Real – Time Rendering

Ambient Occlusion

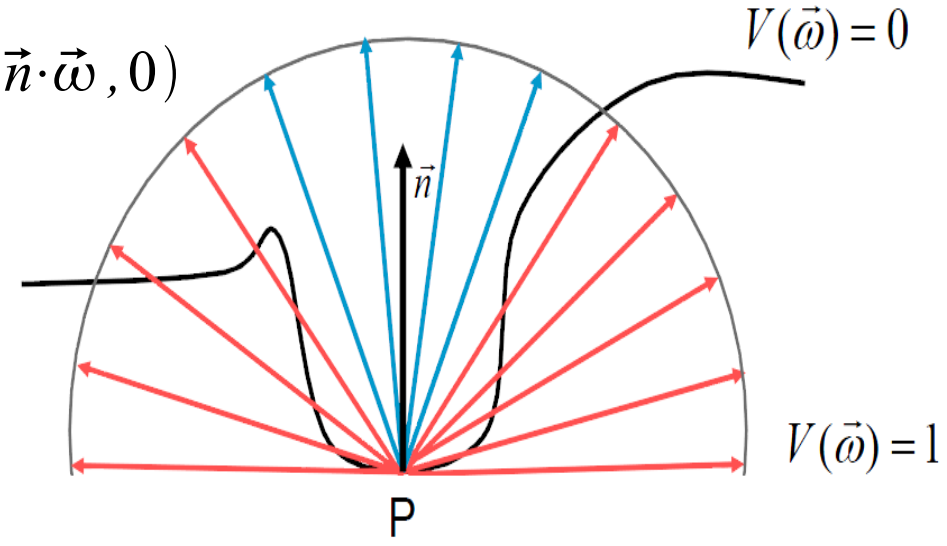
Michal Červeňanský
Juraj Starinský

Motivation

- Definitions of AO
  - diffuse illumination from the sky
  - shadow from the sky illumination
- Ambient light is NOT constant for all points
- Perceptual clues – depth, curvature, spatial proximity

Theory

- For each illuminated point
  - Integrate over hemisphere
  - $$AO(P,\vec{n}) = \frac{1}{\pi}\int_{\Omega} V(P,\vec{\omega})\cdot max(\vec{n}\cdot\vec{\omega},0)\,d\,\vec{\omega}$$
  - $$P - illuminated\ point$$
    $$\vec{n} - normal\ at\ point\ P$$
  - $$V(P,\vec{\omega}) - visibility\ function$$
  - Approximate with sum
  - $$AO(P,\vec{n}) = \frac{1}{\pi}\sum_{\Omega} V(P,\vec{\omega})\cdot max(\vec{n}\cdot\vec{\omega},0)$$
  - Visibility (occlusion) fc.
    - raytracing
  - Time consuming



Simplification

- Monte carlo – few random rays

$$AO(P,\vec{n}) = \frac{1}{n}\sum_{i=0}^{n-1} V(P, rnd\vec{}_i\,\omega)\cdot max(\vec{n}\cdot rnd\vec{}_i\,\omega, 0)$$

$$rnd\vec{}_i\,\omega = i-th\ random\ vector$$

- Still not real-time
  - Static geometry – precompute into AO maps
    - Like light maps
  - Dynamic geometry – more tricks :-)

Dynamic geometry object space

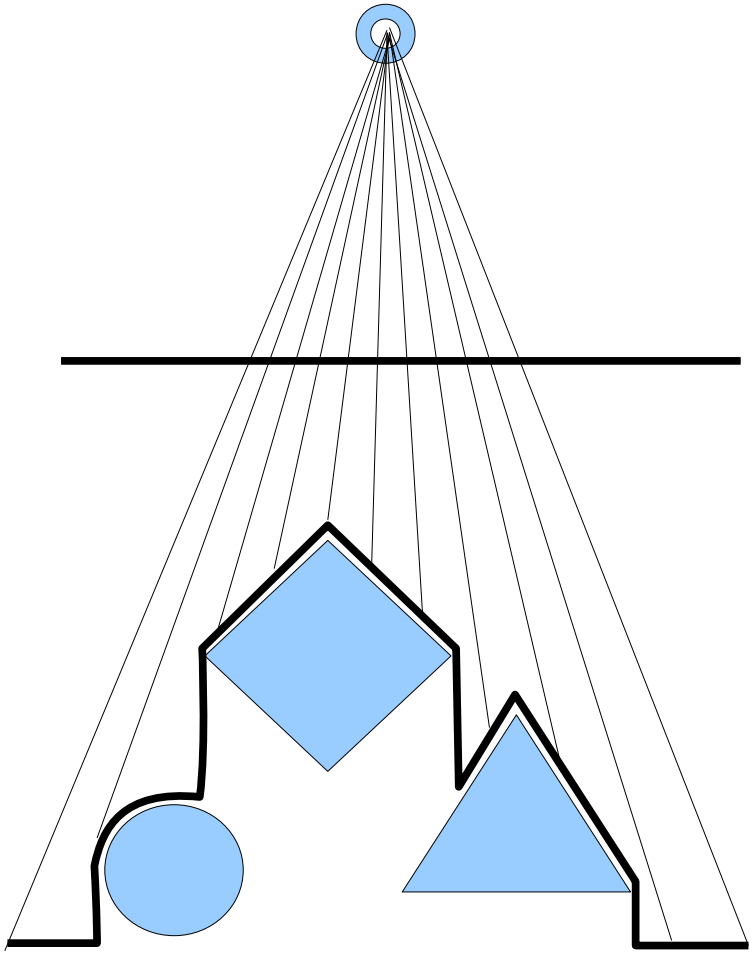
- Simplify geometry
  - Hierarchy of simplifications
- Compute per-vertex occlusion by simplified geometry
  - Hierarchy level according to distance
  - Do not trace distant objects
- Performance dependent on geometry complexity

Dynamic geometry screen space

- State of the Art
- Post-processing effect
  - Geometry independent
  - Processes also static geometry
  - No additional structures or maps
- Requires
  - Depth buffer
  - Normal buffer (normal for every pixel)
    - Calculated / Derived from depth buffer

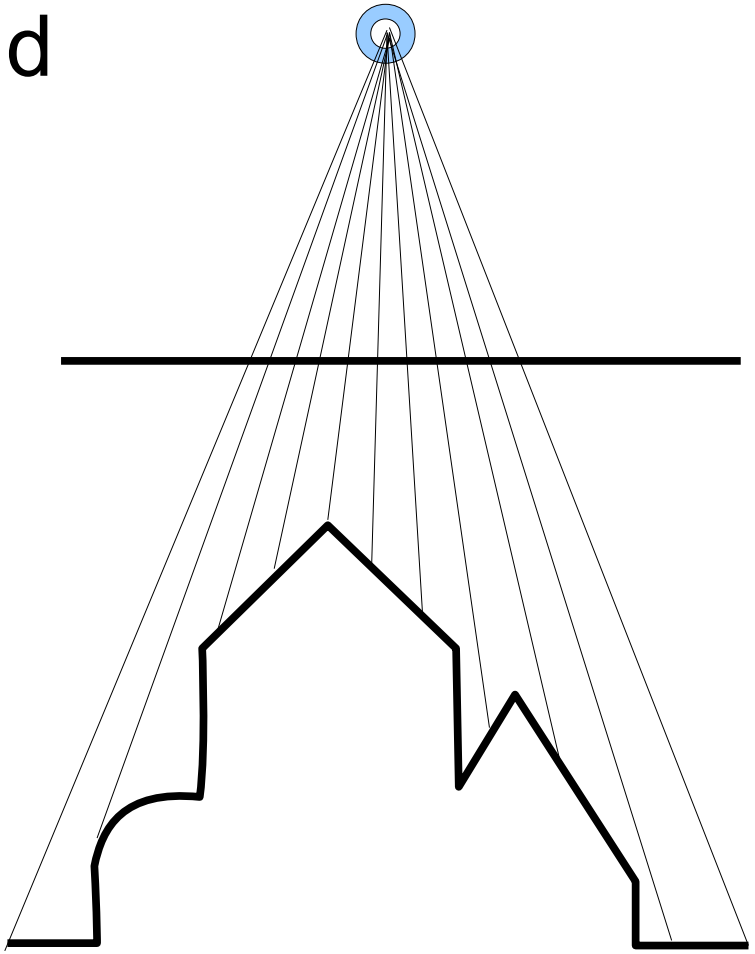
Screen Space AO

- Depth-buffer



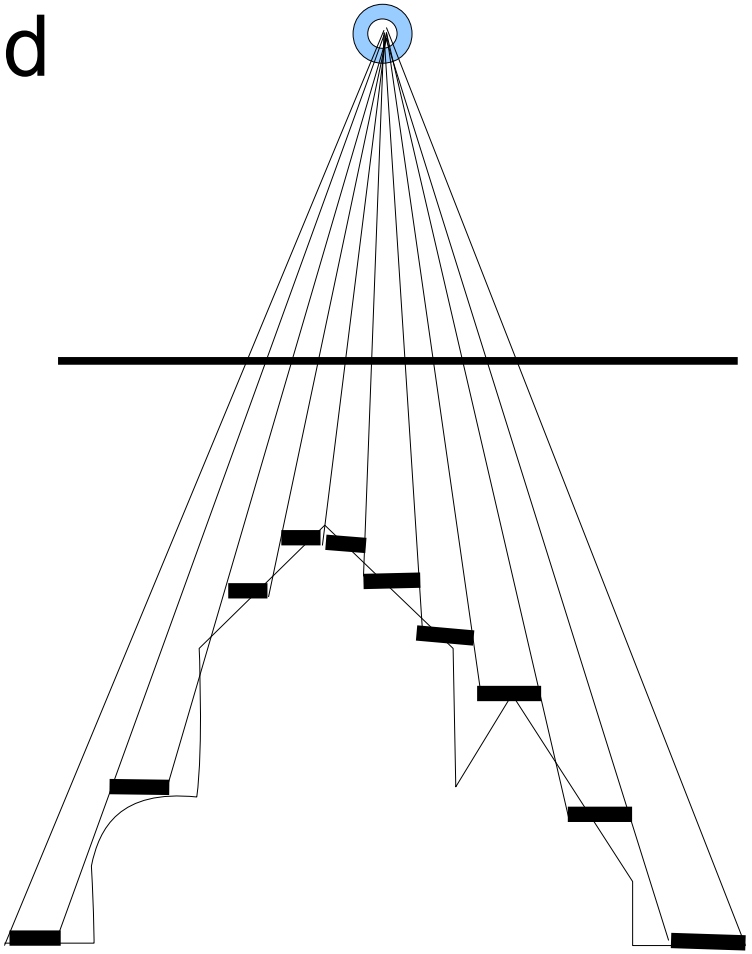
Screen Space AO

- Depth-buffer = height field



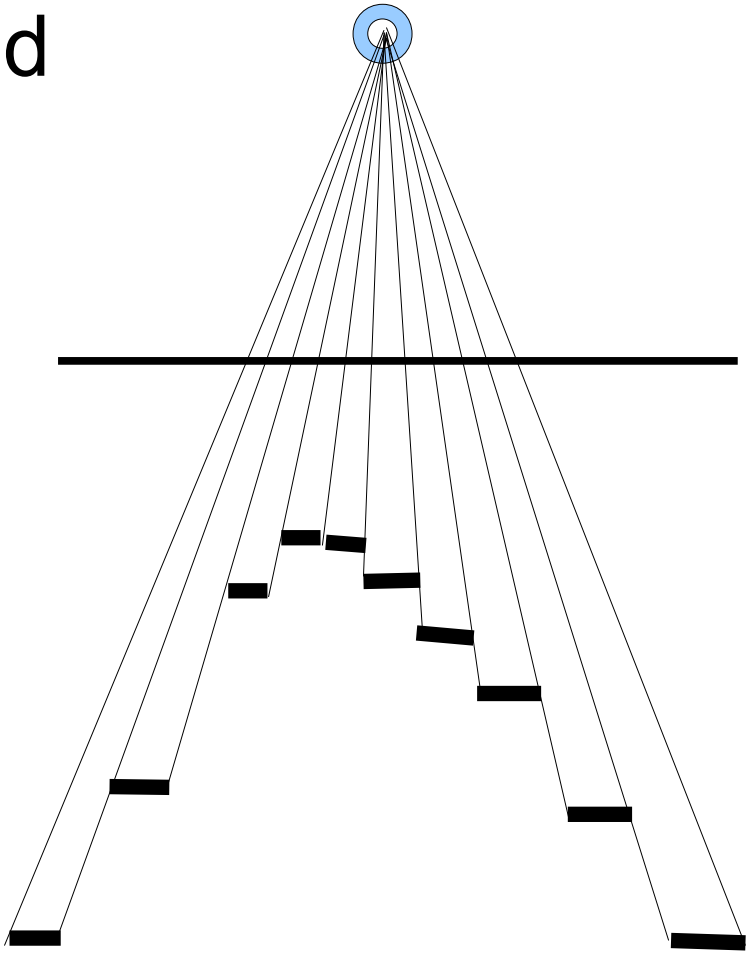
Screen Space AO

- Depth-buffer = height field
  - Every pixel = projected sphere/disc from world space



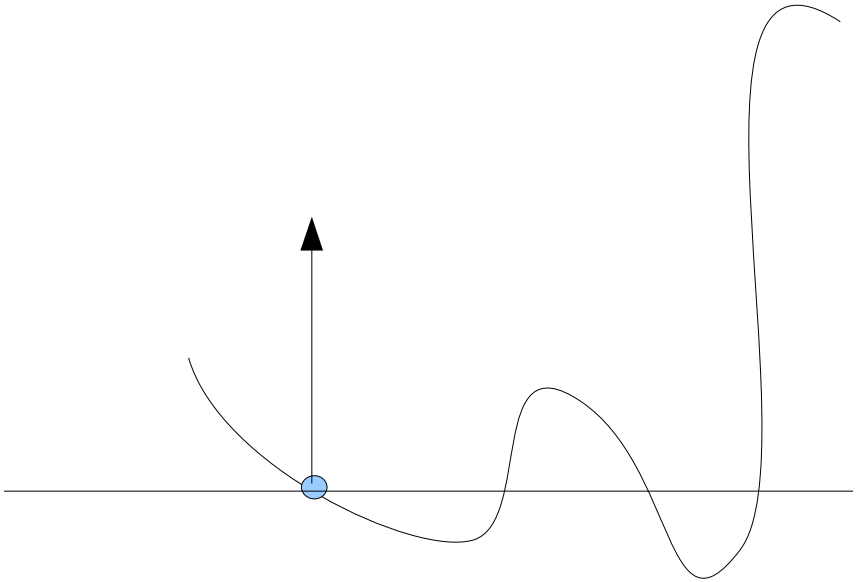
Screen Space AO

- Depth-buffer = height field
  - Every pixel = projected sphere/disc from world space
    - Approximation of scene geometry



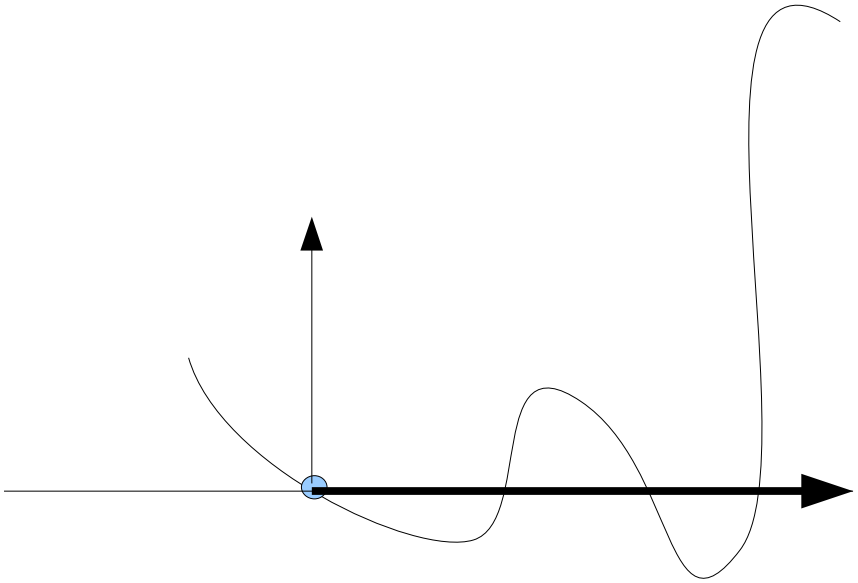
Screen Space AO

- Occlusion in height filed
  - Starting from point P



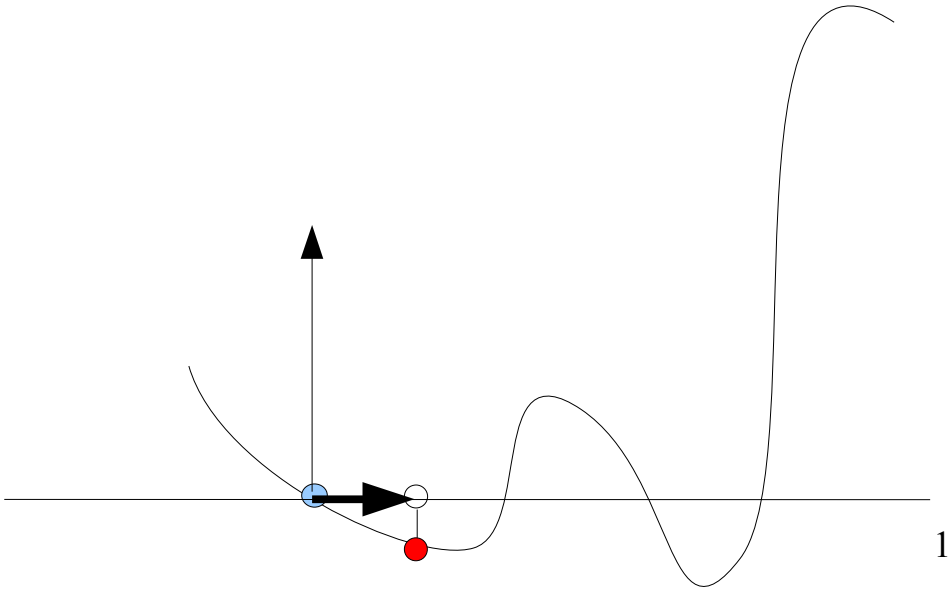
Screen Space AO

- Occlusion in height filed
  - Starting from point P
  - Sampling along ray
    - Ray tracing



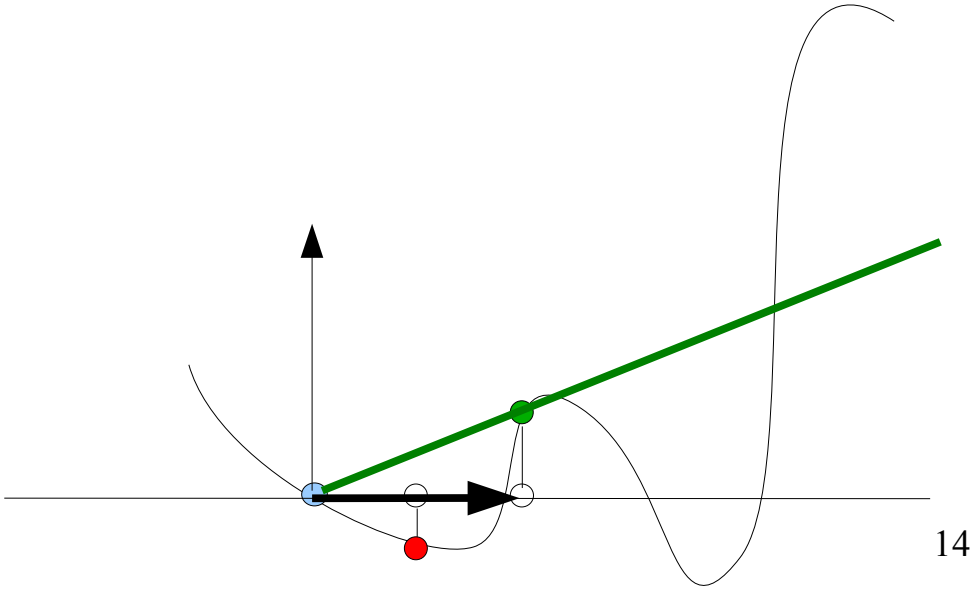
Screen Space AO

- Occlusion in height filed
  - Starting from point P
  - Sampling along ray
    - Ray tracing
  - Search for horizon



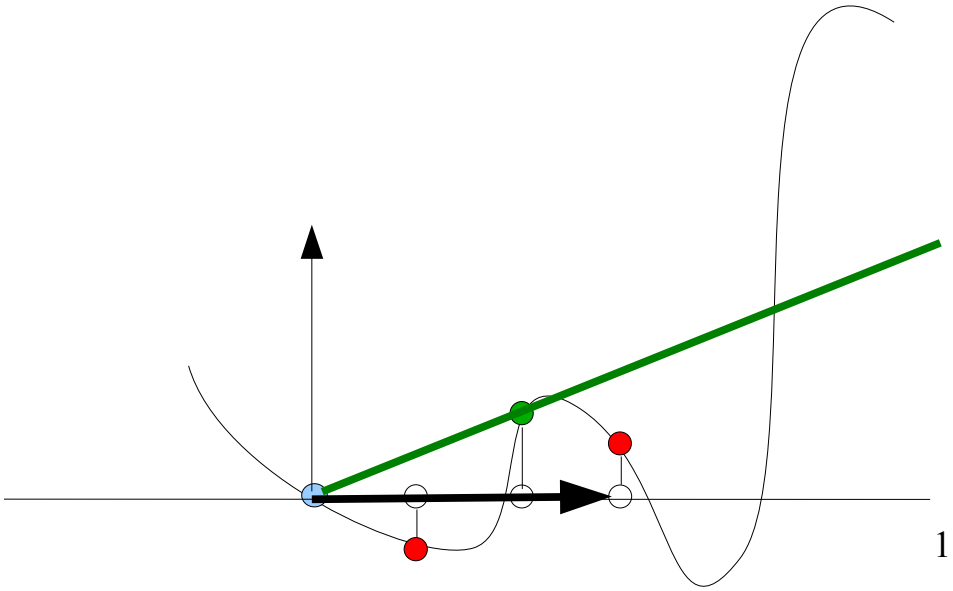
Screen Space AO

- Occlusion in height filed
  - Starting from point P
  - Sampling along ray
    - Ray tracing
  - Search for horizon



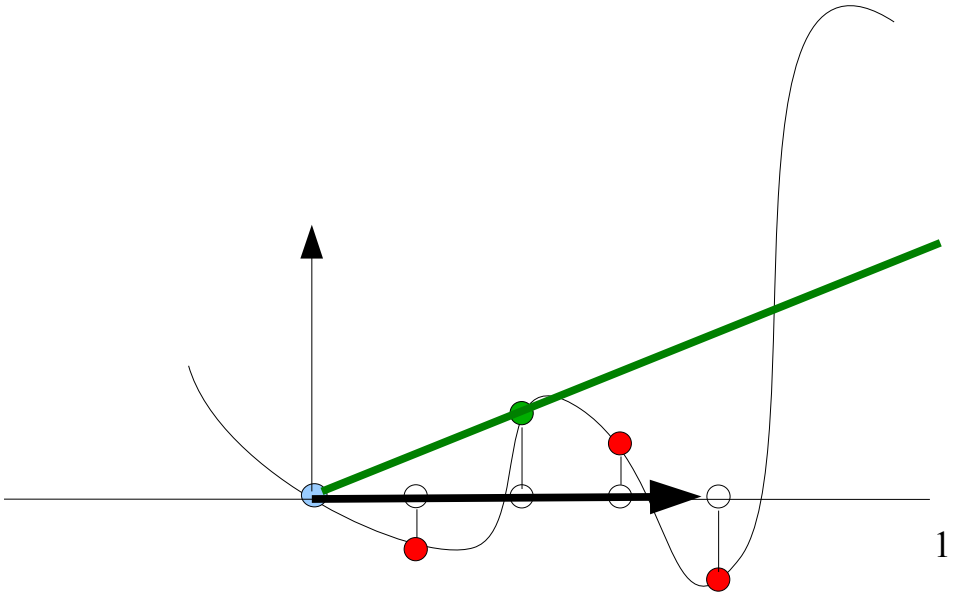
Screen Space AO

- Occlusion in height filed
  - Starting from point P
  - Sampling along ray
    - Ray tracing
  - Search for horizon



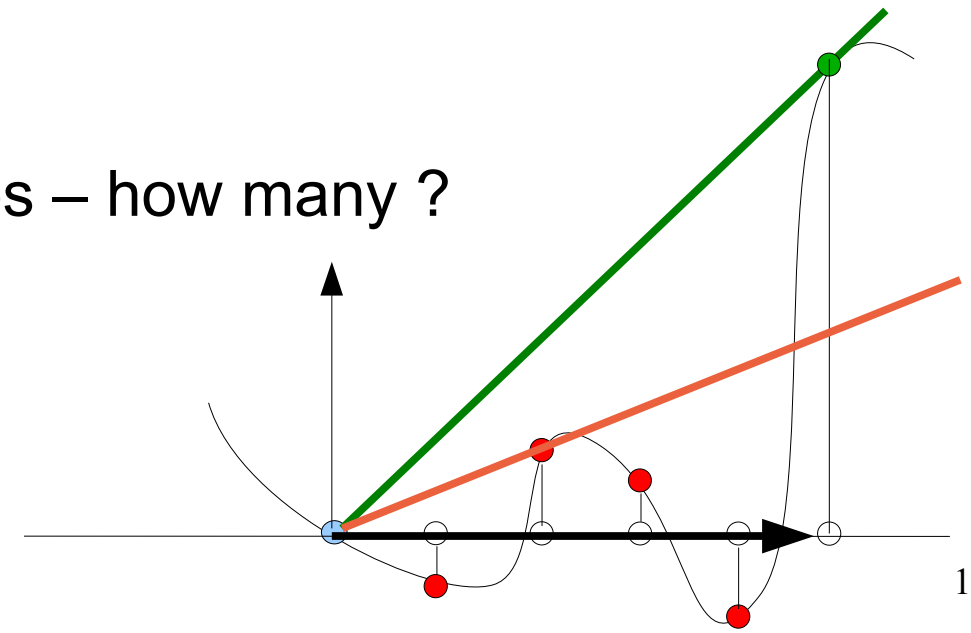
Screen Space AO

- Occlusion in height filed
  - Starting from point P
  - Sampling along ray
    - Ray tracing
  - Search for horizon



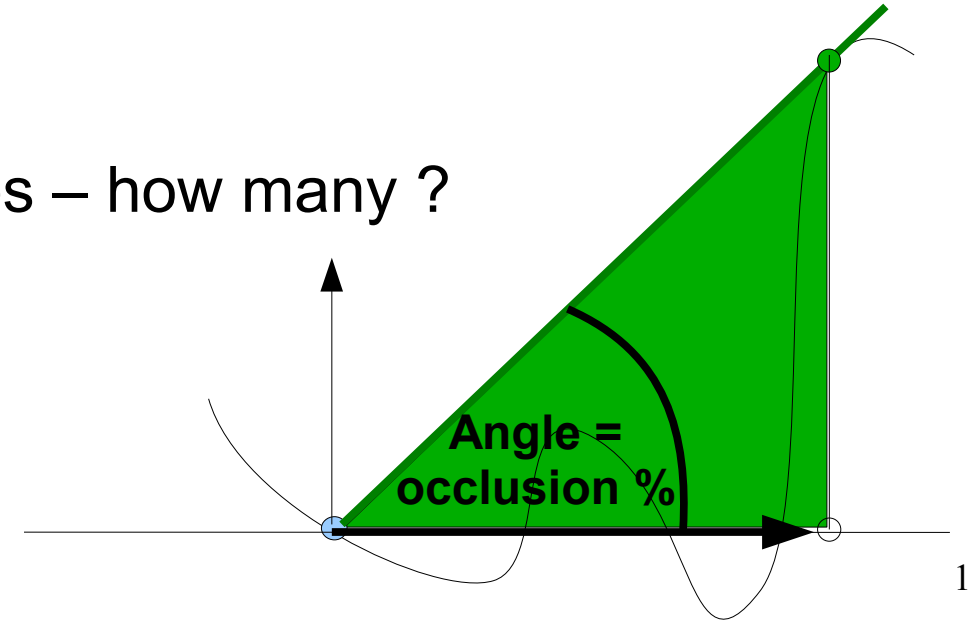
Screen Space AO

- Occlusion in height filed
  - Starting from point P
  - Sampling along ray
    - Ray tracing
  - Search for horizon
    - Few iterations/samples – how many ?



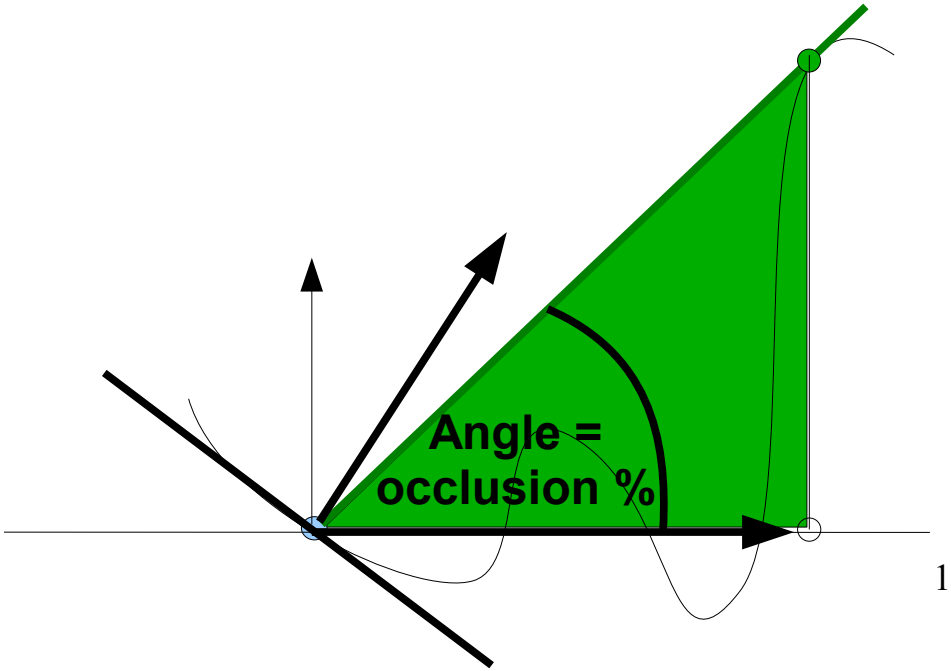
Screen Space AO

- Occlusion in height filed
  - Starting from point P
  - Sampling along ray
    - Ray tracing
  - Search for horizon
    - Few iterations/samples – how many ?
    - Horizon angle (ha)



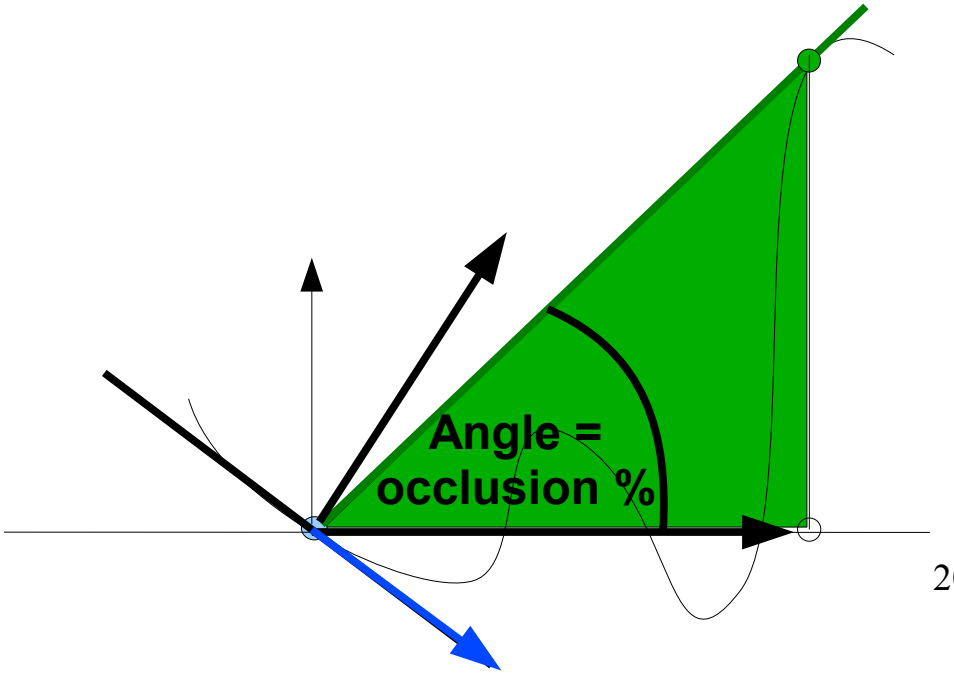
Screen Space AO

- Occlusion in height filed
  - Occlusion dependent on pixel normal



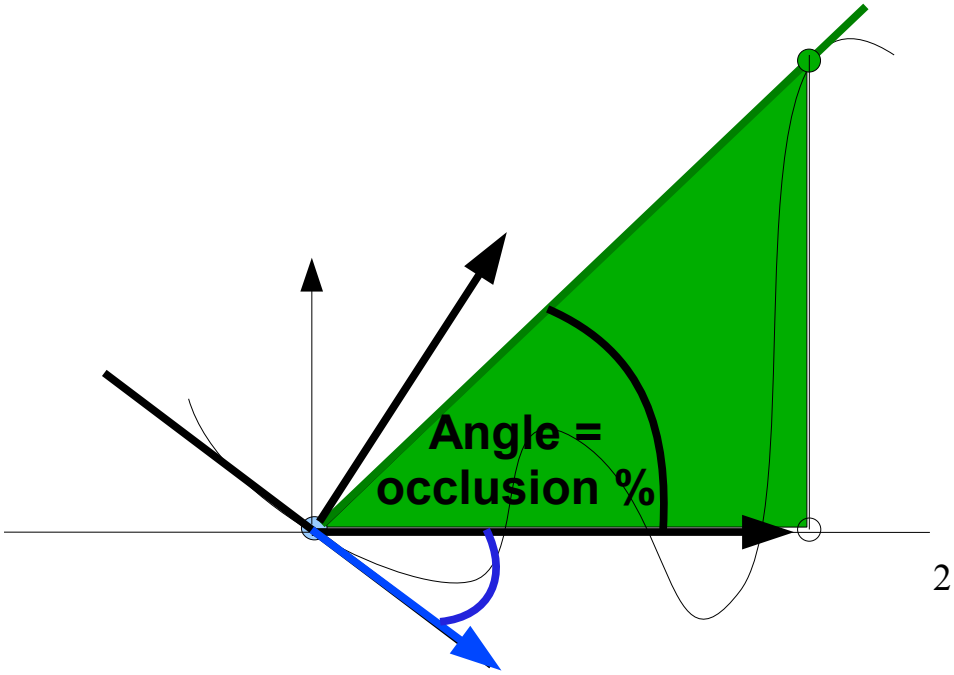
Screen Space AO

- Occlusion in height filed
  - Occlusion dependent on pixel normal
    - Tangent vector



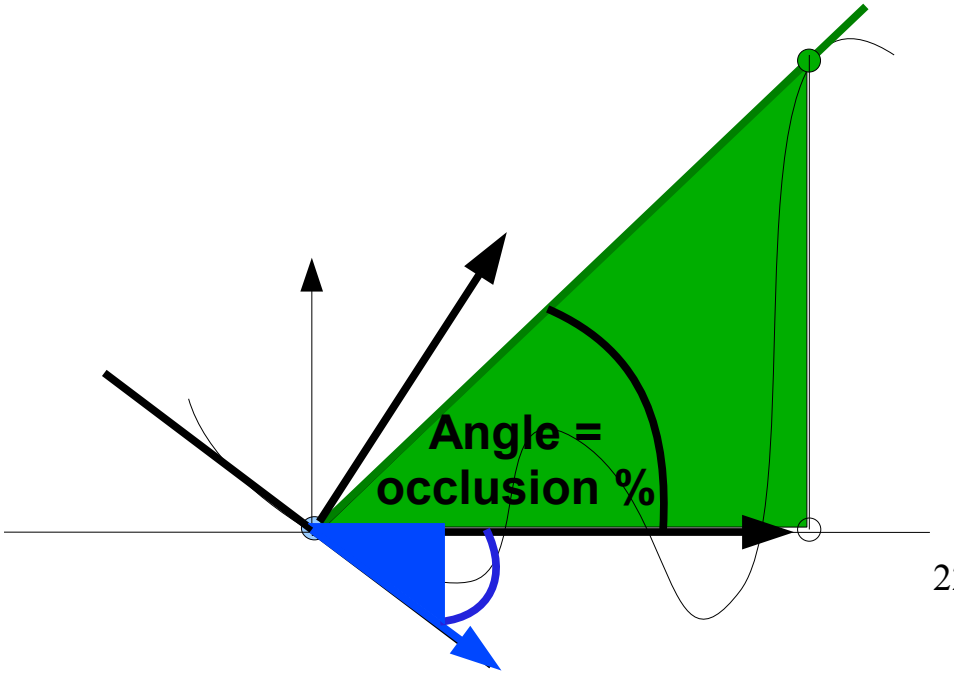
Screen Space AO

- Occlusion in height filed
  - Occlusion dependent on pixel normal
    - Tangent vector
    - Tangent angle
      - Signed value !



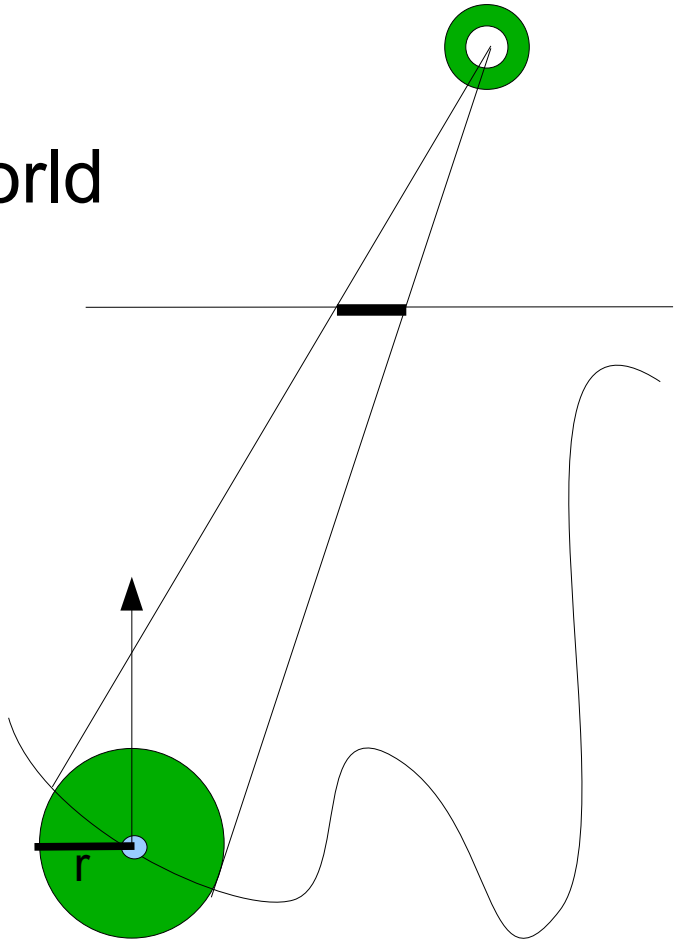
Screen Space AO

- Occlusion in height filed
  - Occlusion dependent on pixel normal
    - Tangent vector
    - Tangent angle (ta)
      - Signed value !
    - AO = ha-ta



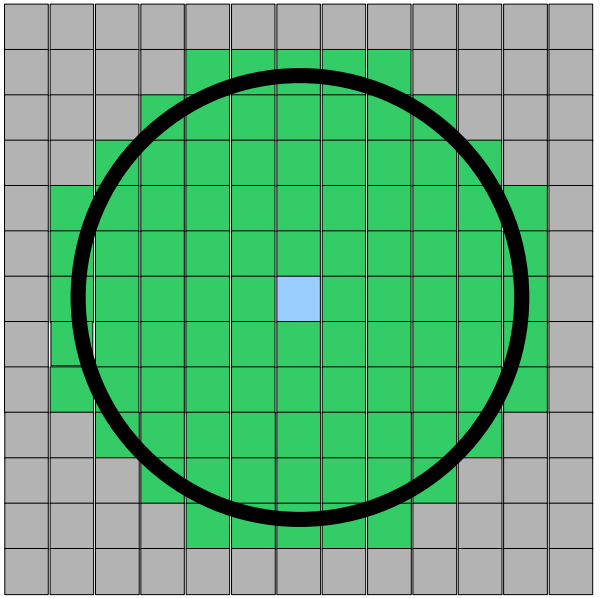
Screen Space AO

- How many iterations/samples per ray ?
  - Limit region of interest in world space by radius r
    - Constant for all points
  - Calculate projection of a sphere with radius r
    - Region of interest in screen space



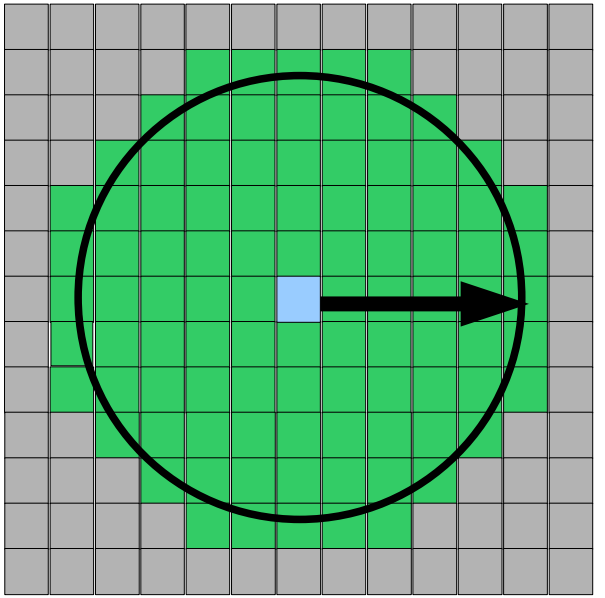
Screen Space AO

- Projected region of interest



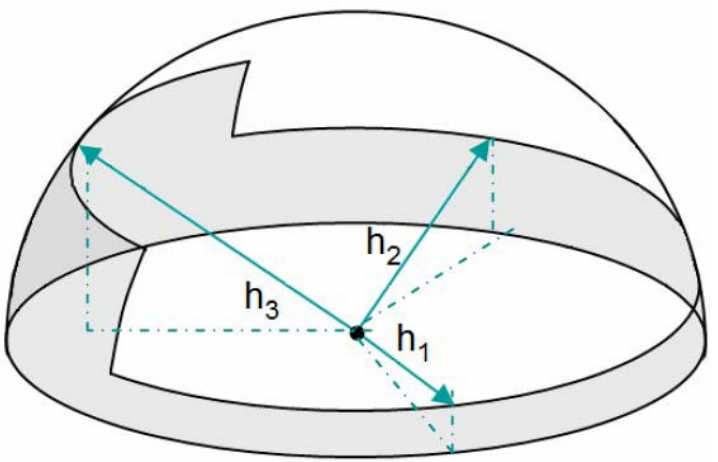
Screen Space AO

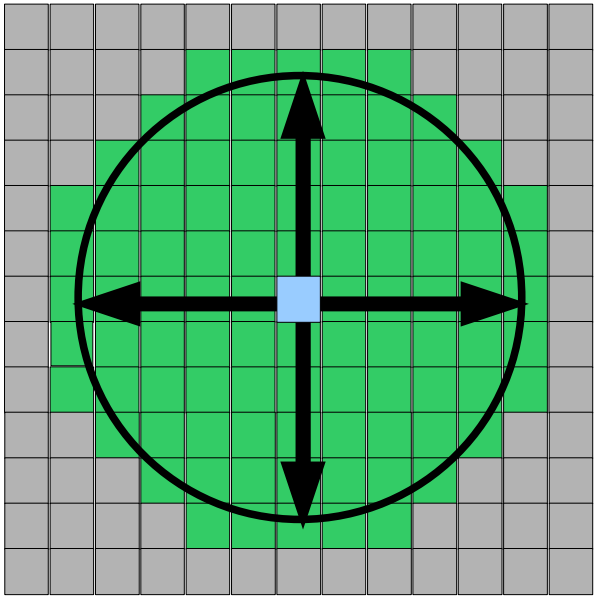
- Projected region of interest
  - Sample along ray



Screen Space AO

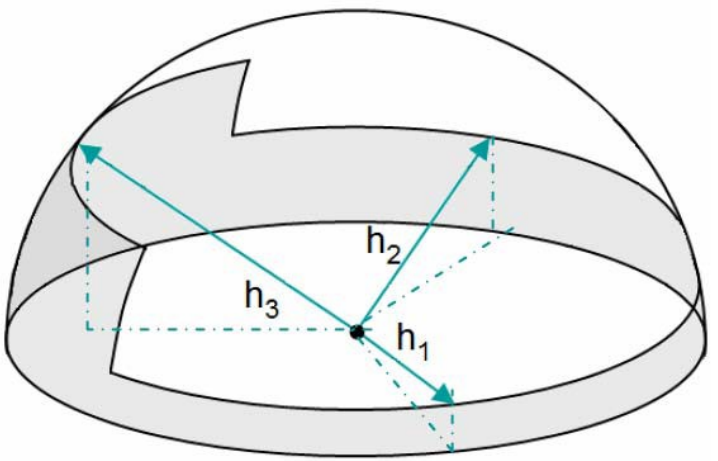
- Projected region of interest
  - Sample along ray
  - Multiple rays
    - User defined number

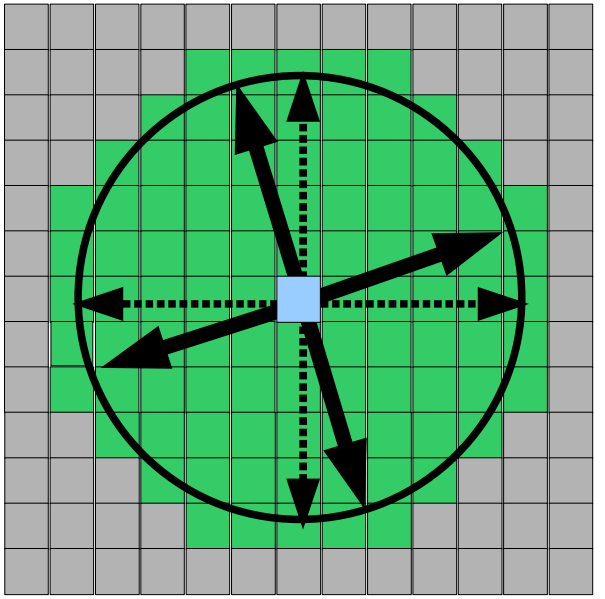




Screen Space AO

- Projected region of interest
  - Sample along ray
  - Multiple rays
    - User defined number
  - Random rotation of rays per pixel
    - Hides aliasing
    - Creates noise
  - Jitter samples along ray



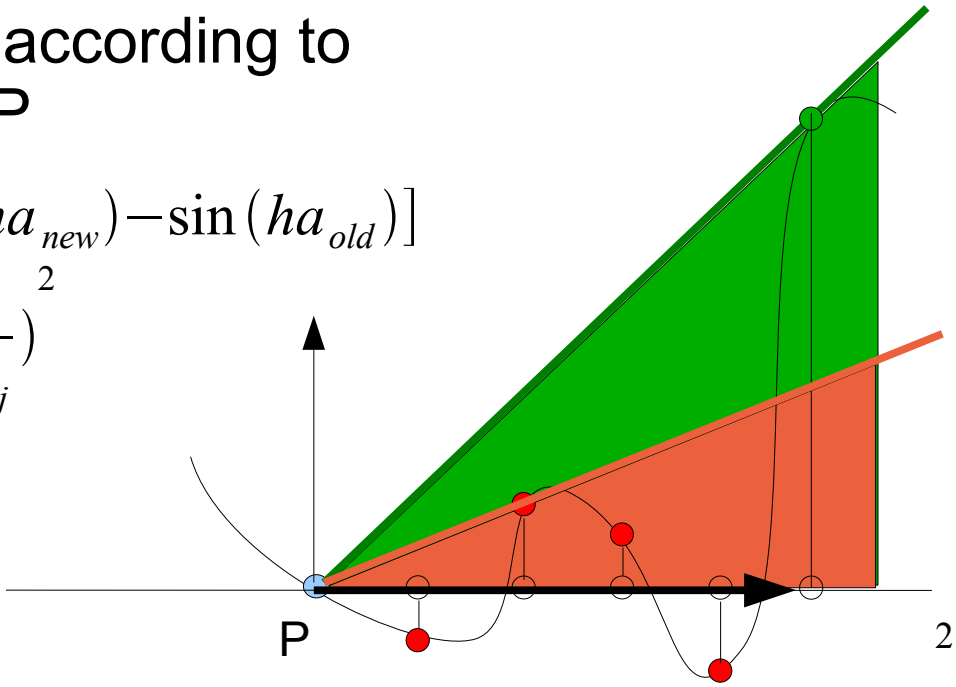


Screen Space AO

- Distance attenuation
  - Sample along ray
  - Every new horizon angle (ha)
    - Attenuate contribution according to sample distance from P

$$AO_{new} = AO_{old} + f_a(d_{new}) \cdot [\sin(ha_{new}) - \sin(ha_{old})]$$

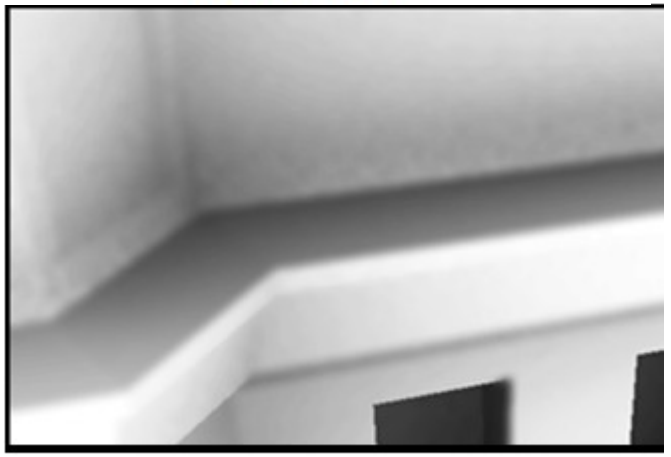
$$f_a(d) = 1 - \left(\frac{d}{r_{proj}}\right)^2$$



Screen Space AO
additional tricks

- Process downscaled depth/normal buffer
  - Better performance
  - Lower quality
- Upscale result
- Use special blur according to depth
  - Depth-dependent Gaussian blur
  - Hides randomization noise
  - Higher quality

Without Blur



With 15x15 Blur

References

- **Simon Green “DirectX10/11 Visual Effects”, Game Developers Conference 2009, March 2009**
  http://www.gdconf.com/conference/Tutorial%20Handouts/100_Advanced%20Visual%20Effects%20with%20Direct
- **Louis Bavoil, Miguel Saintz “Image-space Horizon-based Ambient Occlusion”, SIGGRAPH 2008, Talk Program, August 2008**
  http://developer.download.nvidia.com/presentations/2008/SIGGRAPH/HBAO_SIG08b.pdf
- **Louis Bavoil, “Screen Space Ambient Occlusion”, Whitepaper, August 2008,**
  http://developer.download.nvidia.com/SDK/10.5/direct3d/Source/ScreenSpaceAO/doc/ScreenSpaceAO.pdf
- **Louis Bavoil, “Real-time Depth Buffer Based Ambient Occlusion”, Game Developers Conference 2008, March 2008,**
  http://developer.download.nvidia.com/presentations/2008/GDC/GDC08_Ambient_Occlusion.pdf
- **Megan Fox, Stuart Compton, “Ambient Occlusive Crease Shading”, Game Developer Magazine, March 2008,**
  http://www.shalinor.com/research.html
- **Martin Mittring, “Finding Next Gen – CryEngine 2” SIGGRAPH 2007, Advanced Real-Time Rendering in 3D Graphics and Game Course**
  http://delivery.acm.org/10.1145/1290000/1281671/p97-mittring.pdf?key1=1281671&key2=9942678811&coll=ACM
- **Perumaal Shanmugam, Okan Arikan, “Hardware accelerated ambient occlusion techniques on GPUs” I3D '07 : Proceedings of the 2007 symposium on Interactive 3D graphics and games**
  http://perumaal.googlepages.com/ao.pdf