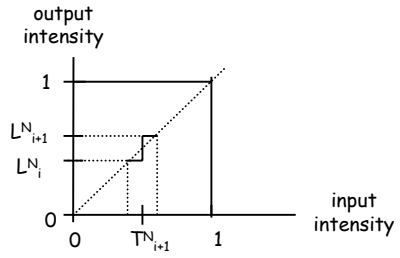
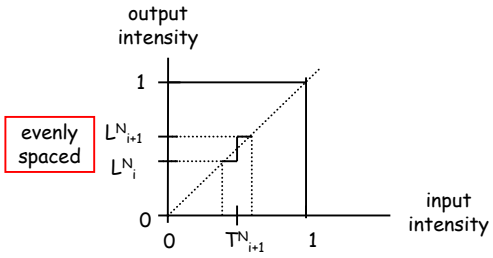
type of techniques

- simple pixel modification
- interpolation/extrapolation
- compositing
- convolution
- **dithering**
- warping
- morphing
- misc. effects

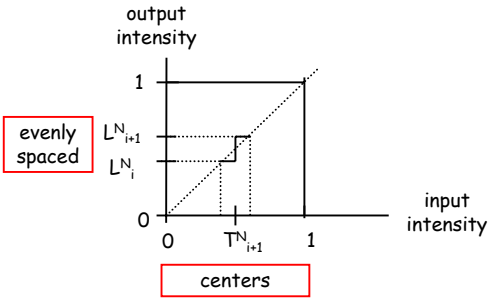
N level uniform quantization



N level uniform quantization



N level uniform quantization



quantization

8 bits per channel per pixel

1 bits per channel per pixel

random dither

add noise to camouflage quantization artifacts

8 bits per channel per pixel

1 bits per channel per pixel

1 bits per channel per pixel noisy

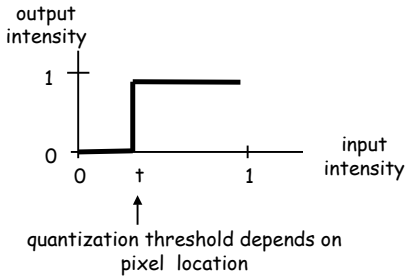
ordered dither

8 bits per channel per pixel

1 bits per channel per pixel

1 bits per channel per pixel dithered

ordered dither: 2 level output



ordered dither

- ordered dither takes an input parameter $m \geq 1$, which controls the number of output levels we want to simulate
- to simulate $M=m^2+1$ levels
  - assign pixel locations to m^2 classes
  - use T^M_i as threshold to quantize pixel in i^{th} class

ordered dither: m=2, M=5

assign pixel locations to $m^2=4$ classes

4	**2**	4	2	4	2	4
1	**3**	1	3	1	3	1
4	2	4	2	4	2	4
1	3	1	3	1	3	1
4	2	4	2	4	2	4

ordered dither: m=2, M=5

compute thresholds T^5_i, i=1, …,4:

$T^5_1=1/8$

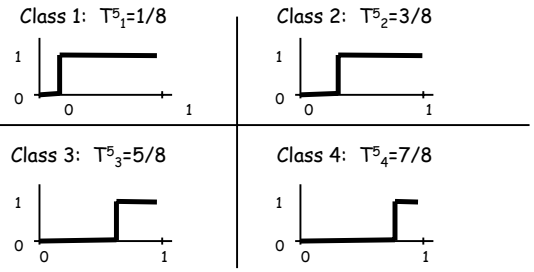
$T^5_2=3/8$

$T^5_3=5/8$

$T^5_4=7/8$

ordered dither: m=2, M=5

assign the thresholds to the pixel classes



ordered dither: example

Input image

$\frac{1}{2}$	$\frac{1}{2}$
$\frac{1}{2}$	$\frac{1}{2}$

→

Output image

0	1
1	0

ordered dither: another view

add "noise" to each pixel depending on its class
then quantize as usual

How much noise?

for m=2 use $\frac{1}{2} - T^m_i$

ordered dither

To simulate $M=m^2+1$ intensity levels:

- assign pixel locations to m^2 classes
- use T^M_i as threshold to quantize pixel in i^{th} class

How do you assign pixel locations to the classes? answer: carefully

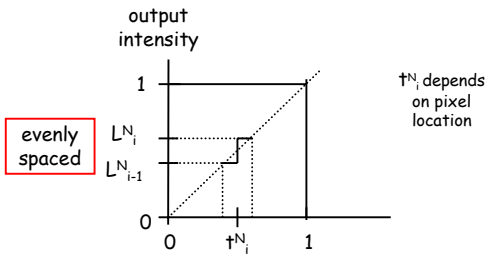
bayer's 2x2 ordered dither

4	2
1	3

bayer's 4x4 ordered dither matrix

16	8	14	6
4	12	2	10
13	5	15	7
1	9	3	11

Ordered dither: n level output



ordered dither: m=2, M=5, N levels

t^N_i for pixel class j is $L^N_{i-1} + T^5_j/2^{N-1}$

intuition: scale the 2 level case for each input intverval

error-diffusion dither

8 bits per channel per pixel

1 bits per channel per pixel

1 bits per channel per pixel dithered

comparison

original

ordered

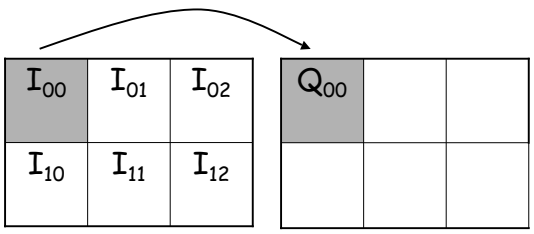
error-diffusion

error-diffusion dither

intuition

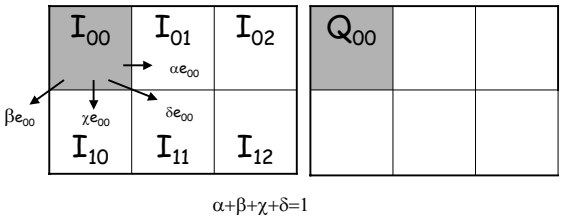
error diffusion dither

quantize I_{00} using uniform quantization



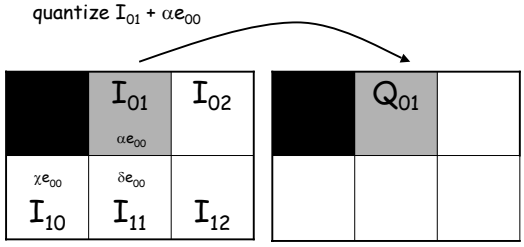
error diffusion dither

distribute error $e_{00}=I_{00} - Q_{00}$ to neighbors not yet quantized



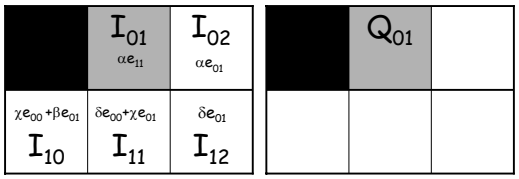
$\alpha+\beta+\chi+\delta=1$

error diffusion dither



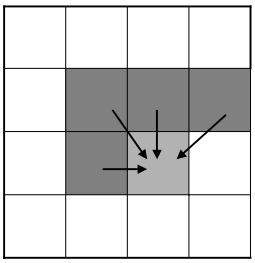
error diffusion dither

distribute error: $e_{01} = I_{01} + \alpha e_{00} - Q_{01}$



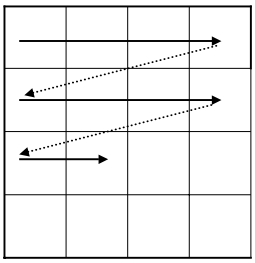
error diffusion dither

error contributions by upper & left neighbors



error diffusion dither

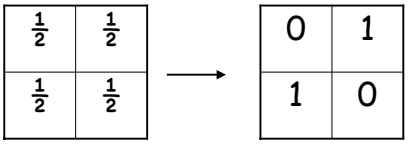
order of quantization is important



floyd-steinberg

$\alpha = 7/16$
$\beta = 3/16$
$\chi = 5/16$
$\delta = 1/16$

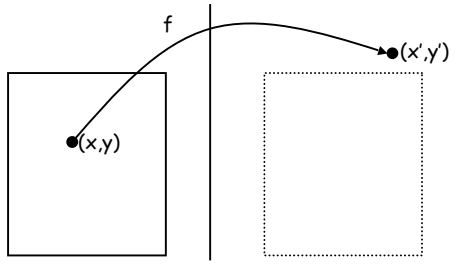
floyd-steinberg: example



types of techniques

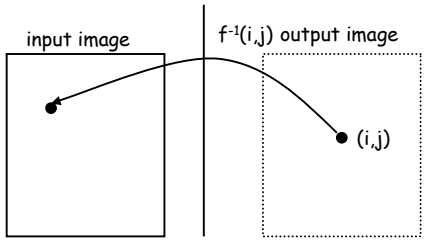
- simple pixel modification
- interpolation/extrapolation
- compositing
- convolution
- dithering
- **warping**
- morphing
- misc. effects

forward warp



f maps points in input image to the plane

forward warp



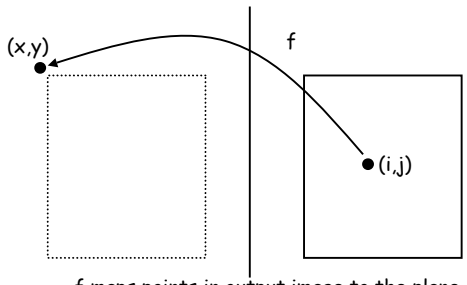
pixel at (i,j) in output image is assigned
the value at location $f^{-1}(i,j)$ in input image

forward warp: problems

if f is not bijective

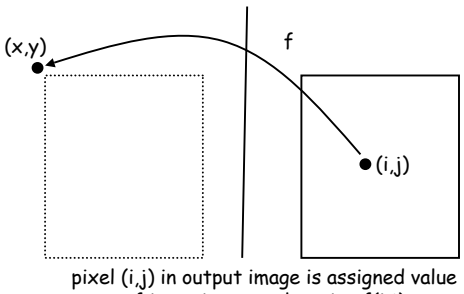
1. $f^{-1}(i,j)$ may not be defined
2. $f^{-1}(i,j)$ may not be unique

backward warp



f maps points in output image to the plane

backward warp

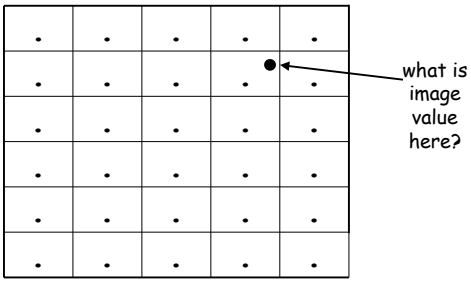


pixel (i,j) in output image is assigned value
of input image at location f(i,j)

backward warp: problems

1. f(i,j) may lie outside the input image area
   solution: give image an infinite, black (or other default) border
2. f(i,j) may not lie on a sample of the input image
   solution: resample input

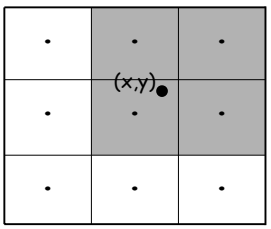
re-sample



re-sample

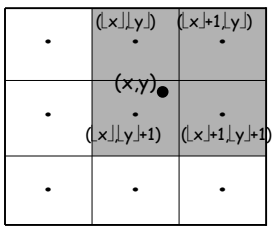
interpolate based on nearby samples
- nearest
- bilinear
- bicubic
- gaussian

which way is up?



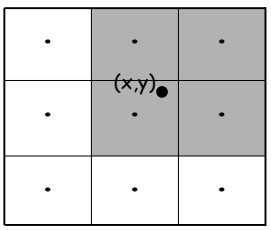
what are the coordinates of the pixels surrounding (x,y)?

which way is up?



what are the coordinates of the pixels surrounding (x,y)?

nearest

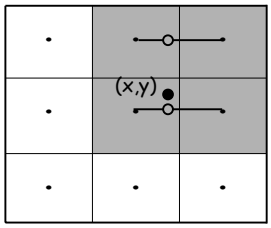


- compute distance between x,y and the locations of the neighboring samples
- set value at x,y to the value of the closest neighbor

re-sample

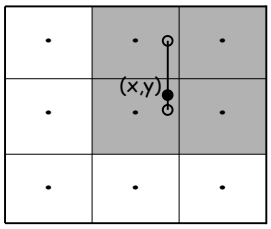
interpolate based on nearby samples
- nearest
- **bilinear**
- bicubic
- gaussian

bilinear interpolation



1. interpolate to find values at $(x\lfloor y\rfloor)$ and $(x\lfloor y\rfloor+1)$

bilinear interpolation

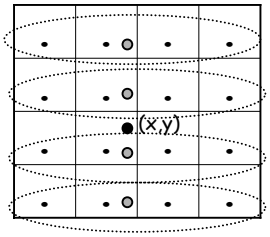


1. interpolate to find values at$(x\lfloor y\rfloor)$ and $(x\lfloor y\rfloor+1)$
2. interpolate to find value at x,y

re-sample

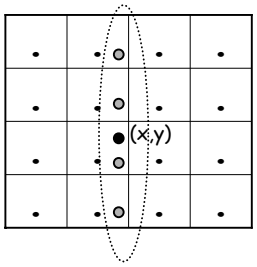
interpolate based on nearby samples
- nearest
- bilinear
- **bicubic**
- gaussian

bicubic



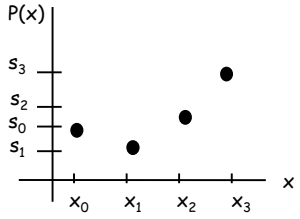
1. interpolate to find values at $(x\lfloor y\rfloor+i)$

bicubic



1. interpolate to find values at $(x\lfloor y\rfloor+i)$
2. interpolate to find value at (x,y)

bicubic: lagrangian



there is a unique cubic polynomial through any four distinct sample point

lagrange cubic polynomial

$P(x) = \Sigma_{i=0,1,2,3}\ s_i\ \Pi_{j=0,1,2,3,j\neq i}\ (x-x_j)/(x_i-x_j)$

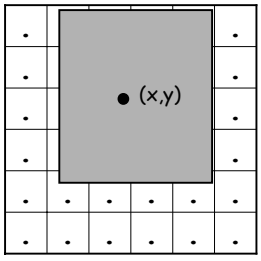
exercise: what is the value of $P(x_i)$ i=0,1,2,3

re-sample

interpolate based on nearby samples

- nearest
- bilinear
- bicubic
- **gaussian**

gaussian



interpolate nearby samples using normalized gaussian weights

unnormalized weight at (i,j) in window is

$\exp[-((x-i)^2+(y-j)^2)/\sigma^2]$

types of techniques

- simple pixel modification
- interpolation/extrapolation
- compositing
- convolution
- dithering
- warping
- **morphing**
- misc. effects