Real-time High Quality HDR Illumination and Tonemapped Rendering

Despina Michael
15th of July, 2010

PhD Thesis Defense

Supervisor: Yiorgos Chrysanthou

Computer Science Department
University of Cyprus

Contents

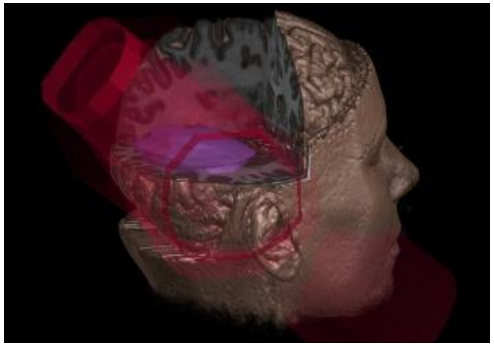
- Introduction
- Illumination
  - Problem statement
  - Approach
  - Proposed algorithm
  - Results
- Tonemapping
  - Problem statement
  - Approach
  - Proposed algorithm
  - Results
- Conclusions & Future Work

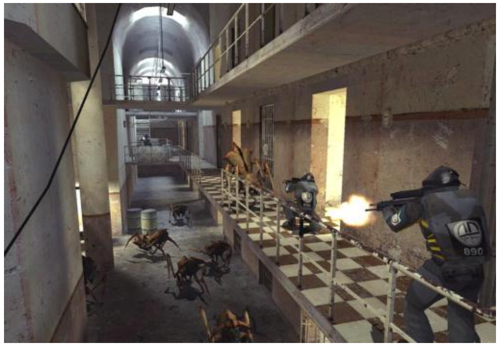
introduction

Computer graphics (CG)

- What is CG?
  It is the field where one utilizes digital data to generate images

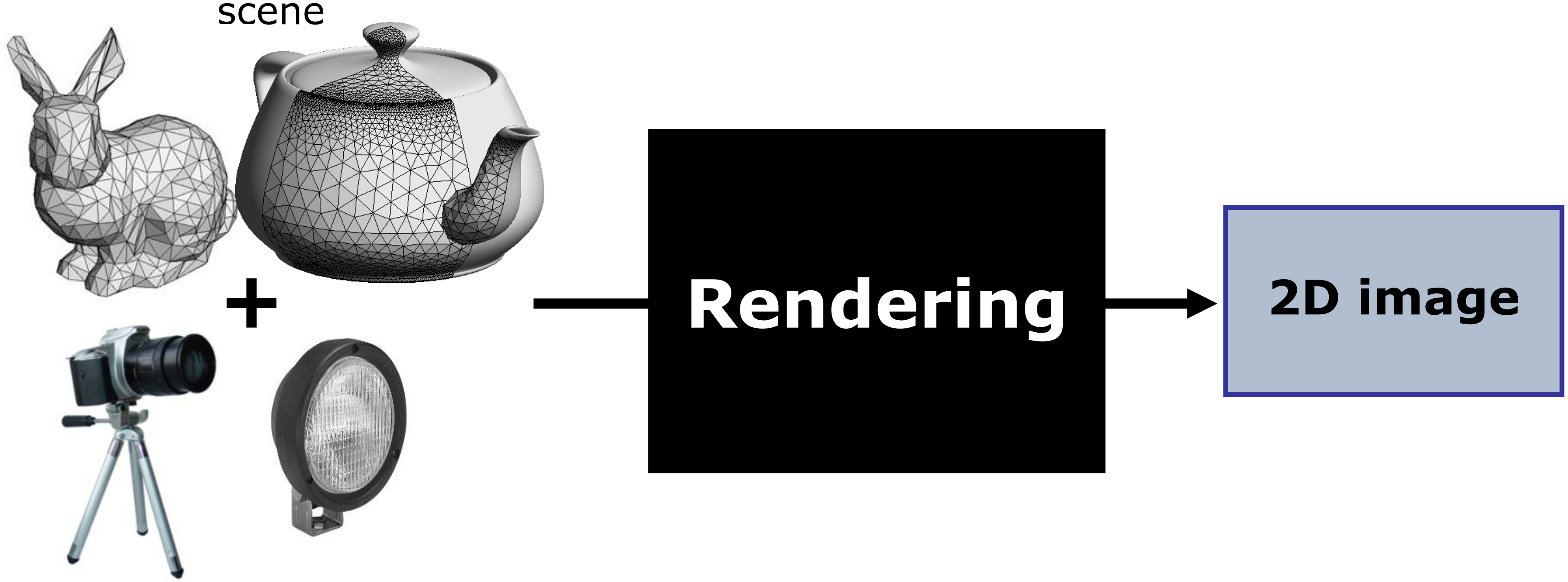
- Applications
  - Training - Simulators
    - military
    - medical
    - aviation
  - Entertainment
    - movies
    - computer games
  - Virtual Worlds
    - virtual museums
    - virtual cities
  - Etc

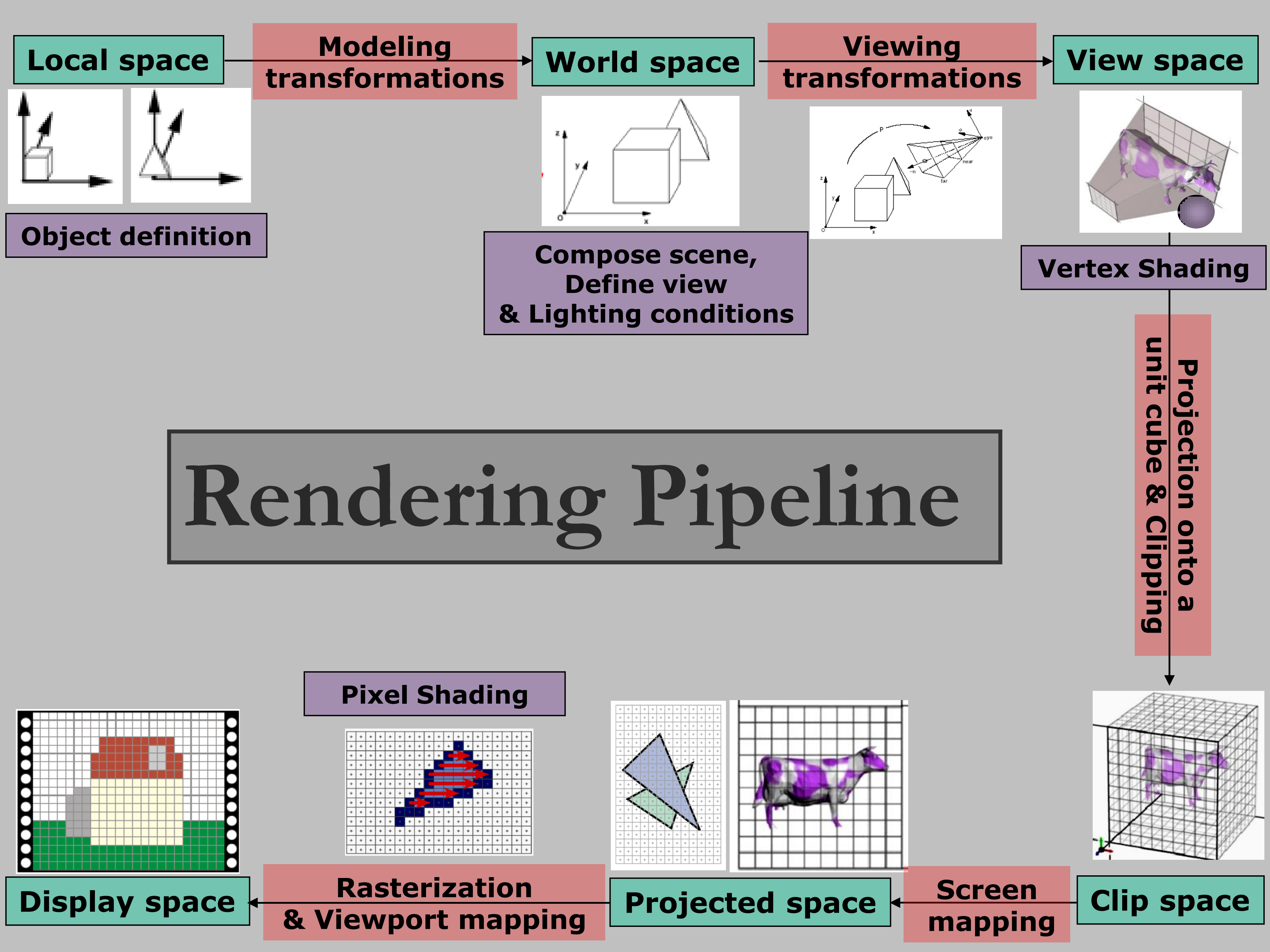





Computer graphics (CG)

- 3D Computer Graphics
  - Geometric modeling
    - the representation of the shapes of objects in the scene
  - Animation
    - the motion and deformation of the objects in the scene
  - Rendering
    - the process of producing 2D images from the description of the 3D scene

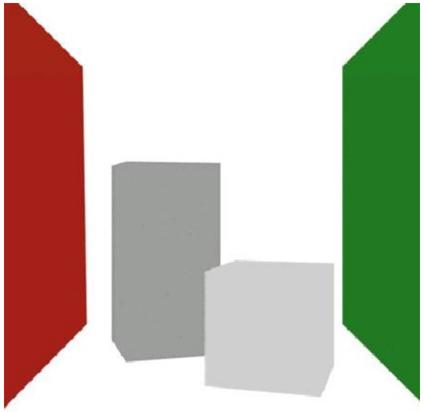



Local space
Modeling transformations
World space
Viewing transformations
View space
Object definition
Compose scene, Define view & Lighting conditions
Vertex Shading
Projection onto a unit cube & Clipping
Rendering Pipeline
Pixel Shading
Display space
Rasterization & Viewport mapping
Projected space
Screen mapping
Clip space

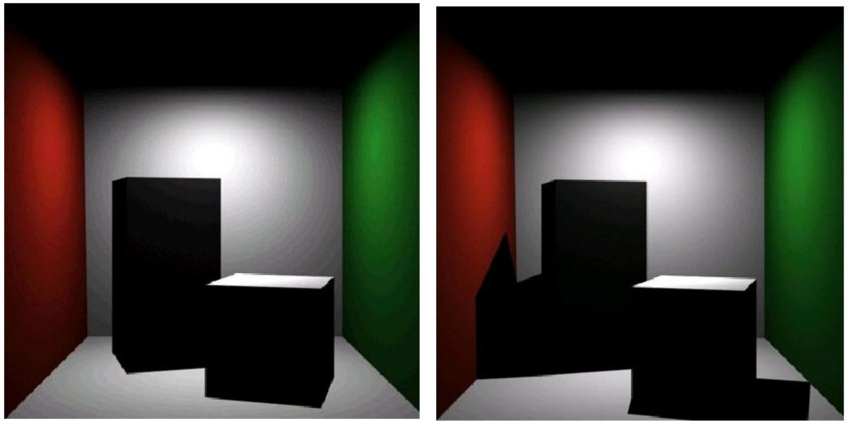
Importance of illumination

- Illumination in real world:
  - Light stimulates sensors to see
  - Without light human eye can not see
- Illumination in Computer Graphics:
  - Provides spatial and geometric information
  - Increases realism in the images

measured

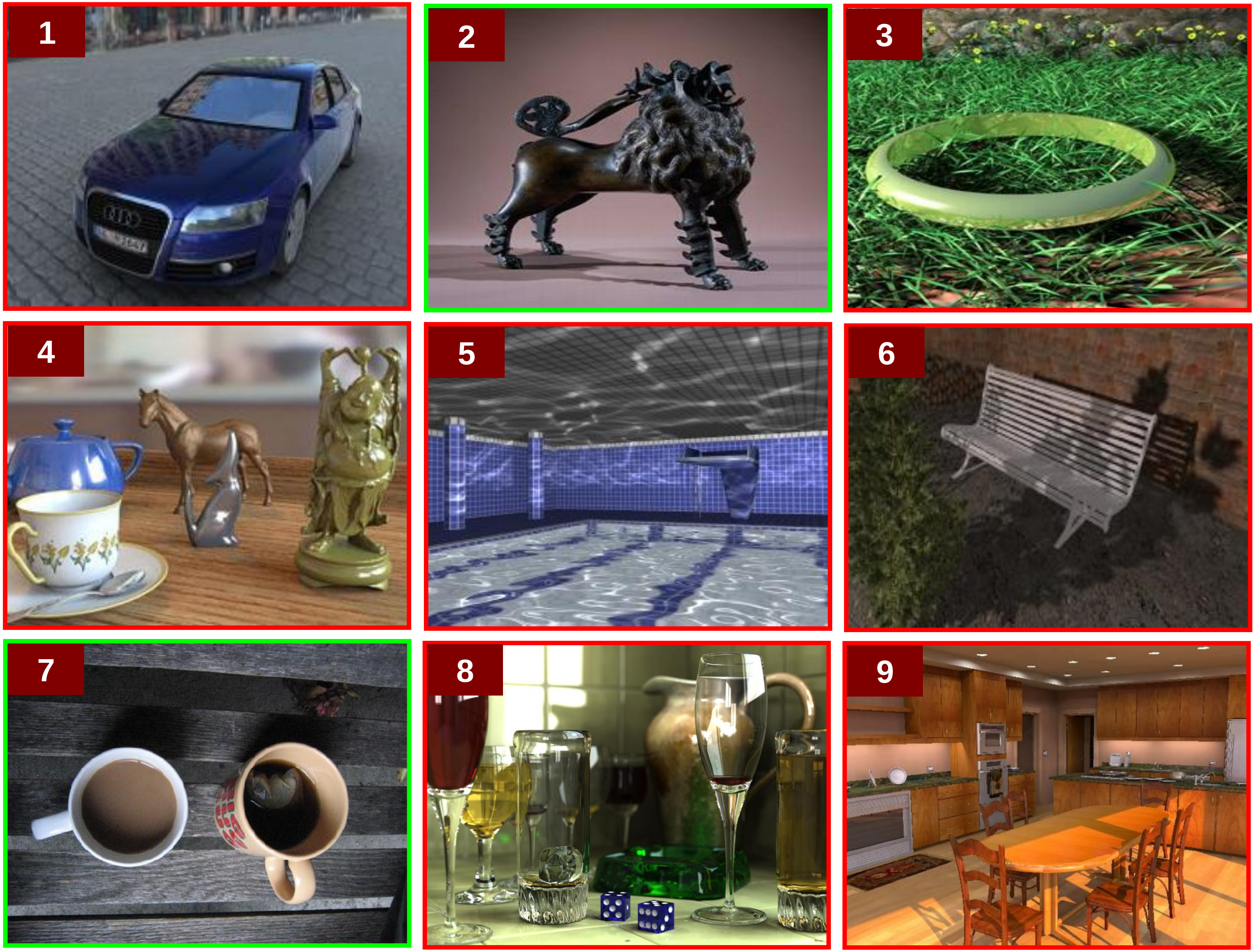


no illumination

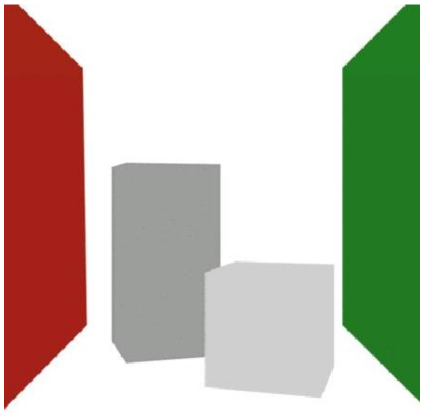
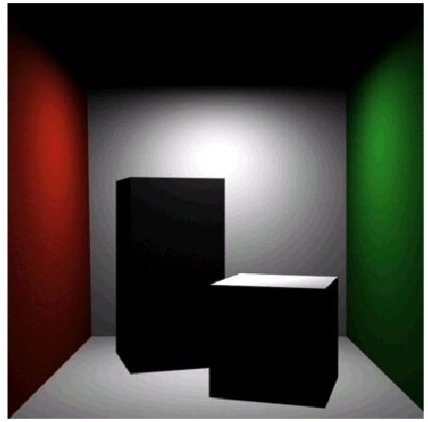
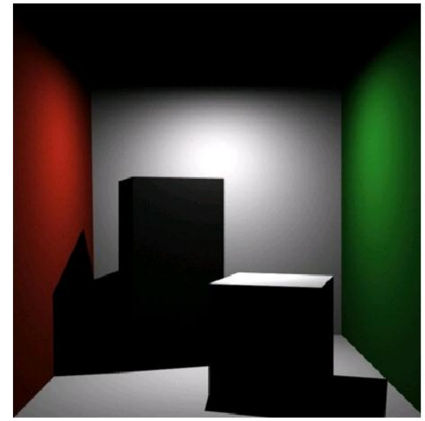


spatial and geometric information

simulated

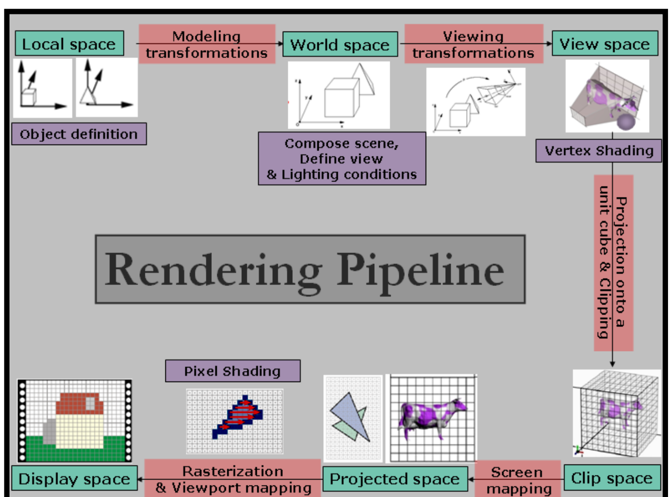

1
2
3
4
5
6
7
8
9

Motivation


- Improve rendering for **real-time** applications

Contribution

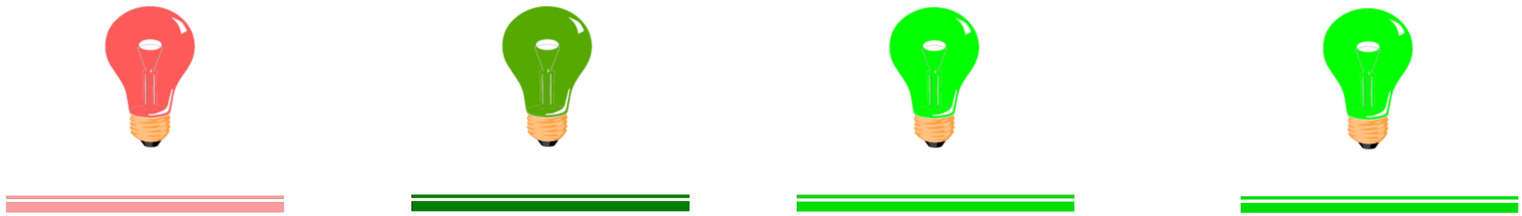


- Illumination
- Tonemapping

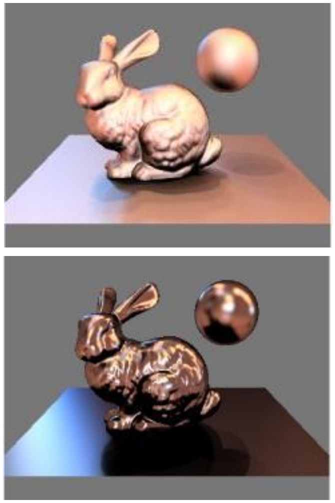
illumination

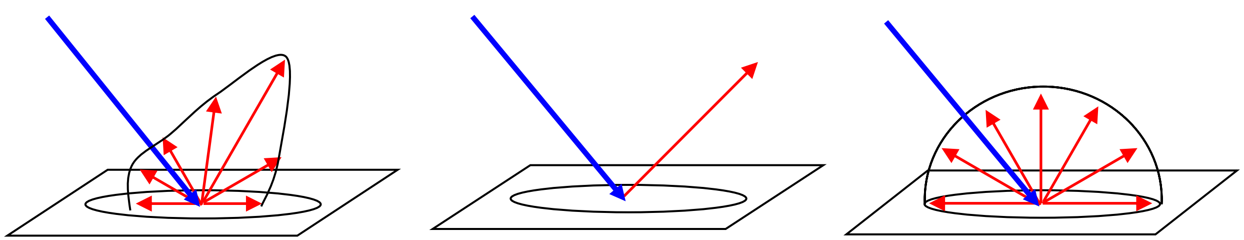
Parameters affecting illumination

- illumination at each point of an object depends on:
  - Incident light energy



  - Reflectance properties of the object material
    - BRDF: Bidirectional Reflectance Distribution Function
      defines how light coming from a given incoming direction is reflected towards an outgoing direction





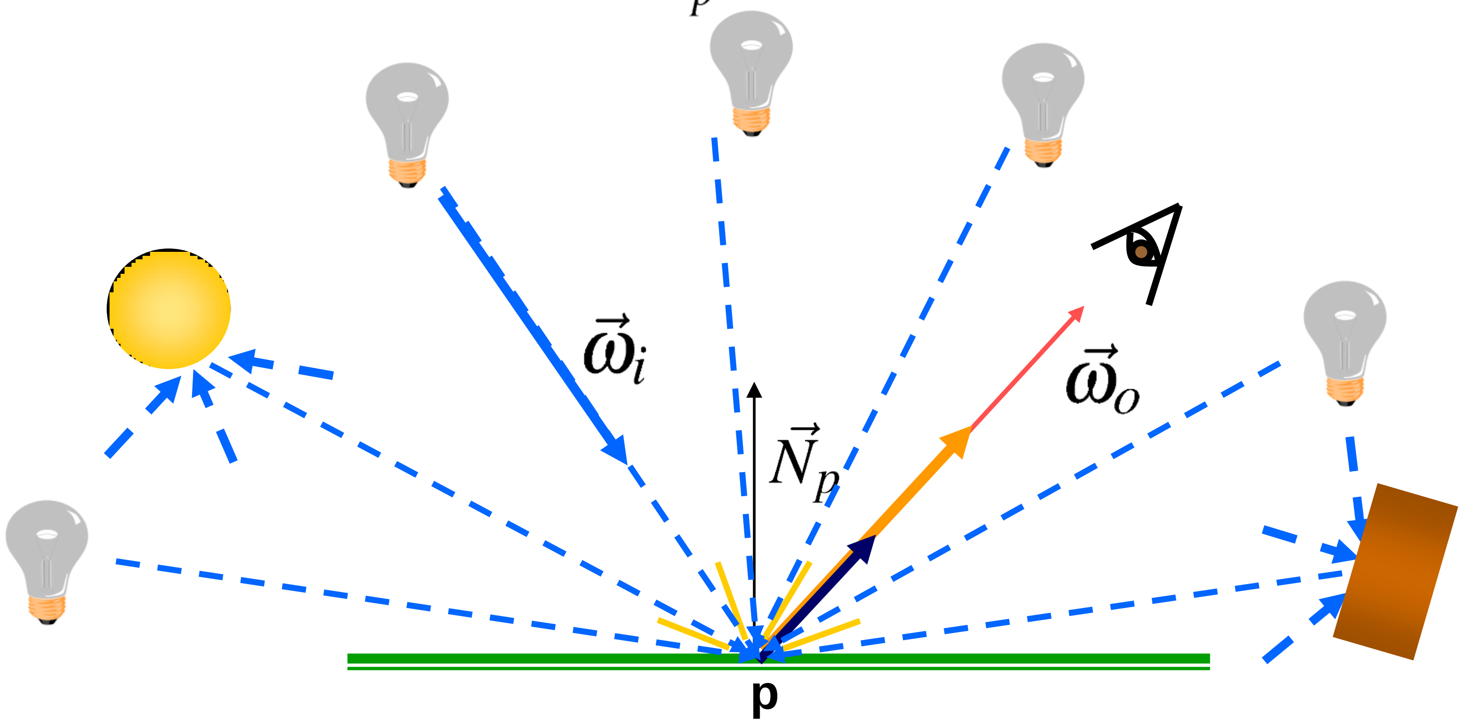
Rendering equation

Outgoing Light = Emitted Light + Reflected Light

BRDF x Incident Light

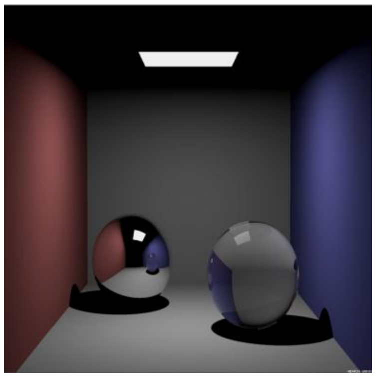
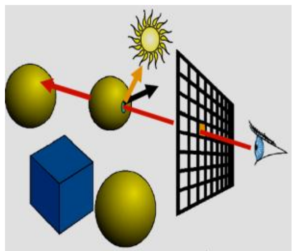
$$L_{out,p}(\vec{\omega}_o) = L_{emit,p}(\vec{\omega}_o) + \int_{\Omega^+_{\vec{N}_p}} f_p(\vec{\omega}_i, \vec{\omega}_o) \cdot L_{in,p}(\vec{\omega}_i) \cdot \cos<\vec{N}_p, \vec{\omega}_i> d\vec{\omega}_i$$

Radiance

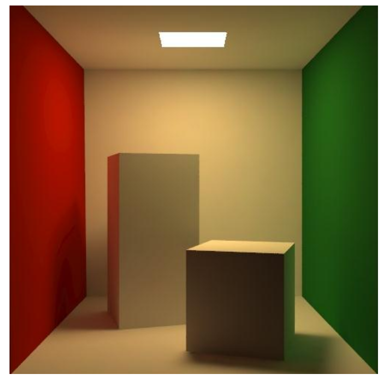
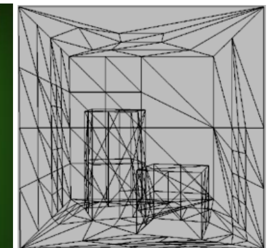


Solving the rendering equation

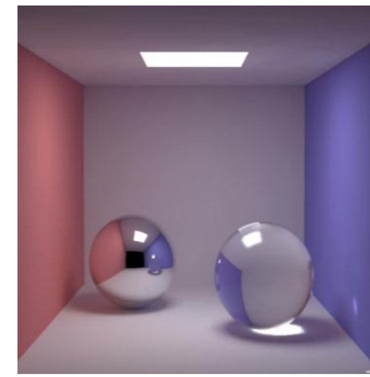
- Computational expensive process
- Emphasis on realism - global illumination (offline methods)

Raytracing

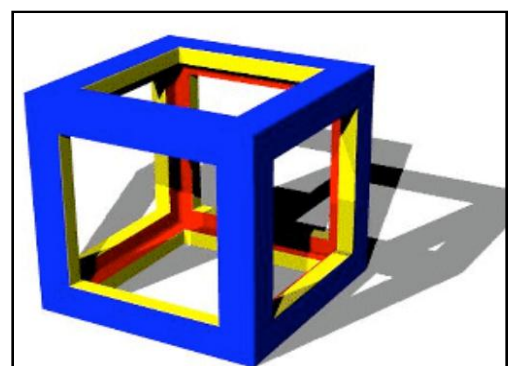
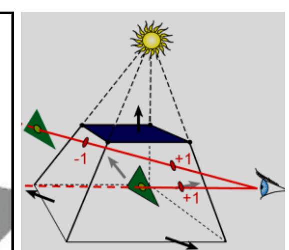
Radiosity



Photon mapping

- Emphasis on frame rate - local illumination

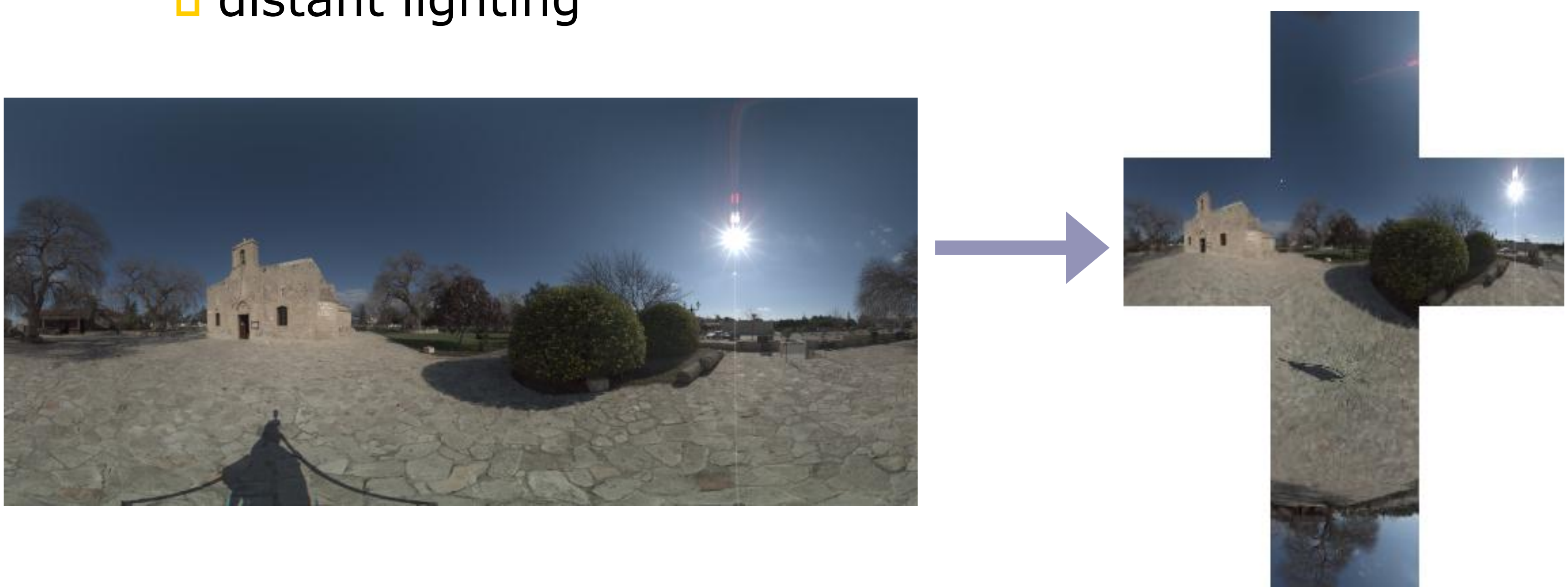

Shadow maps

Shadow volumes

Illumination
Problem statement

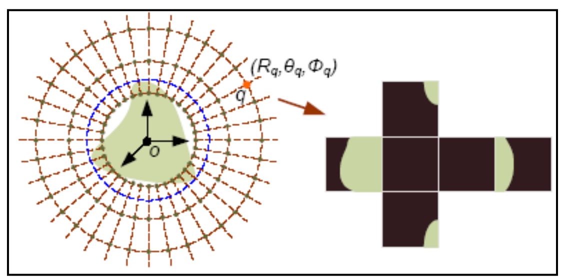
- Towards real-time realism
  - Computation of direct lighting only
  - Taking into account incident radiance from all over the directions
    - represented using environment maps
    - distant lighting



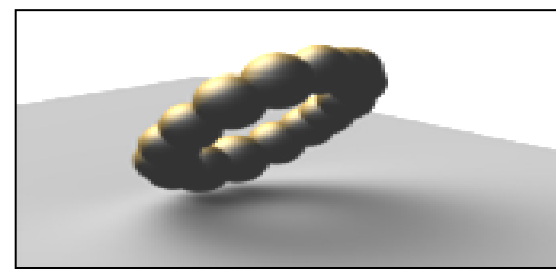
Illumination
Problem statement

- Huge number of incident light directions (e.g 10^3)
  - Run-time computation of $\int$ → not real-time performance
  - Precomputations → with what cost?

PRT [Sloan]
at each vertex
static scenes + memory



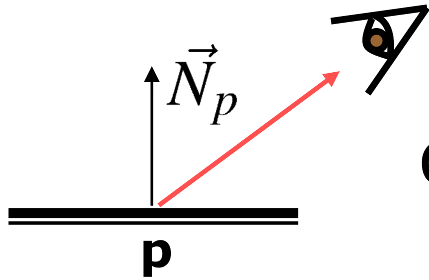
Precomputed shadow fields [Zhou]
rigid objects + memory



Real-time soft shadows [Ren]
geometry approx.

Proposed illumination algorithm
Approach

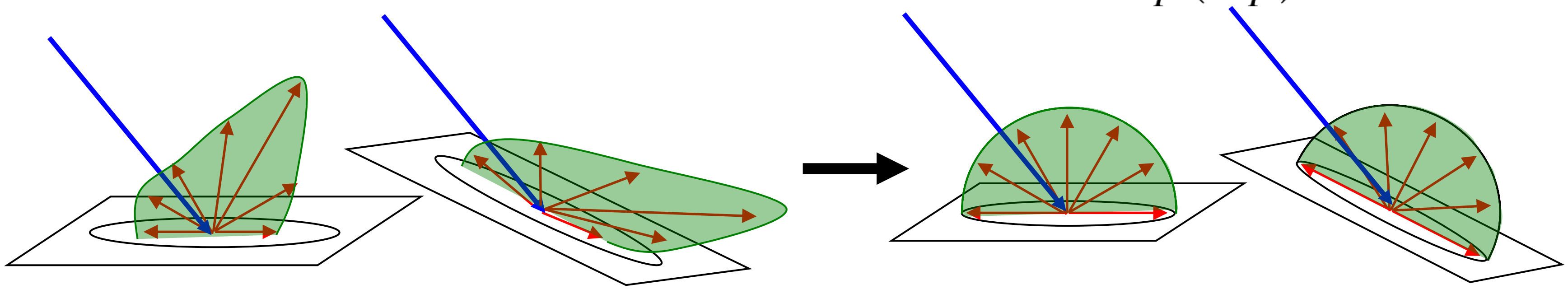
- Precompute the $\int$ in a way that is independent of the geometry of the scene



depends on $\vec{N}_p$

$$L_{out,p}(\vec{\omega}_o) = \sum_{\vec{\omega}_i \in \Omega^+_{\vec{N}_p}} \rho_{dif} \cdot \underbrace{L_{in,p}(\vec{\omega}_i) \cdot \cos\langle\vec{N}_p, \vec{\omega}_i\rangle}$$

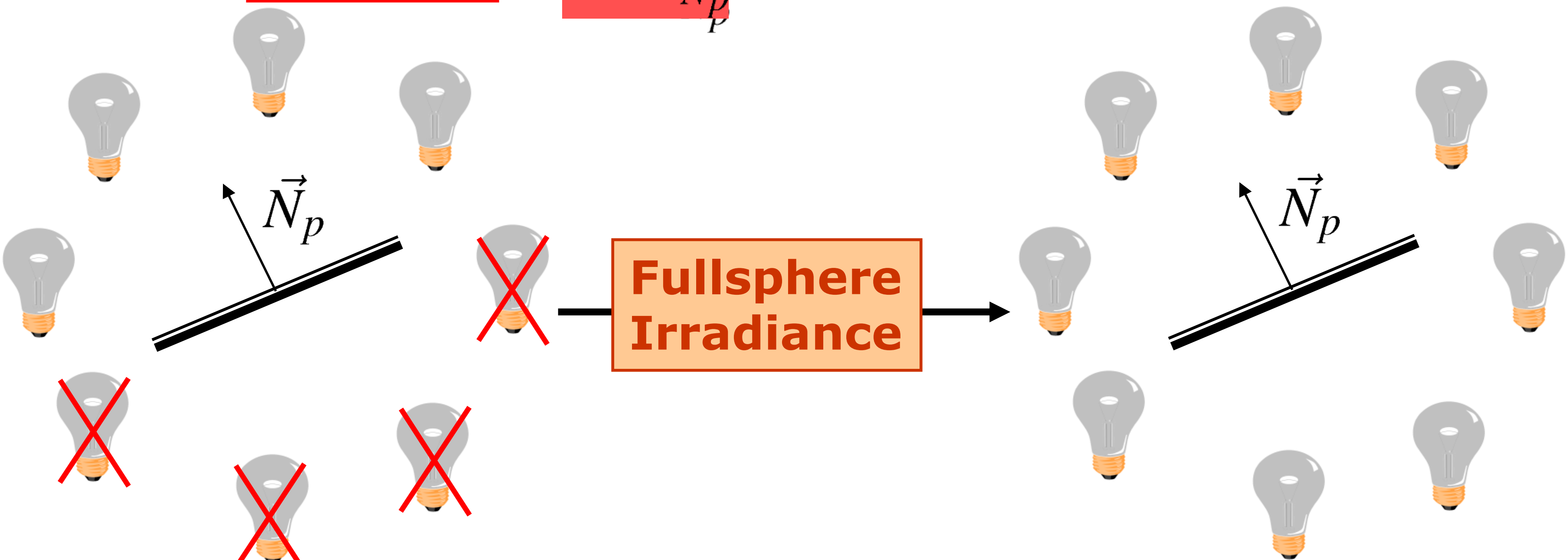
Irradiance $I_p(\vec{N}_p)$

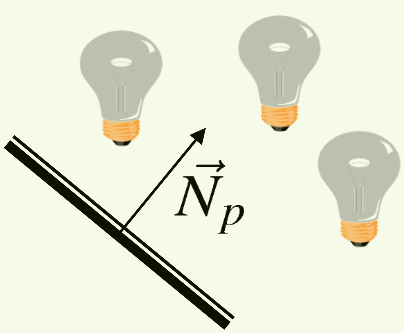


Proposed illumination algorithm
Approach

$$FI_p(\vec{N}_p) = \sum_{\vec{\omega}_i \in \Omega_{\vec{N}_p}} L_{in,p}(\vec{\omega}_i) \cdot \cos\langle \vec{N}_p, \vec{\omega}_i \rangle$$

Irradiance





$$\Omega_{\vec{N}_p} = \Omega^{+}_{\vec{N}_p}$$

$$FI_p(\vec{N}_p) = I_p(\vec{N}_p)$$

Proposed illumination algorithm
Approach

$$FI_p(\vec{N}_p) = \sum_{\vec{\omega}_i \in \Omega_{\vec{N}_p}} L_{in,p}(\vec{\omega}_i) \cdot \cos\langle \vec{N}_p, \vec{\omega}_i \rangle$$

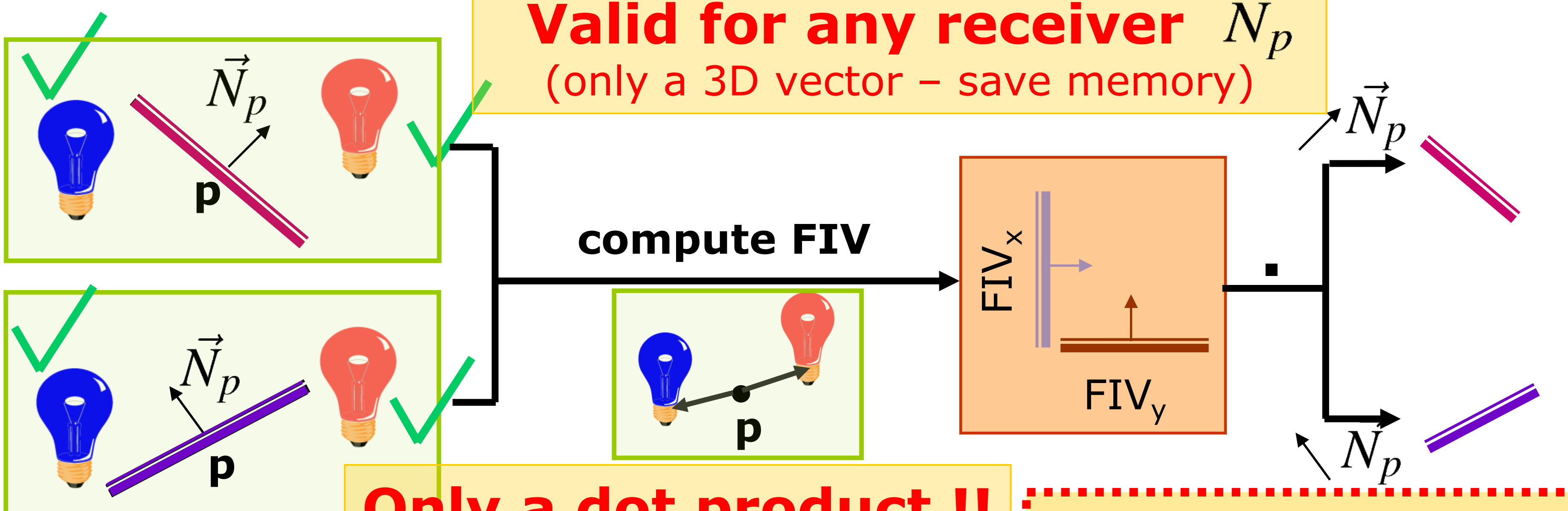
FI Factorization

$$FI_p(\vec{N}_p) = \vec{N}_p \cdot \left(\sum_{\vec{\omega}_i \in \Omega_{\vec{N}_p}} (L_{in}(\vec{\omega}_i)\omega_{i_x}), \sum_{\vec{\omega}_i \in \Omega_{\vec{N}_p}} (L_{in}(\vec{\omega}_i)\omega_{i_y}), \sum_{\vec{\omega}_i \in \Omega_{\vec{N}_p}} (L_{in}(\vec{\omega}_i)\omega_{i_z}) \right)$$

Fullsphere Irradiance Vector – FIV(x,y,z)

Valid for any receiver $\vec{N}_p$

(only a 3D vector – save memory)



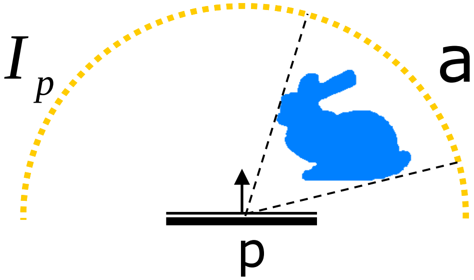
$FI_p(\vec{N}_p)$

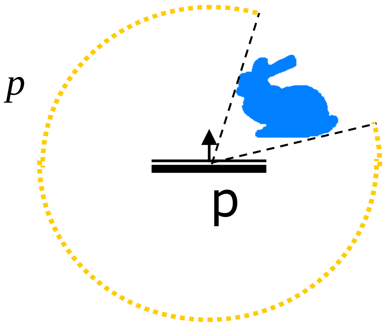
Only a dot product !!

(fast computation)

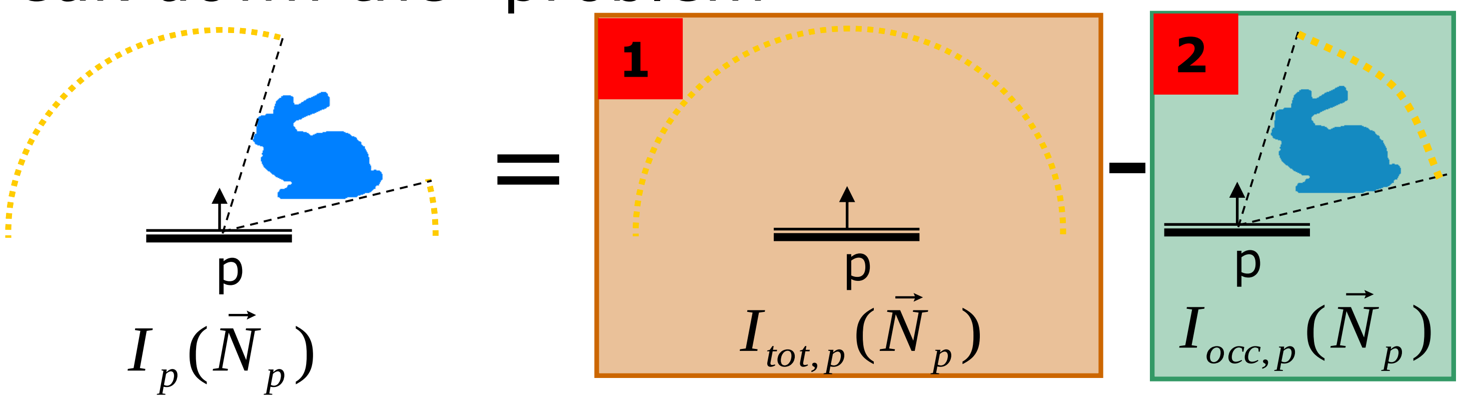
$$FIV \cdot \vec{N}_p = FI_p(\vec{N}_p)$$

Proposed illumination algorithm
Overview

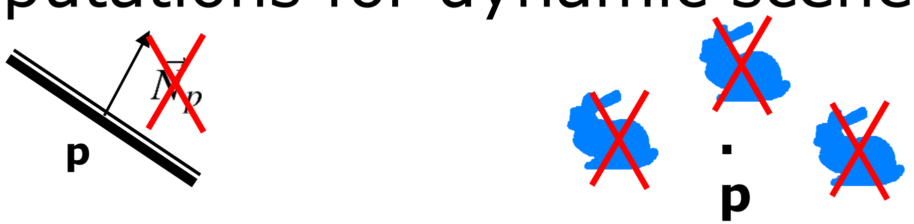
- Compute I_p 

and not FI_p 

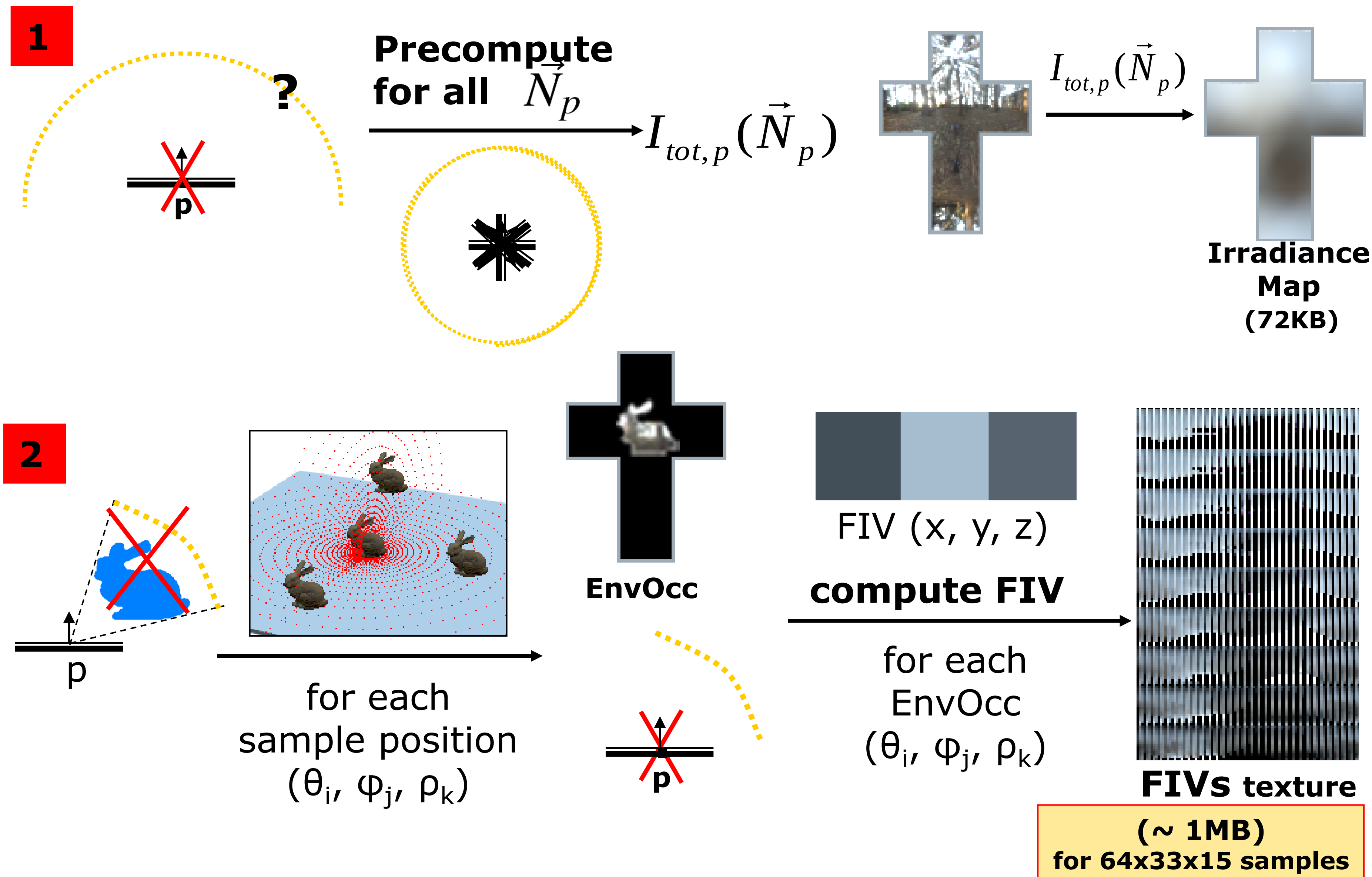
- Use FIV as an intermediate step
- Break down the "problem"



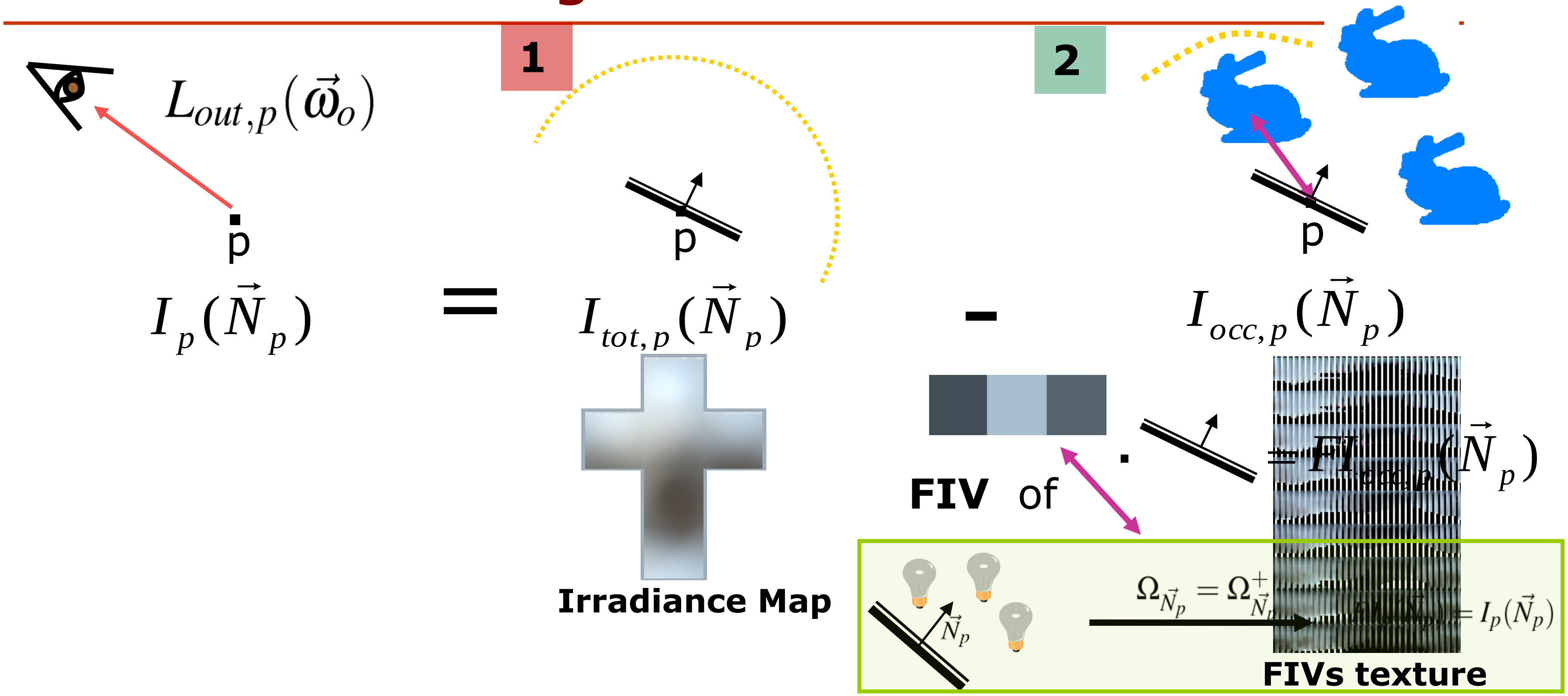
- Precomputations for dynamic scenes



Proposed illumination algorithm
Preprocessing



Proposed illumination algorithm
Runtime: shading

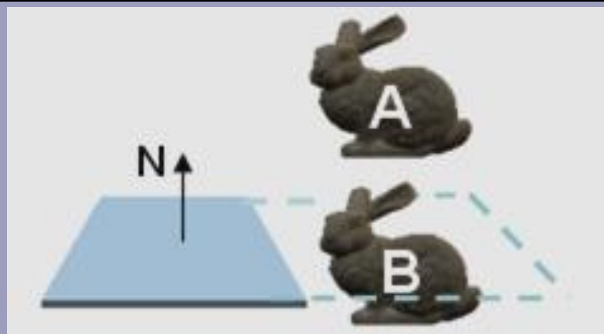


For each pixel *p* of a receiver in the image:

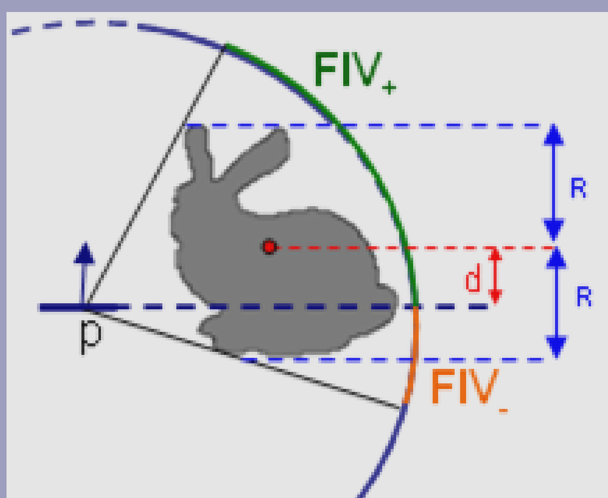
1. Get total irradiance assuming no occlusions $I_{tot,p}(\vec{N}_p)$
2. Compute occluded irradiance $I_{occ,p}(\vec{N}_p)$
3. Compute unoccluded irradiance $I_p(\vec{N}_p) = I_{tot,p}(\vec{N}_p) - I_{occ,p}(\vec{N}_p)$
4. Shade the pixel $L_{out,p}(\vec{\omega}_o) = \rho_{dif} \cdot I_p(\vec{N}_p)$

Proposed illumination algorithm
Runtime: special cases

Partial occluders

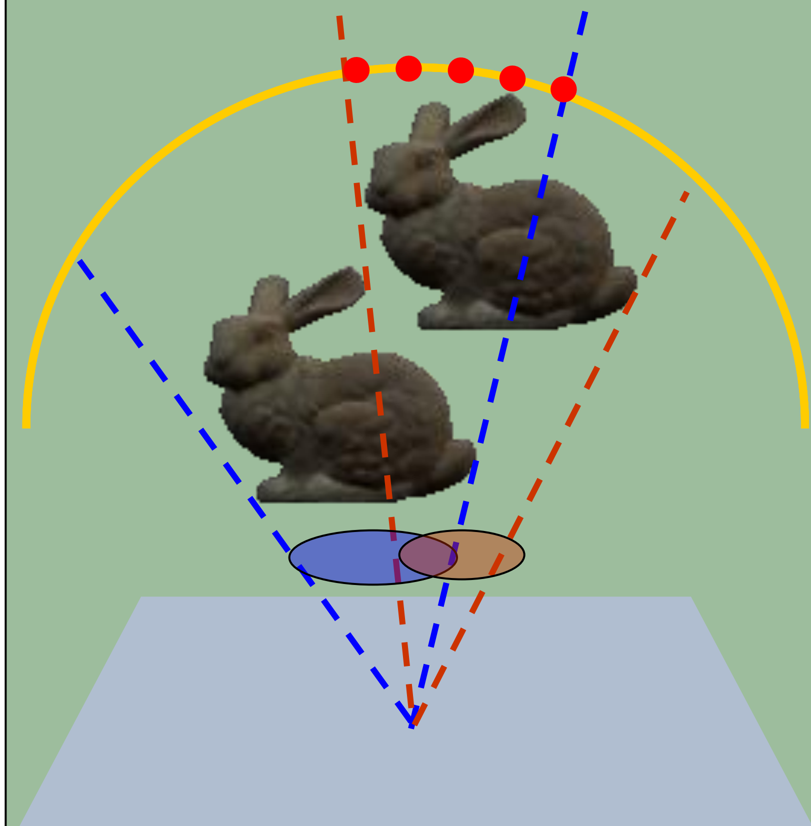


$$FI_p(\vec{N}_p) \neq I_p(\vec{N}_p)$$



Approximate FIV

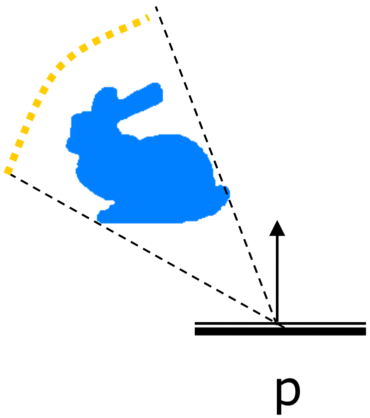
Overlapped occluders



$$I_{occ} = I_{occ_1} + I_{occ_2} - I_{overlapped}$$

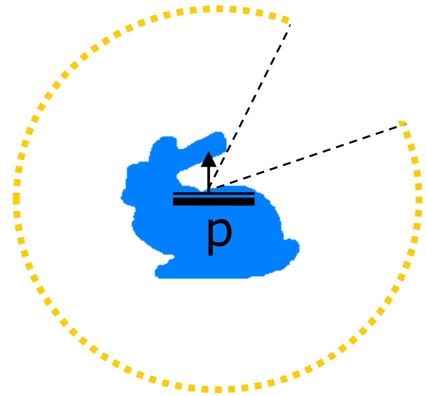
$$I_{overlapped} = Overlapped_{\%} * (I_{occ_1} + I_{occ_2})$$

Proposed illumination algorithm
Runtime: self-shadows



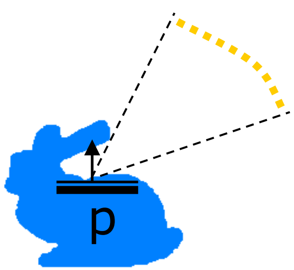
shadows

$$I_{occ,p}(\vec{N}_p) = FI_{occ,p}(\vec{N}_p)$$



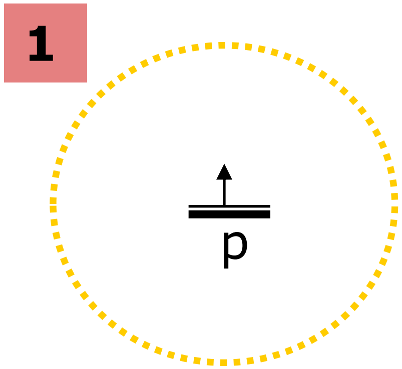
self-shadows

$$I_p(\vec{N}_p) = FI_p(\vec{N}_p)$$

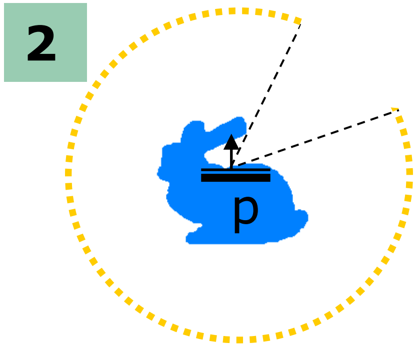


$$FI_p(\vec{N}_p) = FI_{tot_p}(\vec{N}_p) - FIocc_p(\vec{N}_p)$$

$$I_p(\vec{N}_p) = FI_p(\vec{N}_p)$$



$$FI_{tot,p}(\vec{N}_p) = FIV_{tot} \cdot \vec{N}_p$$



$$FI_{occ,p}(\vec{N}_p) = FIV_p \cdot \vec{N}_p$$

Proposed illumination algorithm
Results

- GPU implementation of the algorithm (GLSL)
  - Per-pixel illumination
  - Speed-up the computations
- Intel Core 2 Duo 2.4GHz
  - 2GB RAM
  - NVIDIA GeForce 8800 GTS (512MB Ram)
- Environment map resolution:
  - 32x32x6 pixels
  - Higher resolution – no difference at run-time fps
    - only the preprocessing time
- Window resolution: 512x512
- Trilinear interpolation of the 8 nearest FIVs
- Comparing with ground truth solution
  - Brute force solution
  - Normalized Mean Square Error (NRMSE)

$$NRMSE = \frac{\sqrt{\frac{\sum [p_{ref}(x,y) - p_{ill}(x,y)]^2}{n}}}{p_{max} - p_{min}}$$

Proposed illumination algorithm
Results

Proposed algorithm result (without rendering the occluder)

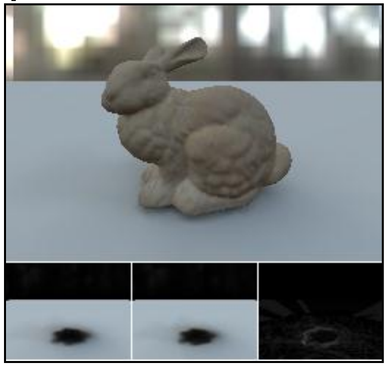
Result of brute force/ground truth solution

Difference between the two

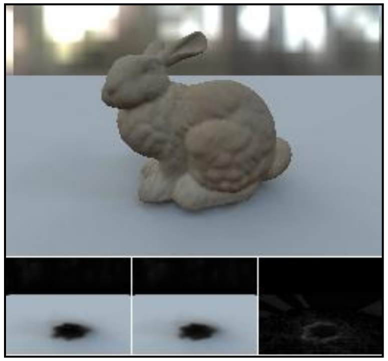
Proposed illumination algorithm
Results

- Different Number of Samples

(θ, φ, ρ) = (32,17,15)
NRMSE = 0.023
0.28 MBytes

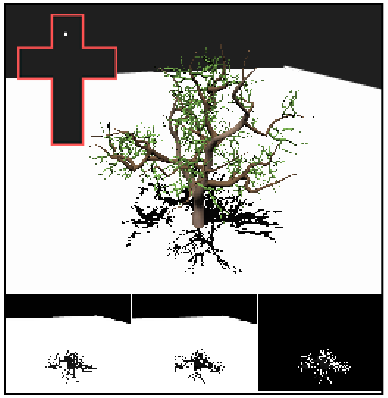


(θ, φ, ρ) = (64,33,30)
NRMSE = 0.008
2.17 MBytes

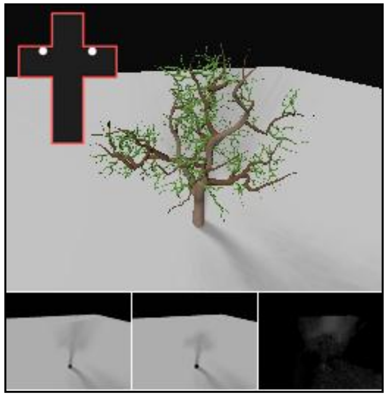


(θ, φ, ρ) = (256,129,30)
NRMSE = 0.007
34 MBytes

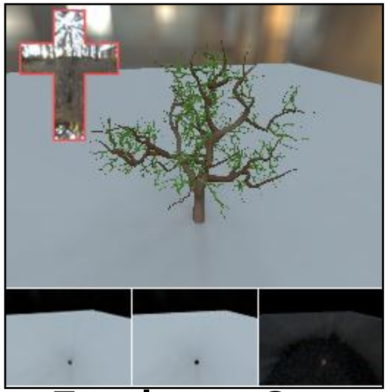
- Different Environment Maps
(samples #: 256x129x30)



1 point light
NRMSE = 0.1



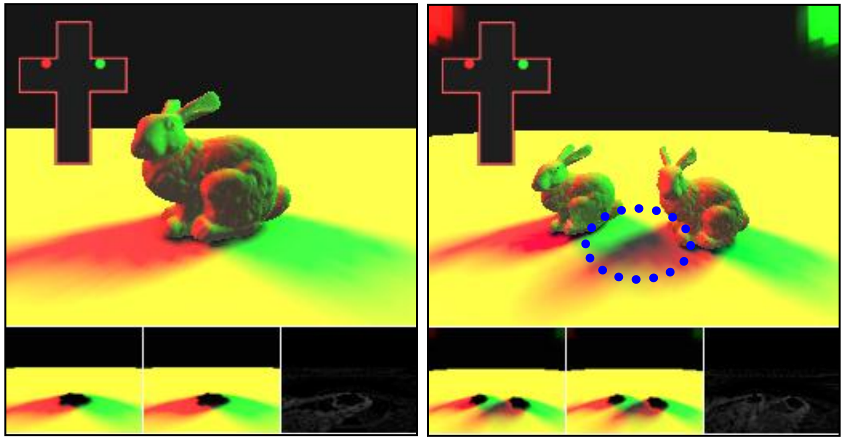
2 area lights
NRMSE = 0.035



Eucalyptus Grove
NRMSE = 0.022

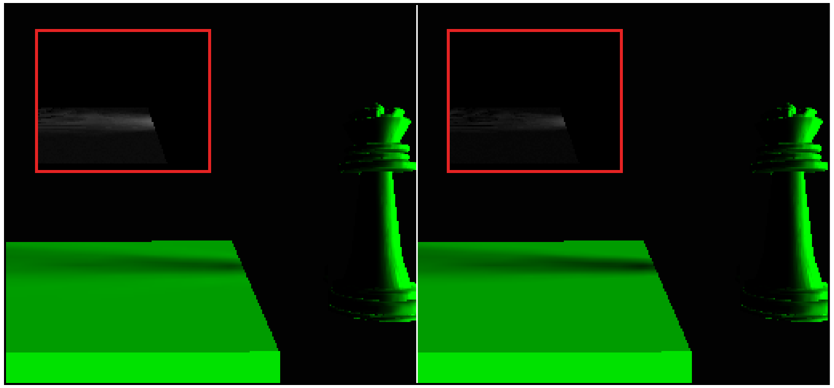
Proposed illumination algorithm
Results

- Multiple occluders



NRMSE = 0.019   NRMSE = 0.019

- Partial Occluders

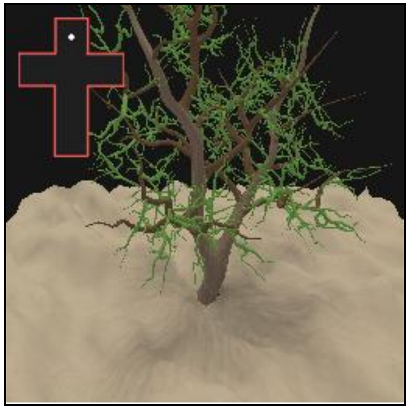


NRMSE = 0.026
Correction: OFF

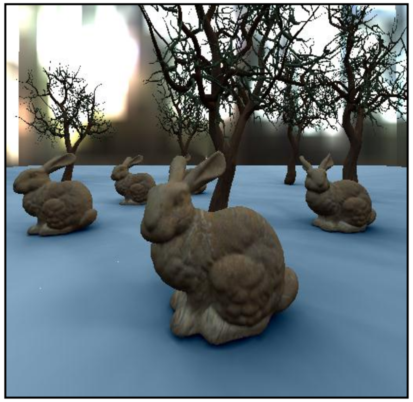
NRMSE = 0.019
Correction: ON

Proposed illumination algorithm
Results

- Other cases
  - Deformable/Dynamic receiver



  - Complex Scene
    10 objects
    ~ 300K vertices



  - Self-shadows

only direct | direct & self-shadows | difference

Proposed illumination algorithm
Results

Precomputation time & memory requirements

Object	Vertices	Samples	Precmp (mins)	Memory (MB)
Bunny	35947	256x129x30	397	34.0
		128x65x30	99	8.56
		64x33x30	24	2.17
		64x33x15	12	1.08
		32x17x15	3	0.28
Tree	20614	256x129x30	222	34.0

(θ, φ, ρ) = (32,17,15)
NRMSE = 0.023
3 mins / 0.28MB

- Memory requirements:
  - independent of the number of vertices
  - linear to the number of samples
- Precomputations time:
  (implemented on CPU)
  - linear to the number of samples

Proposed illumination algorithm
Results

Combination of Objects

Scene	Objects	Vertices	Memory (MB)	FPS
1 bunny	1	35 947	8.56	112
1 tree	1	20 614	8.56	112
10 bunnies	10	359 470	8.56	68
10 trees	10	206 140	8.56	68
5 x (bunny+tree)	10	282 805	17.12	68
10 x (bunny+tree)	20	565 610	17.12	46
15 x (bunny+tree)	30	848 415	17.12	35
20 x (bunny+tree)	40	1 131 220	17.12	29
30 x (bunny+tree)	60	1 696 830	17.12	22

- Number of FPS:
  - Independent of the number of vertices of the object
  - Depends on the number of objects in the scene
- Memory requirements:
  - Depends on the number of different types of objects
    - multiple times the same object does not increase requirements

COMPARISON WITH OTHER METHODS	**PRT** Sloan SIGGRAPH	**Precomputed Shadow Fields** Zhou SIGGRAPH	**Dynamic Soft Shadows** Ren ACM TOC	**FIV** this thesis
Env. Map (EM) Frequencies	Low	Low (dynamic scenes) High (static scenes)	Low	All
Dynamic scenes	Rigid	Rigid	Deformable	Deformable receivers & rigid occluders
Geometry approx.	No	No	Yes	No
Memory requirements dif. obj/vert./EM	>GB	50MB 7 / 30K / low 500 MB 2 / 70K / high	They do not demonstrate any results	<1MB 1 / 1 / all ~10MB (dense sampling) 1 / 1 / all ~10MB (adeq. sampling) 10 / 1 / all
Frame rate obj/vert./EM	~ 100fps* 1/~40K/low *diffuse objects (glossy/fixed VP:~8fps)	0.1 – 10fps 2 /70K/high ~20fps 3-32/28K/ low	~25 fps 2/65K/low ~12 fps 8/120K/low	112 fps 1 / 1 / all 35 fps 30/ 1 /all

Proposed illumination algorithm
HDR illumination

- High-quality illumination
  - incident radiance → natural values can be used

- Real-world
  - Vast range of luminance values
    Examples:
    - sunlight: 10^5 cd/m^2
    - indoor lighting: 10^2 cd/m^2
    - starlight: 10^{-3} cd/m^2

  - ~10 orders of magnitude in luminance values
  - Within a scene we may have High Dynamic Range (HDR) of luminance values
    - dynamic range: ratio of the highest scene luminance to the lowest scene luminance

- Natural illumination
  - Take into account HDR incident radiance from the environment

Proposed Illumination Algorithm
HDR illumination

3D scene

HDR incident radiance

Illumination algorithm →

HDR illuminated scene

- HDR images can not be displayed on standard monitors (LDR)

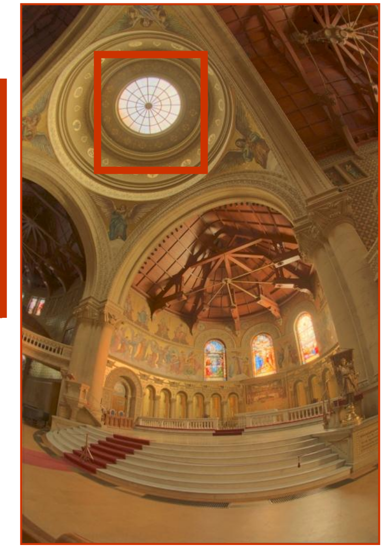
- A final pass is needed to map HDR values to LDR
  - Tonemapping

tonemapping

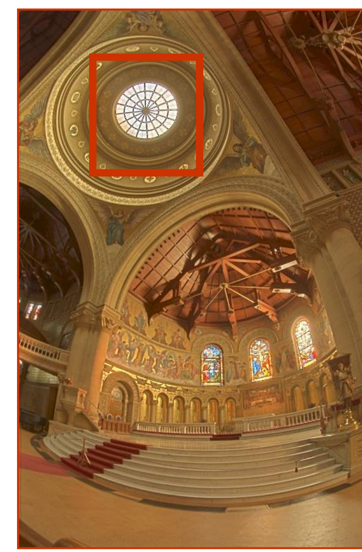
Tonemapping
Problem statement

- Tonemapping operators
  - Global operators
  - Local operators

- Global operators
  - Apply same operation to all pixels
  - Fast Computation
  - Moderate quality results



- Local operators
  - Take into consideration the local properties of individual pixels
  - Preserve the local contrast reproduction
    - local and global component
  - Better quality results
  - More computationally expensive
  - Much slower than global operators



- We need
  - High frame rates
  - High-quality

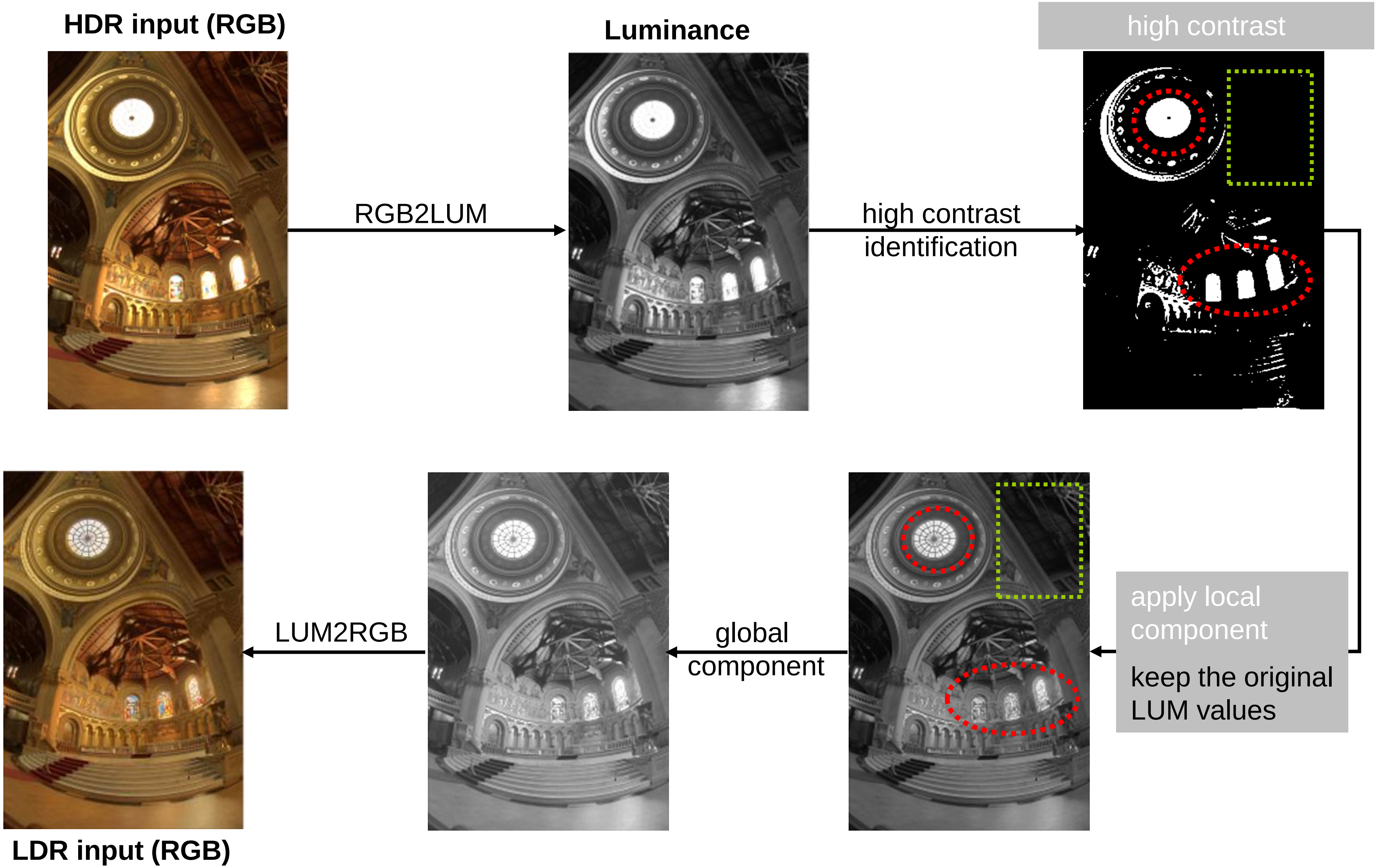
Proposed selective tonemapping Approach

- Apply the computational expensive local component of TMO in a selective way
  - Only on the high contrast areas ("important areas")
    - edge detection algorithm

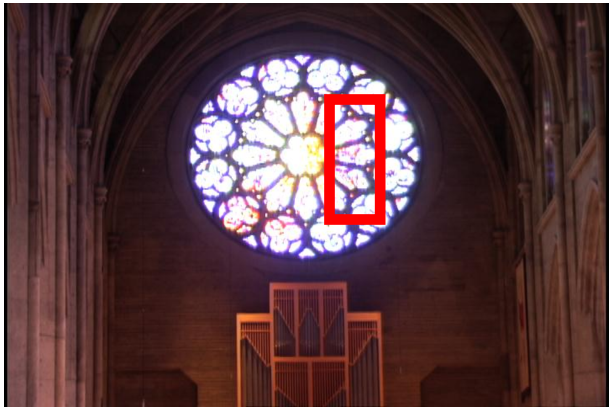
→ Reduce expensive computations

→ Maintain the quality

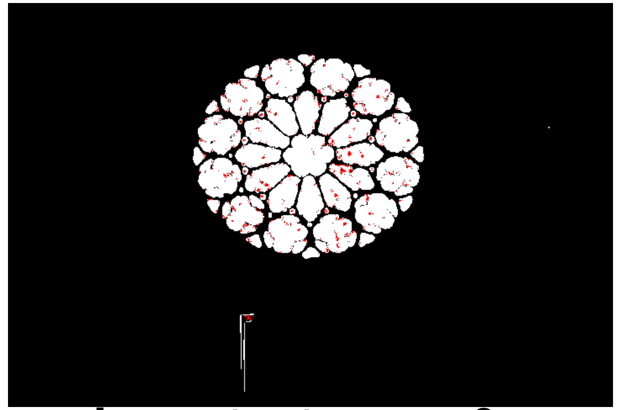
Proposed selective tonemapping Framework



Proposed selective tonemapping
Results



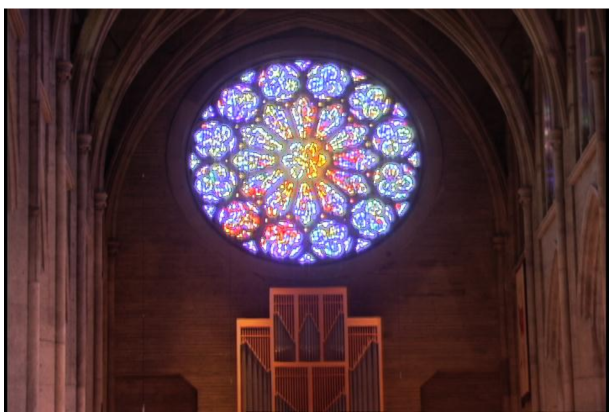
HDR image



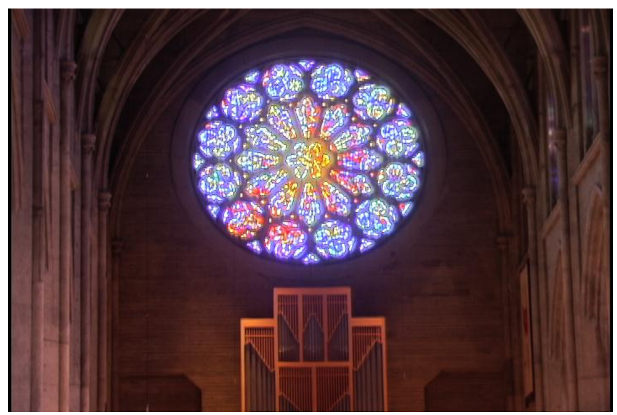
Important areas & superimposed VDP map

GTM

STM



Selective TM



Local TM

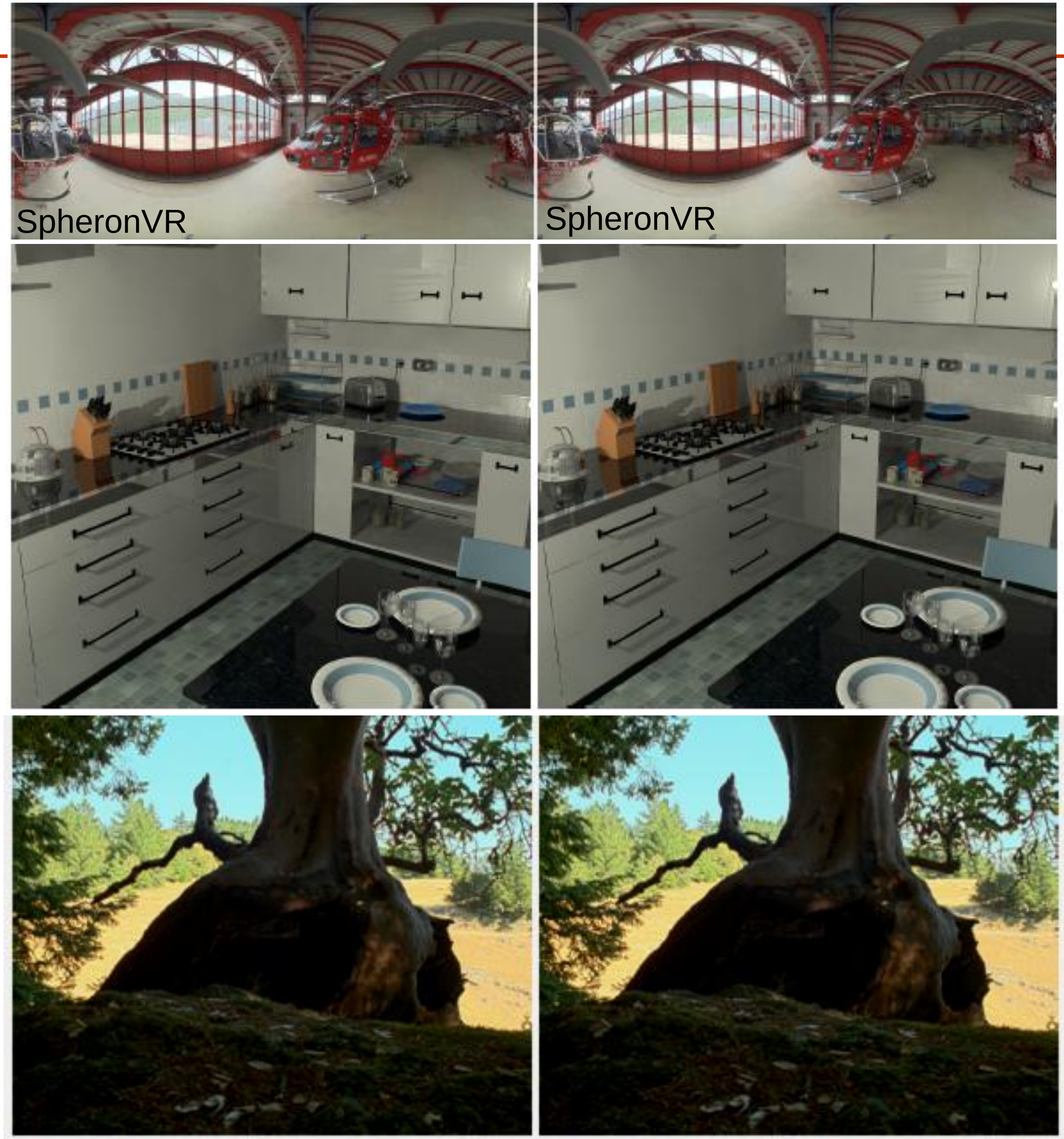
LTM

VDP: Visual Difference Predictor

Proposed selective tonemapping

Results

Selective Tonemapping



Local Tonemapping

Proposed selective tonemapping
Results

HDR image	image resolution	speed (fps) GTM	STM	LTM
Nave	720x480	147	94	29
Rosette	720x480	147	78	29
Memorial	512x768	136	82	25
Desk	644x874	95	31	18
Belgium	1025x768	66	30	13
16RPP	900x900	63	44	12
FogMap	751x1330	63	37	12

- Real-time fps for high resolution images
- Much faster than LTM (~3 times)

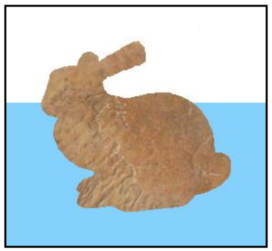
HDR image	$VDP_{\%}$ GTM	$VDP_{\%}$ STM	$Gain_{TM}$ GTM	$Gain_{TM}$ STM
Nave	3.70	1.47	39.72	63.95
Rosette	6.25	0.38	23.52	205.26
Memorial	5.44	2.41	25.0	34.02
Desk	17.66	2.72	5.38	11.40
Belgium	1.63	0.21	40.49	142.86
16RPP	0.08	0.04	787.5	1100.0
FogMap	0.23	0.07	273.91	528.57

- Better quality results than GTM
- Better than GTM taking into account quality & time

$$Gain_{TM} = \frac{speed(fps)}{VDP_{\%}}$$

Unified Rendering Pipeline

3D scene



HDR incident radiance

Illumination algorithm →

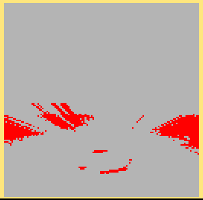
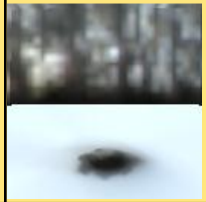
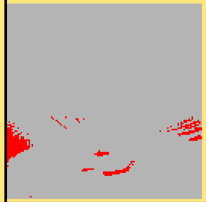
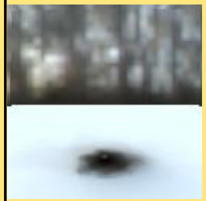
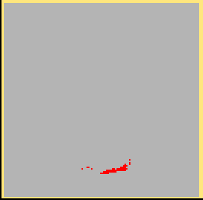
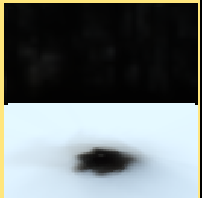
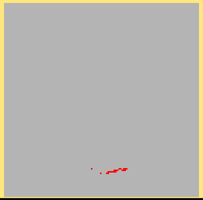
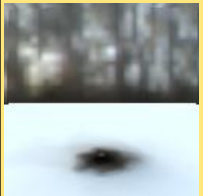
HDR illuminated scene

Tonemapping →

LDR illuminated scene

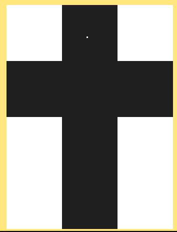
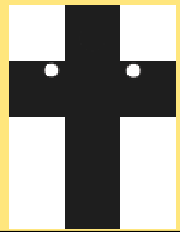
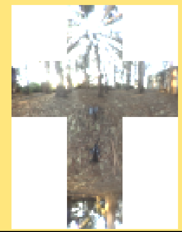
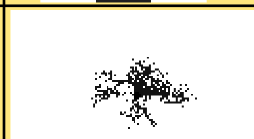
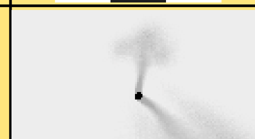
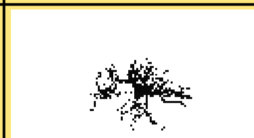
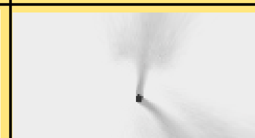
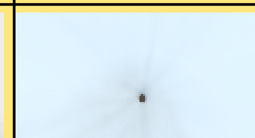
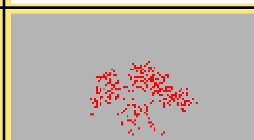
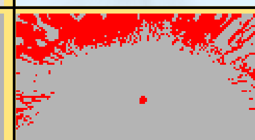
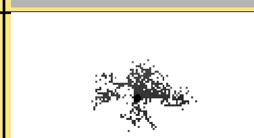
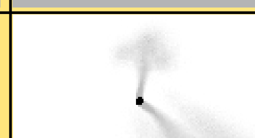
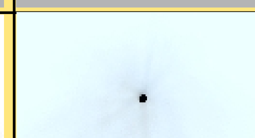
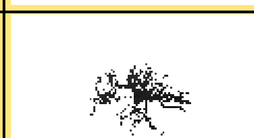
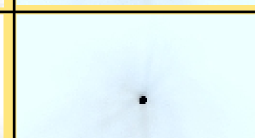
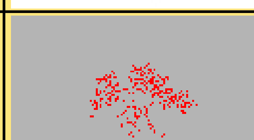
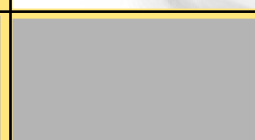
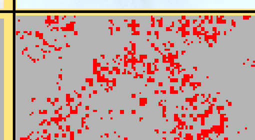
Unified Rendering Pipeline
Results

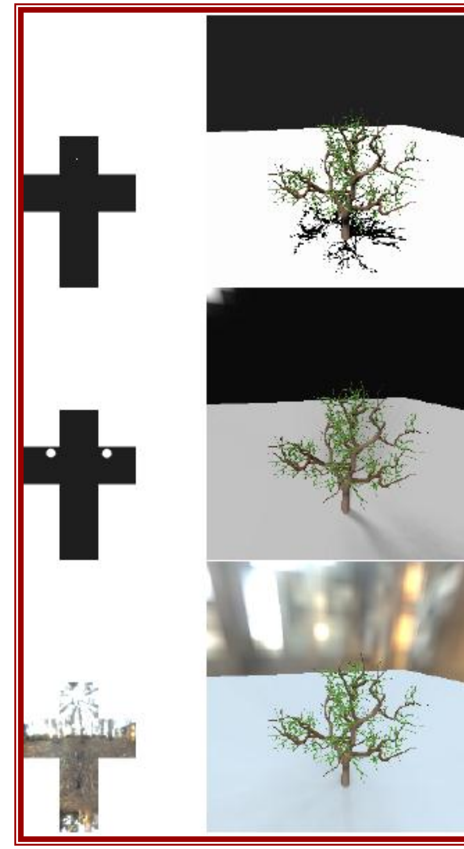
- Visual error of illumination algorithm decreases or increases after tonemapping? (error before TM VS error after TM)

Different number of samples				
	without TM	**VDP map (without)**	**with TM**	**VDP map (with)**
Ground truth				
32x17x15				
64x33x30				
256x129x30				

Samples	without	with
$32 \times 17 \times 15$	5.94	2.59
$64 \times 33 \times 30$	0.20	0.32
$256 \times 129 \times 30$	0.13	0.20

Unified Rendering Pipeline
Results

Different environment maps				
				
with-out TM	**Ground truth**			
	FIV			
	VDP map			
with TM	**Ground truth**			
	FIV			
	VDP map			



Environment map	without	with
1 point light source	1.70	1.70
2 area light sources	0	0
Eucalyptus Grove	11.42	8.31

conclusions
& future work

Conclusions

- Illumination
  - Fullsphere Irradiance (new notion)
  - Fullsphere Irradiance Vector (FIV)
    - Integrate incident light energy within a 3D vector
      - independent of $\vec{N}_p$
      - arbitrary number of light sources
  - Use FIVs to illuminate the scene
    - Soft-shadows
    - Real-time
    - High-quality results
    - Moderate memory requirements

Conclusions

- Tonemapping
  - Selective Tonemapping Framework
    - Speed up local TMOs
    - Real-time for high resolution frames
    - Keep quality (perceptually) of original local TMO
    - Modular framework
      - use any other local TMO
      - use other algorithm for important areas identification step
  - Makes illumination algorithm more tolerant to errors

Future work

- Illumination
  - Specular reflections
    - no direct way to do that
  - Adaptive sampling
  - Online computations of FIVs
    - GPU implementation
    - only at samples needed

    } → Interactive/real-time rates
  - First bounce of inter-reflections

- Tonemapping
  - Exploit more temporal coherence between frames
    - reuse parts of high contrast areas map
  - High contrast areas map → Important areas map
    - what is "important"?
      - e.g. the main character in a game

Publications

1. Despina Michael and Yiorgos Chrysanthou. "Automatic high level avatar guidance based on affordance of movement", Eurographics 2003, In Proceedings Interactive Demos and Posters, pp 221-226, Published by the Eurographics Association, 2003.

2. **Benjamin Roch, Alessandro Artusi, Despina Michael, Yiorgos Chrysanthou, and Alan Chalmers. "Interactive local tone mapping operator with the support of graphics hardware" In ACM Proceedings, SCCG Conference 2007, Published by ACM, 2007.**

3. Petros Patias, Yiorgos Chrysanthou, Stella Sylaiou, Charis Georgiadis, Despina Michael, and Stratos Stylianidis. "The development of an e-museum for contemporary arts". In Virtual Systems and Multimedia, VSMM08, 2008.

4. Samuel Obadan, Andreas Gregoriades, Harris Michail, Vicky Papadopoulou, Despina Michael. "A Robotic System for Home Security Enhancement", ICOST 2010.

5. **Despina Michael and Yiorgos Chrysanthou, "Fullsphere -Irradiance factorization for real- time all-frequency illumination for dynamic scenes." Computer Graphics Forum Journal, accepted for publication, 2010.**

6. Despina Michael, Panagiotis Zaharias and Yiorgos Chrysanthou. "A virtual tour of the Walls of Nicosia: An assessment of childrens' experience and learning effectiveness", VAST 2010.

7. Despina Michael, Nektarios Pelekanos, Isabelle Chrysanthou, Panagiotis Zaharias and Yiorgos Chrysanthou, "Comparative Study of Interactive Systems in a Museum", submitted EuroMed 2010.

8. **"Selective Tonemapping", to be submitted in a CG journal.**

Acknowledgments

- Yiorgos Chrysanthou
- Alessandro Artusi, CaSToRC The Cyprus Institute, Cyprus
  Benjamin Roch, Technical University of Vienna, Austria
- CG Group @ CS.UCY
- CS.UCY
- Family
- Aimilia, Andreas, Anna, Christos, Costas, Georgios I., George S., Marios, Nearchos, Pyrros, Ritsa, Vicky (& little baby coming soon)
- PhD examination committee

Thank you for your attention!