



BOSCH
Invented for life

UC DAVIS
UNIVERSITY OF CALIFORNIA

Procedural Window Lighting Effects for Real-Time City Rendering

Jennifer Chandler
Lei Yang
Liu Ren

UNIVERSITY OF CALIFORNIA, DAVIS
BOSCH RESEARCH NORTH AMERICA
BOSCH RESEARCH NORTH AMERICA

Outline

- **Motivation**
- **Random Window Lighting**
- **Interior Mapping**
- **Results**
- **Conclusion**

Motivation

- We work on large-scale 3D urban scene visualization at Bosch
- Visualizing night scenes with lighting effects is challenging



Requirements

- Randomness



Requirements

- Adjustability 20% lit (12:00AM)



50% lit (6:00PM)

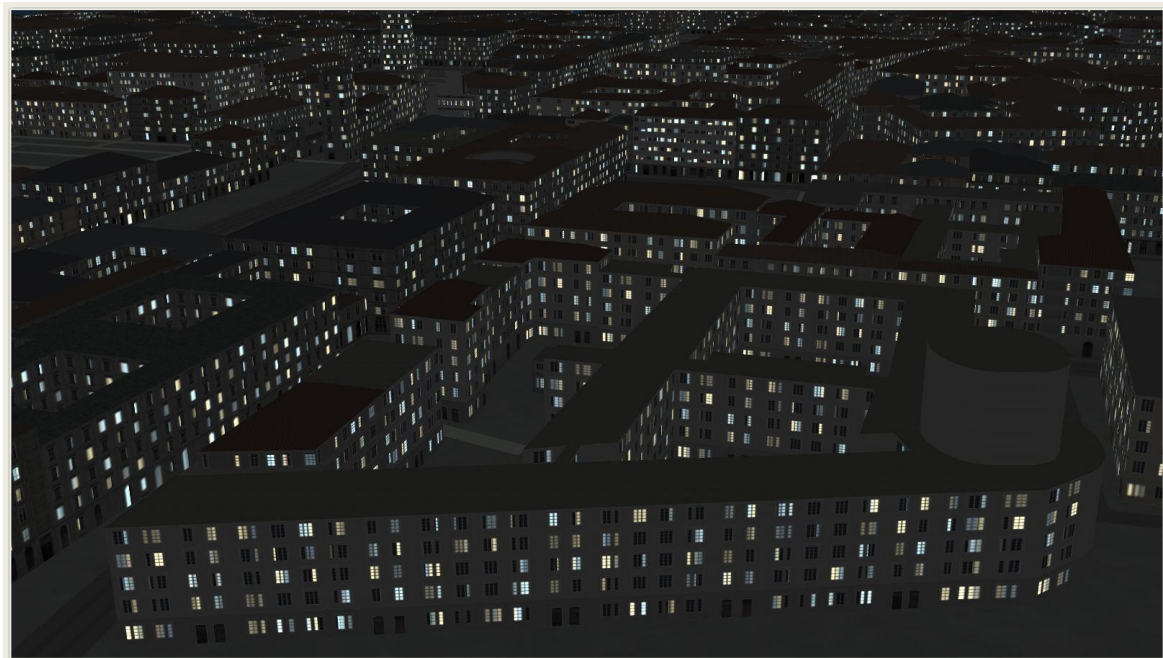


100% lit (8:00PM)



Requirements

- Scalability



Requirements

- Details



Outline

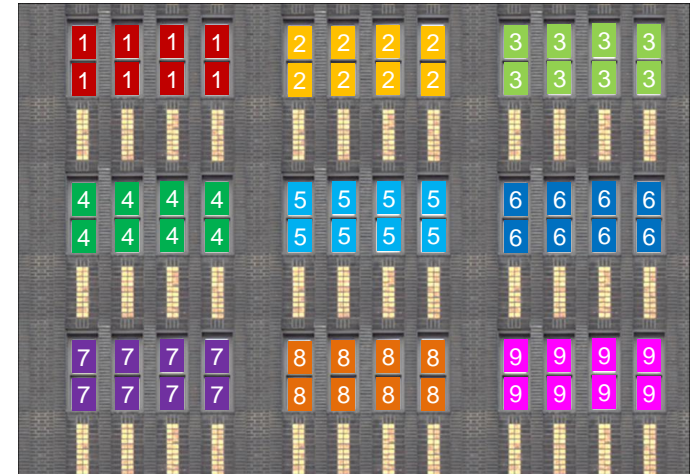
- Motivation
- **Random Window Lighting**
 - Obtaining window ID from texture parameterization
 - Accommodating per-floor façade texture
- Interior Mapping
- Results
- Conclusion

Window ID? Where to store it?

- Per-room window ID can be used to selectively light a random subset of windows
 - E.g. using a hash function like

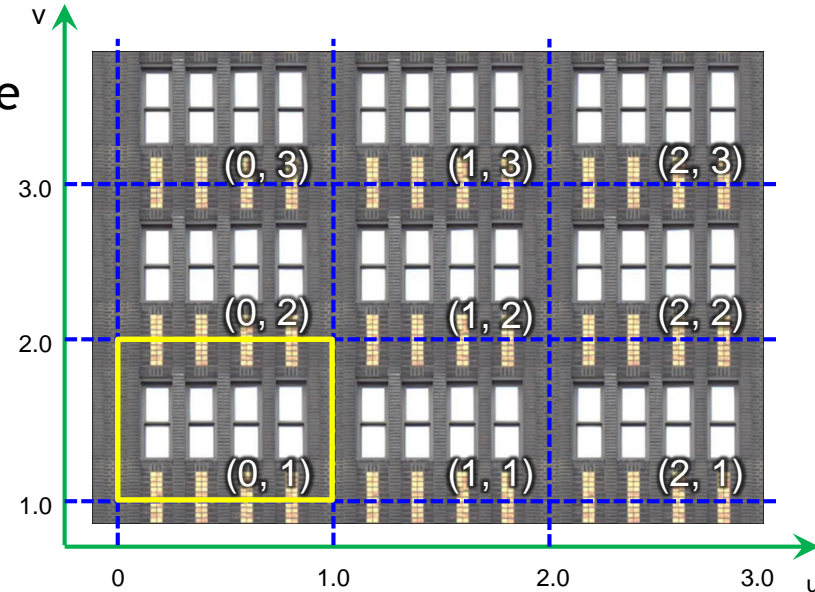

```
bool lit = (id * 107) % 100 < 50
```

 will light roughly 50% of the windows
 - Evaluated per-fragment
 - Window area specified in transparency
- Encoding ID in texture is problematic
 - Extra storage is required
 - Texture filtering is difficult



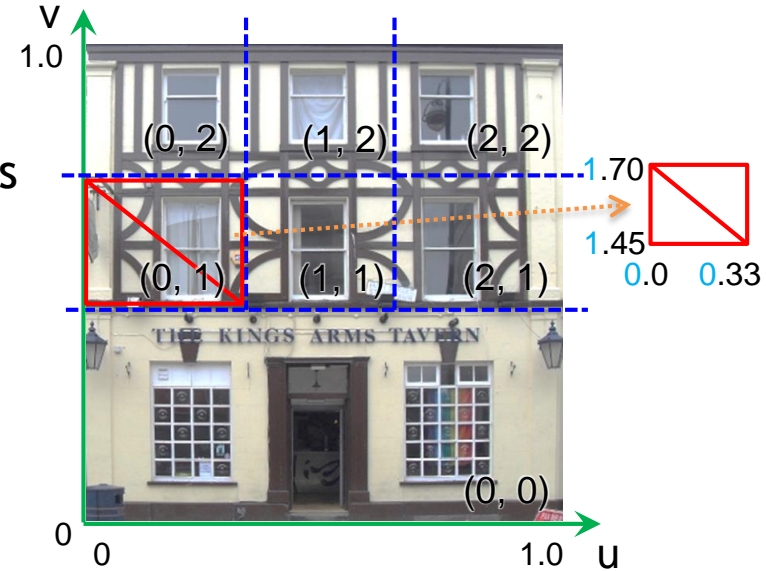
Obtaining ID from texture coordinates

- Secret data with repetitive textures
 - Façades generated with repetitive patches usually use “repeat” texture addressing mode
 - Window/room ID can be obtained from the integer part of UV coordinates
 - Assuming one patch covers a whole room



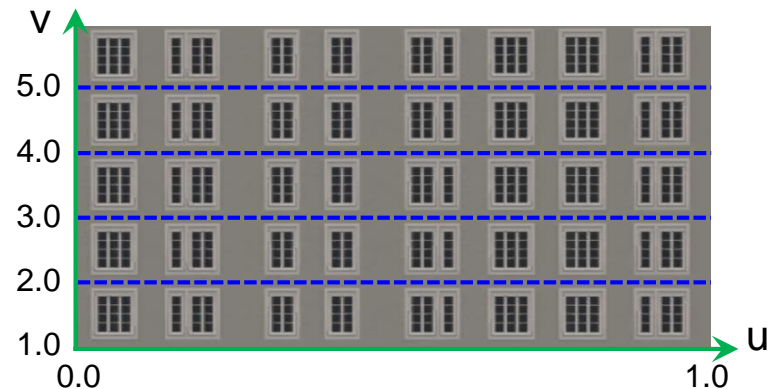
Generalized façade parameterization

- Does it handle non-repetitive façade?
 - Yes, with individual room/window geometry
 - Add integer offset to texture coordinates of different rooms/windows
 - Needs a bit of UV artist work
 - Not common in our city data



Accommodating per-floor facade texture

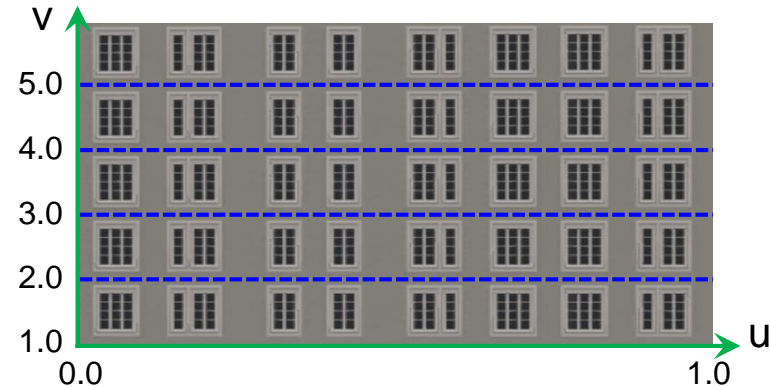
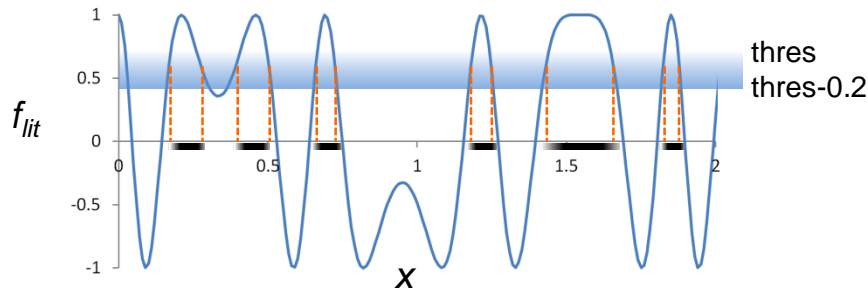
- Many building facades are made by repeating a per-floor texture
 - Improving variety of appearance
 - Keeping geometry simple
- We lost our way to identify rooms
- Need a *floor-plan agnostic* random window-lighting generation method



Pseudo-random “window curtain” function

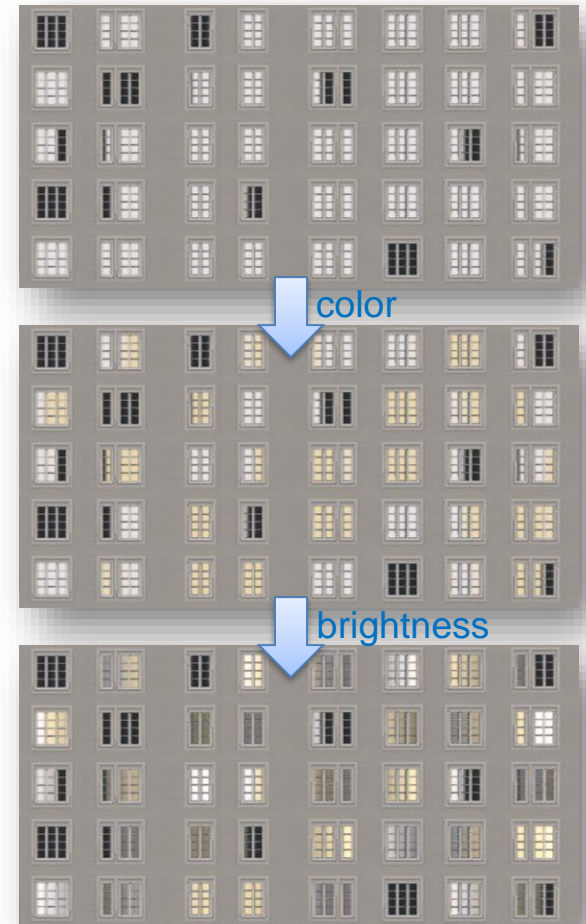
- Place random curtain behind windows
 - Curtain pattern differs among floors
- We use a pseudo-random function:

$$\begin{aligned}
 x &= u + \text{floor}(v) \\
 f_{lit} &= \cos(2(\sin(5x) + \sin(2x))) \\
 lit &= \text{smoothstep}(\text{thres}-0.2, \text{thres}, f_{lit})
 \end{aligned}$$



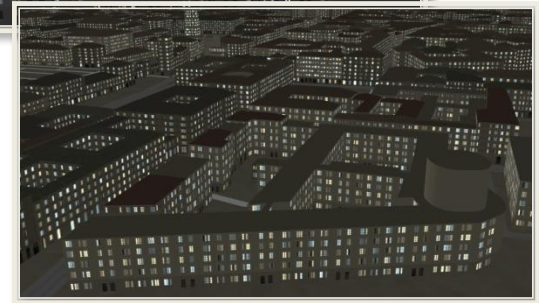
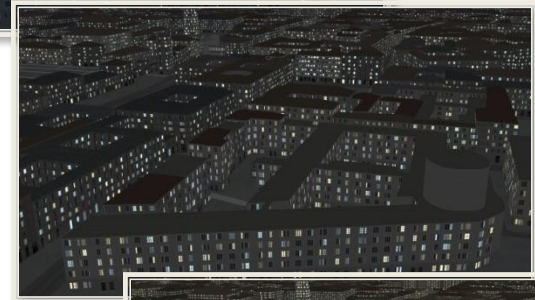
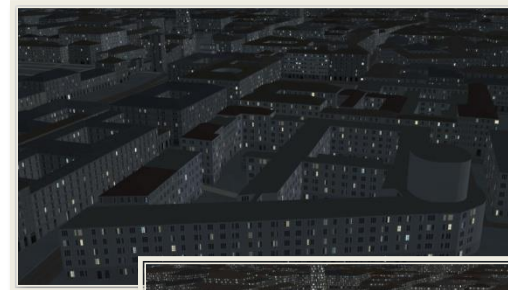
Adding variety

- Adding color variations
 - Alternate between two colors
 - Use a second pseudo-random function to blend
- Adding brightness variations
 - Use a third pseudo-random function to modulate brightness of light



Review requirements

- Properties of our solution
 - **Random:** each row has a different curtain pattern; each building can specify a different texcoord offset
 - **Adjustable:** Percentage of lit window area is adjusted continuously by *thres*
 - **Scalable:** No extra data needs to be stored; function is trivially evaluated in a pixel shader
 - **Details:** interior?



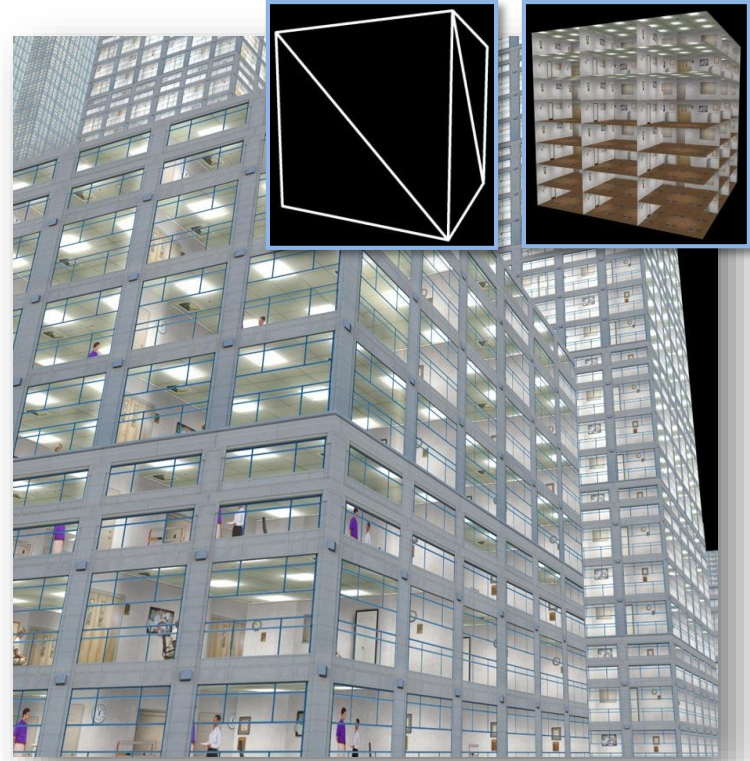


Outline

- Motivation
- Random Window Lighting
- **Interior Mapping**
 - Computing world-space room coordinates
 - Room separation with per-floor texture
- Results
- Conclusion

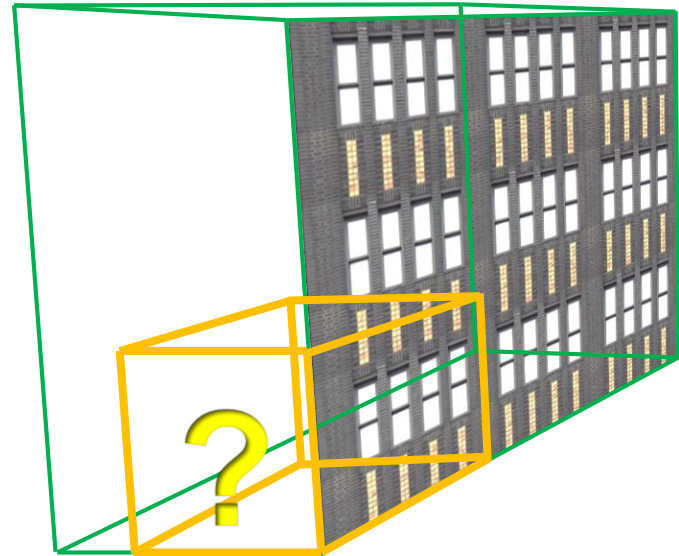
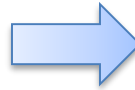
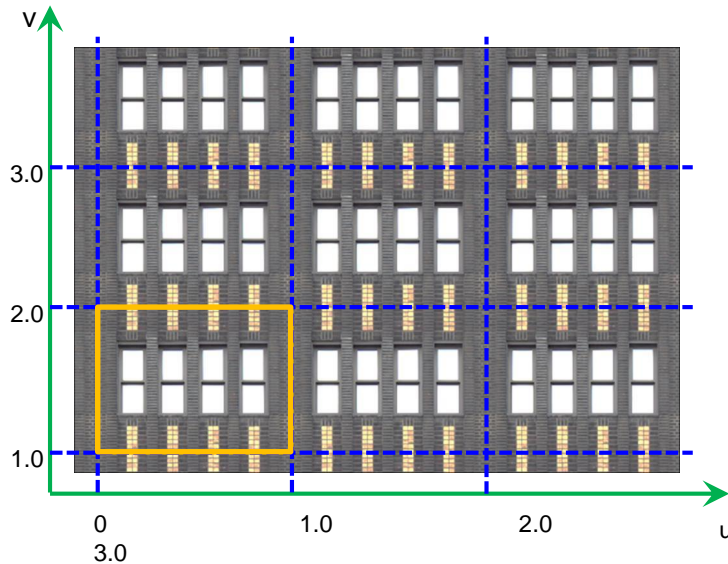
Interior mapping overview

- Parallax effect for rendering building interiors without actual room geometry [Van Dongen 2008]
- Use a cube map texture to represent room interior
 - Each pixel computes an intersection between view ray and room texture
- World space parameterization of rooms
 - Not flexible enough for our purpose



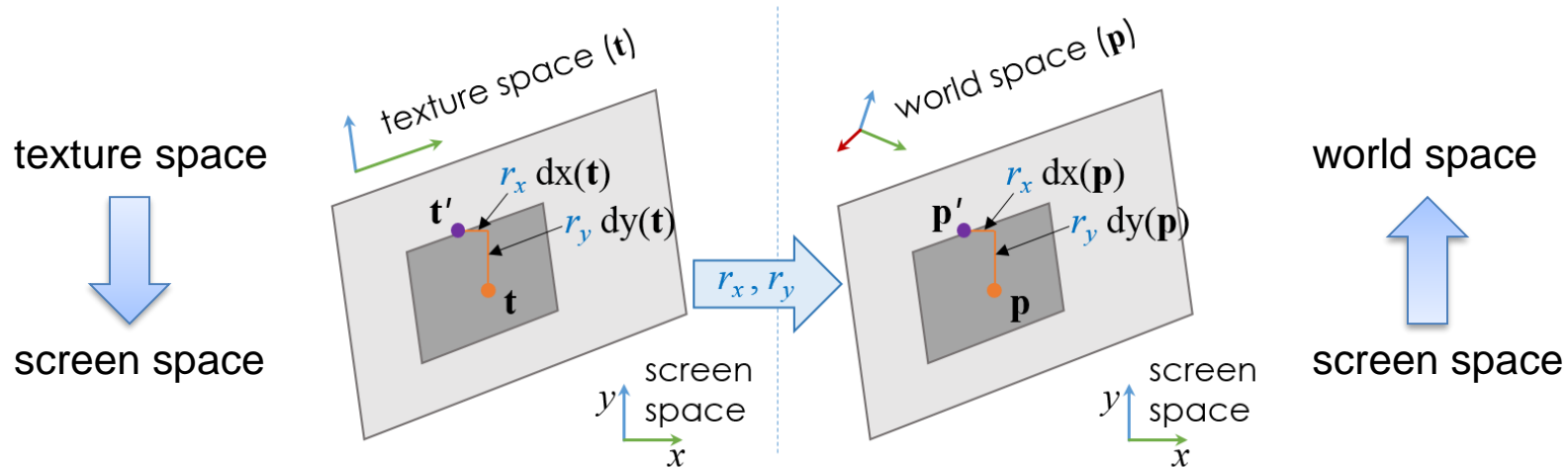
Challenge #1

- How to obtain world-space room coordinates at run time?
 - Required to compute per-pixel ray-cube intersection
 - Our room definition only exists in texture space



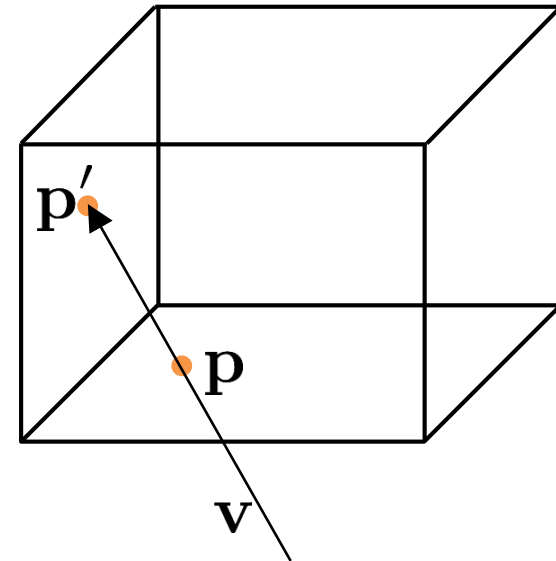
Obtaining world-space room coordinates

- Use screen-space derivatives (pixel shader intrinsics) to map between texture coordinates and world coordinates
- Solve for ratio with known coordinates (texture space)
- Apply ratio to unknown coordinate to get world space position



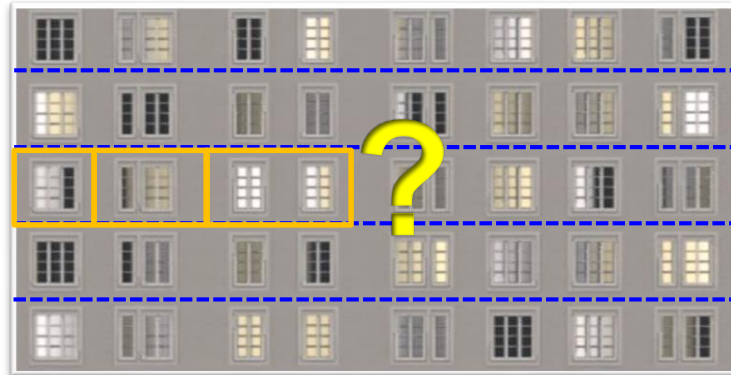
Solving ray-wall intersection

- Need to find point on plane in world space
- Solve ray-plane intersection $t = \frac{(\mathbf{p}' - \mathbf{p}) \cdot \mathbf{n}}{\mathbf{v} \cdot \mathbf{n}}$
- $(\mathbf{p}' - \mathbf{p}) = r_x \mathbf{dx}(\mathbf{p}) + r_y \mathbf{dy}(\mathbf{p})$
- Plug in known texture coordinates for walls
- Find closest intersection for all five walls
- Simplified equations for all cases in the paper



Challenge #2

- How to handle room separation with per-floor texture?
 - Room walls contribute to realistic interior appearance
 - We did not really define rooms with our window-curtain function



Room separation with per-floor texture

- Define wall locations at minimums of window curtain function
 - Correspond to center of curtain
- Wall may appear in middle of window
 - Window curtain can hide this
- Encode wall locations in a texture for easy look-up
 - Window curtain function is periodic, so lookup texture can be bounded



Interior mapping implementation

- Use cube map texture for the room
- Convert intersection to cube map coordinates
 - Compute relative location between each pair of parallel walls
 - Avoid explicit conversion of coordinates to world space
- Add variety
 - Furniture plane
 - Randomly swap walls



Outline

- Motivation
- Random Window Lighting
- Interior Mapping
- **Results**
- Conclusion

Performance

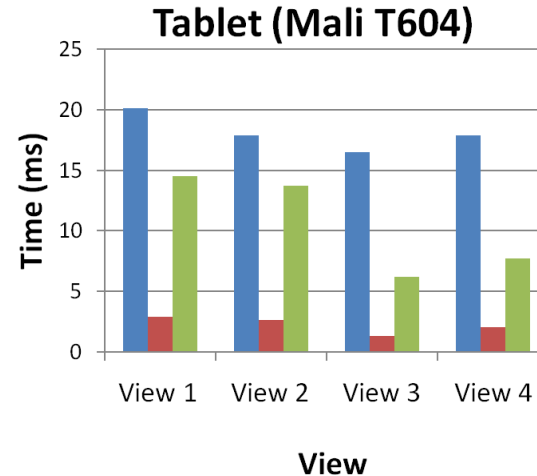
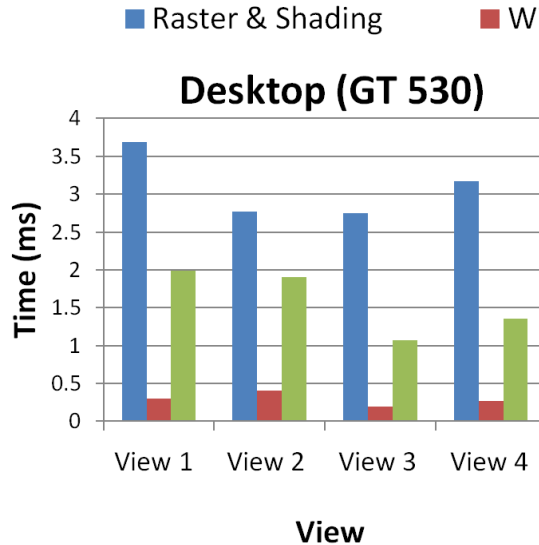
- A fraction of a frame time on desktop and mobile



View 1



View 2



View 3



View 4

Back buffer resolution: Desktop 1600x900, Tablet 1280x720

Practical concerns - pros and cons

Our Method

- 😊 Supports random window lighting w/o extra data storage or pipeline change
- 😞 Needs special facade texture parameterization
- 😊 Visualizes room interior without explicitly knowing room extent
- 😞 Only works on planar windows and piecewise-planar facades
- 😊 Easily scales to large city geometry with image-based algorithm
- 😞 Requires authoring work to create spatial variety of appearance

Conclusion

- Random window lighting with optional interior mapping
- Easy to integrate into existing pipeline
- Flexible texture coordinate façade parameterization
- Scales well for large city scenes





Thanks for your attention!