ART AND ANIMATION STUDIO®

FurryBall GPU renderer

Brand new OptiX based core in FurryBall RT.

Jan Tománek

film Producer, Director and Founder of AAA studio

furryball RT GPU RENDERER

ART AND ANIMATION STUDIO®

Czech animation studio based in Prague

Two CGI animated full length movies for cinema

- **Goat story 1**
  - The first East European CGI feature movie (2008)
  - In year 2008 the movie has immediately become the most successful Czech animated movie ever.
    - Average scene complexity
      3-5 mil polygon
      3-8 Gb textures
- Average render times (**30-40 min per frame**) / using **AIR Renderman compiant**

Chapter I

Pure rasterization

FurryBall 1-3

Inspiration in game engines

Hey guys,
why we render a single frame 40 min and games like Assassin's creed are in realtime!? ;-)

FurryBall development

Pure rasterization

- The first authentic test from 2009
- (GTX 285… ;-)
- Started as a fast pre-visualisation plug-in
  Only in-house. *(like Pixar Renderman)*
- The first tests in 2008-2009
  Why not add other features…
- We found it so powerfully for movie production

Goat story 11

CGI feature 2012

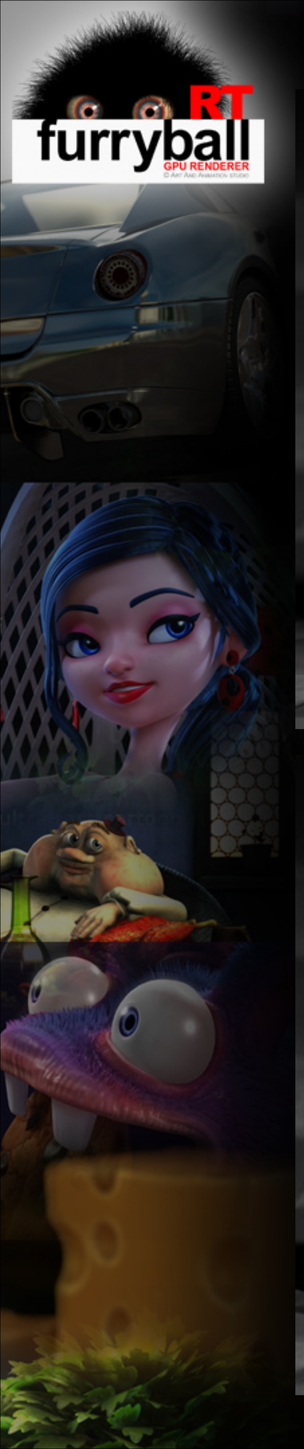
- 980 shots (3D stereo), HD (2K)
- 83 min
- **Whole render on NVIDIA GPUs**
- Average scene complexity
  3-5 mil polygon
  3-8 Gb textures
- Because of stereo we would like to avoid of post production - NO render layers
- DOF, MB, hairs, Fluids and particles
- Average render time (NVIDIA GTX 680)
  3-12 min for 2 HD Stereo frames
  (5x-10x speed up in better quality)

Rasterization vs Raytrace

- **Power of rasterization:**
  Very fast primary "rays"
  Perfect for Cartoon like animation, CGI movies
  Fur and hair, Displacement, Textures *(midmaps)*

- **Weakness of rasterization:**
  Reflections - *very expensive and incorrect, only ONE bounce*
  Indirect lighting - *very expensive and incorrect, only ONE bounce*
  SSS - *Fast, but incorrect*
  No Multi GPU
  No Progressive
  All features has to be SPECIALLY tweaked - settings per object
  Fast but hard to handle

- **Problem of using FurryBall in:**
  Design, Automotive, Architecture rendering, product rendering, hyperreal VFX



Compare GPU and CPU rendering

Rasterize rendering

100x faster than CPU raytrace

Test workstation

***Intel Core i7-930 2,80 GHz**, 16 GB RAM,*
Nvidia Geforce GTX Titan - 6 GB Memory

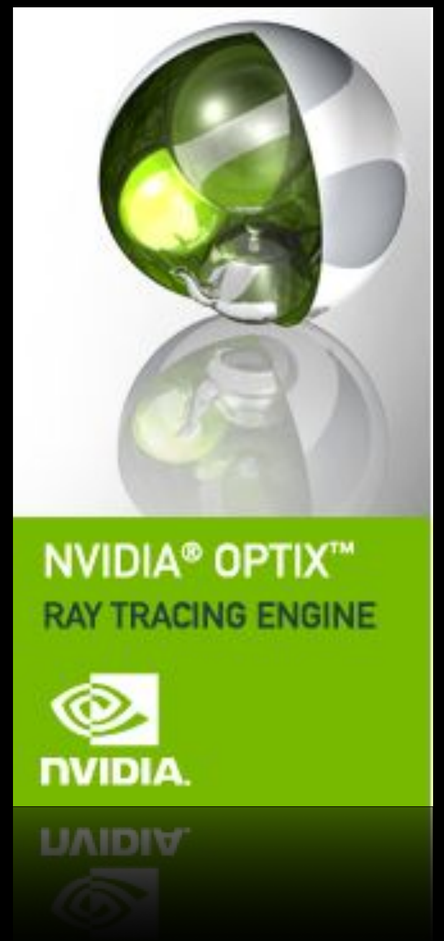
Chapter II

Rasterization with some raytrace effects

FurryBall 4

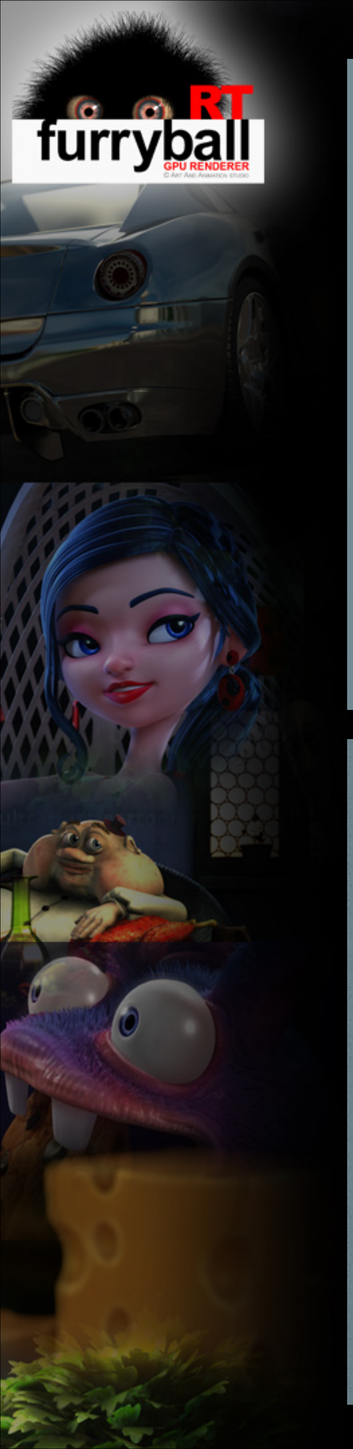
How we started with Nvidia OptiX implementation

- My first OptiX notice was on GTC 2013 - *Dave McAllister talk*
- I come back to Prague - **“Hey guys, let’s trace!!!”** ;-)
- In **ONE month** we implemented traced shadows
- Guys start to love it - “let’s try reflections and indirect”
- Reflection was FASTER than our fake “advanced rasterize” reflection
- In **3 months** we release the first FurryBall 4 beta with OptiX features
- ***Monte Carlo* path tracing**
- **Thanks to *Phil Miller, David McAllister* and *John Ison* from NVIDIA**

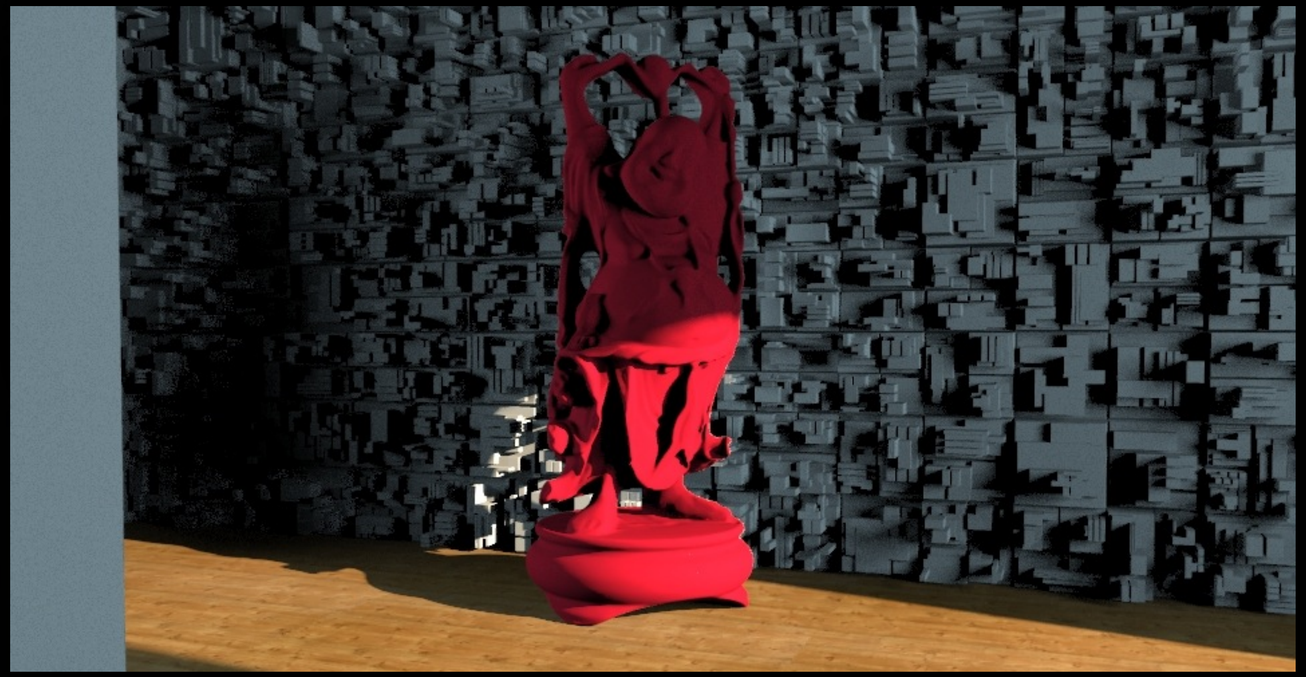


FurryBall 3 - Rasterize only (fake GI) - **5 sec. Full HD**

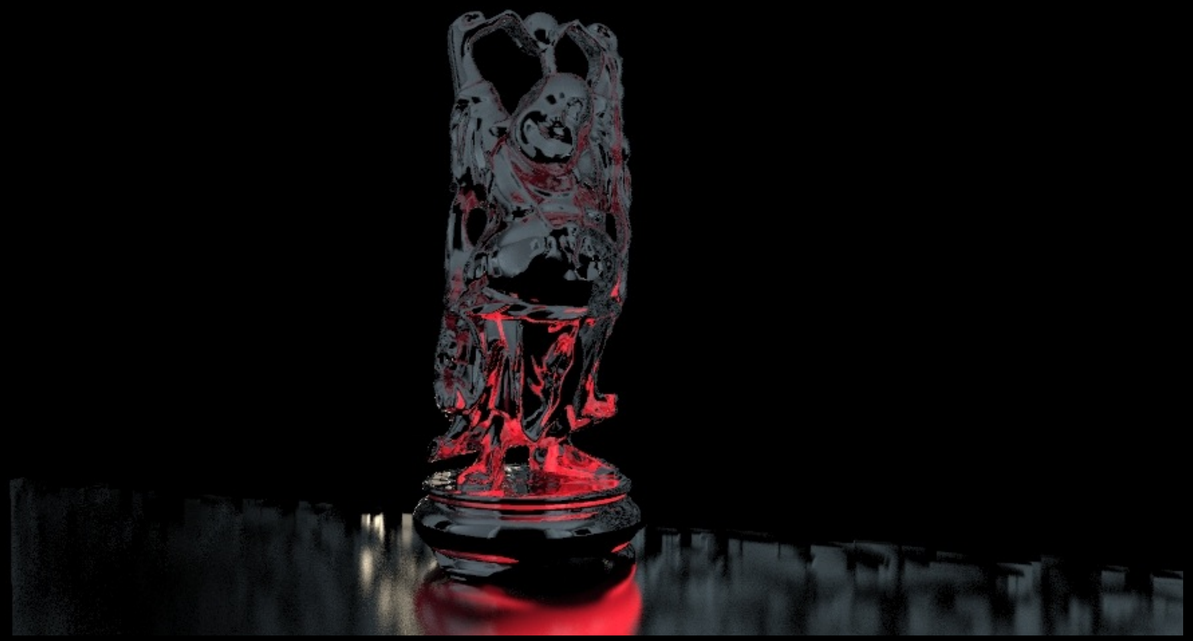
FurryBall 4 - Rasterize and Raytrace indirect - 1:40 min Full HD



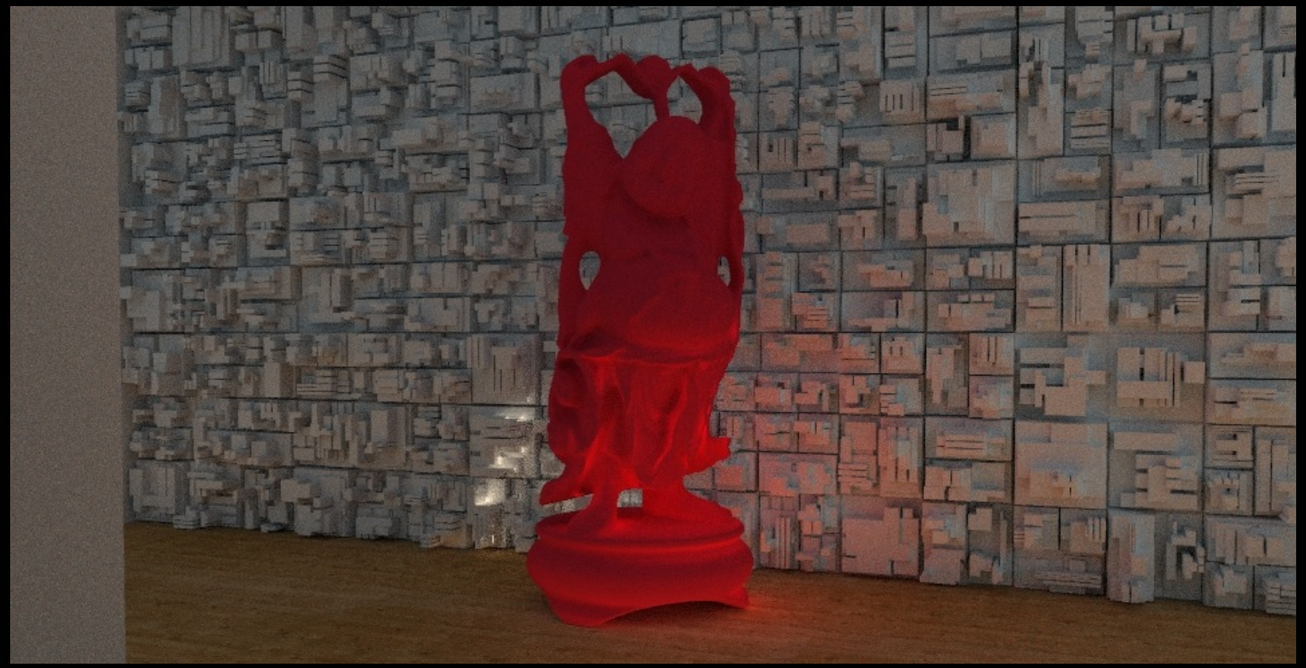
Rasterize - DX render - ONLY
Primary rays
0,1 sec



Rasterize and raytrace shadows
3 sec



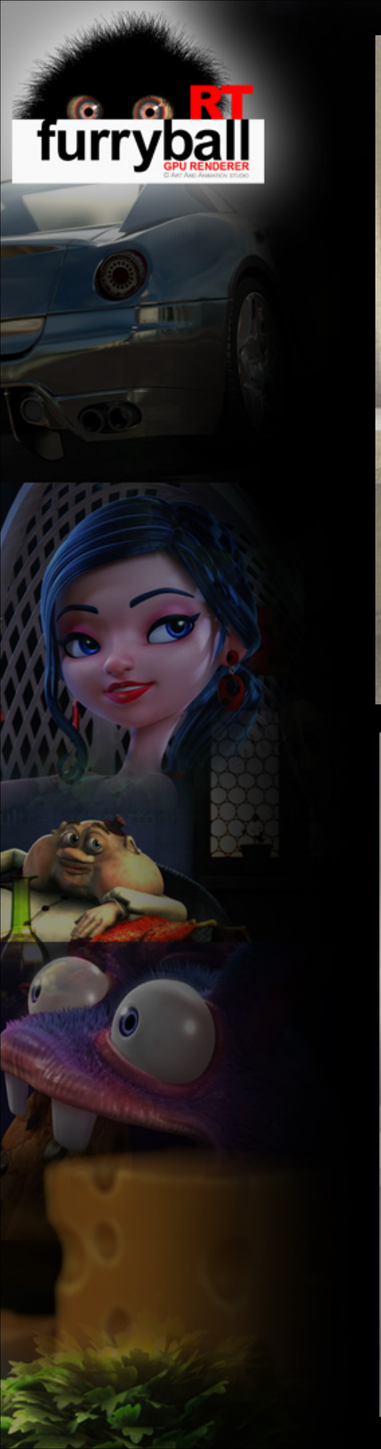
Raytraced reflections
8 sec



Raytraced indirect lighting - *path tracing*
15 sec

Complete render - 27 sec

(Direct light, Indirect, Reflection, Shadows)



Compare GPU and CPU rendering

Lit only by indirect light

Raytracing

Unbiased rendering, Physically based full global illumination, brute force, with 1 primary rays and 3 light bounces - 4 bounces in some renderers

Speed up - up to 10x for raytrace

Test workstation

***Intel Core i7-930 2,80 GHz**, 16 GB RAM, Nvidia Geforce GTX Titan - 6 GB Memory*

FurryBall 4

Combination Rasterize and Raytrace

Rasterize for the **PRIMARY** rays, Raytrace for the **SECONDARY** rays

- **Advantages**
  - Reality and great speed
  - Always clear image in primary rays (rasterized)
  - Lower resolution textures can be used for secondary rays - *indirect light*
  - Scalable features for the users - *turn on/off raytrace*
  - Reflection was EVEN FASTER than our *advanced rasterize reflection*

- **Problems**
  - Sometimes artifacts in combination Rasterize and Raytrace
  - Two separate branch code *(Raster and Raytrace)*
  - Complicated for users - *in fact 2 renders - so robust*
  - Problems with multipass antialiasing *(6x same image in Raster)*
  - No Jitter and Rasterize hardware antialiasing in Raytrace
  - 2x textures in GPU RAM (Raster and RT)
  - Problems in progressive render and multi GPU
  - Huge rendering targets *(many 4K resolution targets for each effect)*
  - Windows only - *DirectX*

Chapter III

Pure Raytrace *(path tracing)*

FurryBall RT

FurryBall RT

Pure raytrace - Monte Carlo path tracing

- **Advantages**

  Same speed for primary rays like Rasterization for huge scenes

  Multi-pass antialiasing for *“free” - hunders of rays*

  Much Faster viewport (*30x faster than FurryBall 4)*

  Pixel reduction for huge scenes

  Progressive rendering

  Easy setup - just one few quality parameters

  Multi GPU - *easy scalable*

  Open for Linux and OSX

- **Problems**

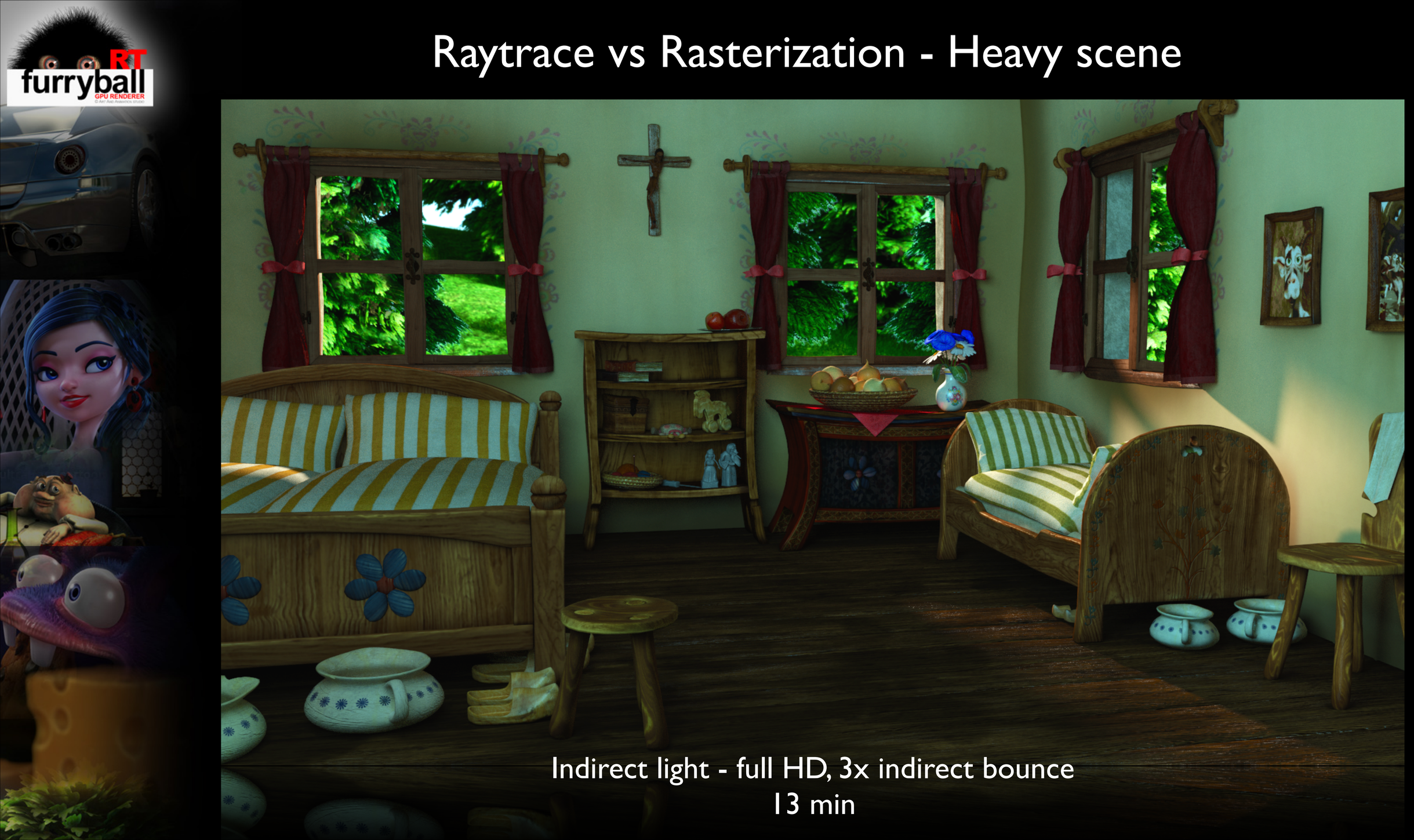
  Memory - all textures and geometry has to be in GPU RAM

  (*Today 6GB or 12GB GPU NVIDIA solves the problem)*

Raytrace vs Rasterization - Heavy scene

No indirect lighting
0:33 min

Raytrace vs Rasterization - Heavy scene

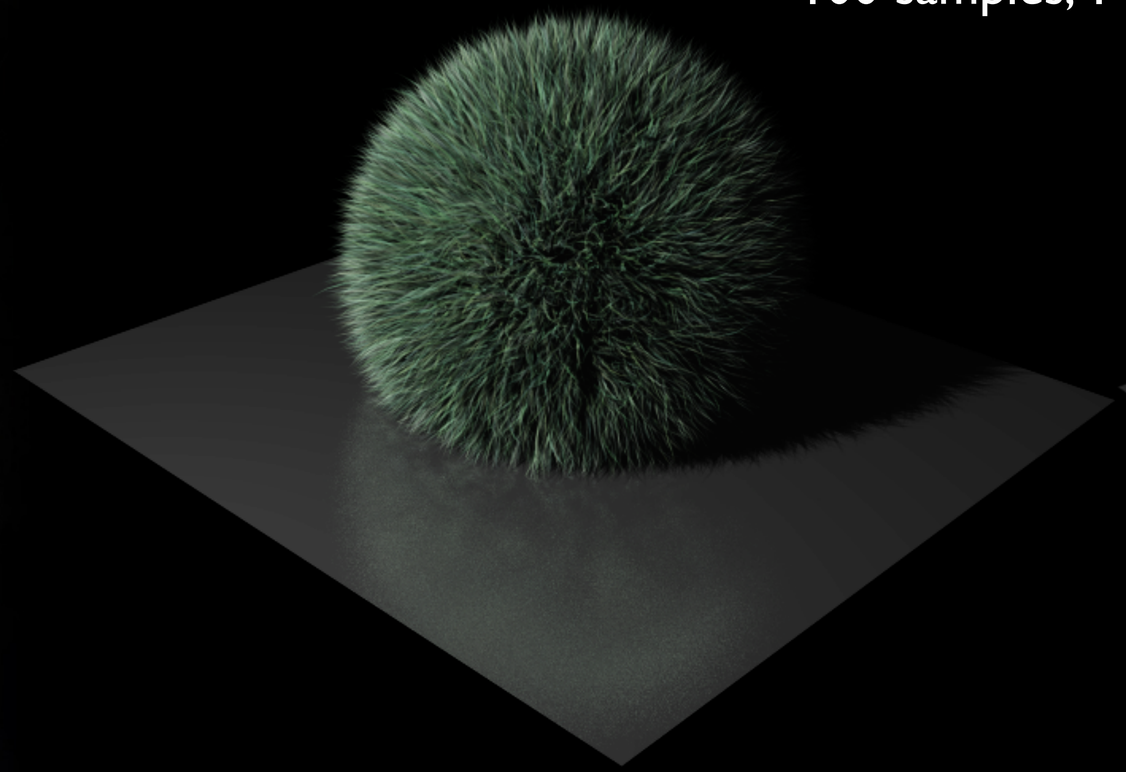


Indirect light - full HD, 3x indirect bounce
13 min

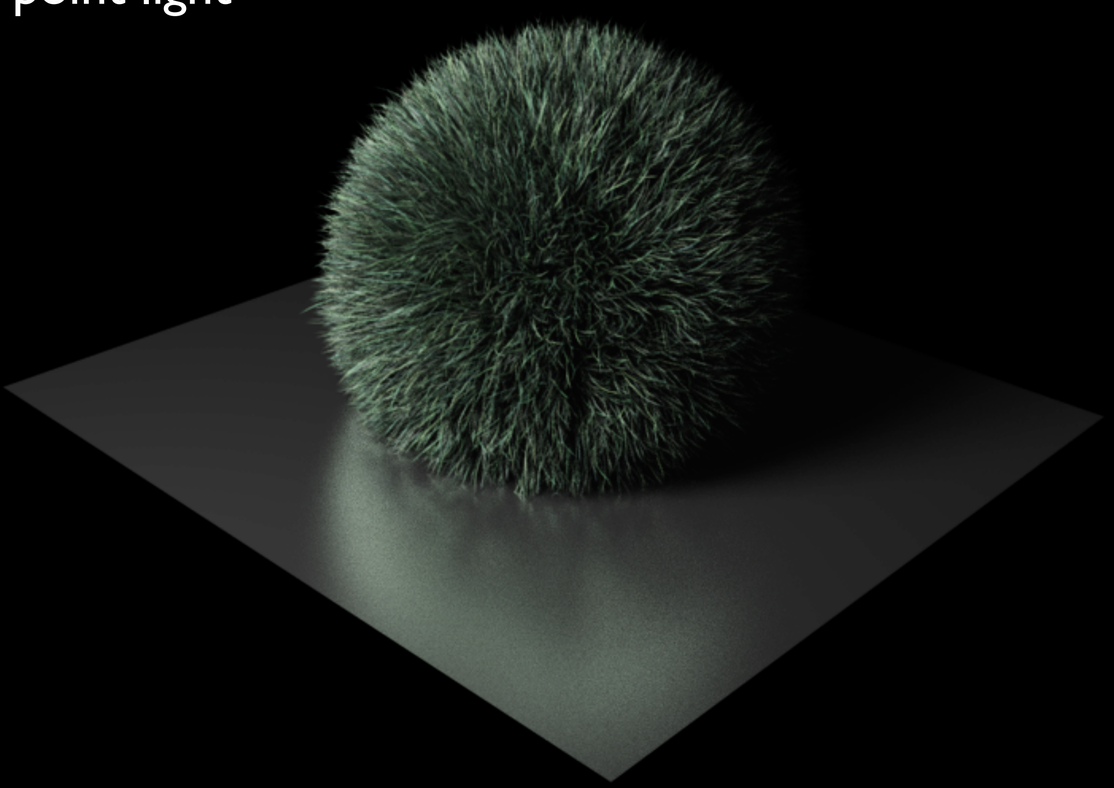
	Rasterization **- FurryBall 4**	***Raytrace*** **- FurryBall RT**
Viewport realtime rotate in HD	1 fps *(no pixel reduction possibe)*	22 fps (with pixel reduction)
Final render HD (no shadows)	59 sec	22 sec
Final render HD (no shadows)	10 sec (hardware jitter antialiasing)	22 sec
Final render HD with shadows (raytraced shadows on rasterize)	1:47 min	33 sec
Full indirect lighting	N/A	13 min

FurryBall RT vs Mental Ray

Shave & Haircut
100 samples, 1 point light



Mental Ray
7:01 min



FurryBall RT
0:20 min

intel i7 2.8 GHz, 18 GB RAM
NVIDIA Titan 6 GB

CG production market - FurryBall 1-4

Realism

Render Time

Frames / s

seconds

minutes

hours

days

Game engines
games, previz

FurryBall 1-4
CGI movies, VFX, previz, advert
Still images

Renderman
CGI movies, VFX, advert

CPU Raytracers
Product design, still images, advert

CG production market - FurryBall RT with Raytrace

Realism

Render Time

Frames / s

seconds

minutes

hours

days

Game engines
games, previz

FurryBall
CGI movies, VFX, previz, advert
Still images

Renderman
CGI movies, VFX, advert

CPU Raytracers
Product design, still images, advert

RT
furryball
GPU RENDERER
© Art And Animation studio

ART AND ANIMATION STUDIO®

raytrace vs rasterization

- Robust and Complicated rasterization become slower and slower
- Raytrace has “almost” consistent speed in huge scenes and shaders
- Raytrace is for some cases even faster than rasterize *(Huge scenes)*
- Faster viewport interactivity *(about 30x faster in some cases)*
  Pixel reduction and multi GPU (fast interaction, effects, layered materials…),

New features in FurryBall RT

RT like RayTrace, RevoluTion, RealTime

- Full support for Maya hairs and Shave & Haircut *(Yeti and other come later)*
- OpenSubdiv and P-Tex support *(later)*
- Maya Fluids and particles *(now support all raytrace features in the scene)*
- Finally ready for OSX and Linux *(No DX problem)*
- Cinema and 3DS Max come later
- IPR render, Unlimited layer shaders, Ramp shader
- New FurryBall lights
- No more compromises and easy interface
- and much more

"LIVE" 10-15 min FurryBall RT presentation on real movie scene

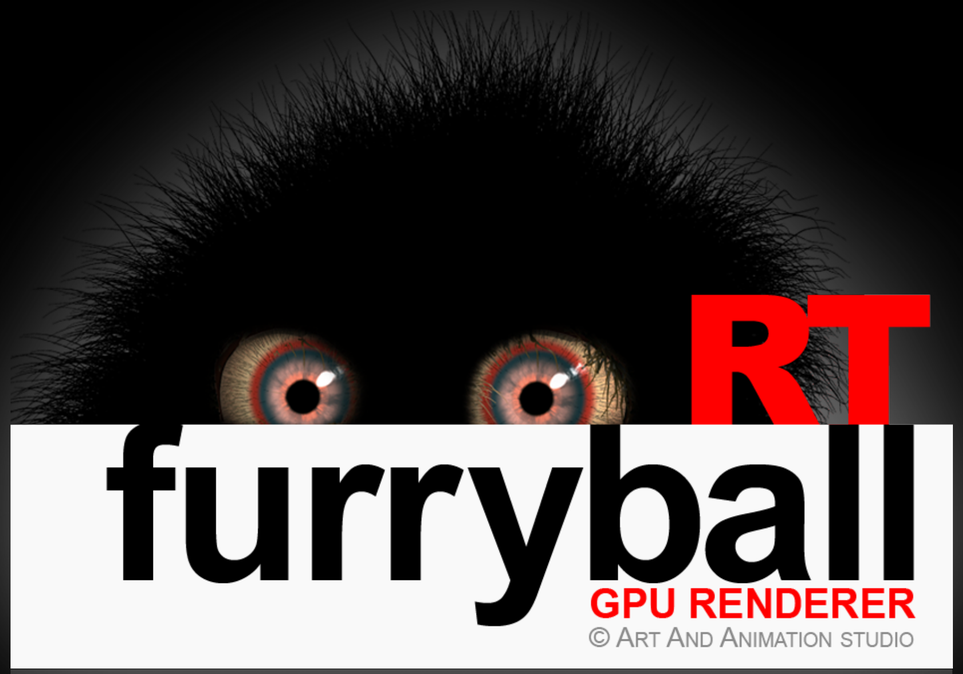
DELL mobile workstation NVIDIA Geforce 680M

(3x-4x slower notebook GPU than GTX TITAN)

ALIENWARE
HIGH-PERFORMANCE SYSTEMS

Thank you for your attention

ART AND ANIMATION STUDIO®



http://furryball.aaa-studio.eu/

@FurryBall_GPU #GTC15

tomanek@aaa-studio.eu