Antialiasing

Prof. George Wolberg
Dept. of Computer Science
City College of New York

Objectives

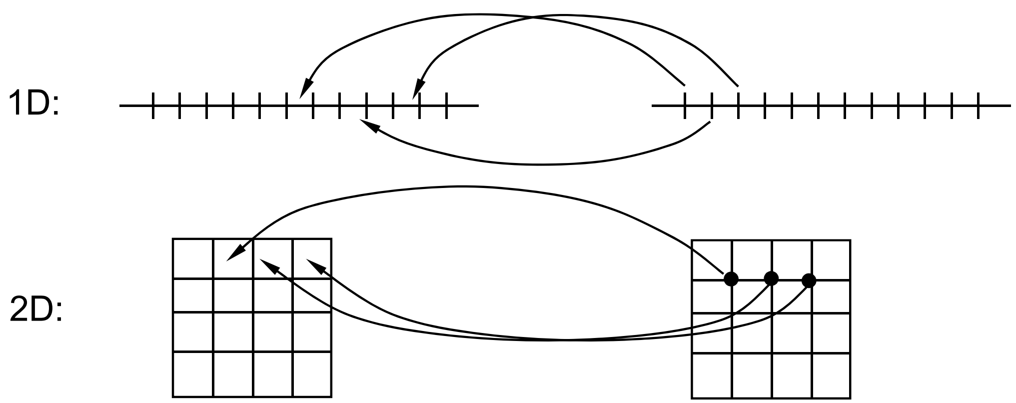
- In this lecture we review antialiasing:
  - Supersampling (uniform, adaptive, irregular)
  - Direct convolution
    - Feibush-Levoy-Cook
    - Gangnet-Perry-Coveignoux
    - Greene-Heckbert (Elliptical Weighted Average)
  - Prefiltering
    - Mip-maps / pyramids
    - Summed area tables

Antialiasing

- Aliasing is due to undersampling the input signal.
- It results in false artifacts, e.g., moire effects.
- Antialiasing combats aliasing. Solutions:
    - Raise sampling rate
    - Bandlimit input
    - In practice, do both in inverse mapping formulation
- Implementations:
    - Supersampling (uniform, adaptive, irregular)
    - Direct convolution
    - Prefiltering

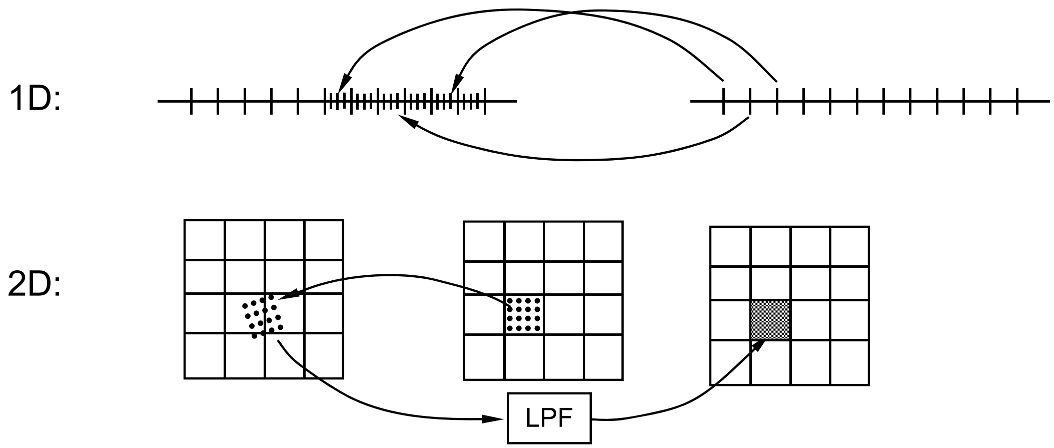
Point Sampling

- One input sample per output pixel
- Problem: information is lost between samples
- Solution: sample more densely and average results

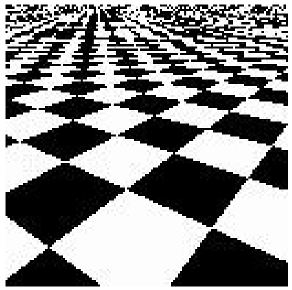


Supersampling (1)

- Multiple input samples per output pixel

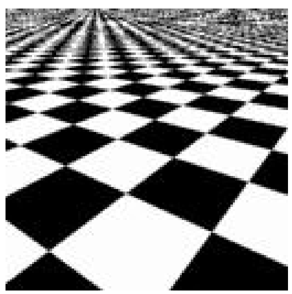


Supersampling (2)



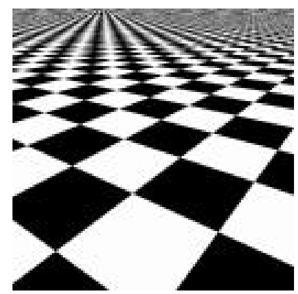
1 sample/pixel

4 samples/pixel



16 samples/pixel

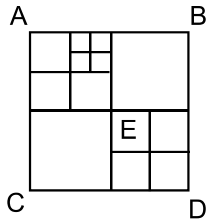
64 samples/pixel



256 samples/pixel

Adaptive Supersampling

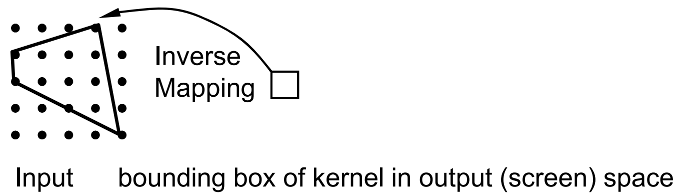
- Collect more samples in areas of high intensity variance or contrast.




```
If (f(A,B,C,D,E)>thr) then subdivide /* quad tree */
```

- To avoid artifacts due to regular sampling pattern (along rectilinear grid), use irregular sampling.
- Three common forms of stochastic sampling:
  - Poisson
  - Jittered
  - Point-diffusion sampling

Direct Convolution: Feibush-Levoy-Cook (1980)



1. All input pixels contained within the bounding rect. of this quadrilateral are mapped onto output.
2. The extra selected points are eliminated by clipping them against the bounding rect. of kernel.
3. Init output pixel with weighted average of the remaining samples. Note that filter weights are stored in a LUT (e.g., a Gaussian filter.)

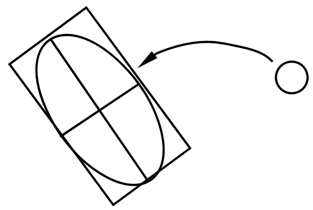
Advantages: 1) Easy to collect input pixels in rect. input region

2) Easy to clip output pixels in rect. output region.

3) Arbitrary filter weights can be stored in LUT.

Direct Convolution: Gangnet-Perry-Coveignoux (1982)

Improve results by supersampling.

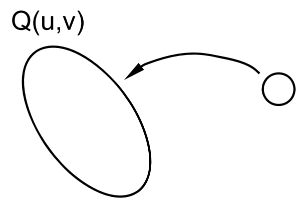


Major axis determines supersampling rate.

They used bilinear interpolation for image reconstruction and a truncated sinc (2 pixels on each side) as kernel.

Elliptical weighted average (EWA): Greene-Heckbert (1986)

EWA distorts the circular kernel into an ellipse in the input where the weighting can be computed directly.



No mapping back to output space.

$$Q(u,v) = Au^2 + Buv + Cv^2 \qquad u = v = 0 \text{ is the center of the ellipse}$$

$$A = V_x^2 + V_y^2$$

$$B = -2(U_x V_x + U_y V_y)$$

$$C = U_x^2 + U_y^2$$

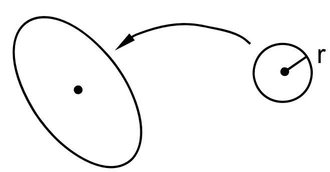
where

$$(U_x, V_x) = \left(\frac{\partial u}{\partial x}, \frac{\partial v}{\partial x} \right) \quad (U_y, V_y) = \left(\frac{\partial u}{\partial y}, \frac{\partial v}{\partial y} \right)$$

$$\text{Point-inclusion test}: Q(u,v) < F \;\; for\ F = (U_x V_y - U_y V_x)^2$$

EWA Advantages

- Only 1 function evaluated; not 4 needed for quadrilateral.
- If $Q<F$, then sample is weighted with appropriate LUT entry.
- In output space, LUT is indexed by r
- Instead of determining which concentric circle the point lies in the output, we determine which concentric ellipse the point lies in the input and use it to index the LUT.

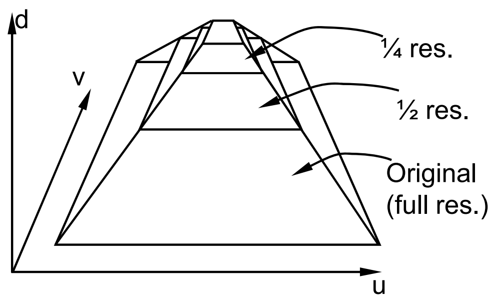


$$r = \sqrt{Q}$$

Comparisons

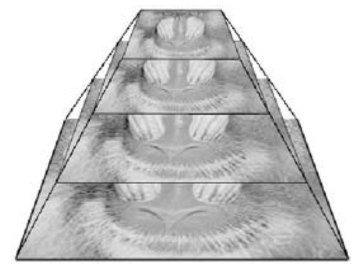
- All direct convolution methods are O(N) where N = # input pixels accessed. In Feibush and Gangnet, these samples must be mapped into output space, EWA avoids this costly step.
- Direct convolution imposes minimal constraints on filter area (quad, ellipse) and filter kernel (precomputed LUT). Additional speedups: prefiltering with pyramids and preintegrated tables -> approx convolution integral w/ a constant # of accesses (indep. of # input pixels in preimage.)
- Drawback of pyramids and preintegrated tables: filter area must be square or rectangular; kernel must be a box filter.

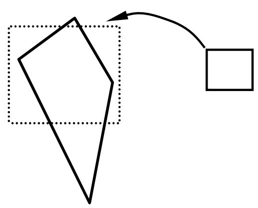
Pyramids



$$\text{memory cost} = 1 + \frac{1}{4} + \frac{1}{16} + \frac{1}{64} + \cdots = \frac{4}{3}$$

only 33% more memory





Size of square is used to determine pyramid level to sample.

Aliasing vs. Blurring tradeoff.

$$d^2 = \max\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2, \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2\right]$$

d is proportional to span of the preimage area.

Summed-Area Table

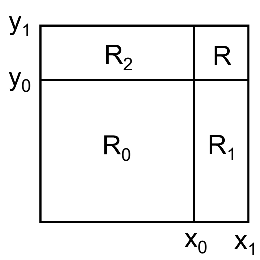


Table stores running sum

$$sum_R = T[x_1, y_1] - T[x_1, y_0] - T[x_0, y_1] + T[x_0, y_0]$$

Restricted to rectangular regions and box filtering

To compute current entry T[x1 , y1] let sum R be current pixel value v:

$$T[x_1, y_1] = v + T[x_1, y_0] + T[x_0, y_1] - T[x_0, y_0]$$

Example

90	10	20	30
50	60	70	80
25	75	200	180
100	50	70	80

Input

265	460	820	1190
175	360	700	1040
125	250	520	780
100	150	220	300

Summed-area table

$$sum_R = T[x_1, y_1] - T[x_1, y_0] - T[x_0, y_1] + T[x_0, y_0]$$

$$T[x_1, y_1] = v + T[x_1, y_0] + T[x_0, y_1] - T[x_0, y_0]$$