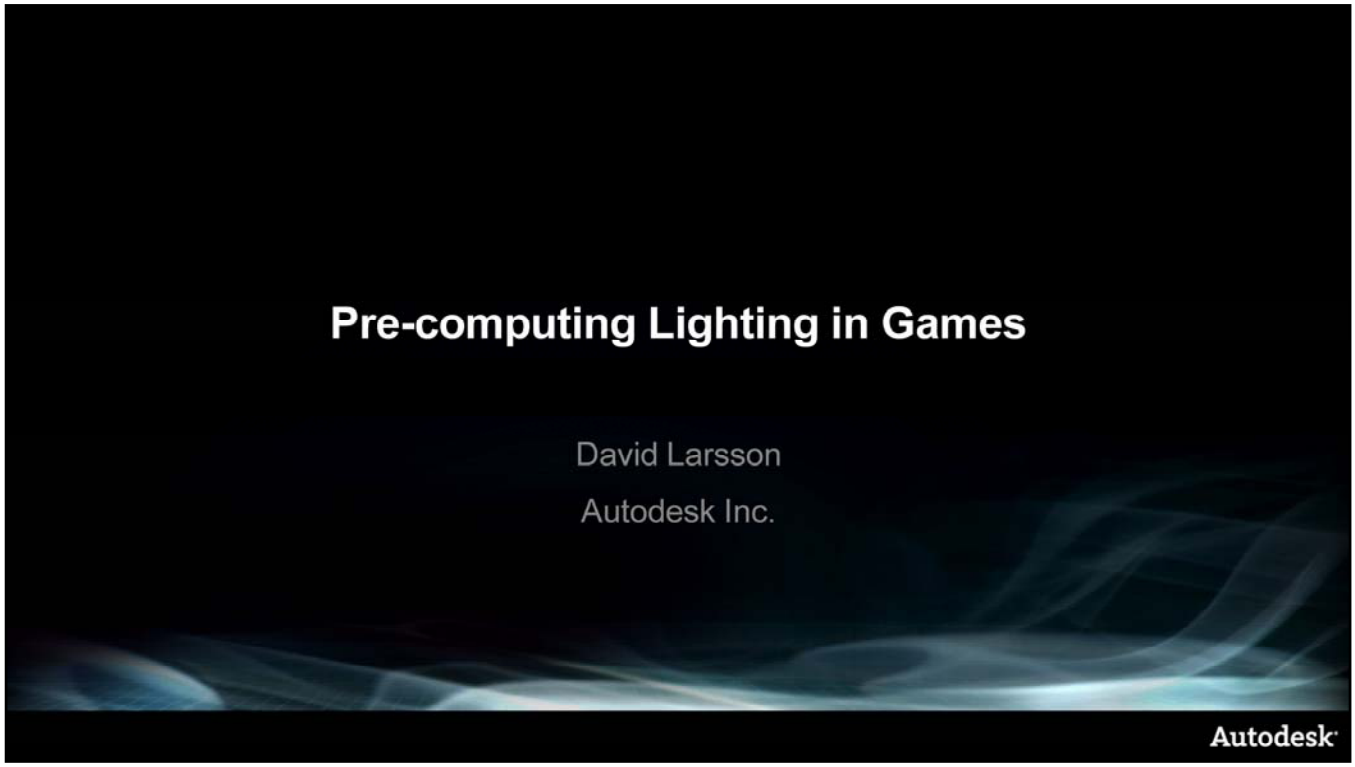
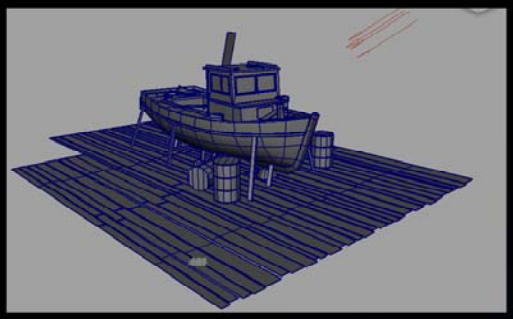
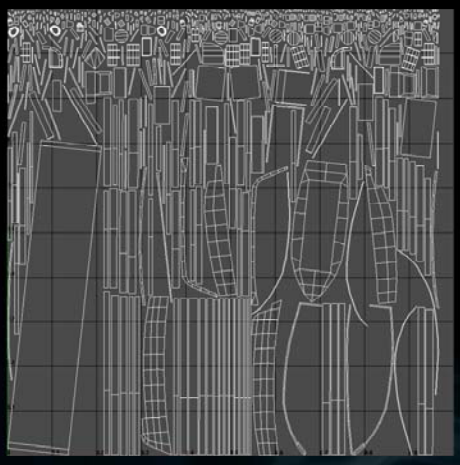

Pre-computing Lighting in Games
David Larsson
Autodesk Inc.
Autodesk

Precomputed lighting is a general concept. This presentation focuses on the static light, static geometry case. Ambient occlusion and PRT, etc are also baked “lighting”.
The idea is to move the heavy part of the light computations into the game studio in order to save runtime cpu and gpu time for other things.
In the end it’s about delivering 30 or 60 frames per second and that doesn’t leave you with much margins, games is still all about smoke and mirrors.
Baked lighting is independent is independent of choice of GI algorithm and can also store direct lighting.

What is baked lighting?



Wireframe scene



Wireframe UV layout

Content courtesy of A2M

Autodesk

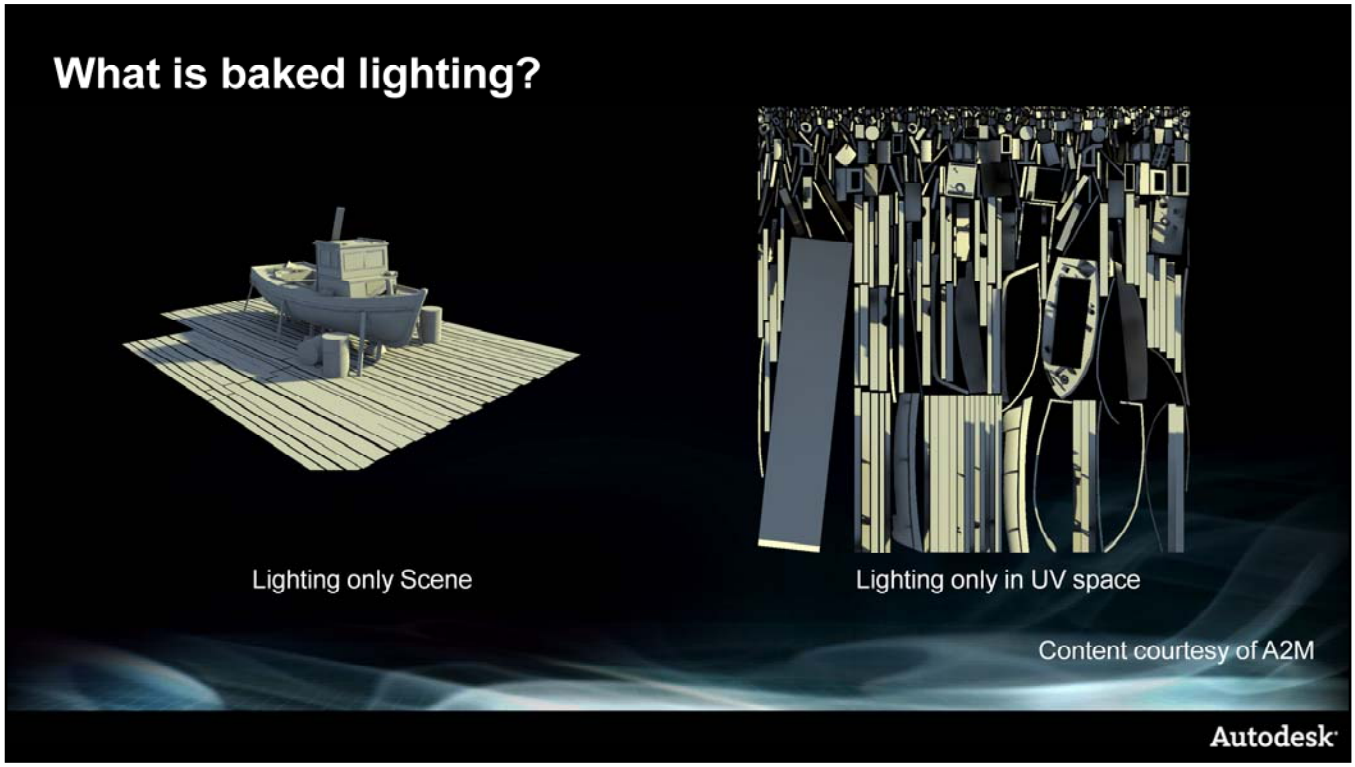
What is baked lighting?

Diffuse Reflectance scene

Diffuse Reflectance in UV space

Content courtesy of A2M

Autodesk



Diffuse light is view independent, suitable for light maps.
Specular effects frome baking are possible. It will be covered later in the presentation

What is baked lighting?

*

=

Content courtesy of A2M

Autodesk

Why bake lighting?

- Quality
- Performance
- Lighting Workflow

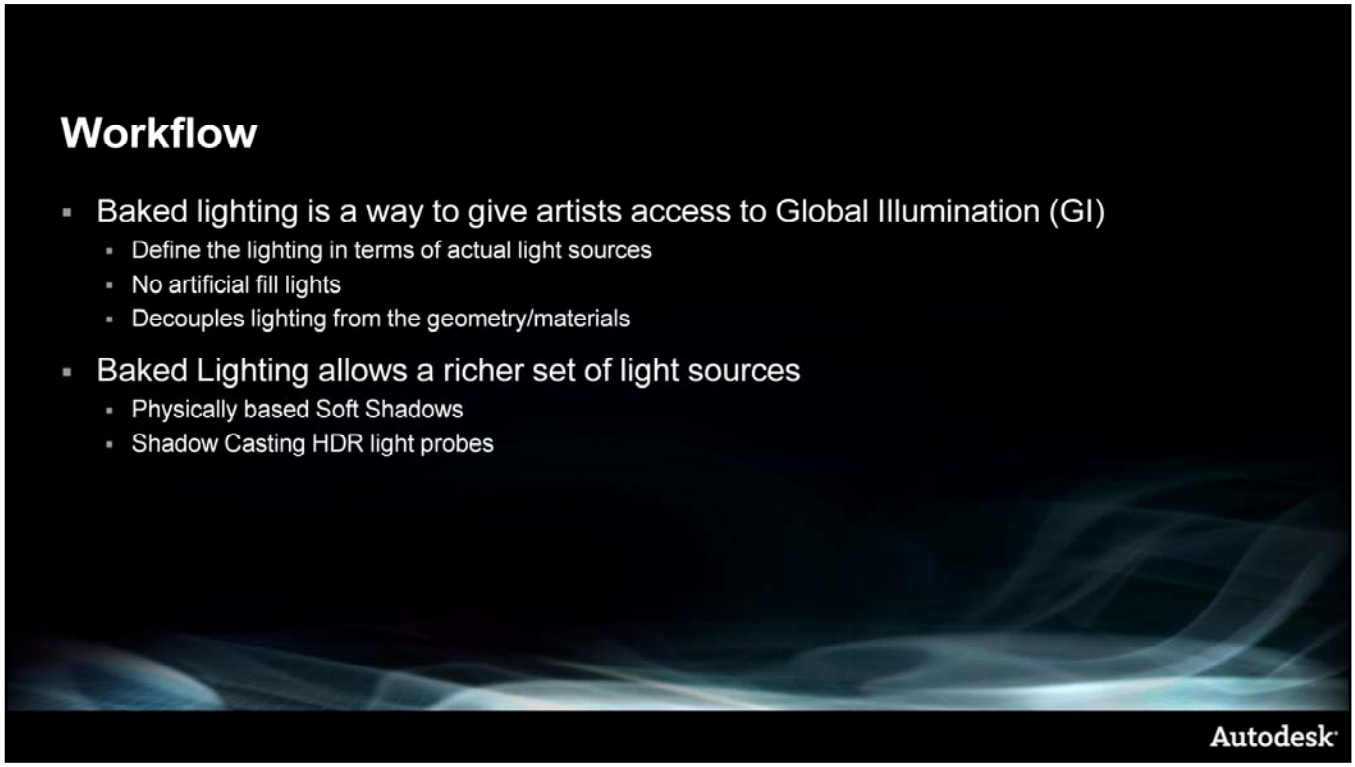
Autodesk

Quality

- Allows the highest quality light simulation algorithms
  - GI effects
  - Multiple bounces
  - Allows high quality direct lighting

Autodesk

Note the color bleeding affecting the white object in its yellow environment.
Note the blue sky lighting that is the major source of lighting in the shadows
Note the shadow penumbra in the sun shadow.
All these effects would be expensive to simulate in realtime



GI is a powertool for lighting. Less Manual work.
We have seen examples of literally thousands of light sources to simulate an overcast sky environment for an outdoor level
Placing manual bounce lights forces change of lighting if you change involved materials and geometry.
GI encourages reuse of scenes since you can change the look of an environment by just changing the lighting conditions
Examples of light sources that are hard to run in realtime: Emissive objects, lights with soft shadows, HDR domes with shadow casting.

The runtime performance is about one or a few texture fetches per shading sample. The fetches are done in a cache friendly way, it's not about abusing a texture for a data structure that was never meant to be stored in a texture.

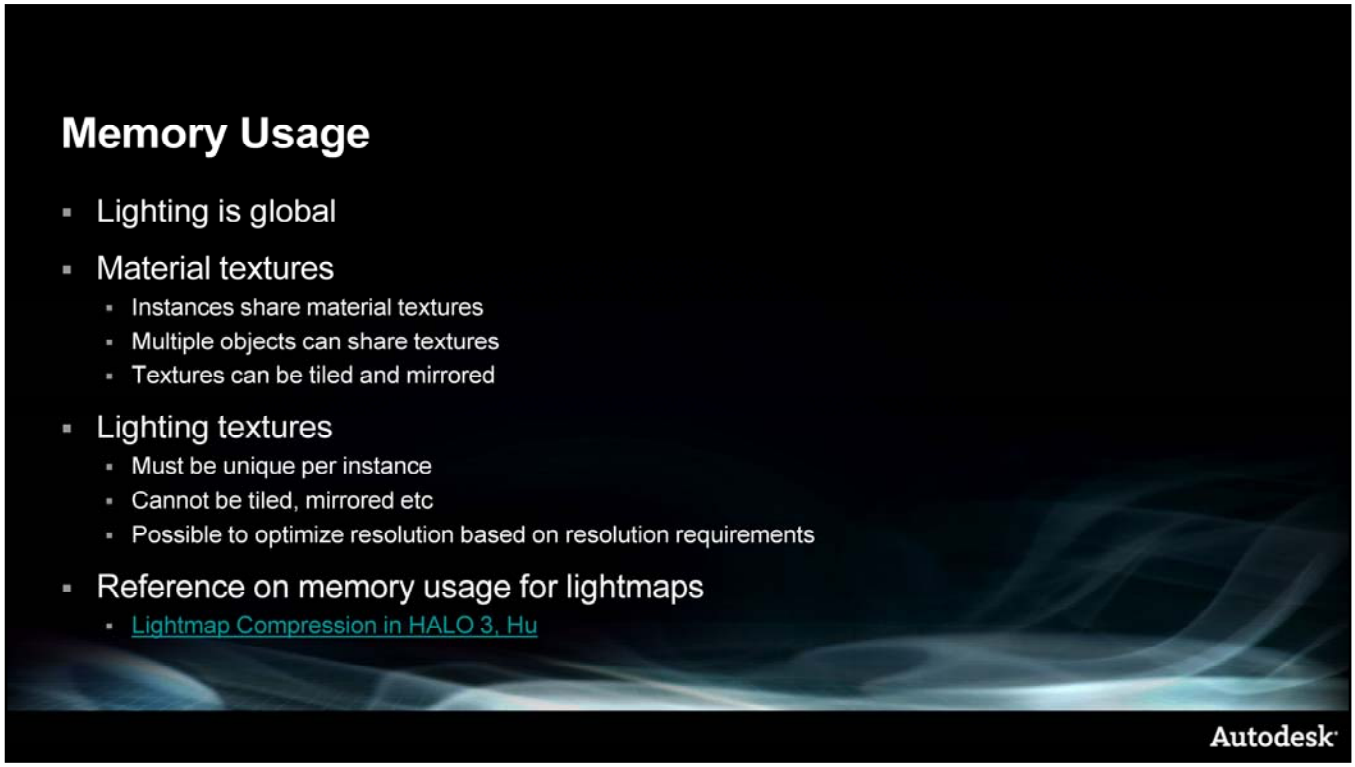
Predictable performance is important.
Don’t want the game runtime performance to be different because an artist happened to add too many lights in a small area. Game Engines often have hard limitations on how many lights that can affect an object which can give strange results.
Some games uses a higher sun direction for the shadows than for lighting in order to make sure the shadows aren’t cast over the entire scene at sunrise/sunset times. With baked lighting the performance is the same regardless of this.

On low end platforms baked lighting is the most effective way to get good looking lighting.

Challenges

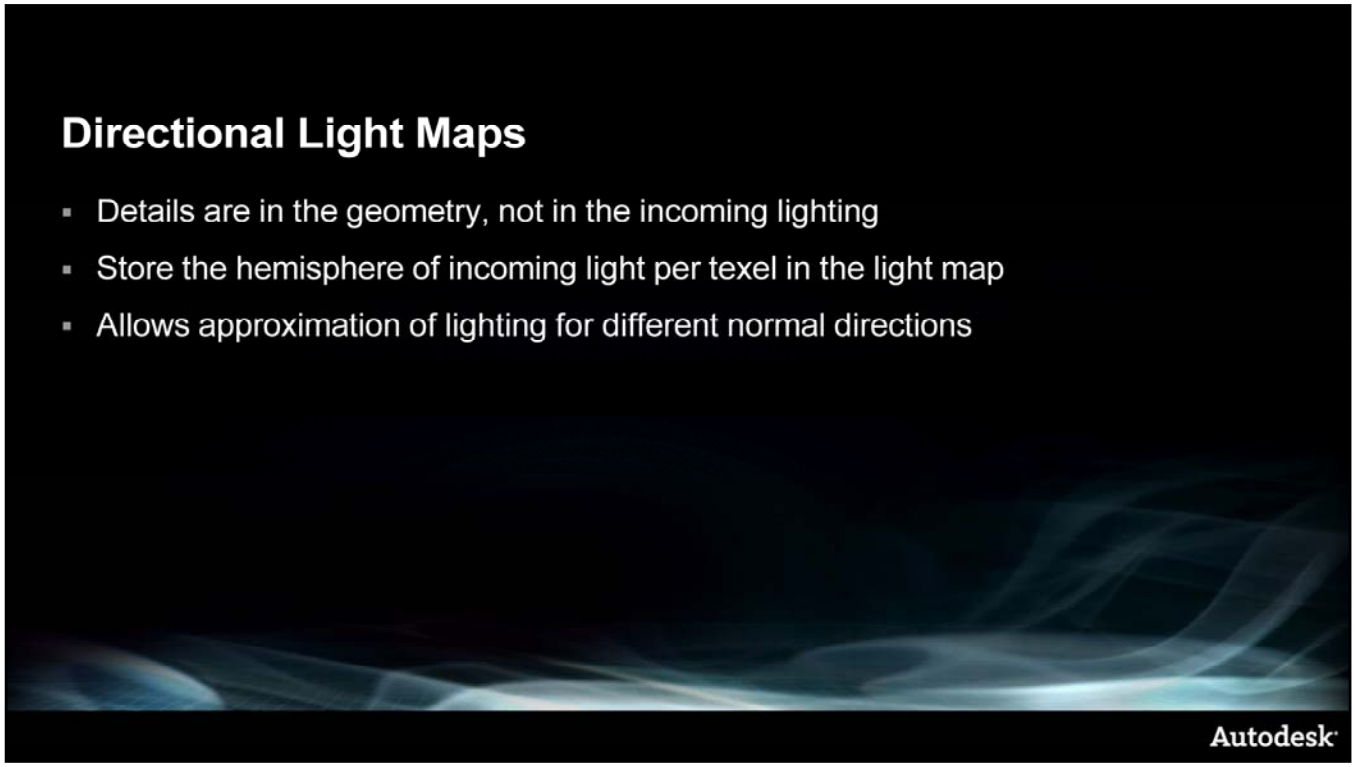
- Changing Light Setups
- Moving/Deformed Geometry
- Memory Usage for the Baked Lighting
- Light Rebuild Times

Autodesk



Texture memory is scarce on game consoles and making good use of it is important.
Lighting is different from many other kinds of textures.
Lighting is HDR too which makes texture format choice or encoding even more important.

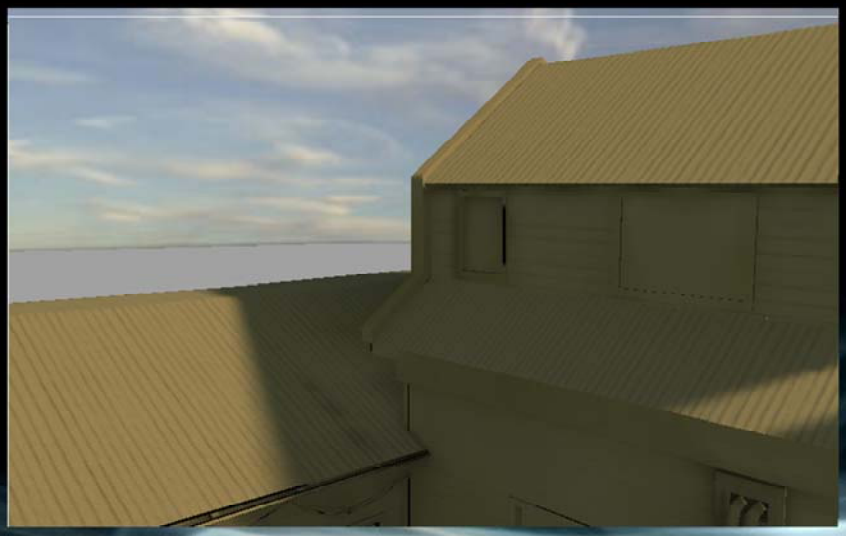
Normal maps forces the resolution of the light map to the same as the normal map in order to capture the details.



The first point is somewhat simplified, there is of course a self shadowing aspect of the normal map and fine geometric details

Directional Light Maps

Content courtesy of A2M



Low resolution Directional Lightmaps combined with normal maps

Autodesk

Directional Light Maps

- Typical Encodings
  - Radiosity Normal Maps (RNM)
  - SH (generally 2 bands, 4 components)
  - Per pixel ambient and directional light
  - H-basis
- Allows real BRDF:s
  - Hemisphere is blurry but it's possible to get reasonable specular effects from it too
- References
  - Half-Life 2 / Valve Source Shading, Gary McTaggart
  - An Efficient Representation for Irradiance Environment Maps, Ramamoorthi et al
  - Efficient Irradiance Normal Mapping, Habel, et al

Autodesk

Daylight cycles are manageable to precompute since they tend to be the main source of lighting in outdoor scenes.
There is no problem combining dynamic and baked lighting so if there are lights that just can't be baked, it's easy to treat them independently.

Changing Geometry

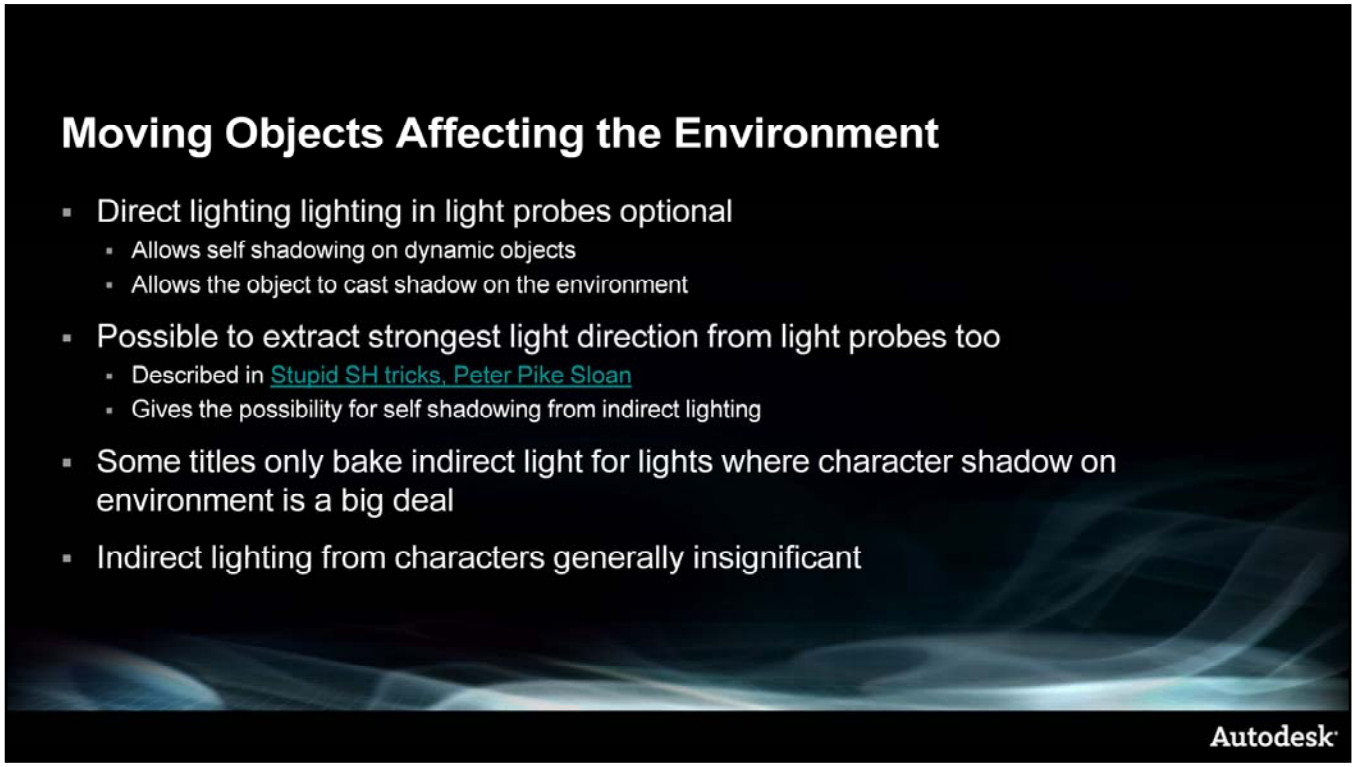
- Separate between local and global changes
- Local
  - Characters moving in a room
  - Small furniture
  - Bullet holes
- Global
  - Destroyed buildings
  - Destroyed walls

Autodesk

Generally solved without considering how the object affect the indirect lighting in its surrounding.

The Naïve approach is just adding direct lighting to moving objects. Tends to make characters to look out of place. Also in regions where no direct light reaches, the characters are completely black.
Light probes solves these problems elegantly and gives a nice pipeline for lighting characters and other moving objects.
Some games keep key lights out of the light probes and add them as more traditional direct lights.

A light probe in this case means a representation of the incoming light for an arbitrary point in space.

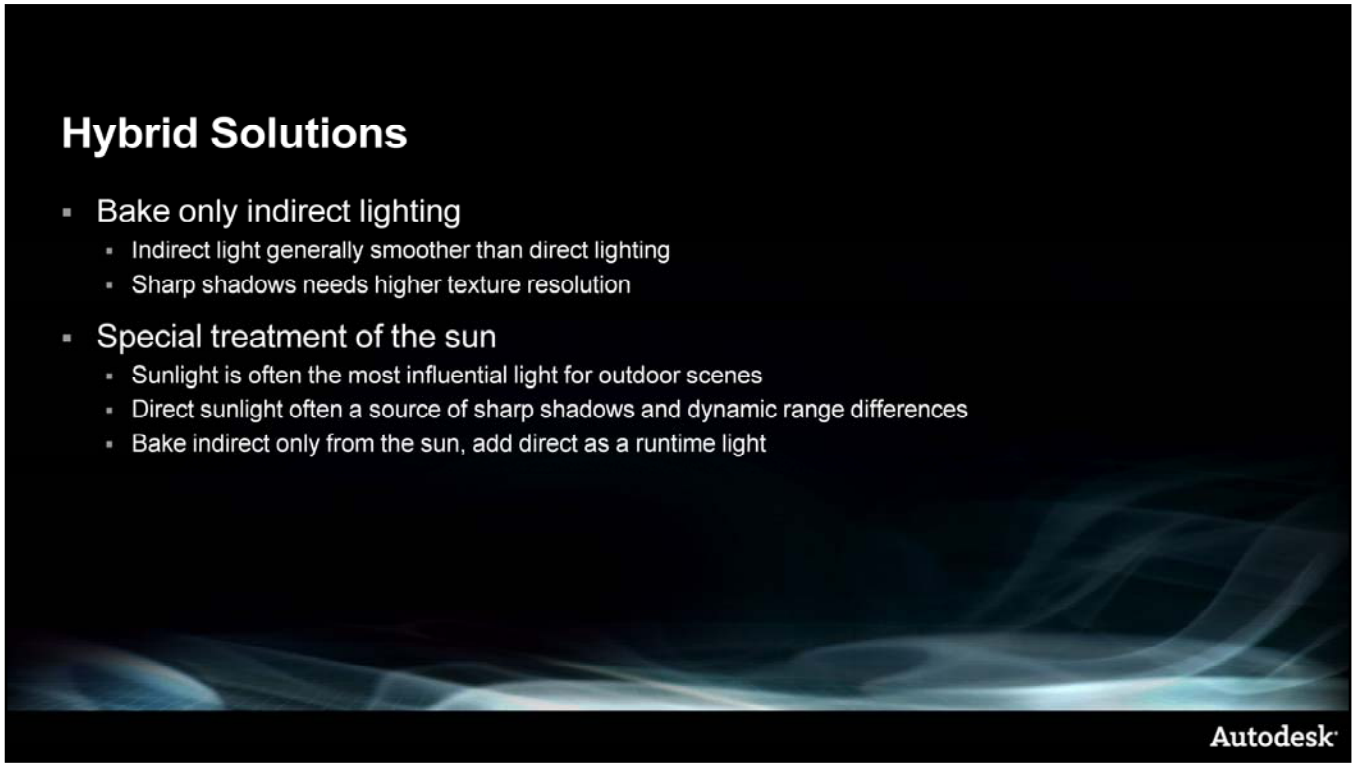


Subtracting a dynamic shadow of a light baked into a light map is not trivial.
Common to just multiply the light map color with a darkening factor.
More advanced would be to subtract the shadow from the lighting in case the light is not already occluded by the environment but it can give seams since the shadow border from the realtime and the bake shadow doesn't necessarily match up perfectly.

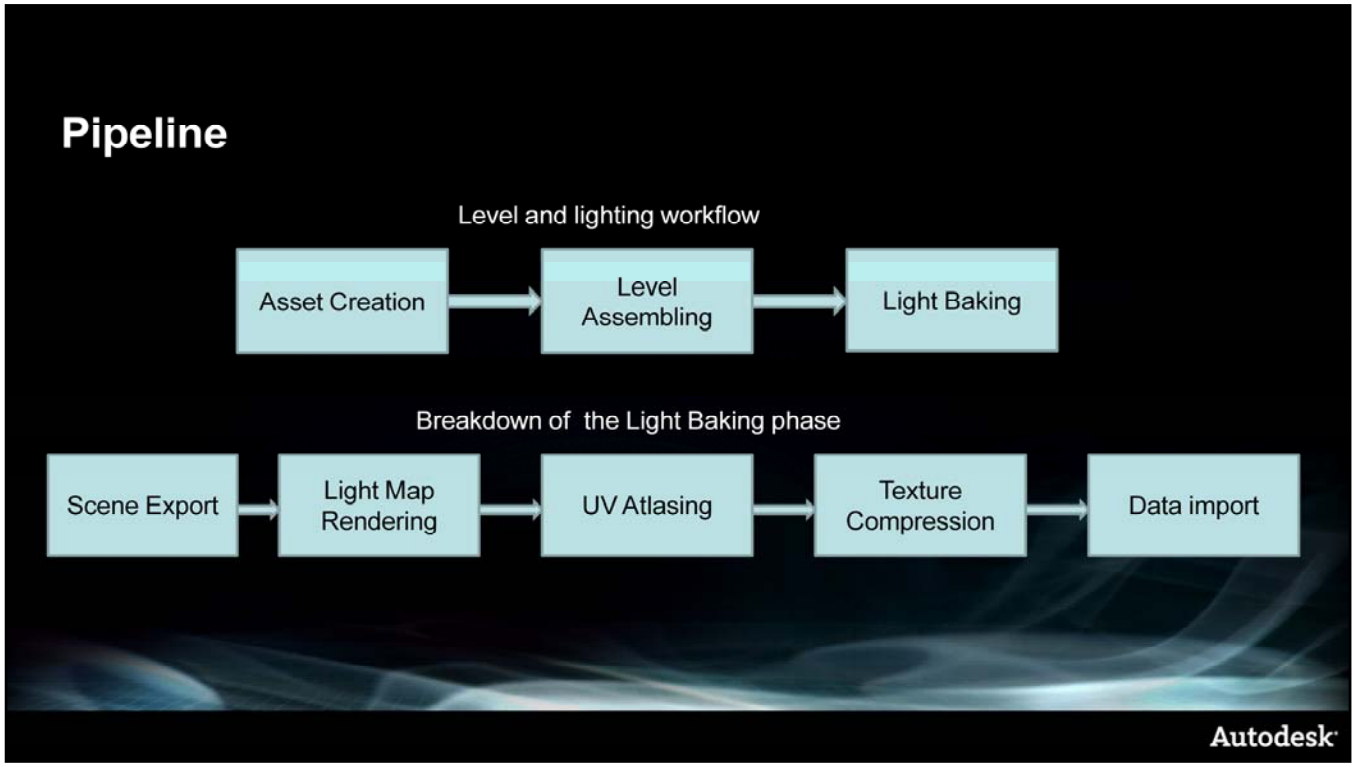
Global Geometry Changes

- Highly dynamic games tend to avoid global baked lighting
- Other subsystems tends to rely on or perform better on static geometry as well
  - Path Finding
  - Collision Detection
  - Game story often requires players following certain paths

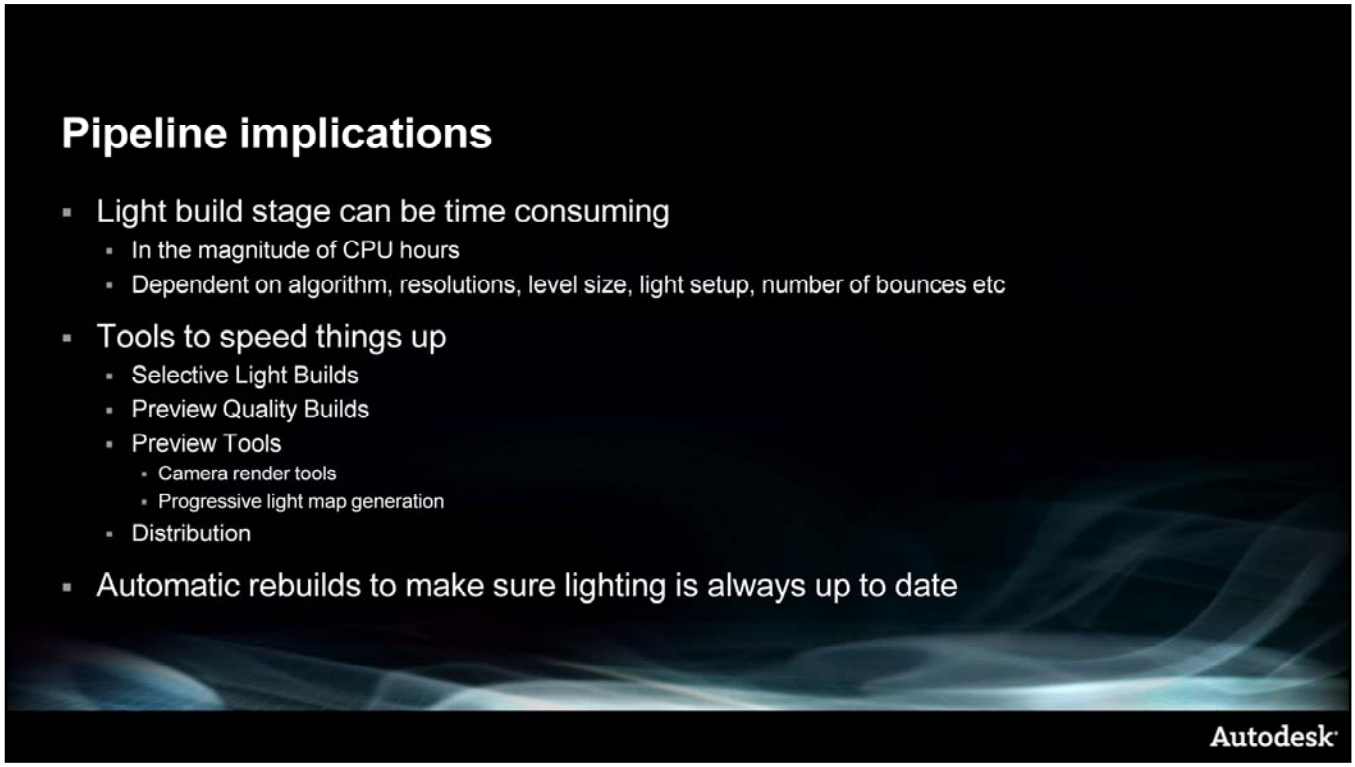
Autodesk



Most games are hybrids to some extent. Some lights or components of some lights are computed as dynamic lights.
Games having key lights outside the light maps often uses standard one component (as opposed to directional) light maps for the baked lighting. This saves them memory and the key lights will contribute with normal map and specular effects.



Not necessarily a one way process but changes in the assets or the scene assembly forces light builds. Scene assembly and lighting are iterative processes.
Asset Creation, tends to happen inside Maya, Max etc for meshes and paint programs for textures.
Scene Assembly, happens in a dedicated game editor or Maya, Max, or other modeling programs.
Lighting, happens inside the game editor and must happen when the entire scene is assebled since it is a global effect where everything can affect everything.
UV Atlasing refers to the process of creating a few large textures from many small. This gives no limitations on texture sizes, reduces number of texture switches when rendering and allows different objects uv spaces to overlap as long as no baked information overlaps.

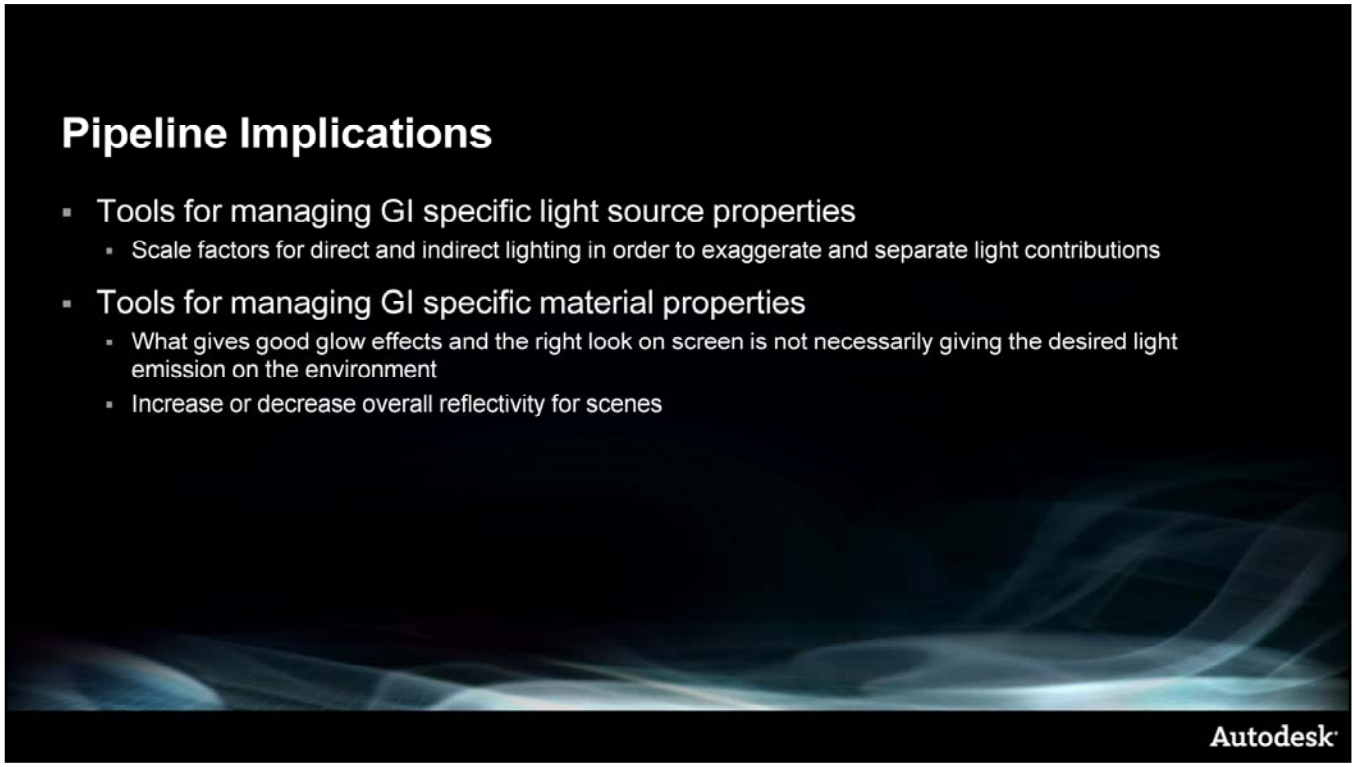


Tools to shorten the iteration times makes a big difference for artist productivity.
Being able to work on a room at the time can be an effective way of shortening iteration times.
Doing camera renderings inside the offline renderer can give fast previews. It also exposes any differences in what was exported and what the GI result was.
Distribution can dramatically cut render times.
<ShamelessProductPromotion>Check out Ernst from Autodesk for a cool preview tool for light map baking</ShamelessProductPromotion>
Automatic rebuilds is a convenient way to make sure there is always up to date lighting for the scene available after geometry modifications.

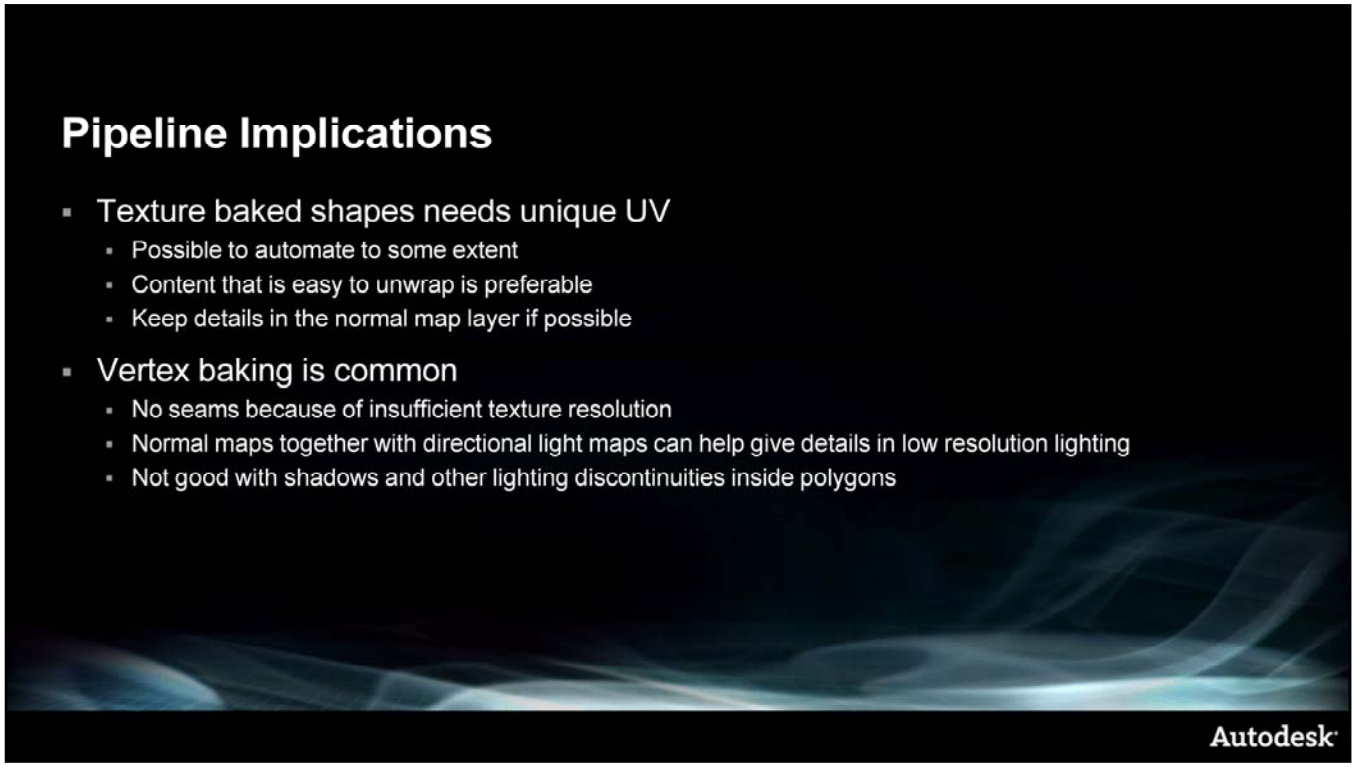
Pipeline Implications

- Clear separation of what is static and dynamic
  - Both for Lights and Geometry
- Tools for placing and managing light probes in levels
  - Grids
  - Hierarchical grids
  - Arbitrary points
- Tools for managing texture resolutions and bake type

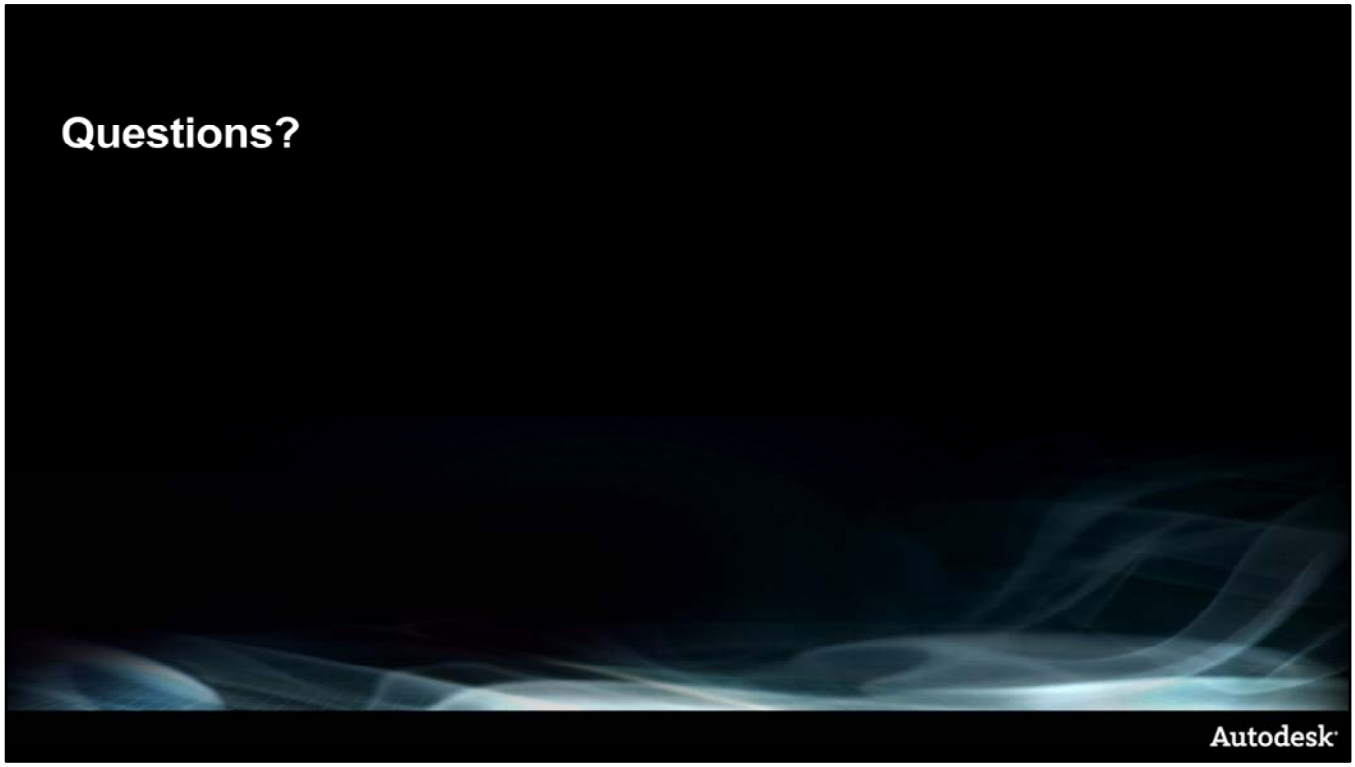
Autodesk



These are just examples. Even if GI should give automatic realism, artists will always want control of the exact end results.
Too much artist control can make things less realistic. Lots of SIGGRAPH 2010 presentations praised energy conserving materials since they guarantee a consistent look and levels for content.



Unwrapping can be done automatically, but good artists can improve the results with manual work. Ideally the UV layout should be continuous where the surface is smooth and discontinuous where the surface isn't smooth to avoid bilinear filtering over sharp edges in the lighting.
Unwrapping tends to work better on low polygon normal mapped geometry as opposed to high detail meshes since folds and wrinkles should preferably end up as seams in the uv layout. If doing this the UV layout will be constructed from lots of small islands which will waste texture memory on padding.
Vertex baking is more common than most people thinks and supports directional light maps too.


Questions?
Autodesk