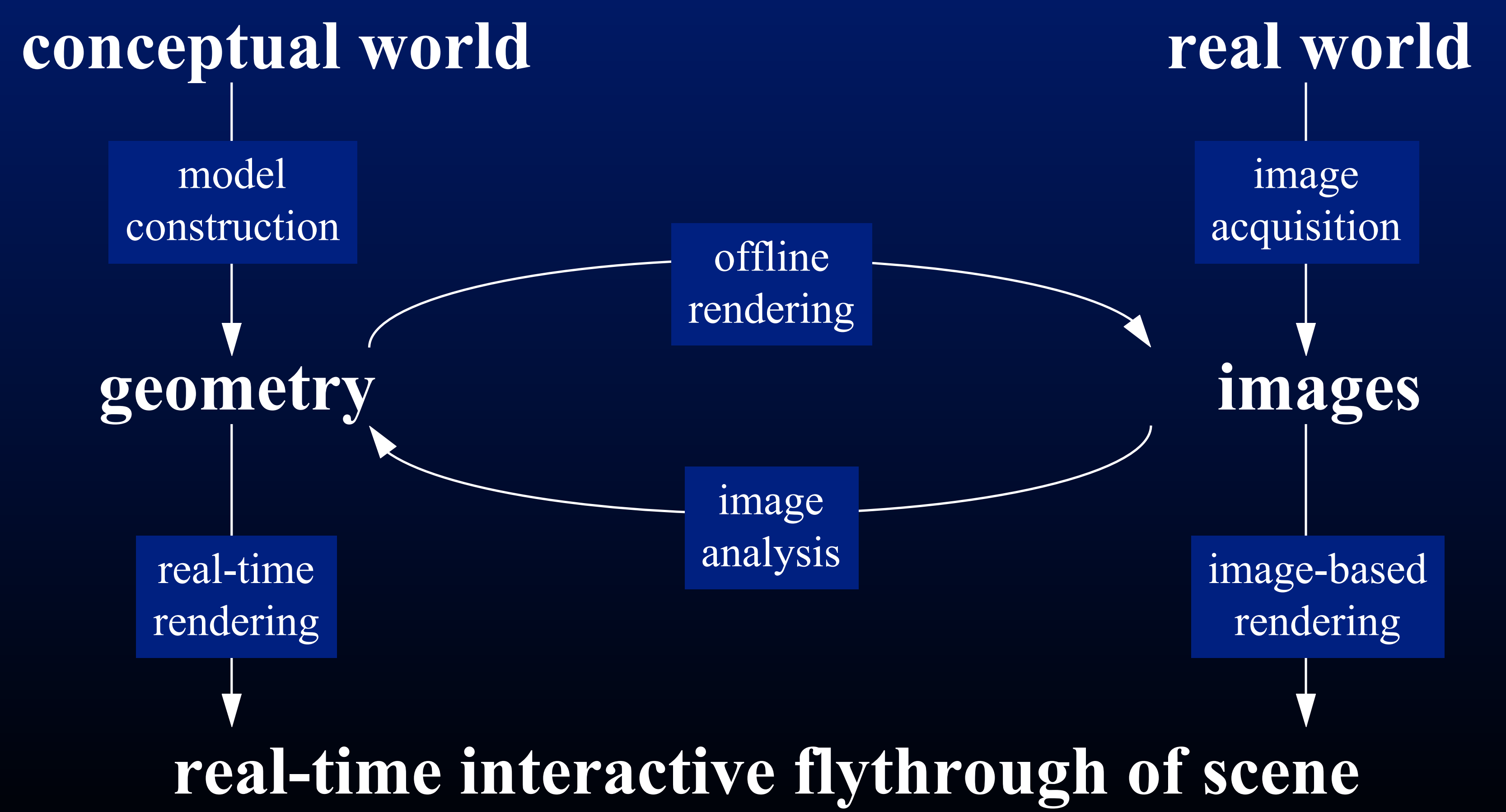
Expanding the horizons of image-based modeling and rendering

Marc Levoy

Computer Science Department
Stanford University

Geometry-based versus image-based rendering



IBR: a definition

*"The study of image-based modeling
and rendering is the study of
sampled representations of geometry."*

Sampled representations of geometry

- classic representations
  - textures, sprites, shadow maps, reflectance maps, range images, disparity maps, optic flow fields, panoramas, movie maps, density volumes, epipolar volumes, etc.
- recent additions
  - arrays of images, light fields / Lumigraphs, inconsistent arrays of range images, hierarchical image caches, etc.

So what's new?

increasing memory and CPU speeds

$\Downarrow$

greater dimensionality and size

$\Downarrow$

algorithms that operate without geometry

Where are the boundaries of this paradigm?

1. What other sample representations are possible?
2. What operations are possible on image-based representations?
3. What new applications exist for these techniques?

Game #1: increase the dimensionality

2D rgb	texture
2D rgbz	range image
2.5D rgbαz	layered depth images
4D rgb	light field / Lumigraph
4D rgbz	array of range images
4.5D rgbαz	layered light fields

Game #2: replace the quantity represented

4D rgb — light field / Lumigraph

$\{u, v, s, t\}$

5D rgb — plenoptic function

$\{x, y, z\} \times \{\theta, \varphi\}$

6D ρ — free-space BRDF field

$\{u, v, s, t\} \times \{\theta_i, \varphi_i\}$

7D ρ — BRDF volume

$\{x, y, z\} \times \{\theta_i, \varphi_i, \theta_r, \varphi_r\}$

Other quantities

- opacity
- importance
- vector irradiance
- geometric transformations

Game #3: make a taxonomy

- Operation: changing the viewpoint

	reflectance	radiance
geometry-based	classic ray tracing (Whitted, 1980)	ray tracing + radiance interpolation (Teller et al., 1996)
image-based	**?** (requires 4.5D ρ z)	light field / Lumigraph (Levoy & Hanrahan, 1996; Gortler et al., 1996)

Other operations

- Changing the illumination
- Changing the reflectance
- Changing the shape

forward methods

inverse methods

more info

less info

output

input

	obj refl	obj rad	LF+z refl	LF+z rad	LF refl	LF rad	im+z refl	im+z rad	im refl	im rad
obj refl	3D paint	lighting	LF build	0	LF build	0	render	0	0	0
obj rad	inverse lighting?	1	0	LF build	0	LF build	0	render	0	0
LF +z refl	occup- ancy grid	0	LF zpaint?	LF lighting?	drop z	0	LF view	0	0	0
LF +z rad	0	occup- ancy grid	LF inverse lighting?	1	0	drop z	0	LF view	0	0
LF refl	shape from LF ?	0	find z from LF ?	0	LF paint?	0	photo- metric stereo	0	LF view	0
LF rad	0	shape from LF ?	0	find z from LF ?	pick refl	1	0	0	0	LF view
im +z refl	0	0	0	0	0	0	zpaint	0	drop z	0
im +z rad	0	0	0	0	0	0	pick refl	1	0	drop z
im refl	0	0	0	0	0	0	shape from shading	0	paint	
im rad	0	0	0	0	0	0	0	0	pick refl	1

Will IBR take over the world?

"In 10 years, all rendering
will be volume rendering."

- Jim Kajiya, 1991

- Yes
  - simple, brute force
  - universal meta-primitive
  - geometry is implicit
  - independent of scene complexity
  - multiresolution is trivial
  - hardware acceleration
- No
  - discards structure of model
  - rasterized geometry
  - big and slow