nvision 08

THE WORLD OF VISUAL COMPUTING

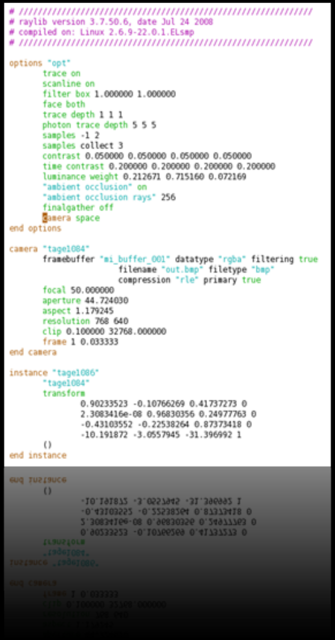
The Future of Rendering

Rolf Herken, CEO and CTO of mental images

What is Rendering?

Definition

Rendering is the process of converting data into visually perceivable form.

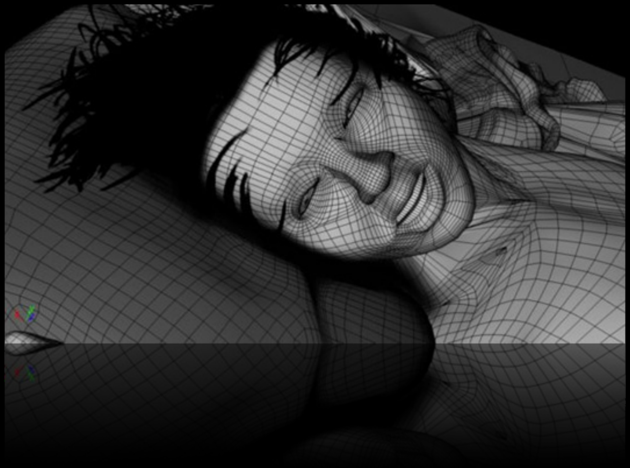


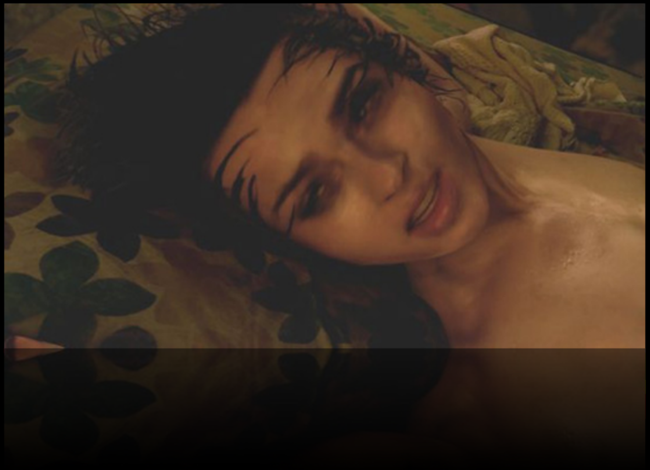
```
# ////////////////////////////////////////////////////////////////////
# raylib version 3.7.50.6, date Jul 24 2008
# compiled on: Linux 2.6.9-22.0.1.ELsmp
# ////////////////////////////////////////////////////////////////////

options "opt"
        trace on
        scanline on
        filter box 1.000000 1.000000
        face both
        trace depth 1 1 1
        photon trace depth 5 5 5
        samples -1 2
        samples collect 3
        contrast 0.050000 0.050000 0.050000 0.050000
        time contrast 0.200000 0.200000 0.200000 0.200000
        luminance weight 0.212671 0.715160 0.072169
        "ambient occlusion" on
        "ambient occlusion rays" 256
        finalgather off
        camera space
end options

camera "tage1084"
        framebuffer "mi_buffer_001" datatype "rgba" filtering true
                        filename "out.bmp" filetype "bmp"
                        compression "rle" primary true
        focal 50.000000
        aperture 44.724030
        aspect 1.179245
        resolution 768 640
        clip 0.100000 32768.000000
        frame 1 0.033333
end camera

instance "tage1086"
        "tage1084"
        transform
                0.90233523 -0.10766269 0.41737273 0
                2.3083416e-08 0.96830356 0.24977763 0
                -0.43103552 -0.22538264 0.87373418 0
                -10.191872 -3.0557945 -31.396992 1
        ()
end instance
```





What is Rendering?

Application Examples

- 2D graphics
  - fonts
  - user interfaces
- 3D graphics
  - scientific visualization
  - product visualization
  - dynamic content, e.g. visual effects
  - interactive content, e.g. Games
  - 3D Web applications

The Rendering Space

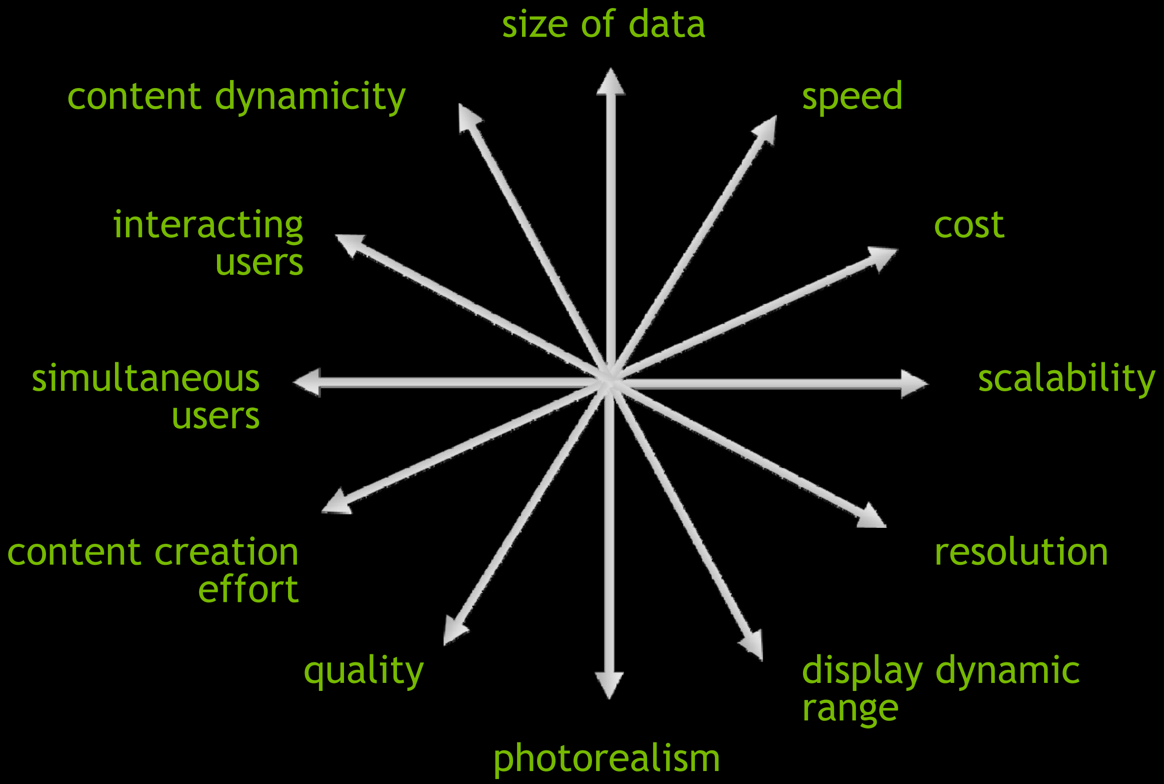
Dimensions and Variables of Rendering

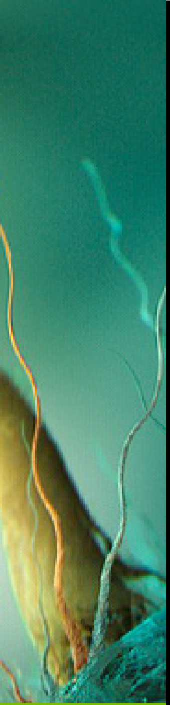
- data complexity: static, dynamic, interactive, in-core, out-of-core
- algorithms and parallelization: memory footprint and scalability
- image data: dynamic range, resolution, pixel layout, stereo
- performance: image quality and correctness versus speed
- visual richness: programmability, content creation effort
- photorealism: physical correctness, precision, approximations
- cost: per image, per hour, per user
- coherence amortization: simultaneous users, interaction among users

The Rendering Space

Dimensions and Variables of Rendering

- impossible to optimize all at the same time (and all are “logarithmic” scale)





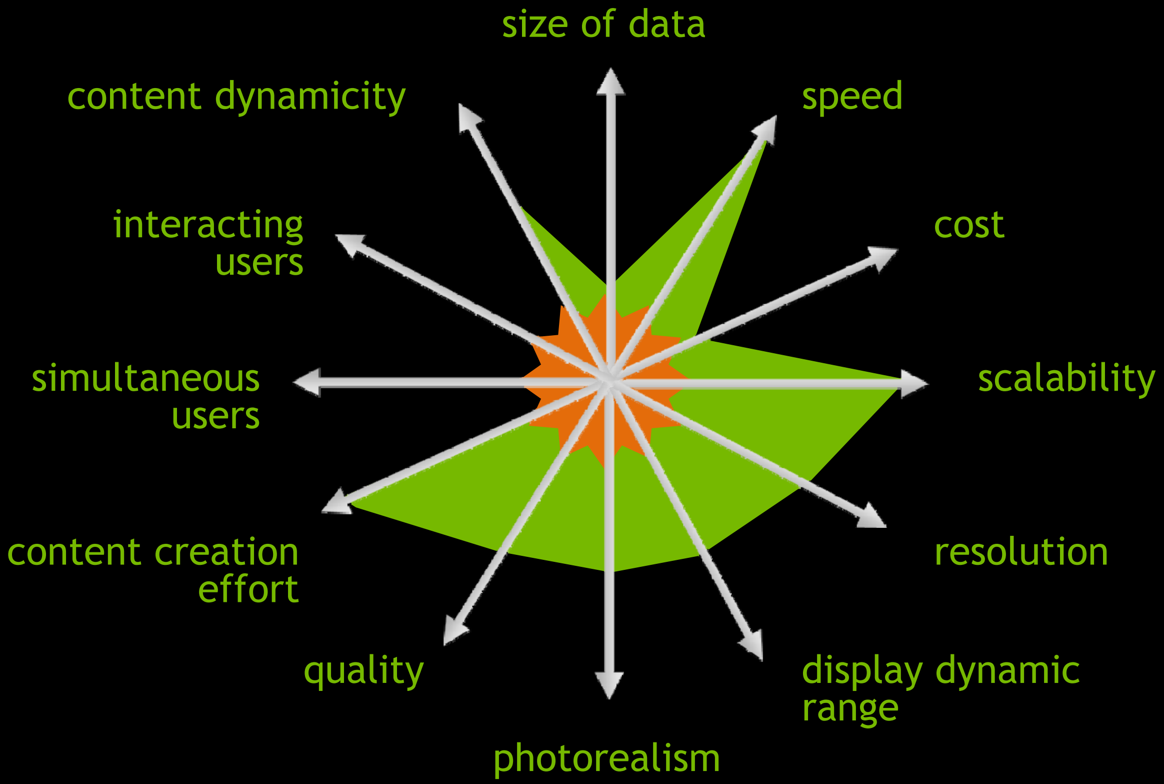
Rendering Technologies

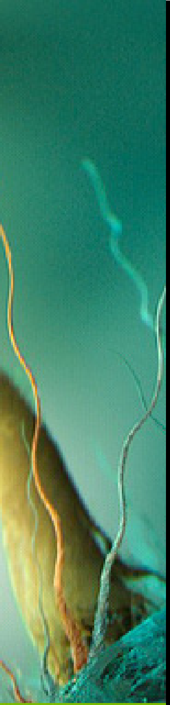
Rasterization

- rapid image generation by exploiting regular structure of pixel matrix
- GPUs realize rasterization in hardware
    - minor part of the GPU is used for rasterization
    - most of the GPU is dedicated to other computations, such as shading, geometry, and physics simulation
    - inherently parallel architecture
- programmable shading in several languages
- higher order illumination (“Global Illumination”) difficult or impossible to achieve

The Rendering Technologies

Navigating the Rendering Space: Real-time Rasterization





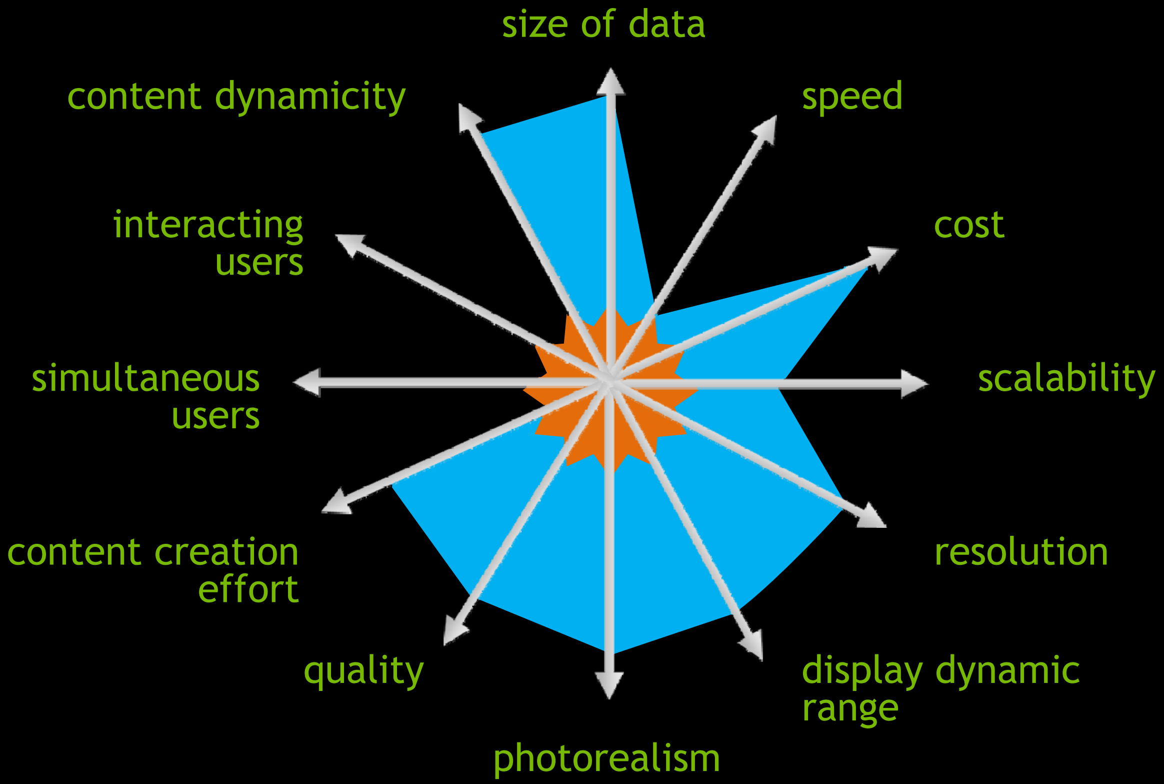
Rendering Technologies

Ray Tracing

- allows for higher order illumination (Global Illumination)
  - easy to simulate all transport paths of light
  - real shadows, reflections and refraction, and indirect illumination
- programmable shading from the beginning
- extremely versatile and (ideally) simple to use
- to date usually CPU based software only

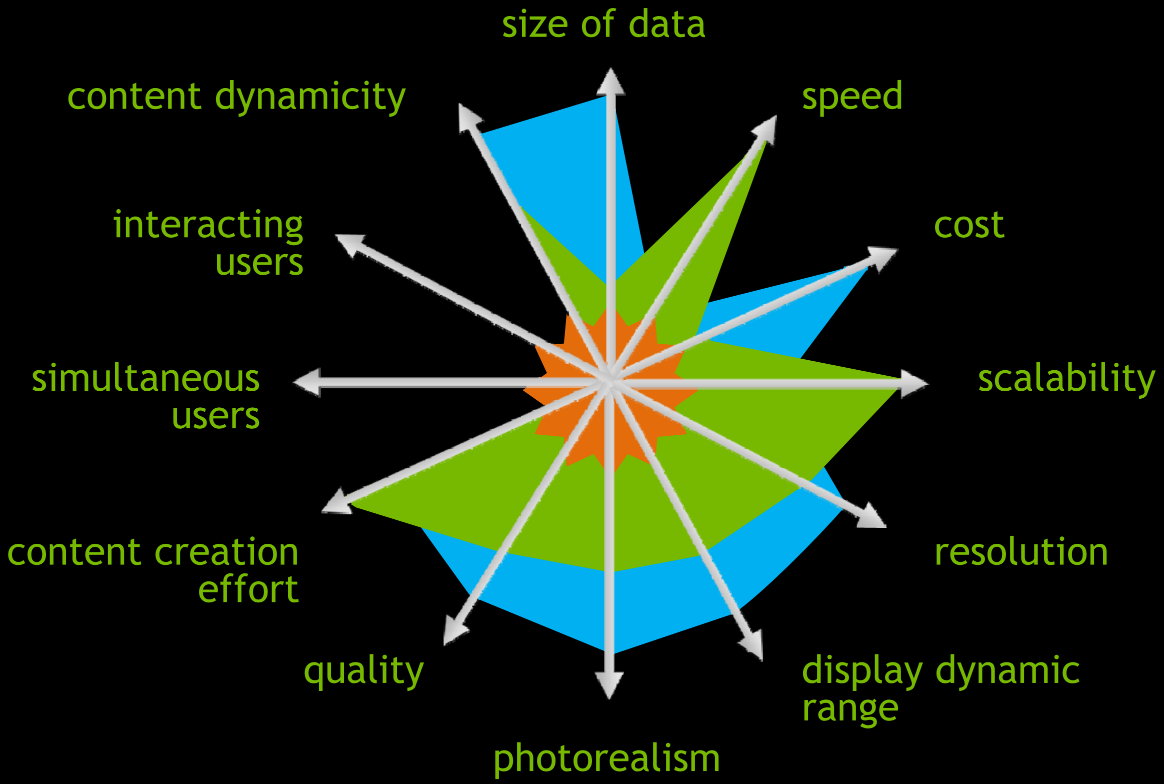
The Rendering Technologies

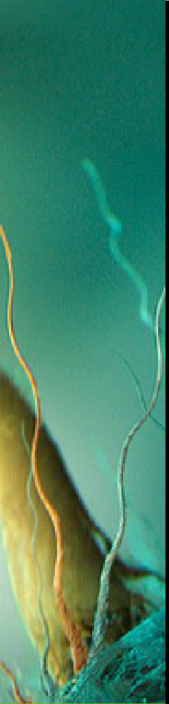
Navigating the Rendering Space: Today's Ray Tracing



The Rendering Technologies

Today's Ray Tracing versus Real-time Rasterization





Ray Tracing: State-of-the-Art

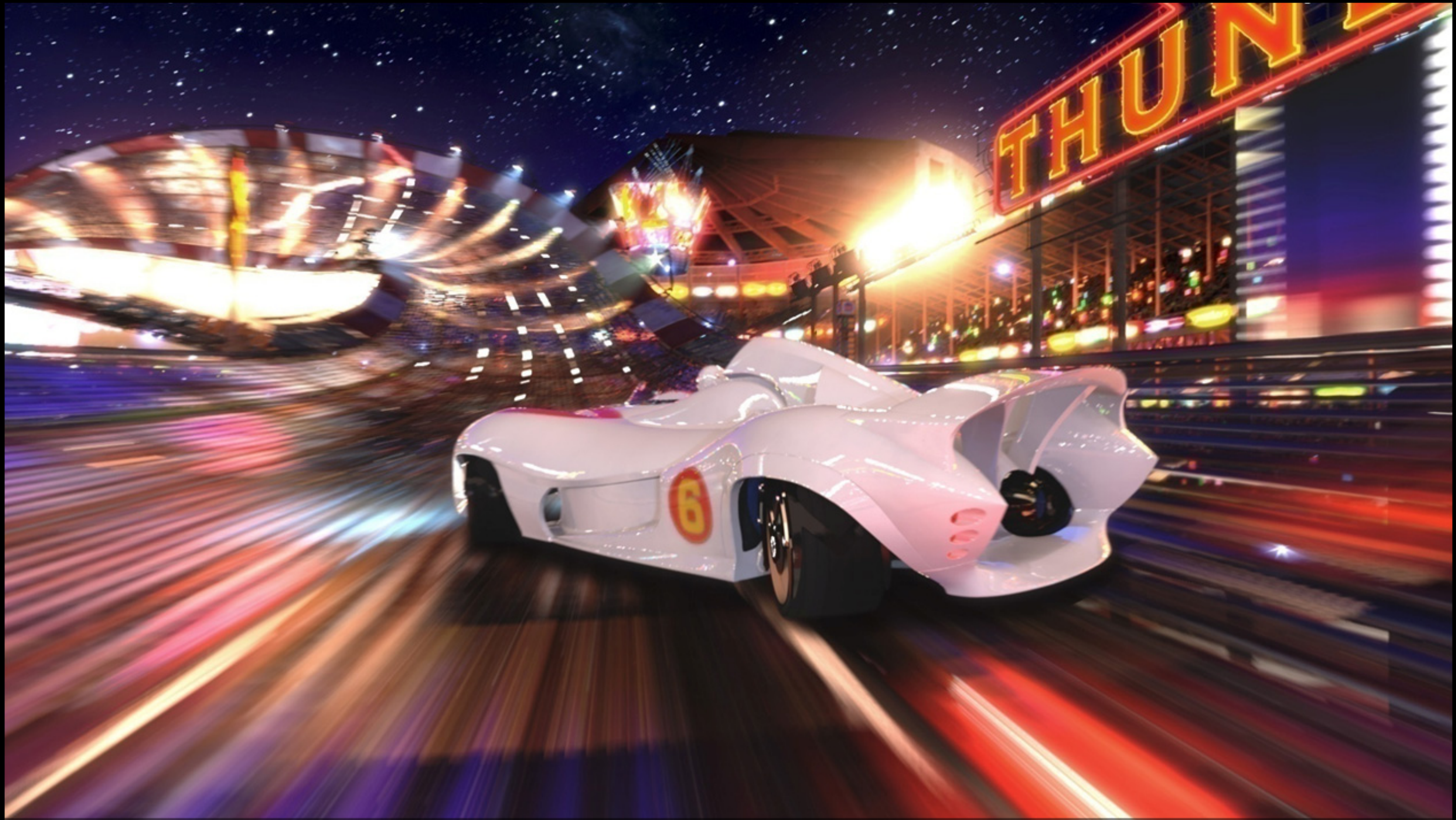
mental ray®

- industry standard since 15+ years
  - 10+ million install base (VFX/animation studios, Maya, 3ds Max, Softimage|XSI, Catia, SolidWorks, AutoCAD, Revit, Inventor, etc. …)
- most versatile and powerful ray tracer in the market
  - unsurpassed image quality
  - can render extremely complex content
  - quasi-Monte Carlo technology: deterministic, reproducible, faster, correct
  - fully programmable
    - hear Andy Kopra talk on mental ray

Ray Tracing: State-of-the-Art

SPEED RACER
Image rendered with *mental ray*® by Digital Domain

Ray Tracing: State-of-the-Art



SPEED RACER
Image rendered with *mental ray*® by Digital Domain
© 2008 Warner Brothers. All Rights Reserved.

Ray Tracing: State-of-the-Art

SPEED RACER
Image rendered with *mental ray*® by Digital Domain
© 2008 Warner Brothers. All Rights Reserved.

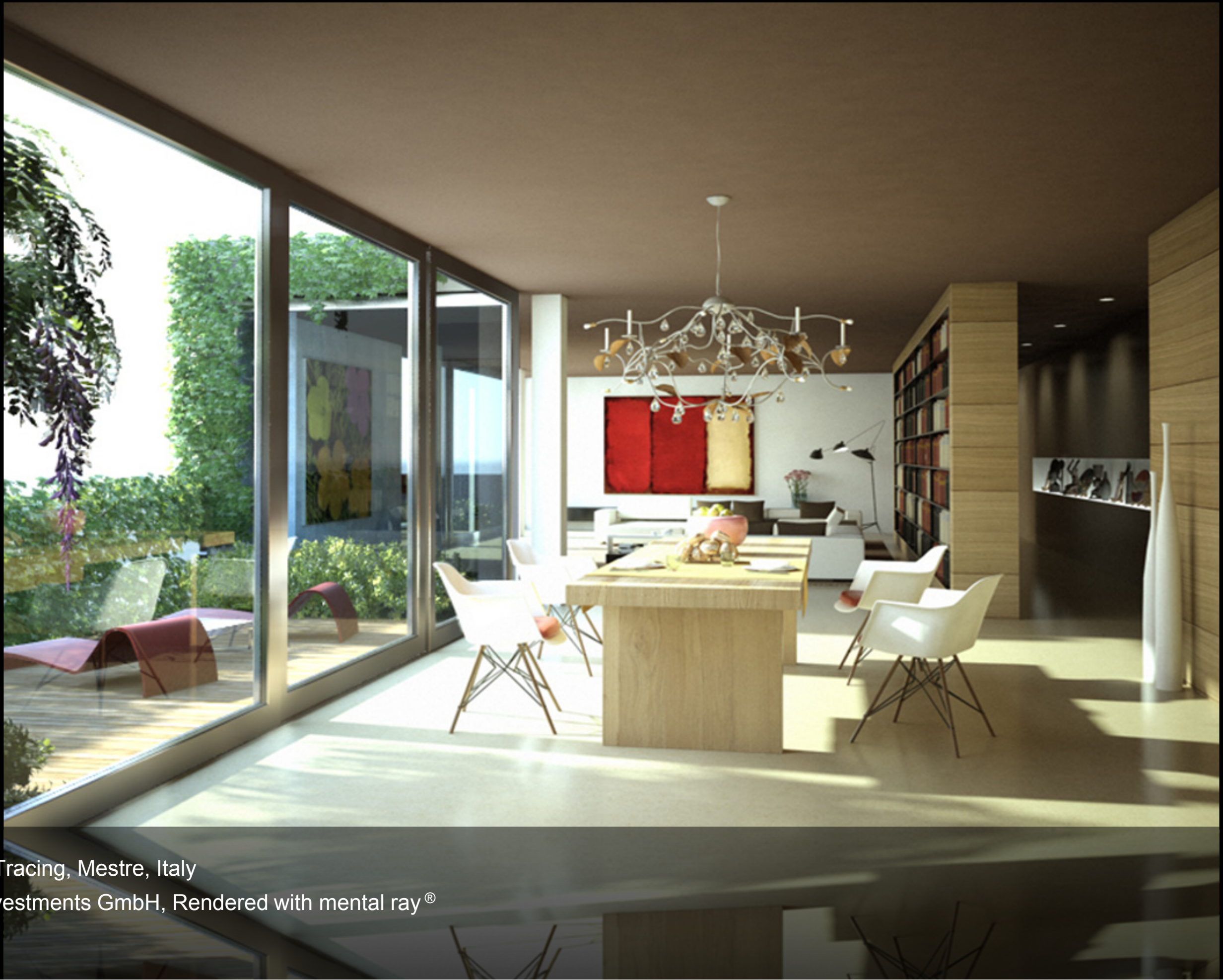
Ray Tracing: State-of-the-Art

SPEED RACER
Image rendered with *mental ray*® by Digital Domain
© 2008 Warner Brothers. All Rights Reserved.

Ray Tracing: State-of-the-Art

SPEED RACER
Image rendered with *mental ray*® by Digital Domain

Ray Tracing: State-of-the-Art



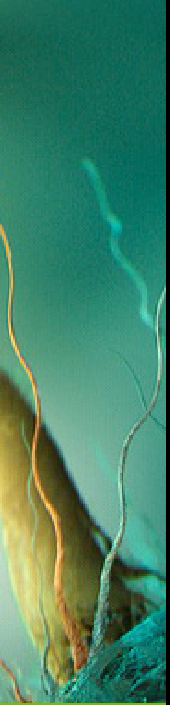
Visualization: Delta Tracing, Mestre, Italy

Design: Stofanel Investments GmbH, Rendered with mental ray®

Future Rendering Platforms

Technology Trends

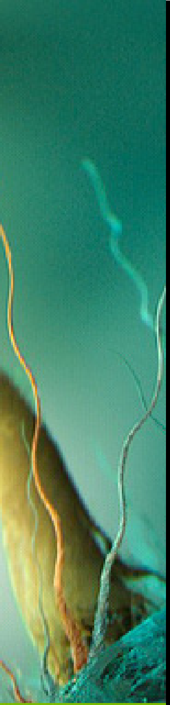
- CPU computational power, both in the PC and on the server, has increased along the path indicated by Moore’s Law beyond an inflection point
- the increasing incorporation of GPUs into PCs and server systems will allow new levels of performance to be achieved
- through a combination of these two trends, increased computational power of PCs and servers will transform 3D rendering



Future Rendering Platforms

Technology Trends

- ray-tracing technology will find its way into five main future rendering technology platforms
  - real-time game engines
  - interactive visualization engines
  - offline photorealistic rendering systems
  - online interactive photorealistic rendering systems
  - interactive photorealistic engines



Future Rendering Platforms

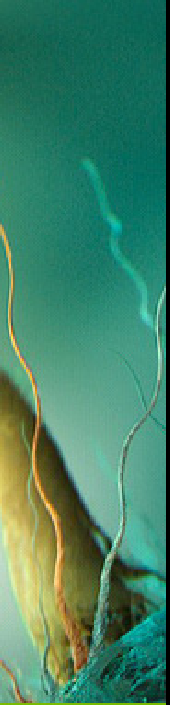
Real-time Game Engines

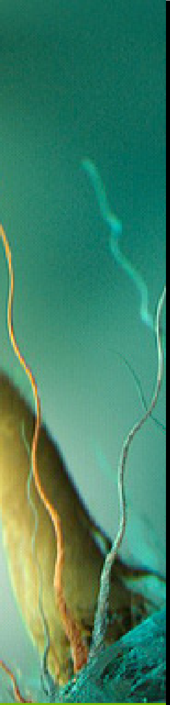
- future game engines will include ray-tracing features
  - either entirely based on ray-tracing, or on
  - ray tracing in combination with rasterization algorithms
  - frame rates of 30 to 60 frames per second, at high resolution
  - highly-tailored scenes of commercially viable artistic complexity
- *Example: id Software ray-casting work*

Future Rendering Platforms

Interactive Visualization Engines

- allow application specific real-time or near real-time (10-40 frames per second) rendering of scenes.
    - these scenes, and the rendering effects, can be strongly application specific
    - visualization areas include automotive styling and design, architectural walk-throughs, large scene visualization in plant design and city planning, seismic data exploration, scientific visualization, etc.
- *Examples: NVIDIA Siggraph ray-tracing demos, neuray® 2.2 (Q4 08)*
    - hear David Luebke and Steven Parker about interactive ray tracing

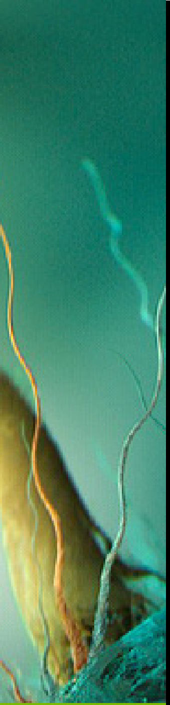




Future Rendering Platforms

Offline Photorealistic Rendering Systems

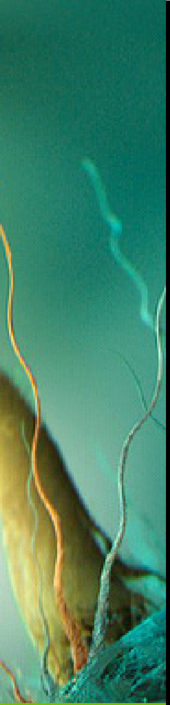
- are used to produce the highest-quality images or sequences of these images by performing offline processing
  - in VFX and animation production, coupling the availability of all possible photo-realistic and non-photo-realistic effects with complete artistic flexibility
  - *Examples*: *mental ray ®*, *RenderMan®*
  - in visualization applications, providing extreme ease of use in generating physically correct renderings for visualization applications
  - *Examples*: *mental ray 4.0 (2009), neuray 3.0 (2009)*



Future Rendering Platforms

Online Interactive Photorealistic Rendering Systems

- allow for interactive or near real-time (1-10 frames per second) manipulation and photorealistic rendering of extremely complex and visually rich scenes
    - built specifically for visual effects and feature animation pre-visualization and look development.
- *Examples: proprietary VFX and animation pre-visualization and pre-lighting systems, RealityServer® 3.0 with mental ray quality interactive neuray rendering module (2009/10)*



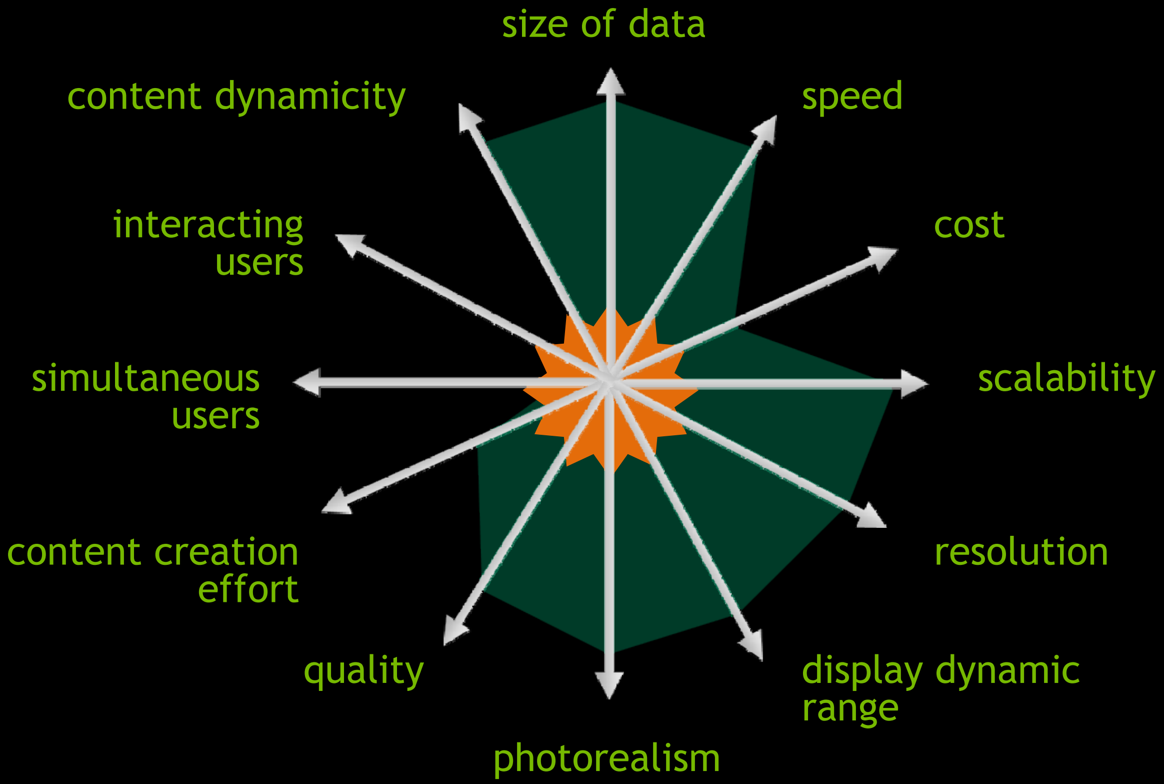
Future Rendering Platforms

Interactive Photorealistic Engines

- will comprise a new category of rendering platform
  - small object code and run-time memory footprint – suitable for integration into widely distributed 3D players
  - cover most of the effects capabilities of an off-line photorealistic rendering system, while still delivering interactive frame rates
  - smooth trade-off between realism and frame rate
  - potentially limited scene complexity, matched to reasonable web or document delivery database sizes
- *Example: iray® 1.0 (2009)*

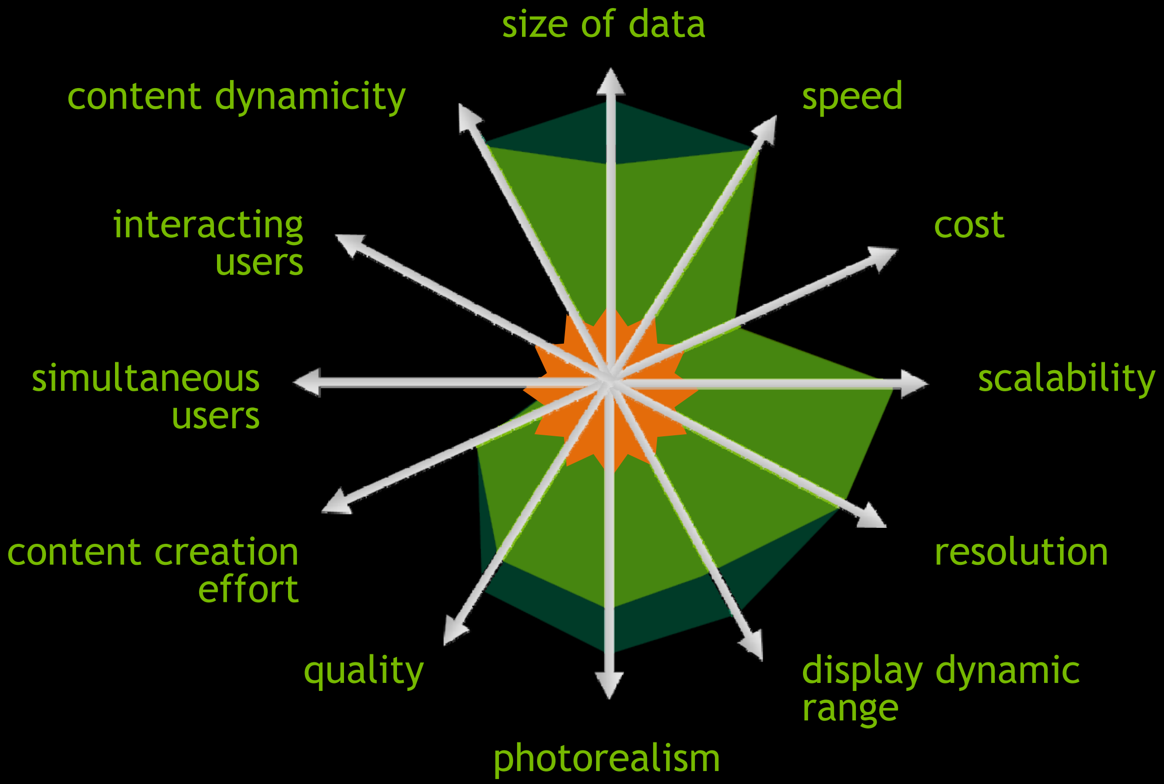
Future Rendering Technology

The Next Generation Rendering Technology



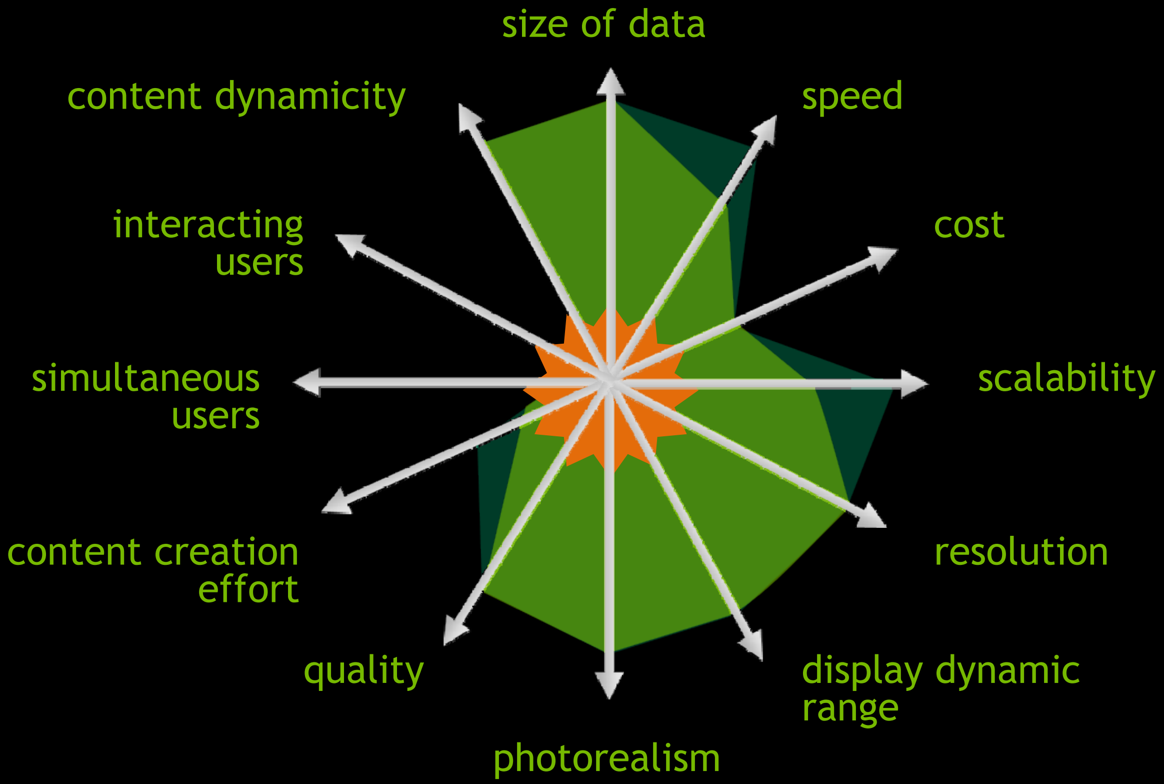
Future Rendering Technology

Real Time Game Engines



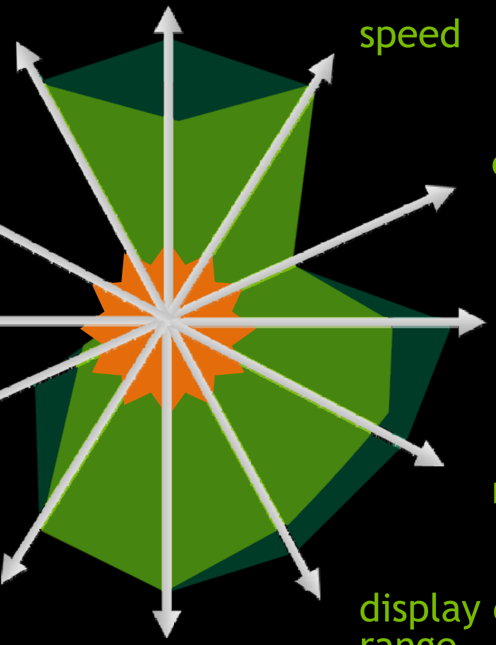
Future Rendering Technology

Offline Photorealistic Rendering Systems



Future Rendering Technology

Interactive Photorealistic Engines



size of data

content dynamicity

speed

interacting users

cost

simultaneous users

scalability

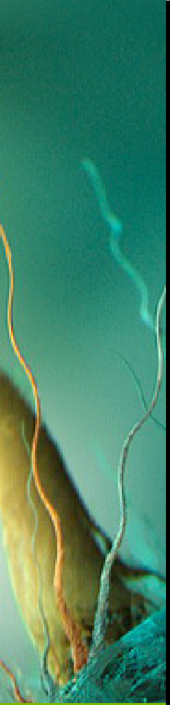
content creation effort

resolution

quality

display dynamic range

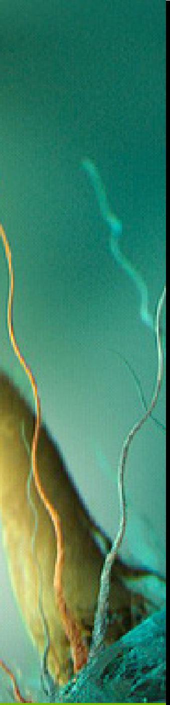
photorealism



Future Rendering Technology

Why More and More Ray Tracing?

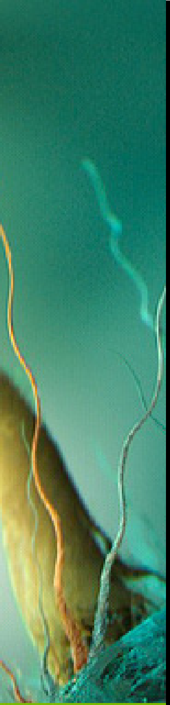
- everything is just simple, provided the compute resources will be available
    - physically correct simulation of light
        - find all light paths that connect pixels and light sources and sum up their contribution
            - correct (soft) shadows, even over real reflection and refraction
            - correct participating media scattering
            - correct subsurface scattering/translucency
            - wave length dependent phenomena
            - consistent adaptive sampling



Future Rendering Technology

Why Leveraging the GPU?

- enormous compute power due to inherent massive on-chip parallelism with increasingly general purpose processing units
  - increasingly suitable for general purpose computation (via CUDA)
  - ray tracing can be accelerated by the GPU



Future Rendering Technology

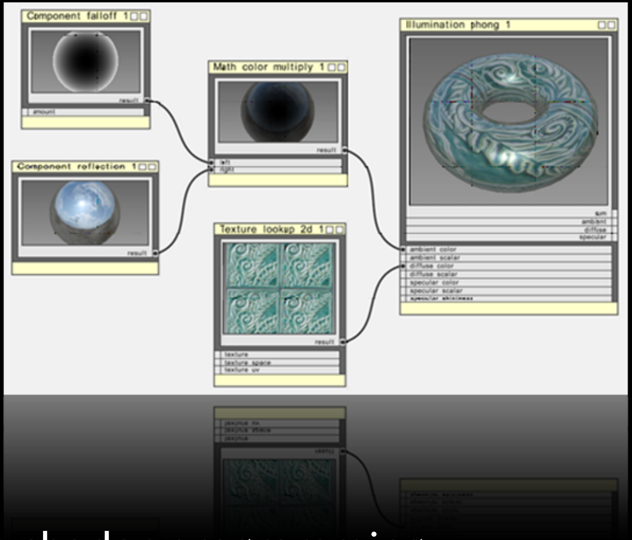
Ray Tracing and Rasterization will Converge

- rasterization is a subset of ray tracing
- shared data structures for acceleration, same ingredients and primitives
  - hierarchical culling
- z-buffer is an immediate mode algorithm: fixed memory, arbitrary number of polygons
- ray tracing can also be done in immediate mode with a different approach to generate acceleration data structures
- both feature programmable shading
- to facilitate the transition and convergence to richer visual possibilities
    - need for a Meta Shading Language

Future Rendering Technology

MetaSL™: Platform Independent Shader Representation

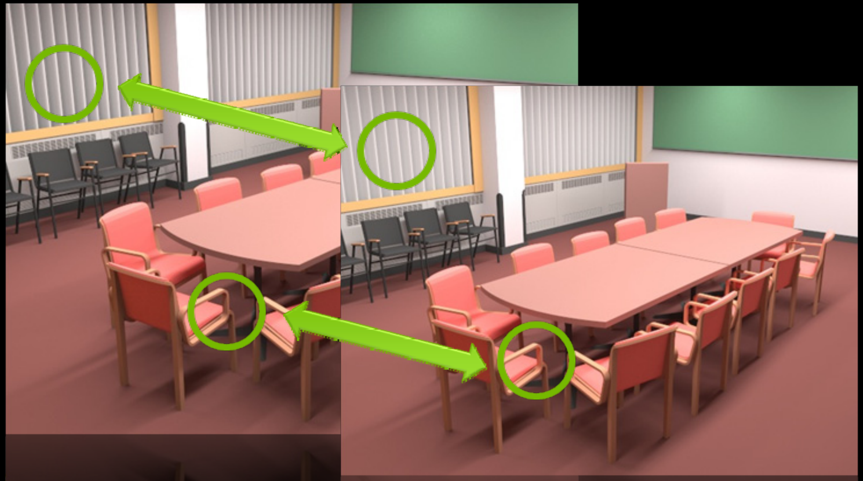
- encompass expressive power of current and future shading languages
    - one future-proof asset compiles to any other shading language
    - platform independent look development
    - will be incorporated in open 3D standards
    - adopted and backed by NVIDIA
    - will be broadly supported by industry
- visual programming via **mental mill®**
    - shader creation made easy for artists
    - enormously increased productivity
        - hear Laura Scholl talk on cross-platform shader programming



Future Rendering Technology

Looking Ahead: Consistent Predictive Rendering

- consistent: it will converge
- predictive: you can rely on physical correctness

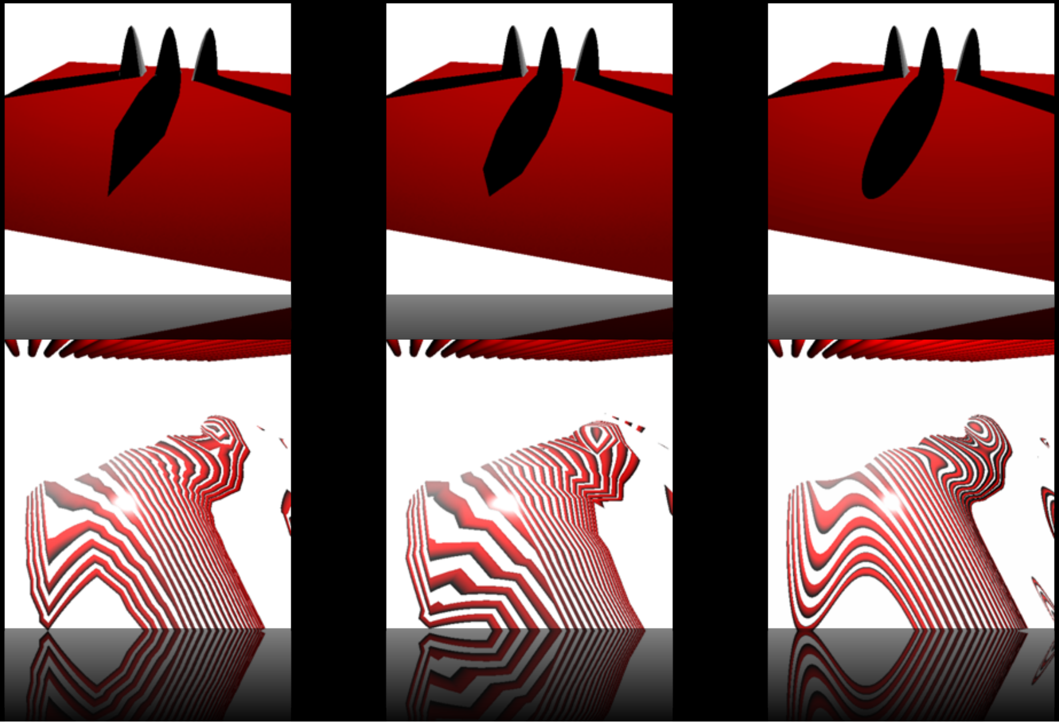


- full spectral rendering including measurement and color management

Future Rendering Technology

Looking Ahead: There is more than Polygons

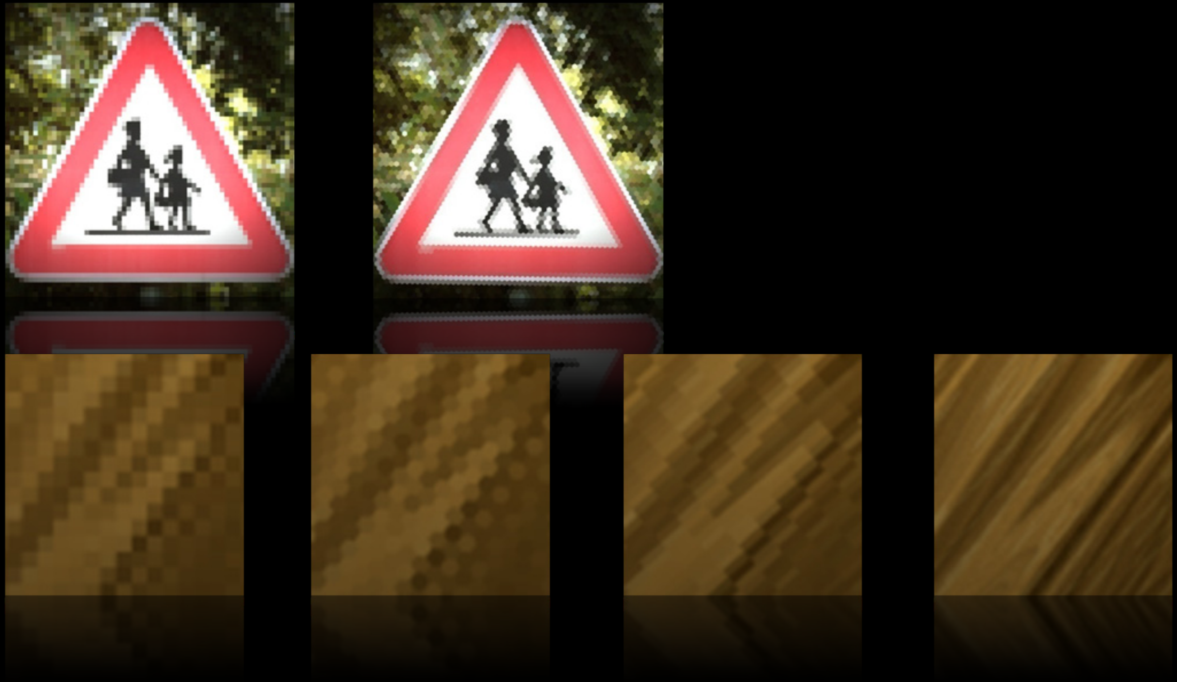
- going up to floating point precision



Future Rendering Technology

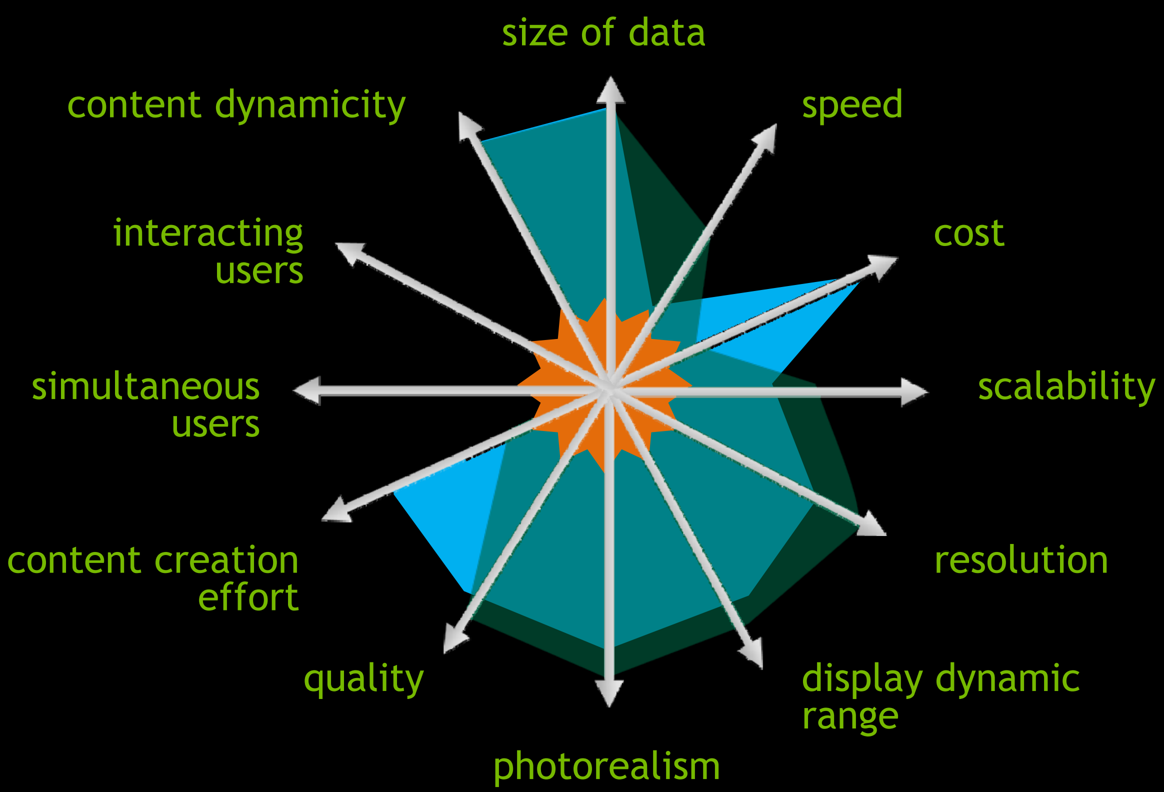
Looking Ahead: Better Textures

- images and textures on (anisotropic) rank-1 lattices



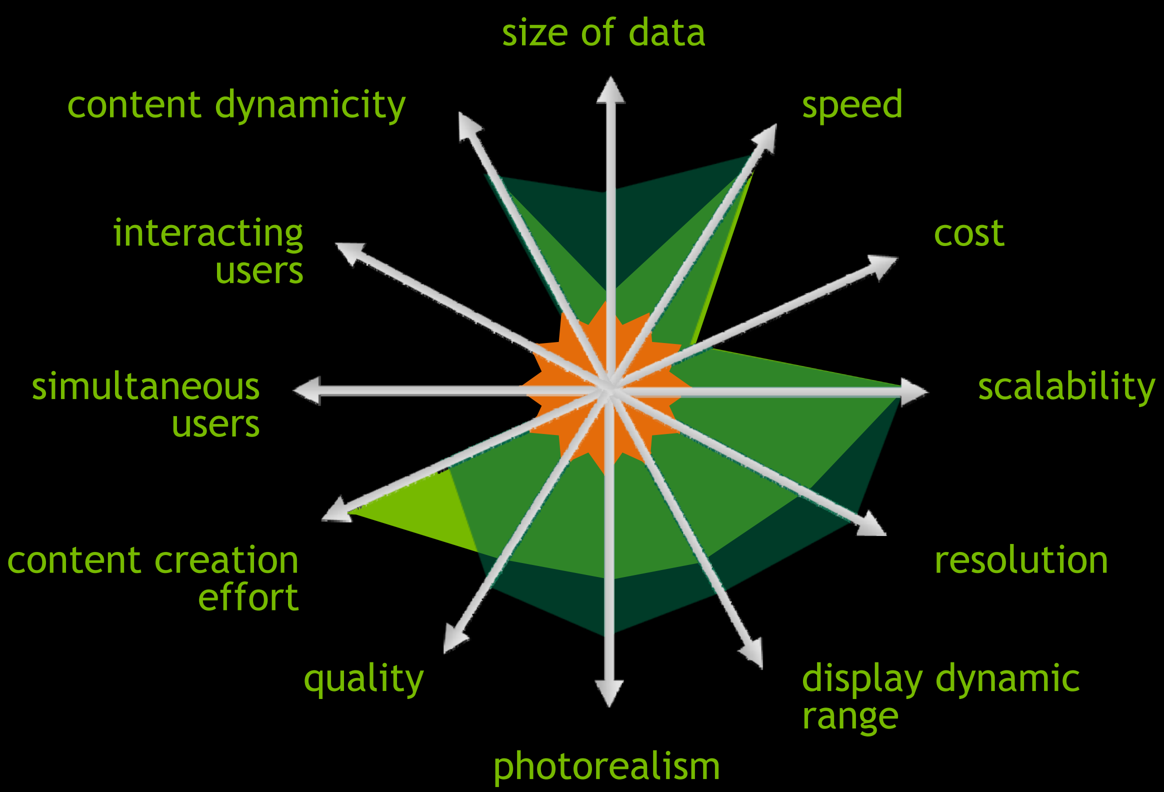
Future Rendering Technology

Next Generation versus Today's Offline Ray Tracing



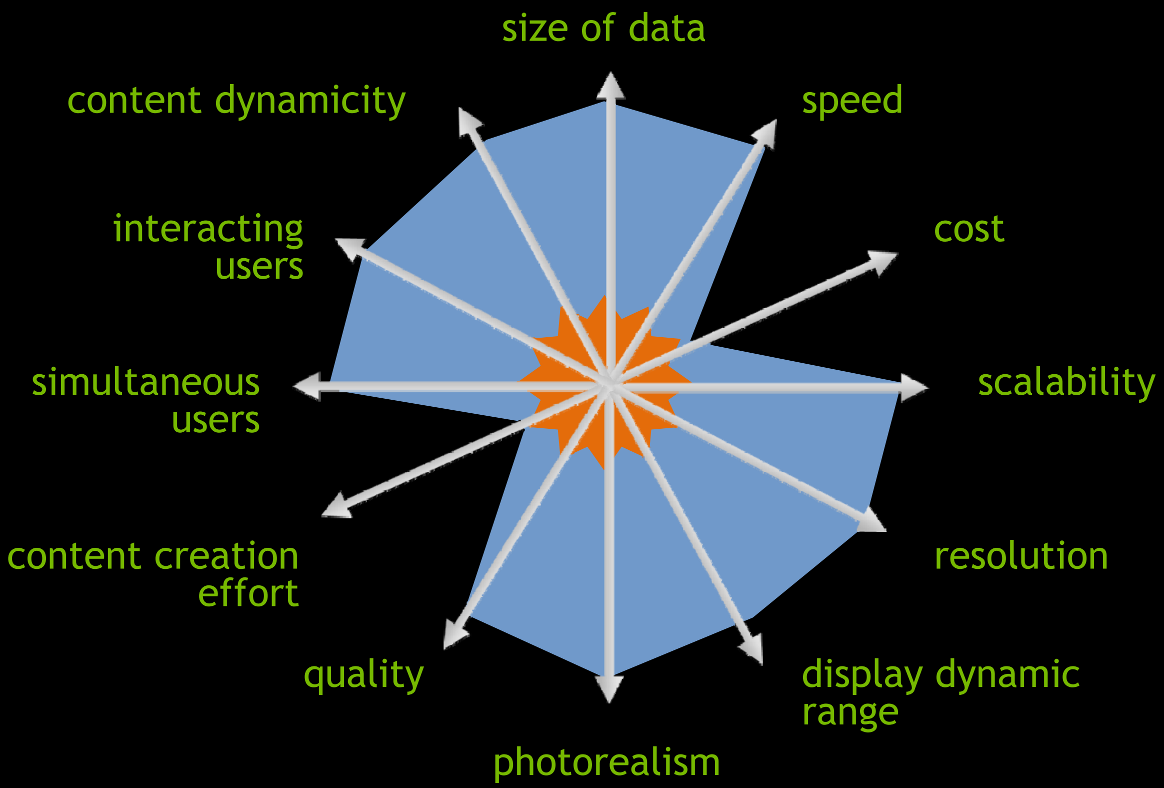
Future Rendering Technology

Next Generation versus Today's Real-time Rasterization



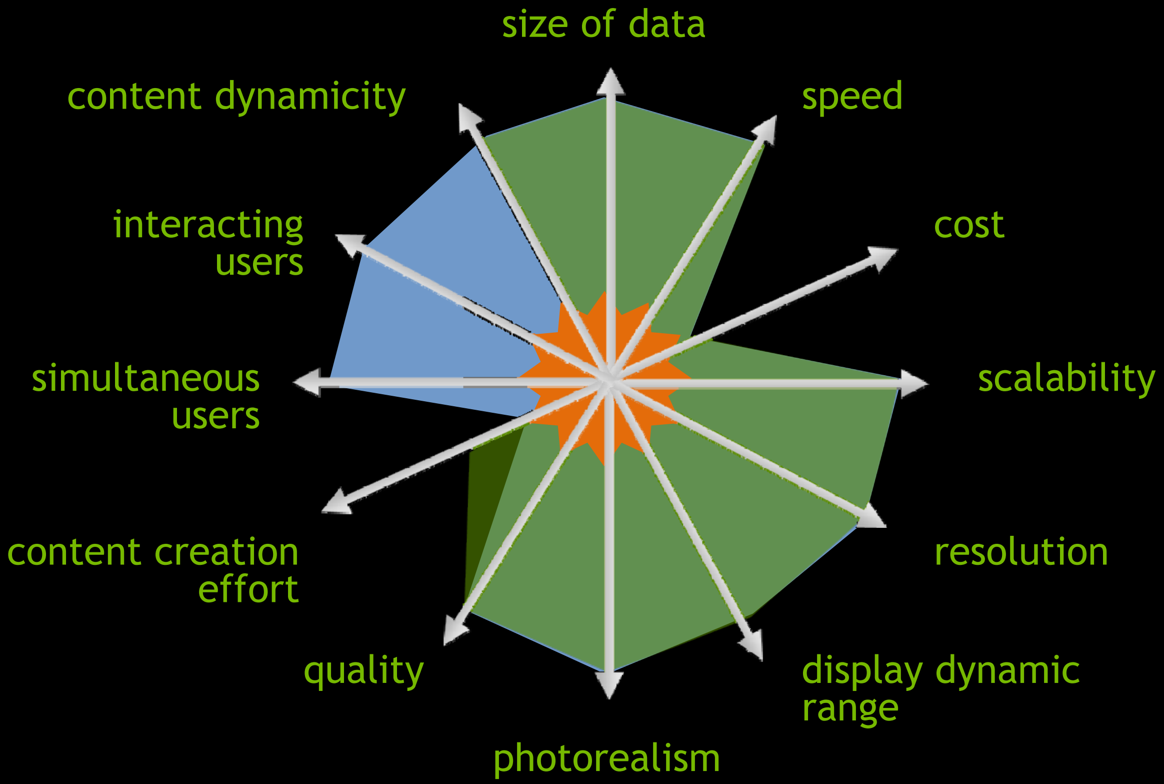
Future Rendering Technology

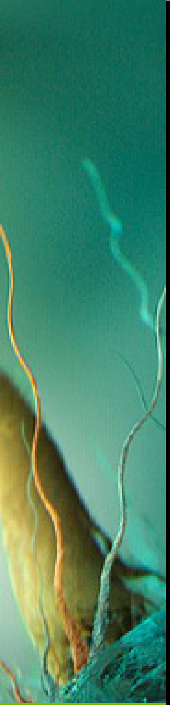
The Ultimate Platform: Multi-user, Collaborative, Ubiquitous



Future Rendering Technology

Next Generation versus The Ultimate Platform

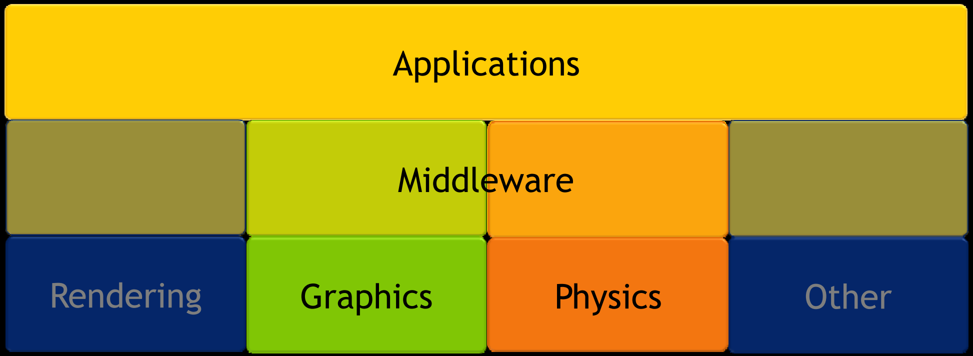




The Future of Rendering

Integration of Simulation and Visualization

- visual computing middleware becomes the foundation of application development



Applications

Middleware

Rendering

Graphics

Physics

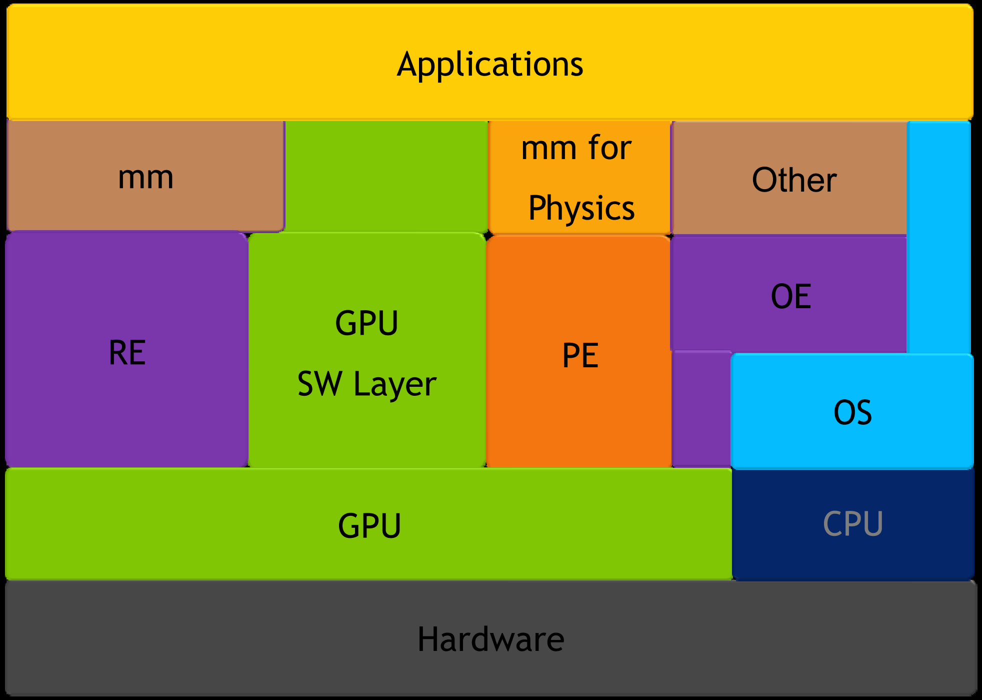
Other

- convergence of image synthesis and physics simulation
  - collision detection without auxiliary acceleration data structure

The Future of Rendering

Visual Computing Middleware

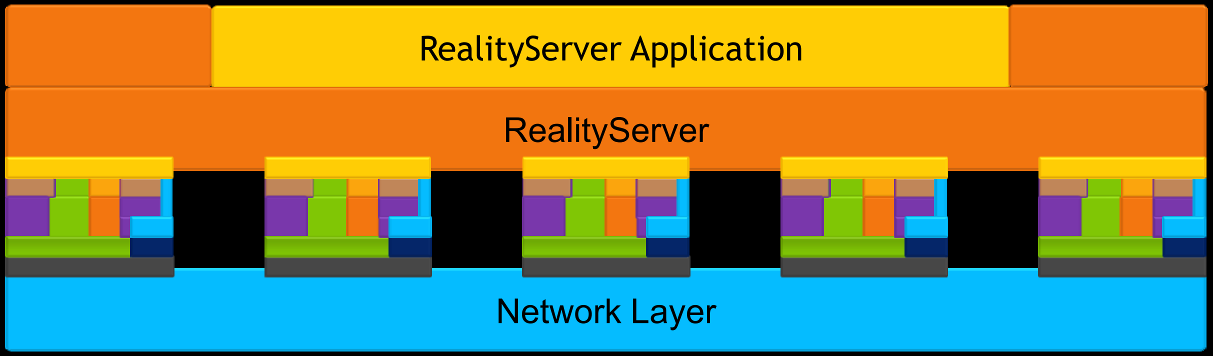
- data management
  - scene graph
- simulation
  - rendering (simulation of physical appearance)
  - physics (dynamics)
  - behavior (AI)
  - character animation (motion synthesis)
  - organic shape growth
  - etc.
- math libraries



The Future of Rendering

Reality Server®

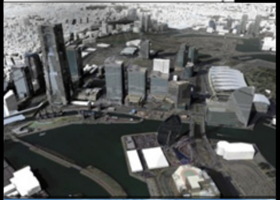
- first visual computing meta middleware
- failsafe distributed compute environment
- server-based scalable platform for 3D web applications

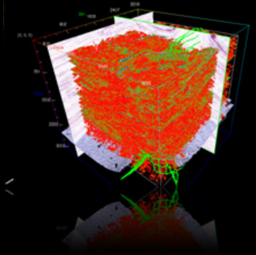


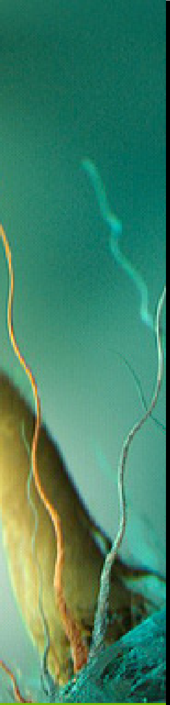
The Future of Rendering

Reality Server®

- server-based scalable platform for 3D web applications
  - interact with arbitrarily complex content and collaborate
  - stream images from everywhere to any browser
  - develop failsafe and distributed applications
  - optimize all dimensions of rendering
- based on mental images rendering technology portfolio
- realize the applications you **nvision** ...
  - hear Ludwig von Reiche talk on Reality Server



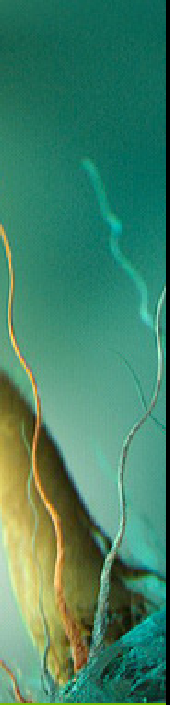




The Future of Rendering

A Short History of the Future of Rendering

- there was more and more ray tracing
- the GPU provided the compute power needed
- ray tracing and rasterization had converged
- all rendering became (real-time) interactive
- rendering became more precise
- rendering became distributed
- rendering became available on the web clouds
- NVIDIA GPUs, CUDA, MetaSL, mental ray, neuray, iray, and RealityServer got us there



The Future of Rendering

The Future of Rendering

August 25, 2008

Rolf Herken
CEO and CTO
mental images®

www.mentalimages.com

office@mentalimages.com

mental images GmbH
Fasanenstrasse 81
10623 Berlin
Germany

mental images, Inc.
One Embarcadero Center, Suite 500
San Francisco, CA 94111
USA

mental images Pty. Ltd.
1/140 Bourke Street
Melbourne, Victoria 3000
Australia