Chapter 8 Sampling and Aliasing

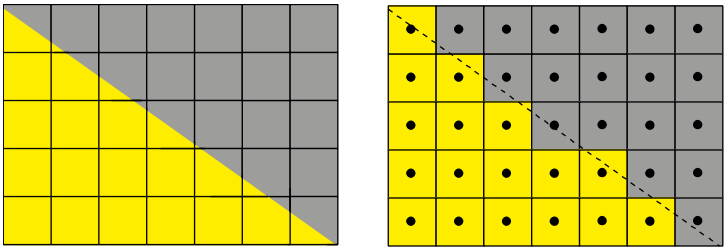
- Introduction
- Aliasing effects
- Remedies
- Antialiasing
- Filtering

Introduction

- Computers are **discrete** devices that display a **finite** number of pixels, work with a finite number of colors, and in the case of ray tracing, sample scenes at finite number of discrete points.
- As such, most ray-traced images are subject to **aliasing**, where an alias means a **substitute**. Where images are substitutes for the real scenes we are trying to render.
- Most obvious effects of aliasing are **jaggies**, which are the staircase appearance of sharp edges.

Aliasing Effect

- Many of the aliasing effects visible in ray-traced images are caused by the fact that ray tracing is a **point-sampling** process, where we sample scenes with infinitesimally thin rays.
- To see how this produces aliasing, look at the figure below.

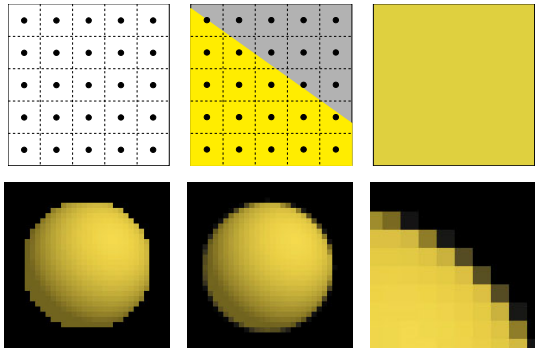


Remedies

1. Increase the Image Resolution
   - The simplest antialiasing technique is to render the scene at a higher pixel resolution, because that requires no additional programming.
   - However, this technique has the following problem: it doesn't eliminate the aliasing. The jaggies would still be there.
   - Fortunately, due to the limited angular resolving power of our eyes, there limits to how high the resolution has to be before we can't see any aliasing.
   - There are, however many variables involved, including whether we look at the image on a computer screen or as a printed image, the lighting conditions, viewing distance.

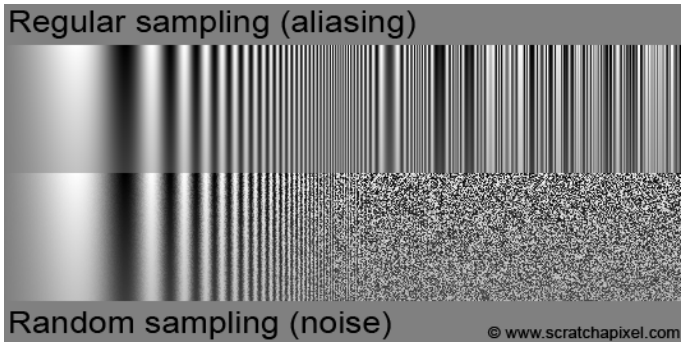
Remedies-cont.

2. Regular Sampling
   - With regular sampling, we shoot rays on a regular grid inside each pixel as below



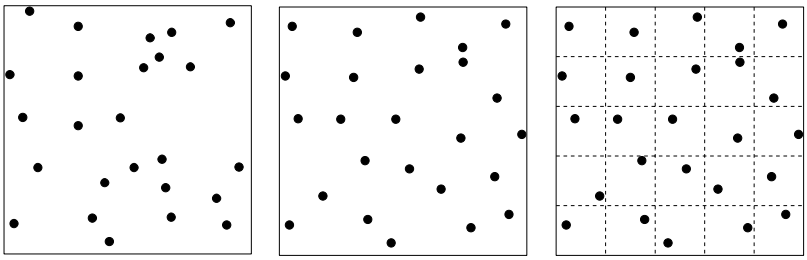
Remedies-cont.

3. Random Sampling
   - Most aliasing can be replaced by **noise** if we use rays that are **randomly** distributed over the pixel surfaces.



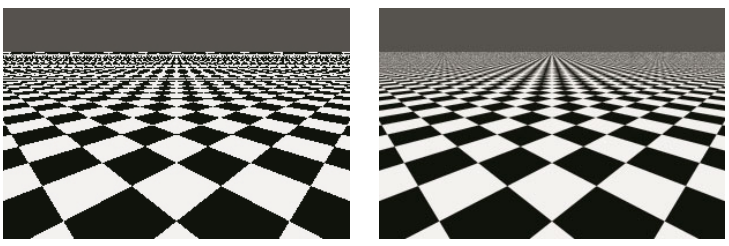
Remedies-cont.

4. Jittered Sampling
   - **Random sampling** is not the best way to distribute rays over the pixel because the samples can clump together and leave gaps.
   - A better strategy is force a more even distribution of the samples over the pixel, while still maintaining the randomness.



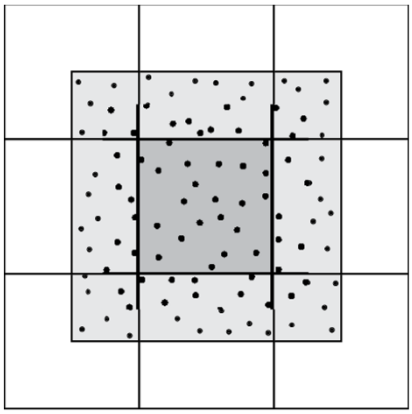
Antialiasing Fine Detail

- **Textures** are often difficult to antialias because they can contain fine detail. In fact, some images contain texture detail that is infinitely small.
- The Figure below shows a perspective view of a plane with checker texture. We use **64 jittered** samples per pixel, which improves the image, but the checkers still break up near the horizon.



Filtering

- In ray tracing **filtering** involves computing a pixel color by using rays that are **outside** the pixel boundary, as well as **inside**.
- In this figure, the dark-gray square in the center is the pixel being ani-aliased, and the filter area is the pixel are plus the surrounding light-gray area.



Types of Filters

- **Unweighted filtering** because each super samples to compute a pixel, irrespective of its position, it has equal influence in determining the pixel's color.
- **Weighted filter**, each super-sample is multiplied by its corresponding weight and the products are summed to produce a weighted average which is used as the pixel color. The weighting is given to each sample should depend in some way on its distance from the center of the pixel.

Filter Examples

- **Filtering** means that eliminating the high frequencies, combining the super-samples to compute a pixel color. Here are some known ones:
  - **Box** filter
  - **Gaussian** filter
  - **Motion Blur** filter
  - **Radial Blur** filter

  and more…