

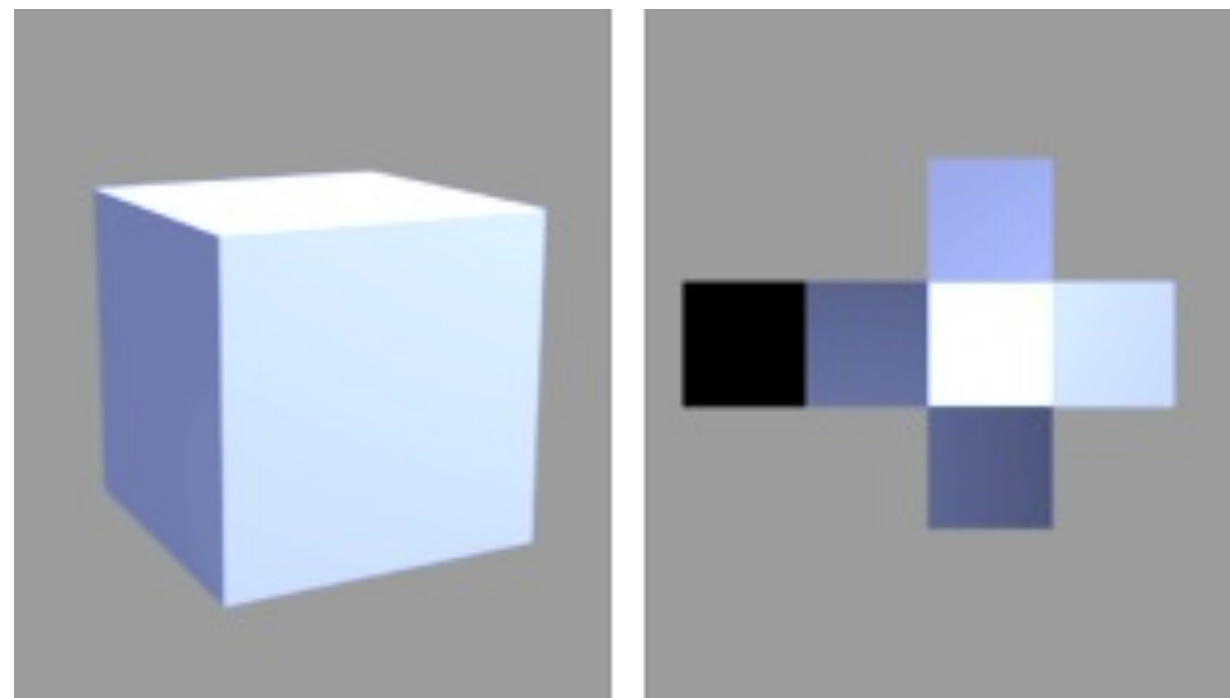


Light mapping

Applying pre-calculated lighting to a model

Saves real-time processing time for models with static lighting

Allows high-quality lighting with high performance



(Image from Wikipedia)

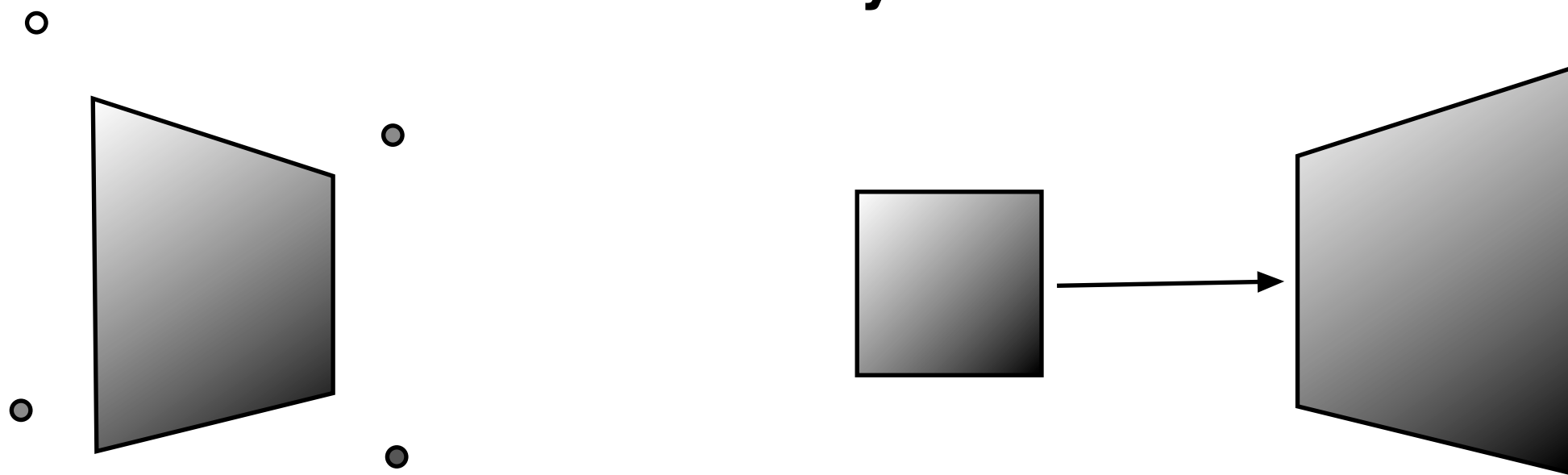


Light mapping

Two approaches:

- **Vertex-level light mapping**
- **Light map textures**

Both methods are high-performance and require little memory!





Vertex-level light mapping

Calculate lighting per vertex

Apply with glColor

Render with Gouraud shading

Trivial to use with textures. Very fast and low memory demand. Limited quality,

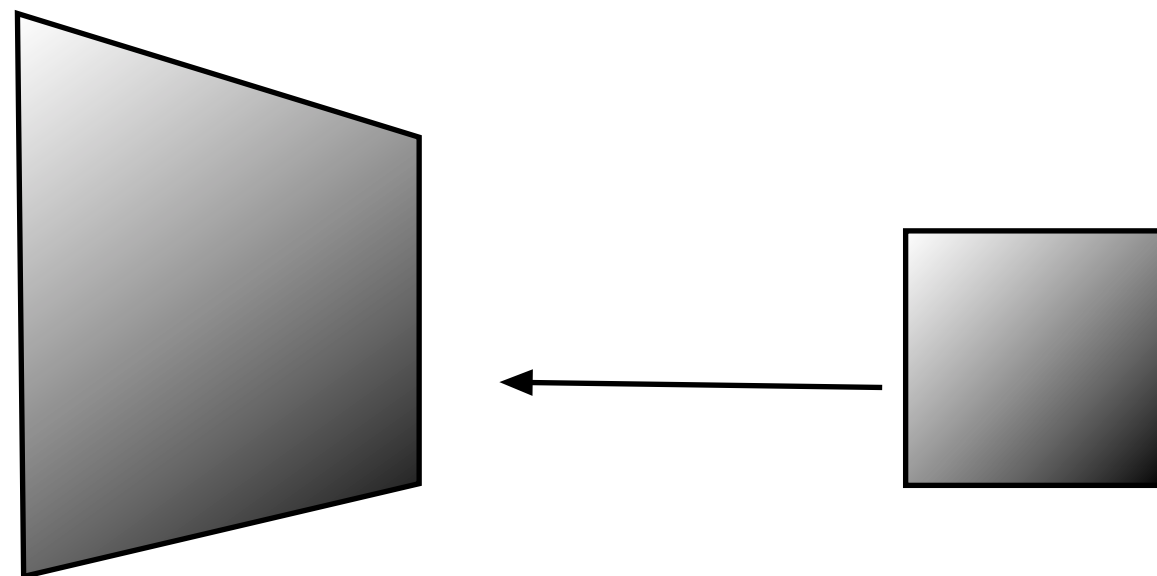


Light map textures

Pre-calculate image any way you like (e.g. radiosity)

Allows arbitrary precision with low processing demand

Images usually very small, but can be made large when needed

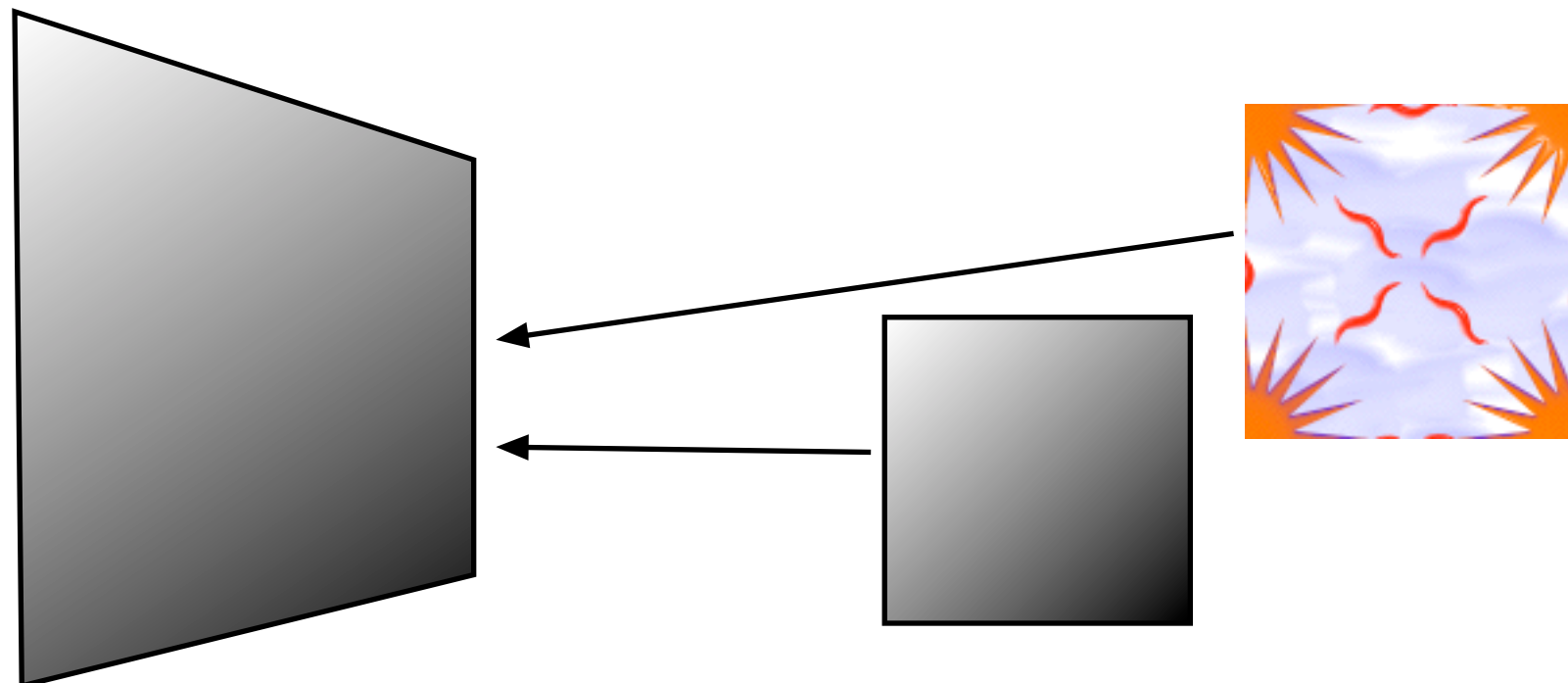




Light map textures

Texture maps and light maps can be applied to the same surface by multitexturing.

Texture values (texels) are multiplied by light map values.



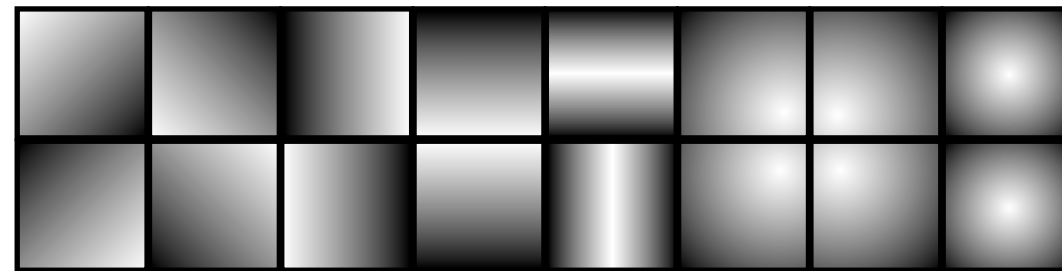


Light map textures

Generation:

- 1. Hand-painted**
- 2. Ray-tracing**
- 3. Radiosity**
- 4. Provided by 3D tools**

Generate a palette of light maps, reuse within some tolerance





Light mapping today

Usually not the most vital technique today, but can save performance by pre-calculating light that does not change over time (like shadows from static objects).