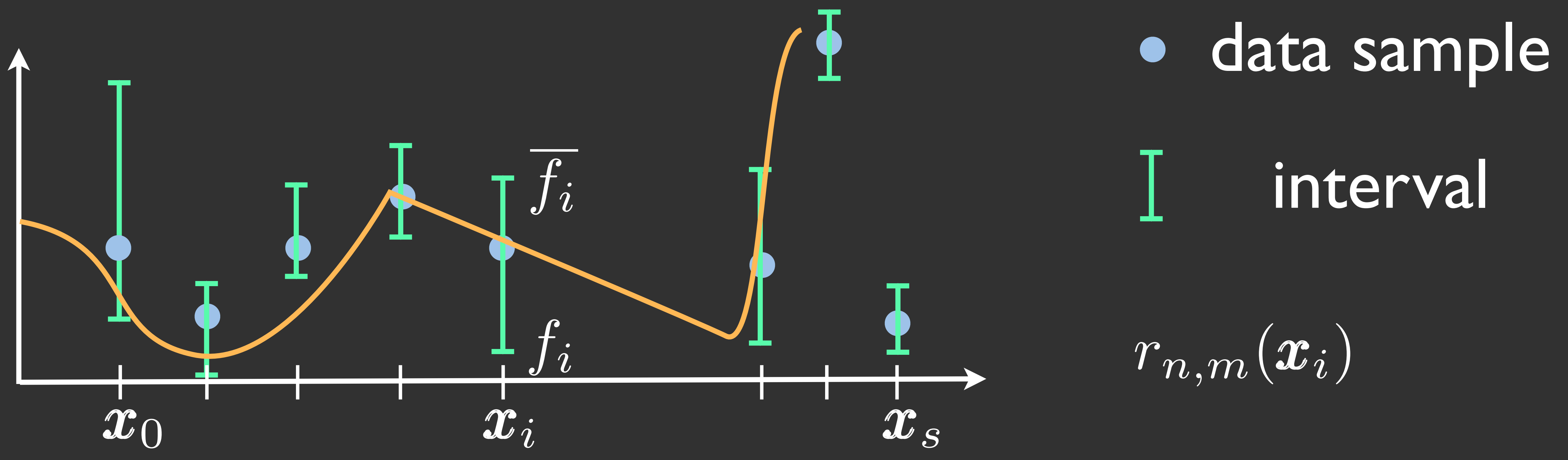

- data sample

$r_{n,m}(\boldsymbol{x}_i)$

# Fitting data: Rational Function Approach



# Fitting data: Rational Function Approach




- Find the Rational Function such that:

$$\forall i = 0, \ldots, s \qquad \underline{f_i} \leqslant r_{n,m} = \frac{p_n(\boldsymbol{x}_i)}{q_m(\boldsymbol{x}_i)} \leqslant \overline{f_i}$$

$$\text{with} \quad n + m \ll s$$

$$\text{and} \quad q_m(\boldsymbol{x}_i) > 0$$

# Overview of our fitting algorithm

1. Choose $C_{max}$=(n+m) coefficients for the Rational Function
2. Set intervals' size $F_i$ = $|\underline{f_i} - f_i|$
3. S = fitRationalFunction( $C_{max}$, $F_i$, data)
4. while S == ∅
   4.1. Increase intervals' size $F_i$
   4.2. S = fitRationalFunction( $C_{max}$, $F_i$, data)
5. Return S

# Overview of our fitting algorithm

1. Choose $C_{max}$=(n+m) coefficients for the Rational Function
2. Set intervals' size $F_i$ = $\left|\underline{f_i} - f_i\right|$
3. S = fitRationalFunction( $C_{max}$, $F_i$, data)
4. while S == ∅
   
   4.1. Increase intervals' size $F_i$
   
   4.2. S = fitRationalFunction( $C_{max}$, $F_i$, data)
5. Return S

# Fitting a Rational Function

Based on the algorithm from [Celis2007]

- Quadratic Problem *P(n,m)*
  - Unique solution
  - Pole-free methods

    $$\forall x_i, \quad q_m(\boldsymbol{x_i}) > 0$$

  - Global Convergence
    - Convex problem
  - Size(A) $\propto$ (n+m) X (2*s+2)

$$\mathscr{P}(n,m):$$

$$\arg\min_{\boldsymbol{c}\in\mathbb{R}^{n+m+2}} |\boldsymbol{c} = (p_0,\ldots,p_n,q_0,\ldots,q_m)^t|_2$$

subject to

$$\mathbf{A}^{(j)}_{n,m}\boldsymbol{c} - \delta|\mathbf{A}^{(j)}_{n,m}|_2 \geqslant 0, \qquad j = 1,\ldots,2s+2$$

$$\mathbf{A}_{n,m} =$$

$$\begin{pmatrix} b_0(\boldsymbol{x}_0) & \ldots & b_n(\boldsymbol{x}_0) & -\underline{f_0}\,b_0(\boldsymbol{x}_0) & \ldots & -\underline{f_0}\,b_m(\boldsymbol{x}_0) \\ \vdots & & \vdots & \vdots & & \vdots \\ b_0(\boldsymbol{x}_s) & \ldots & b_n(\boldsymbol{x}_s) & -\underline{f_s}\,b_0(\boldsymbol{x}_s) & \ldots & -\underline{f_s}\,b_m(\boldsymbol{x}_s) \\ -b_0(\boldsymbol{x}_0) & \ldots & -b_n(\boldsymbol{x}_0) & \overline{f_0}\,b_0(\boldsymbol{x}_0) & \ldots & \overline{f_0}\,b_m(\boldsymbol{x}_0) \\ \vdots & & \vdots & \vdots & & \vdots \\ -b_0(\boldsymbol{x}_s) & \ldots & -b_n(\boldsymbol{x}_s) & \overline{f_s}\,b_0(\boldsymbol{x}_s) & \ldots & \overline{f_s}\,b_m(\boldsymbol{x}_s) \end{pmatrix}$$

# Fitting a Rational Function

Our Algorithm to find a RF

- For (n+m) coefficients
- Test all possible combinations for numerator and denominator

e.g., n+m=6

(1,5) (2,4) (3,3) (4,2) (5,1)

- If multiple solutions exists
  - Keep the "most" stable

⇔ Lowest condition number of A

$$\mathscr{P}(n,m):$$

$$\arg \min_{\boldsymbol{c} \in \mathbb{R}^{n+m+2}} |\boldsymbol{c} = (p_0, \dots, p_n, q_0, \dots, q_m)^t|_2$$

subject to

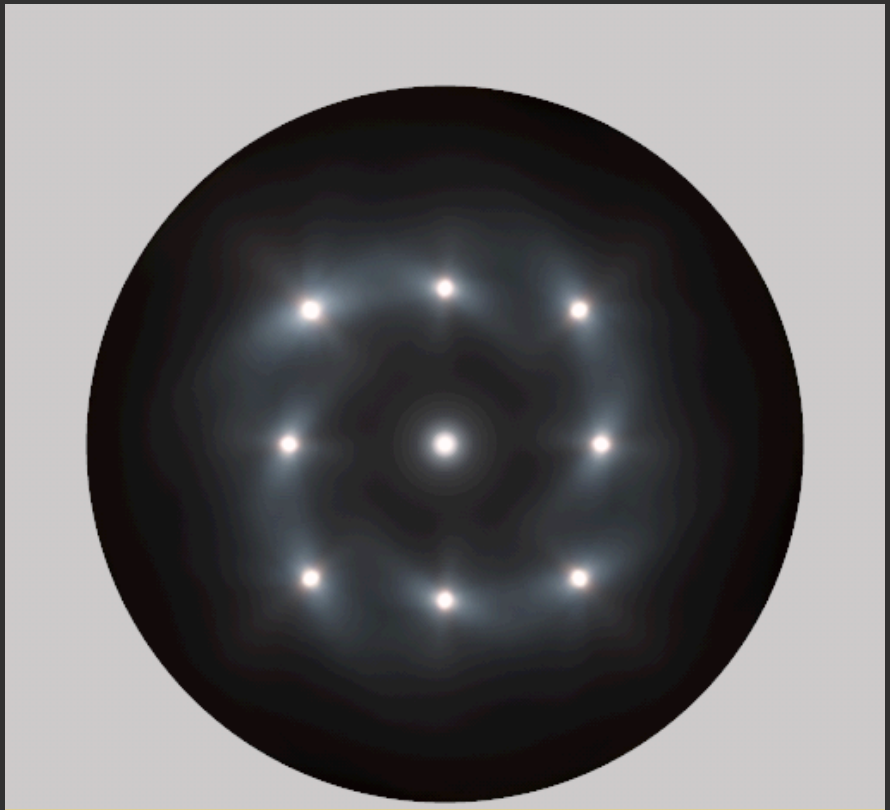
$$\mathbf{A}_{n,m}^{(j)} \boldsymbol{c} - \delta |\mathbf{A}_{n,m}^{(j)}|_2 \geqslant 0, \qquad j = 1, \dots, 2s+2$$

$$\mathbf{A}_{n,m} = \begin{pmatrix} b_0(\boldsymbol{x}_0) & \dots & b_n(\boldsymbol{x}_0) & -\underline{f_0}\, b_0(\boldsymbol{x}_0) & \dots & -\underline{f_0}\, b_m(\boldsymbol{x}_0) \\ \vdots & & \vdots & \vdots & & \vdots \\ b_0(\boldsymbol{x}_s) & \dots & b_n(\boldsymbol{x}_s) & -\underline{f_s}\, b_0(\boldsymbol{x}_s) & \dots & -\underline{f_s}\, b_m(\boldsymbol{x}_s) \\ -b_0(\boldsymbol{x}_0) & \dots & -b_n(\boldsymbol{x}_0) & \overline{f_0}\, b_0(\boldsymbol{x}_0) & \dots & \overline{f_0}\, b_m(\boldsymbol{x}_0) \\ \vdots & & \vdots & \vdots & & \vdots \\ -b_0(\boldsymbol{x}_s) & \dots & -b_n(\boldsymbol{x}_s) & \overline{f_s}\, b_0(\boldsymbol{x}_s) & \dots & \overline{f_s}\, b_m(\boldsymbol{x}_s) \end{pmatrix}$$

# Results: BRDF fitting

- Tested on the MERL-MIT data base
  - Data noisy
    - at grazing angle
    - at center of the specular lobe
  - Isotropic BRDF
  - 3D parametrization [Rusinkiewicz 1997]

$$\rho(\boldsymbol{v}, \boldsymbol{l}) = \rho(\theta_h, \theta_d, \phi_d)$$


3D Data

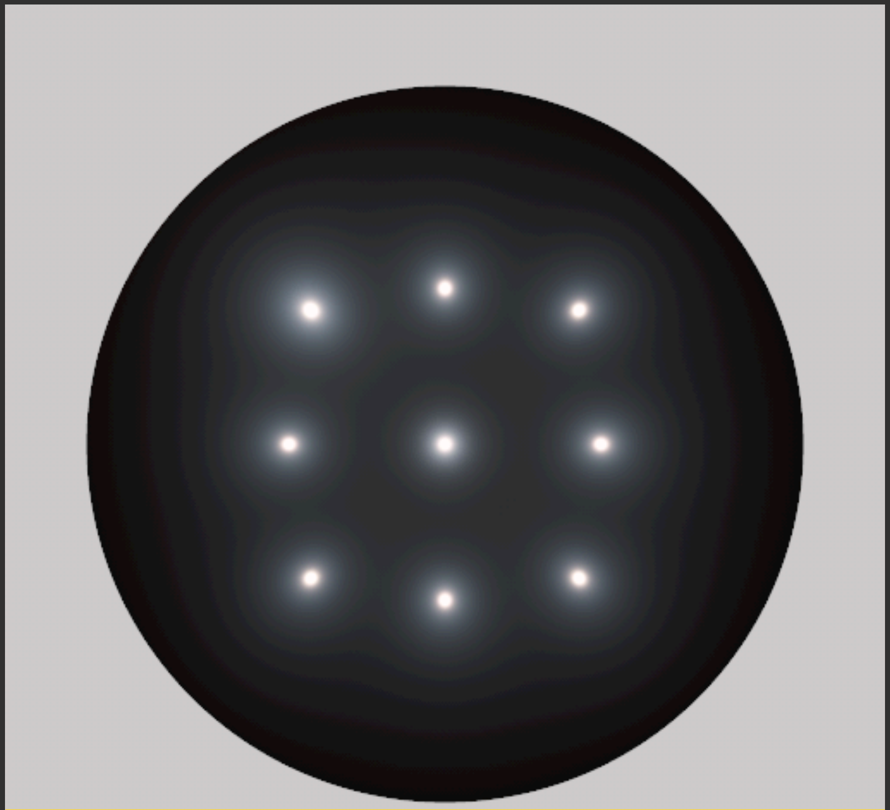
- BRDF are approximated as 2D Rational Functions
  - 2D parametrization [Romeiro 2007]

$$\rho(\boldsymbol{v}, \boldsymbol{l}) = \rho(\theta_h, \theta_d, \phi_d) \approx \rho(\theta_h, \theta_d)$$

# Results: BRDF fitting

- Tested on the MERL-MIT data base
  - Data noisy
    - at grazing angle
    - at center of the specular lobe
  - Isotropic BRDF
  - 3D parametrization [Rusinkiewicz 1997]

$$\rho(\boldsymbol{v}, \boldsymbol{l}) = \rho(\theta_h, \theta_d, \phi_d)$$



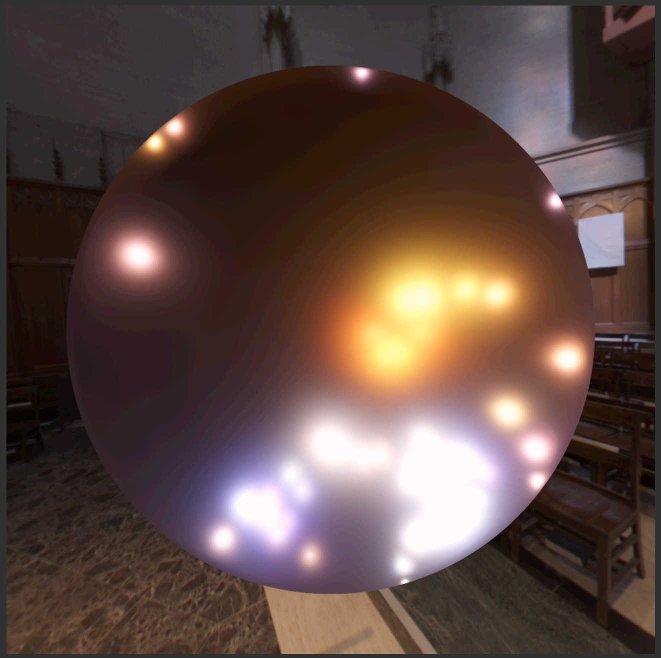
2D Data

- BRDF are approximated as 2D Rational Functions
  - 2D parametrization [Romeiro 2007]

$$\rho(\boldsymbol{v}, \boldsymbol{l}) = \rho(\theta_h, \theta_d, \phi_d) \approx \rho(\theta_h, \theta_d)$$

# Results: BRDF fitting

- Nickel Material
- Very specular material

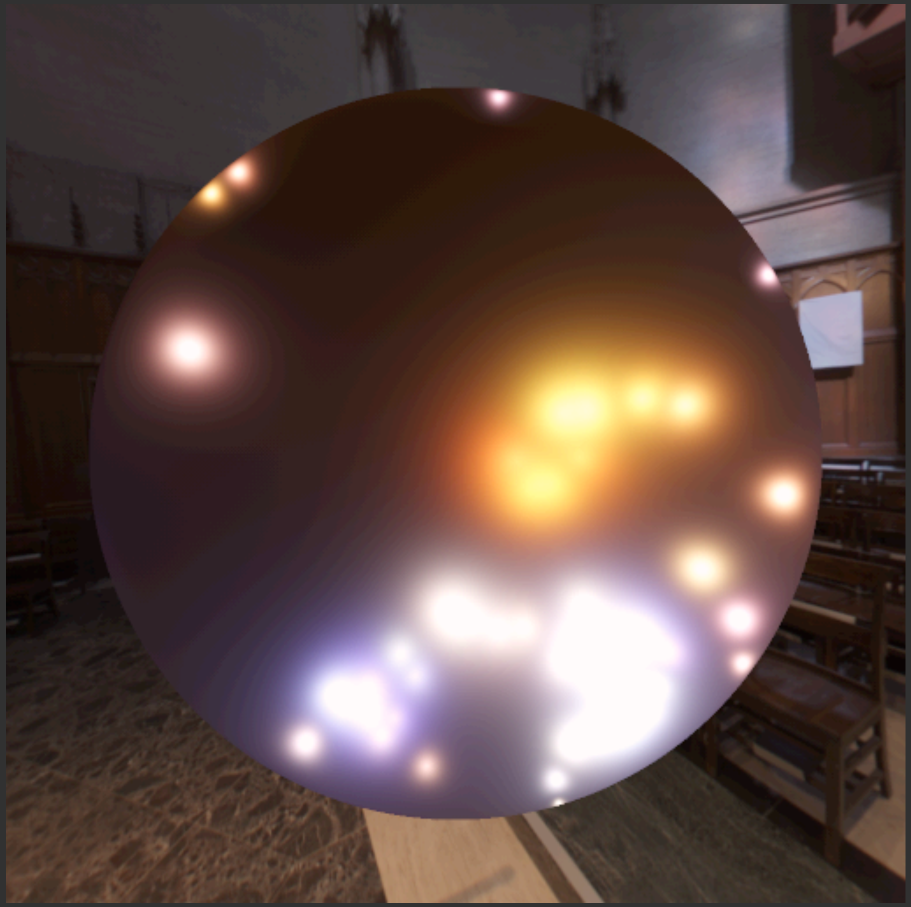


| Maximum Relative Error (Blue Channel) | | |
|---|---|---|
| Rational | Non-linear A&S BRDF | Polynomial |
| 2.45 | 4.11 | 49 399 |
| Memory Footprint | | |
| 0.19KB | 0.0625KB | 0.19KB |

# Results: BRDF fitting

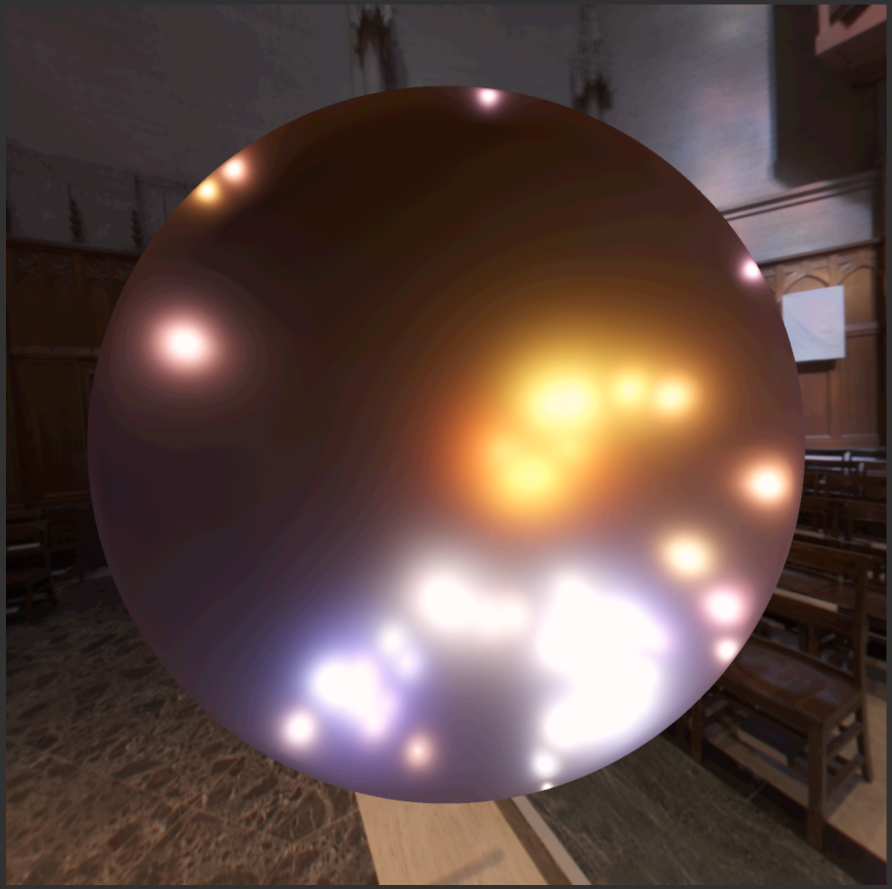
- Nickel Material
- Very specular material

Rational Function

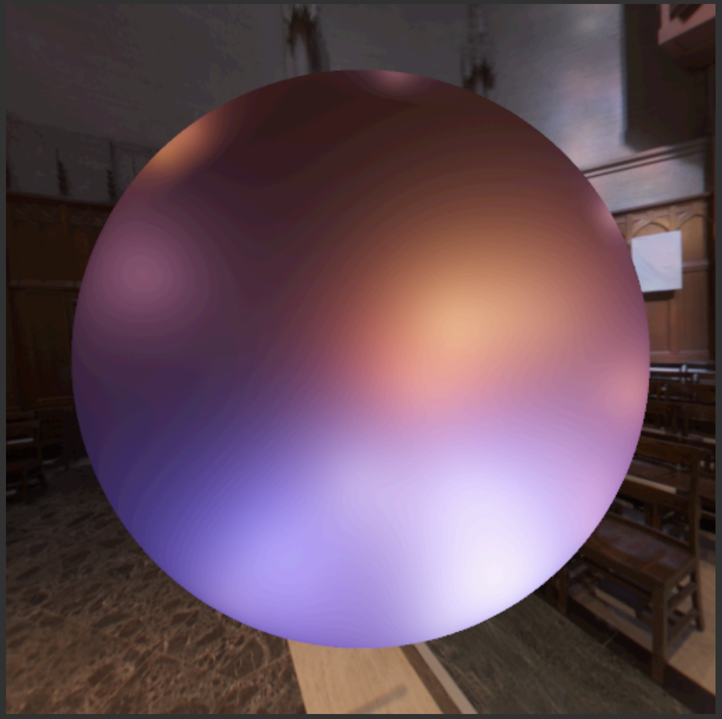


Mean Lab Error: 1.52

Original 3D Data



# Results: BRDF Fitting Time



- Metallic-blue material
- 2D Fit for each color channel
  - Grazing angles removed (> 78 degrees)
  - 6308 data samples
  - Max relative error is 0.2

- Multi-threaded C++ algorithm

- Looking for all possible solutions up to 42 coefficients:
  - (1,41) (2,40) ... on i7-3820@3.60 GHz
- 22 seconds to find 2 solutions with 42 coefficients

# Results: GPU Frame rates

- Direct Implementation in CUDA on GTX 580
- Horner factorization (as done in C++)

$$x^3 + x^2 + x + 1 = ((x+1)x+1)x+1$$

- Image resolution: 1024x768 pixels
- 256 directional light sources
- 1K polygons Mesh

| | blue-metallic | nickel | chrome |
|---|---|---|---|
| #coefficients | (48, 46, 32) | (43, 49, 48) | (101,61, 76) |
| fps | 110 | 70 | 19 |

# Rational Functions Framework

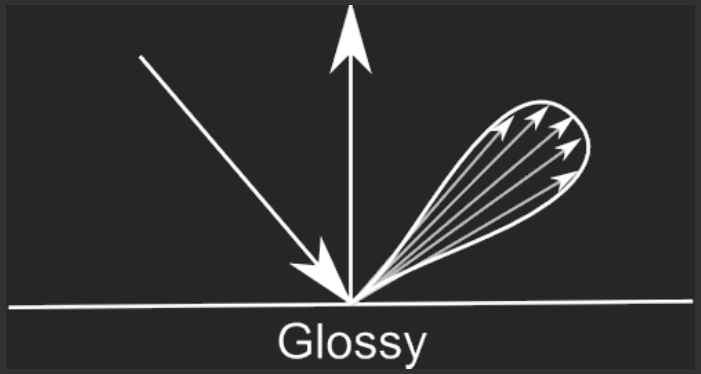
- Rational Functions as representations for
  - BRDF
  - Inverse CDF (for BRDF importance sampling)

- Approximation technique
  - *A priori* error control
  - Global convergence guaranteed

- New Estimator for Monte-Carlo Importance Sampling
  - No probability density function (pdf) storage needed

# Monte-Carlo Importance Sampling

$$L(\boldsymbol{v}) \approx \frac{1}{K} \sum_{k=1}^{K} \frac{\boldsymbol{n} \cdot \boldsymbol{l}_k}{\mathrm{PDF}(\boldsymbol{v}|\boldsymbol{l}_k)} \rho(\boldsymbol{l}_k, \boldsymbol{v}) \; L(\boldsymbol{l}_k)$$




- Generation of samples $\boldsymbol{l}_k$
  - Inverse CDF to generate $\boldsymbol{l}_k$
  - Associated PDF

- PDF proportional to BRDF scaled by cosine factor $\boldsymbol{n} \cdot \boldsymbol{l}_k$
  - Inverse of a Rational Function is hard too find automatically
  - Tabulated approaches work well [Lawrence2004, Montes2008]

# Importance Sampling with Rational Functions

1. Generate tabulated data for the BRDF × cosine
   1. 2D Table:
   2. 3D Table:

$$\begin{cases} \theta_l = \mathrm{CDF}^{-1}(\theta_v, \mu) \\ \phi_l = \mathrm{CDF}^{-1}(\theta_v, \tau | \theta_l) \end{cases}$$

2. Approximate inverse CDFs with Rational Functions

$$\begin{cases} \theta_l = \mathrm{CDF}^{-1}(\theta_v, \mu) \approx r_{n_\theta, m_\theta}(\theta_v, \mu) \\ \phi_l = \mathrm{CDF}^{-1}(\theta_v, \tau | \theta_l) \approx r_{n_\phi, m_\phi}(\theta_v, \theta_l, \tau) \end{cases}$$

3. Deduce the PDF from the inverse CDFs

# New Monte-Carlo Estimator

$$L(\boldsymbol{v}) \approx \frac{1}{K} \sum_{k=1}^{K} \frac{\boldsymbol{n} \cdot \boldsymbol{l}_k}{\mathrm{PDF}(\boldsymbol{v}|\boldsymbol{l}_k)} \rho(\boldsymbol{l}_k, \boldsymbol{v}) \; L(\boldsymbol{l}_k)$$

- Goal: avoid fitting and storing the PDF
- Derivative of the inverse CDF
  - Equal to 1.0/PDF
  - Evaluated on the fly during rendering

$$L(\boldsymbol{v}) \approx \frac{1}{K} \sum_{k=1}^{K} \alpha_{\boldsymbol{v}}(\mu_k, \tau_k) \; \rho(\boldsymbol{v}, \boldsymbol{l}_k) \; (\boldsymbol{n} \cdot \boldsymbol{l}_k) \; L(\boldsymbol{l}_k)$$

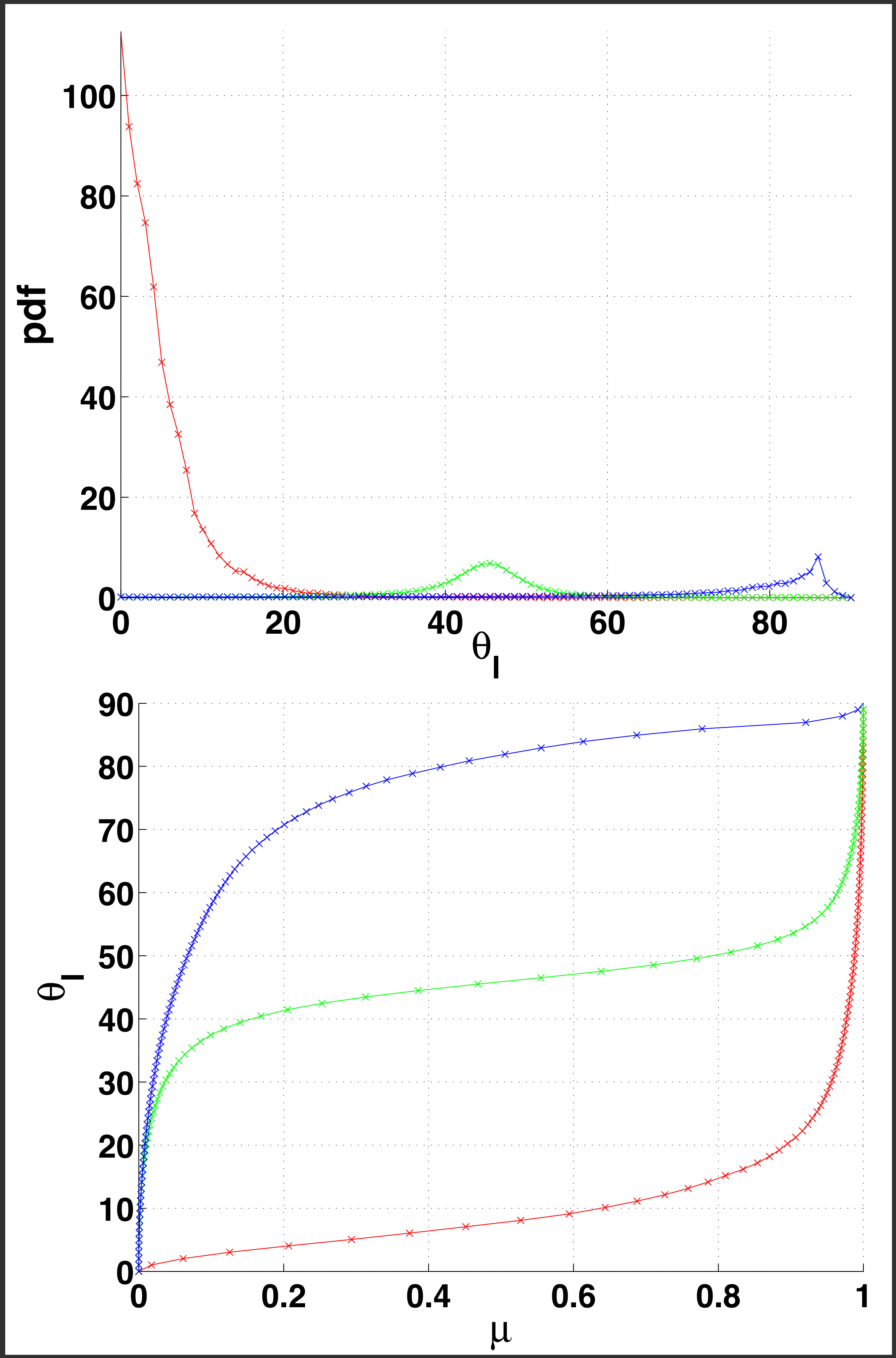
$$\text{with } \alpha_{\boldsymbol{v}}(\mu, \tau) = \frac{\partial \mathrm{CDF}_{\boldsymbol{v}}^{-1}}{\partial \mu}(\mu) \, \frac{\partial \mathrm{CDF}_{\boldsymbol{v}}^{-1}}{\partial \tau}(\tau \,|\, \theta_l) \; \sin \theta_l$$

# (Inverse) CDFs fitting

CDF is

- Normalized Integral of the PDF
  - ➪ simpler than the PDF

- Algorithm should return a CDF
  - ➪ Quadratic Problem modified:
    - Boundary constraints
    - Monotonicity constraints
    - Symmetry constraints



# Boundary Constraints for CDF fitting

For 2D inverse CDF

$$\mathrm{CDF}^{-1}(\theta_v,\ \mu\!=\!0) = 0$$

$$\mathrm{CDF}^{-1}(\theta_v,\ \mu\!=\!1) = \frac{\pi}{2}$$

For 3D inverse CDF

$$\mathrm{CDF}^{-1}(\theta_v\!=\!0,\ \theta_l,\tau) = \pi\tau$$

$$\mathrm{CDF}^{-1}(\theta_v,\ \theta_l\!=\!0,\tau) = \pi\tau$$

# Boundary Constraints for CDF fitting

For 2D inverse CDF

$$\mathrm{CDF}^{-1}(\theta_v,\ \mu = 0) = 0$$

$$\mathrm{CDF}^{-1}(\theta_v,\ \mu = 1) = \frac{\pi}{2}$$

For 3D inverse CDF

$$\mathrm{CDF}^{-1}(\theta_v = 0,\ \theta_l, \tau) = \pi\tau$$

$$\mathrm{CDF}^{-1}(\theta_v,\ \theta_l = 0, \tau) = \pi\tau$$

⇒ Embedded in the models:

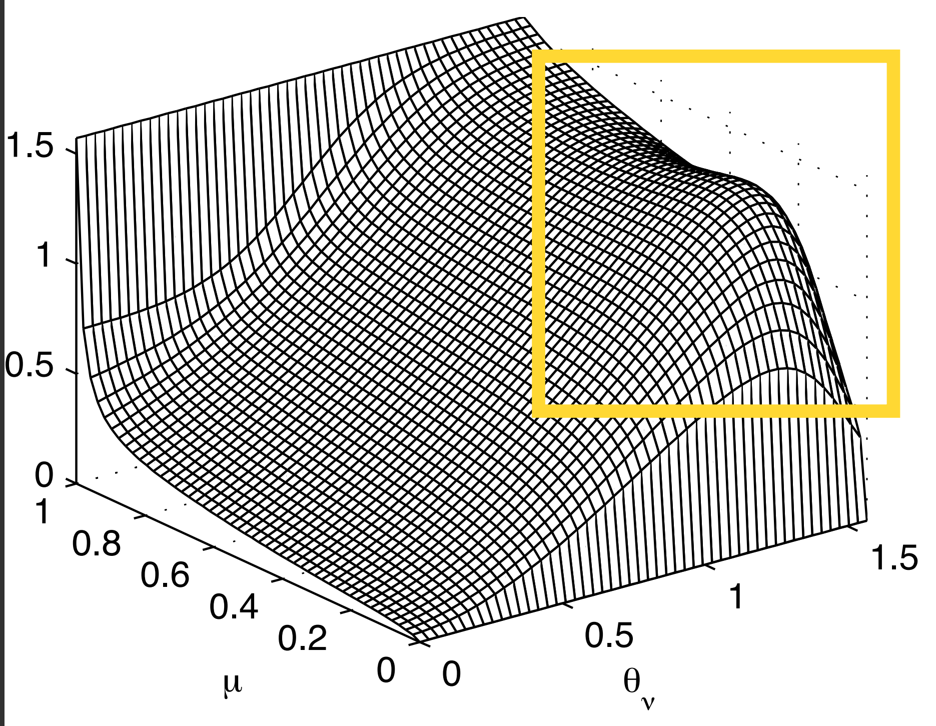
$$r_{n_\theta, m_\theta}(\theta_v, \mu) = \frac{\pi}{2}\mu + \mu(1-\mu)\,\frac{p_{n_\theta}(\theta_v, \mu)}{q_{m_\theta}(\theta_v, \mu)}$$

$$r_{n_\phi, m_\phi}(\theta_v, \theta_l, \tau) = \pi\,\tau + \tau(1-\tau)\,\theta_v\,\theta_l\,\frac{p_{n_\phi}(\theta_v, \theta_l, \tau)}{q_{m_\phi}(\theta_v, \theta_l, \tau)}$$

# Monotonicity Constraints for CDF fitting

- CDFs is monotonically increasing :

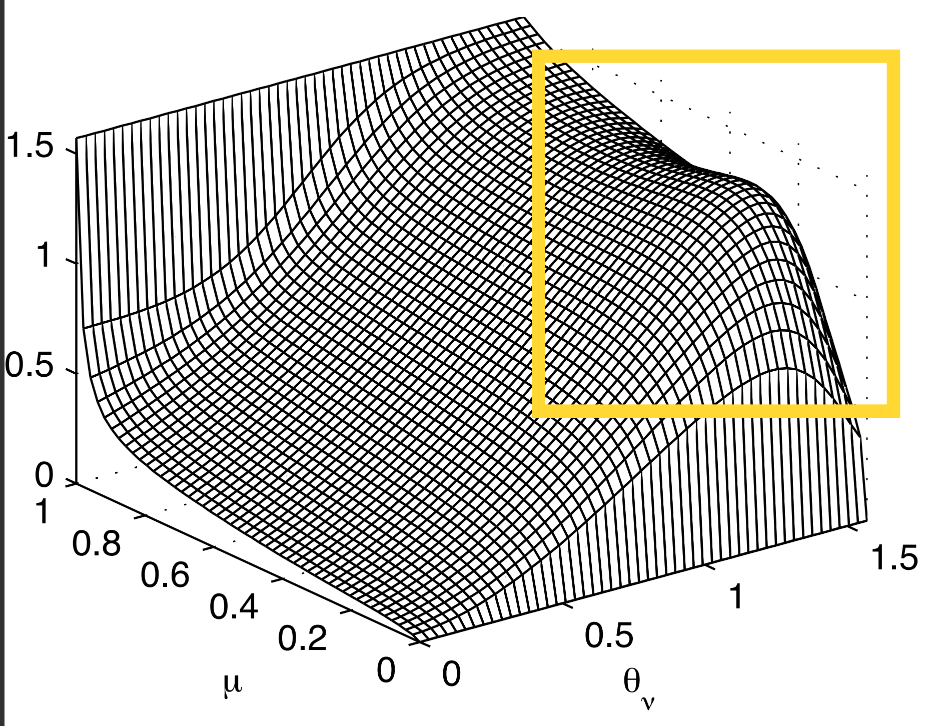
$$\frac{\partial r_{n,m}}{\partial x_j}(\boldsymbol{x}) \geqslant 0$$



# Monotonicity Constraints for CDF fitting

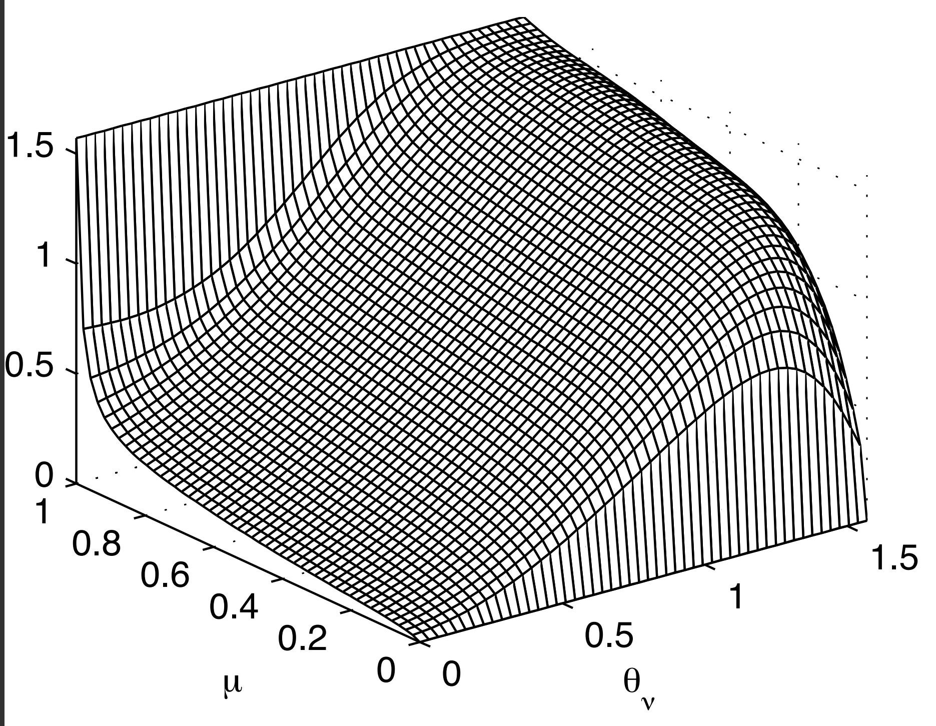
- CDFs is monotonically increasing :

$$\frac{\partial r_{n,m}}{\partial x_j}(\boldsymbol{x}) \geqslant 0$$



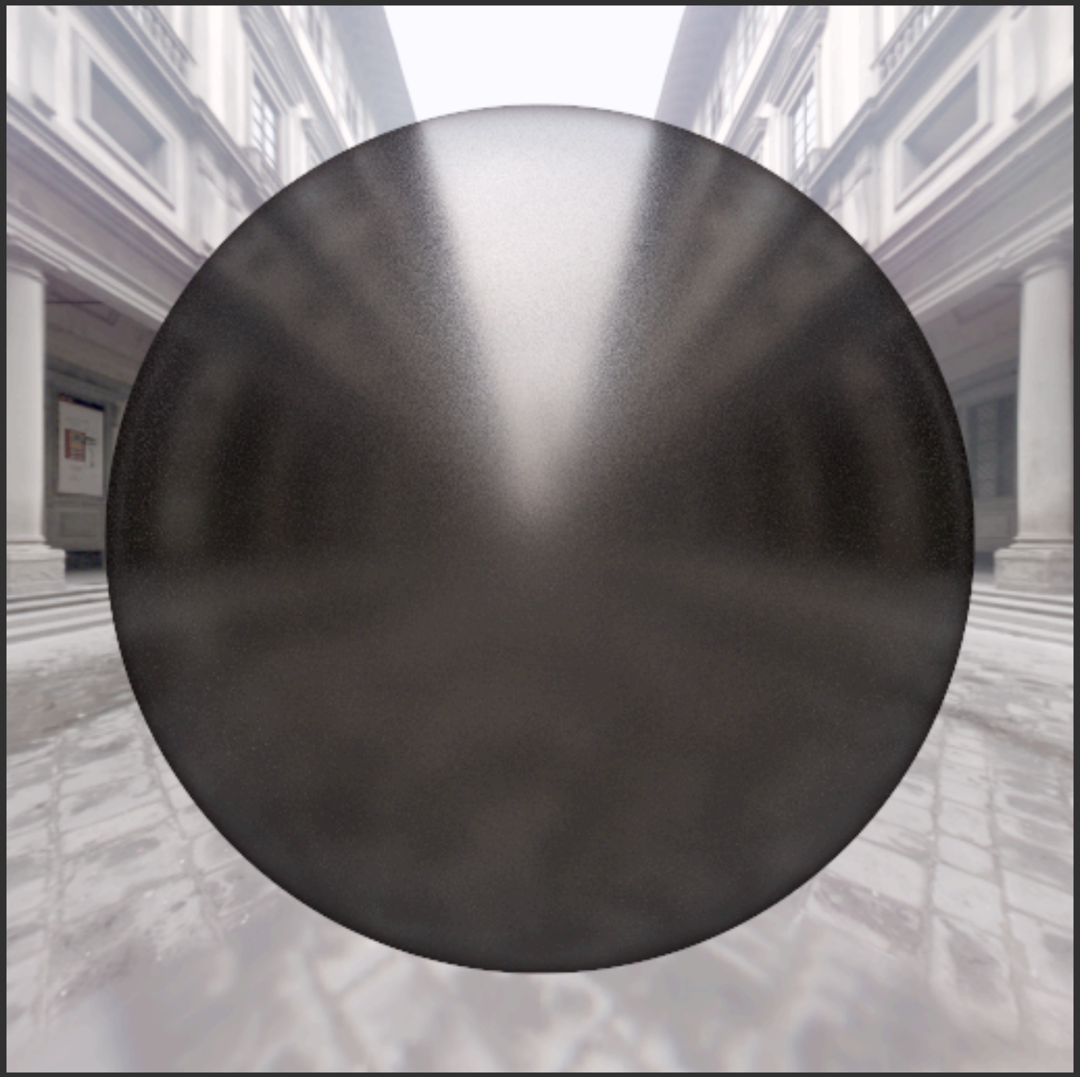

- Adding linear inequalities for each data sample:

$$\begin{cases} \dfrac{\partial p_{n,m}}{\partial x_j}(\boldsymbol{x}_i) \geqslant \dfrac{\partial q_{n,m}}{\partial x_j}(\boldsymbol{x}_i)\, \underline{f_i} \\ \dfrac{\partial p_{n,m}}{\partial x_j}(\boldsymbol{x}_i) \geqslant \overline{f_i}\, \dfrac{\partial q_{n,m}}{\partial x_j}(\boldsymbol{x}_i) \end{cases}$$



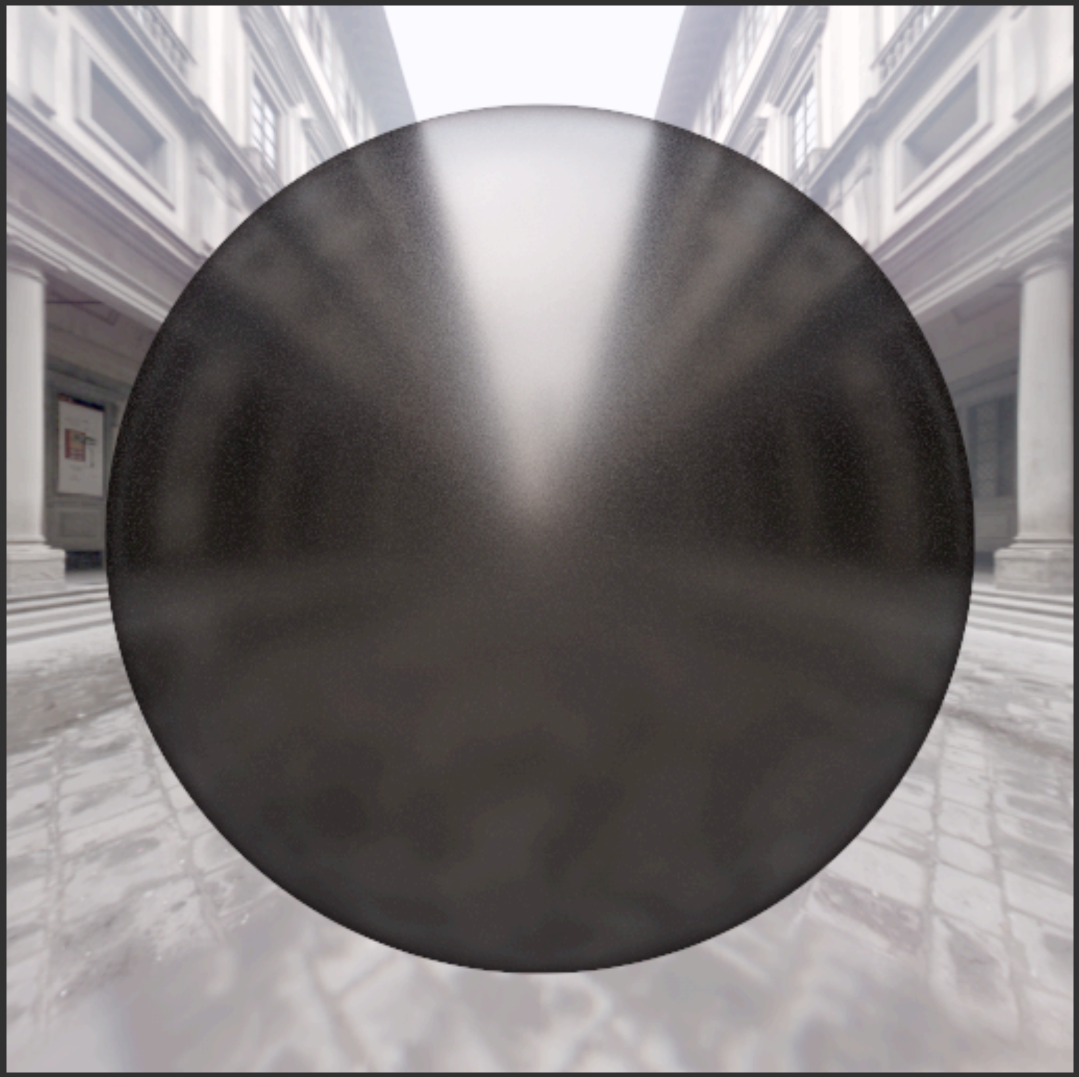
# Results: Nickel inverse CDFs

Rational Functions



Variance
(0.0017,0.0141,0.0156)
Mean Lab Error: 0.44

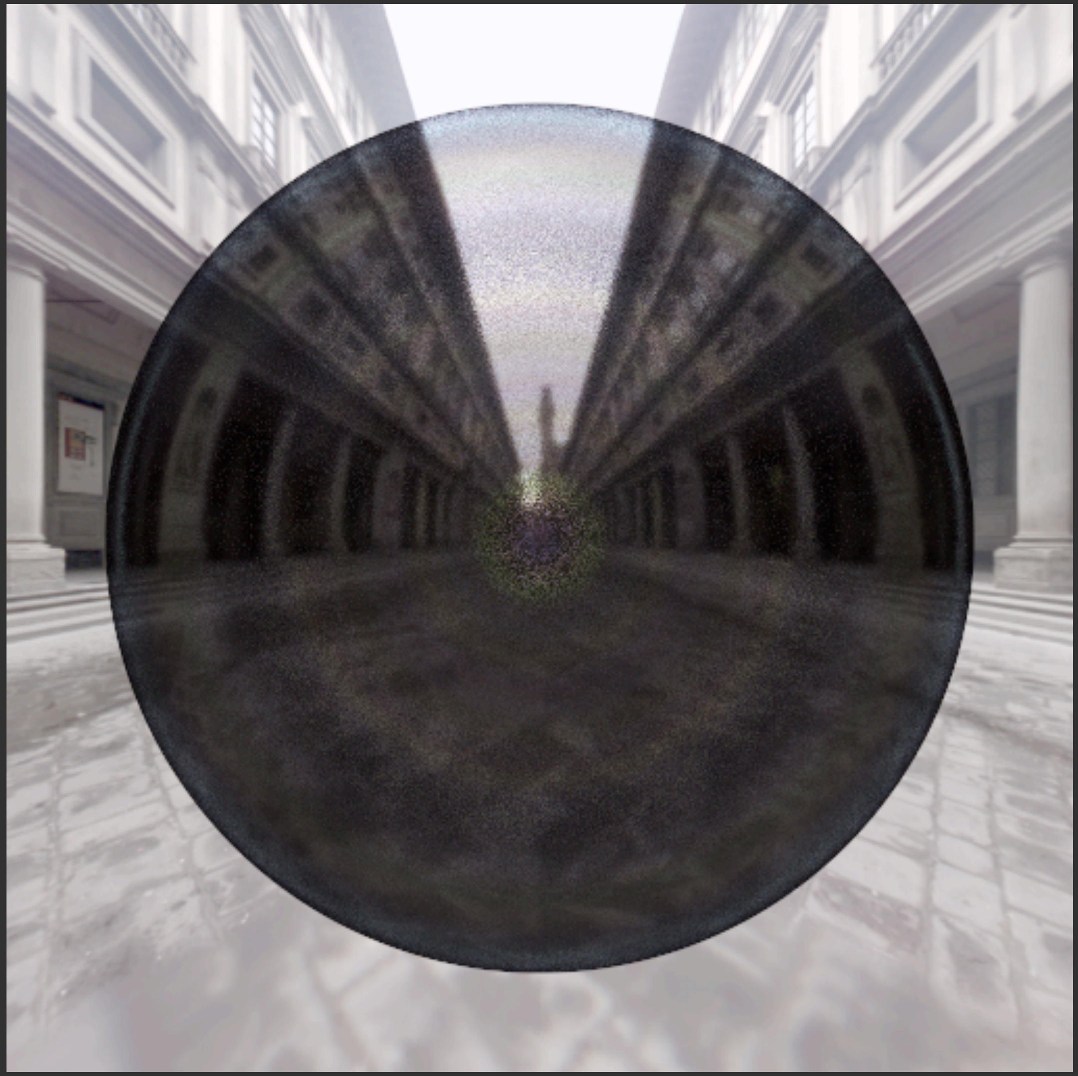
Tabulated Data



Variance
(0.0025,0.0020,0.0022)

# Results: Chrome inverse CDFs

Rational Functions

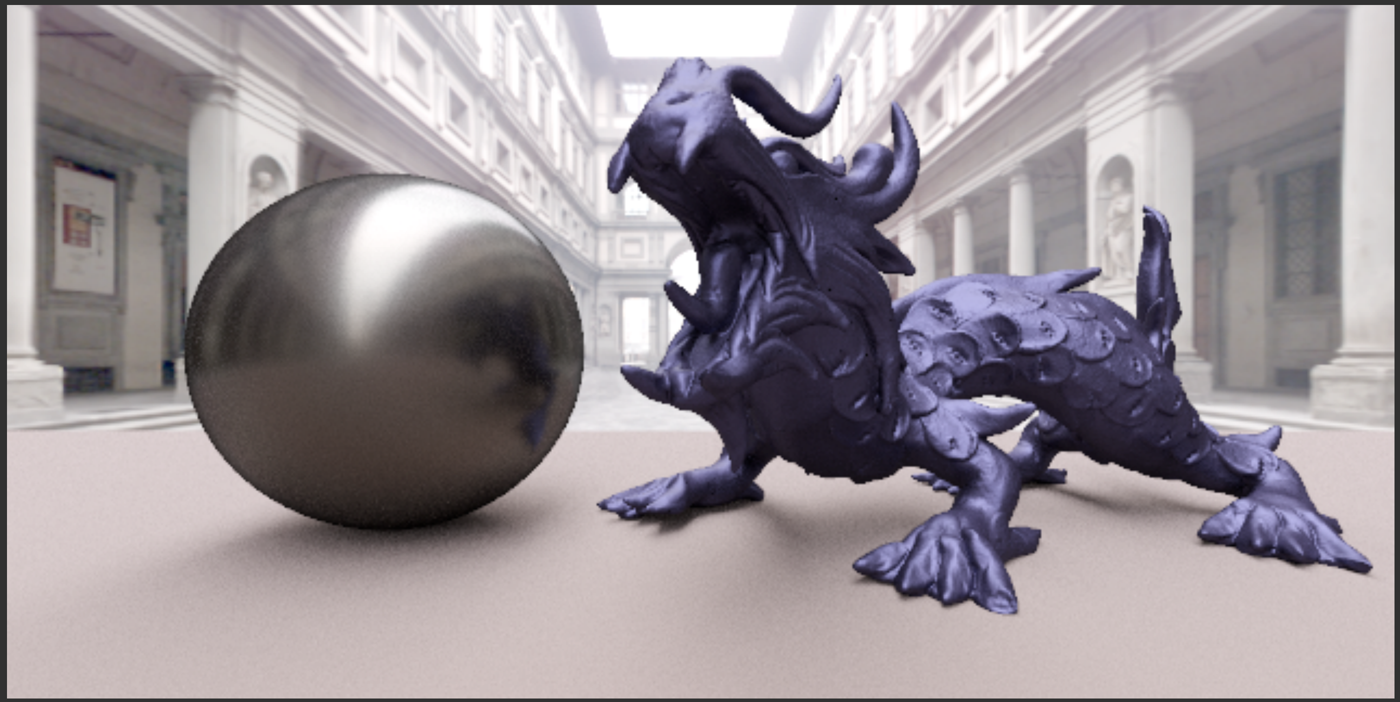


Variance
(0.0701,0.0631,0.0937)
Mean Lab Error: 1.13

Tabulated Data

Variance
(0.0077,0.0062,0.0095)

# Results: Global illumination

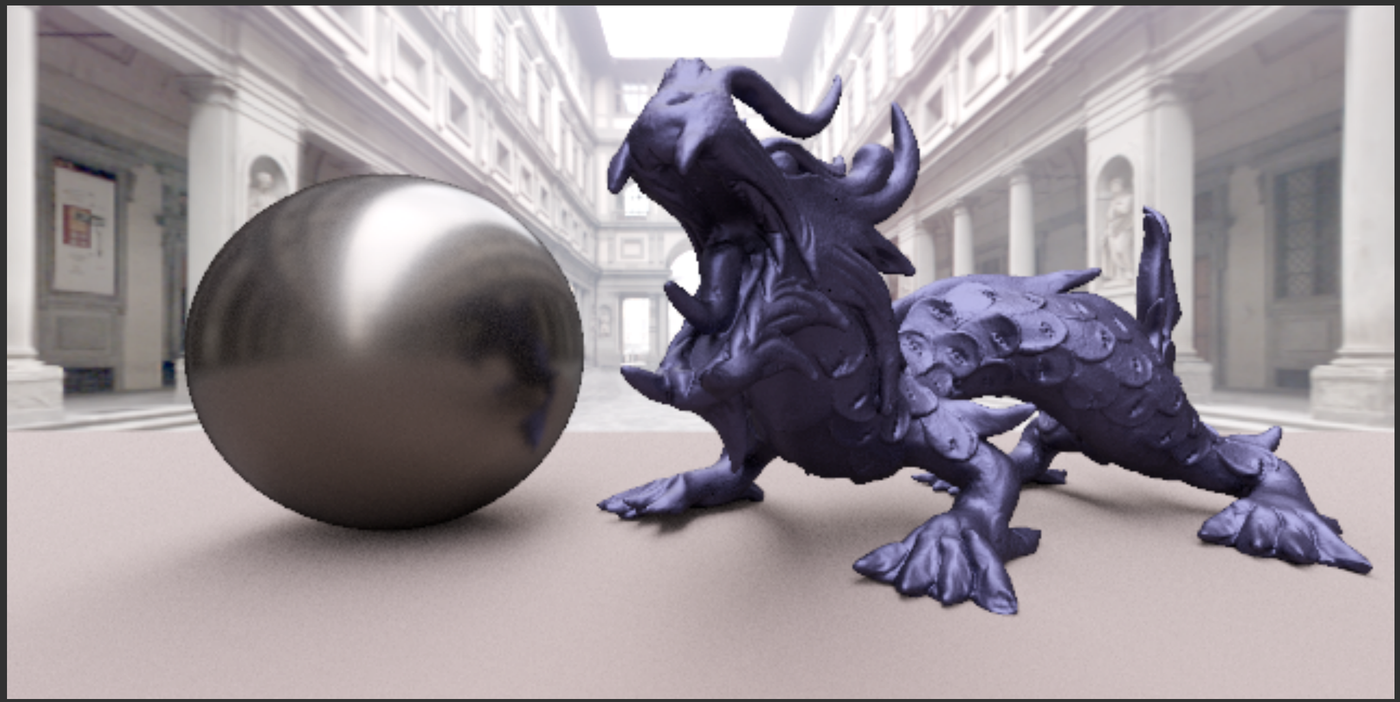


Rational Functions

BRDFs : 1.67KB
Inverse CDFs: 0.600KB
Mean Lab Error: 0.77

Tabulated Data

BRDFs: 480 KB
CDFs + PDFs: 30 MB



# Conclusion

- Rational Functions Framework:
  - Representation for BRDF and inverse CDF
  - Associated Fitting Technique
    - Global convergence
    - A priori error control
  - New Monte-Carlo estimator without PDF storage
- Limitations:
  - Poles between data samples
    - Happen for sparse data
    - Very rare for the MERL-MIT data base
    - Solution: generate artificial samples with intervals
  - No straightforward way to do Multiple Importance Sampling

# Future Work

- Color Compression
- Controllable Rational Functions for artists/users
  - Separable diffuse, specular, fresnels effects
- Speed improvement for the fitting algorithm
- General Fitting Library
  - Non-linear, linear and quadratic approaches
  - C++ and Cuda
  - Free and Open Source

# Thank your for your attention

Acknowledgements:


- MERL-MIT BRDF data base
- Kenny Mitchell (Disney Interactive Studios Research)
- Research supported by the project ALTA (ANR-11-BS02-006)
- TVCG Editor in Chief and Reviewers
- I3D Chairs for the invitation
- Collaborators: H. Lu and L. Belcour