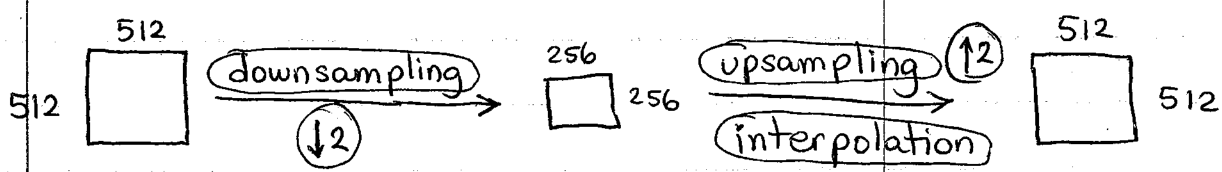
Interpolation in 2-D

continuous signal $f(x,y)$ $\xrightarrow{\text{sampling}}$ sampled signal $f_p(x,y)$ $\xrightarrow{\text{interpolation}}$ cont. signal $\hat{f}(x,y)$ $\xrightarrow{\text{sampling}}$ sampled signal $f_2(x,y)$

$f_p(x,y) \xrightarrow{\text{resampling}} f_2(x,y)$

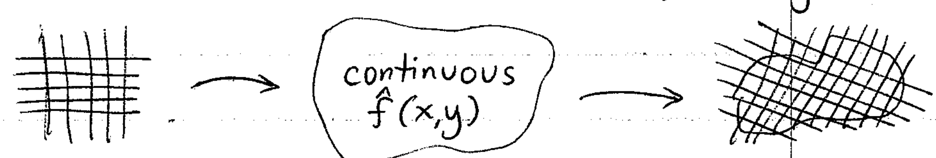
Terminology:

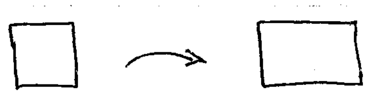


Why go back to continuous?

(1) Need analog video for CRT display

(2) Geometric transformations other than integer translations & 90° rotations, e.g. image resizing





- correction of aspect ratio
- 4/3 expansion in the x direction

(A) Geometric corrections of an individual image

(B) Registration of one image to another

In Matlab : imresize , imrotate

(3) Filling in missing samples

(A) JPEG compression $RGB \rightarrow YCrCb$, downsample Cr, Cb

(B) Bayer color filter array : color filter affixed to CCD sensor. MxN array. Need to interpolate to generate full

G R G R G
B G B G B

Ideal Interpolation

Recall: $\underbrace{f_p(x,y)}_{\text{sampled im}} = \underbrace{f_I(x,y)}_{\text{cont. ideal im}} \cdot \underbrace{S(x,y)}_{\text{sampling fct}}$

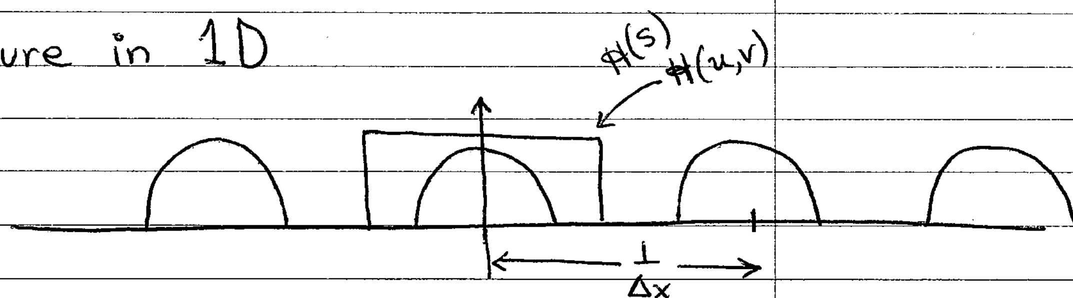
$$f_p(x,y) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f_I(j\Delta x, k\Delta y)\, \delta(x - j\Delta x, y - k\Delta y)$$

In Fourier domain:

$$F_p(u,v) = F_I(u,v) * F_s(u,v)$$

$$= \frac{1}{\Delta x \Delta y} \sum_j \sum_k F_I(u - j f_{xs}, v - k f_{ys})$$

Picture in 1D



Perfect Reconstruction:

$$F_R(u,v) = F_p(u,v) \cdot H(u,v)$$

F.T. of recon. image — F.T. of sampled im — rect fct $\begin{cases} K & |u| \leq \frac{f_{xs}}{2} \\ & |v| \leq \frac{f_{ys}}{2} \\ 0 & \text{otherwise} \end{cases}$

$$= \frac{1}{\Delta x \Delta y} H(u,v) \sum_j \sum_k F_I(u - j f_{xs}, v - k f_{ys})$$

In spatial domain:

$$f_R(x,y) = f_p(x,y) * h(x,y)$$

$$= \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f_I(j\Delta x, k\Delta y)\, \delta(x - j\Delta x, y - k\Delta y)\Big] * h(x,y)$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \sum_j \sum_k f_I(j\Delta x, k\Delta y)\, \delta(\alpha - j\Delta x, \beta - k\Delta y)\, h(x-\alpha, y-\beta)\, d\alpha\, d\beta$$

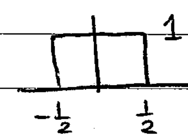
$$= \sum_j \sum_k f_I(j\Delta x, k\Delta y)\, h(x - j\Delta x, y - k\Delta y)$$

1D ideal interpolation: convolution with a sinc function

$$\operatorname{sinc} \frac{x}{(\Delta x)} \longleftrightarrow \Delta x\, \Pi(s\,\Delta x)$$

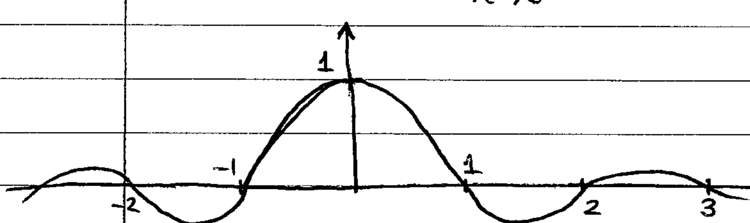
Note: $\operatorname{sinc} \frac{x}{\Delta x} = 0$ for $x = n\Delta x$

Taking $\Delta x = 1$: $\operatorname{sinc} x \longleftrightarrow \Pi(s)$



$$\operatorname{sinc} x = \frac{\sin \pi x}{\pi x}$$

Exact Reconstruction
Spatially Unlimited



$\operatorname{sinc} 0 = 1$

$\operatorname{sinc} n = 0 \quad \forall n \in \mathbb{Z}, \; n \neq 0$

sinc goes on forever → want to find convolution kernels which don't.

Convolution kernels which satisfy $h(0) = 1$ and $h(n) = 0$ for $n \neq 0$ are called interpolators.

Otherwise: approximator

$$f_R(x,y) = \sum_j \sum_k f_I(j,k)\, h(x-j,\, y-k)$$

f_I was sampled at integer positions because $\Delta x = 1$ $\Delta y = 1$

$$f_R(n,m) = \sum_j \sum_k f_I(j,k)\, h(n-j,\, m-k)$$

but $h(0,0) = 1$ $h(n,m) = 0$ if $(n,m) \neq (0,0)$

so:

$$f_R(n,m) = f_I(n,m)$$

Guarantees that the image is not modified if it is resampled onto the same grid

$$f_I(x,y) \xrightarrow[\times S(x,y)]{\text{sampling}} f_P(x,y) \xrightarrow[* h(x,y)]{\text{interpolation}} f_R(x,y) \xrightarrow[\times S(x,y)]{\text{re-sampl.}} f_P(x,y)$$

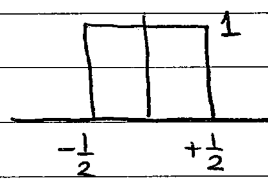
$$f_I(n,m) = f_P(n,m) = f_R(n,m)$$

$\boxed{\text{Zero-order Hold}}$

The order of an interpolation scheme, by one definition, is one less than the # of pts involved in defining an output value

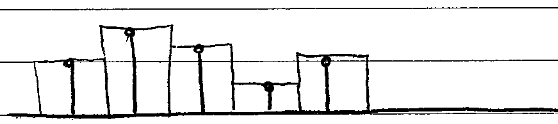
Convolve w/ square pulse

In 1D

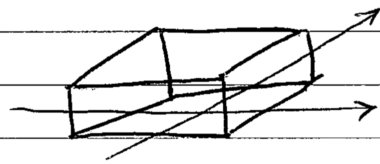


$h_1(x)$

meets condition $h_1(0) = 1$ $h_1(n) = 0$ to be interpolator



In 2D, use product of 2 1D rect functions



← Square peg filter

$$\underset{\text{small}}{\begin{matrix} 10 & 12 \\ 20 & 24 \end{matrix}} \xrightarrow[\text{with zeros}]{\text{upsample}} \overbrace{\begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 12 \\ 0 & 0 & 0 & 0 \\ 0 & 20 & 0 & 24 \end{matrix}}^{a} \xrightarrow[\text{with } \underset{b}{\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}}]{\text{convolve}} \begin{matrix} 10 & 10 & 12 & 12 \\ 10 & 10 & 12 & 12 \\ 20 & 20 & 24 & 24 \\ 20 & 20 & 24 & 24 \end{matrix}$$

c = conv2(a,b,'same');
c = imresize(small, 2, 'nearest');

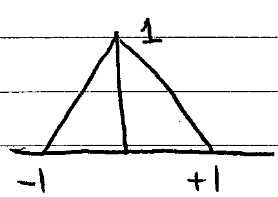
Also called pixel replication

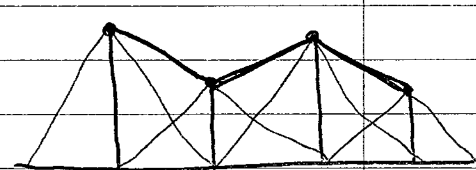
First-Order Hold

In 1D : convolve with a triangle function

$h_2(x) = h_1(x) * h_1(x)$ 2 rect functions convolved

$$= \begin{cases} 1-|x| & 0 \le |x| < 1 \\ 0 & \text{elsewhere} \end{cases}$$





Equivalent to linearly connecting adjacent sample peaks. Called linear interpolation

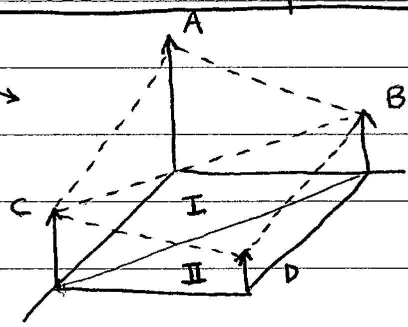
Meets condition $h(0) = 1$ $h(n) = 0$ $n \neq 0$ to be interpolator

First Order Interpolation in 2d

The simple extension of linear interpolation to 2d does not hold because, in general, it is not possible to fit a plane through 4 adjacent samples.

Piecewise linear interpolation

Planar fit → in a piecewise fashion

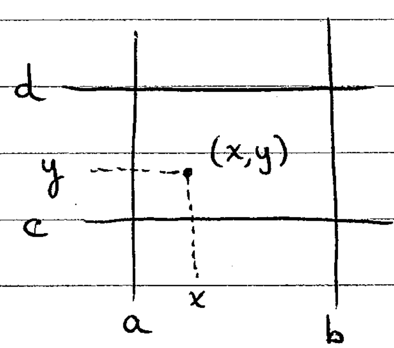


In Region I, points are linearly interp. in the plane defined by pixels A, B, C

Region II : B, C, D

Bilinear interpolation :

- → computationally simpler
- → 1d interp along rows, then 1d interp along columns
- → Resultant interpolated surface generally non-planar



4 discrete points are known

$f(a,c)$ $f(b,c)$

$f(a,d)$ $f(b,d)$

$$\Delta_x = \frac{x-a}{b-a}$$

$0 \leq \Delta_x \leq 1$

fraction of the distance across

$$\Delta_y = \frac{y-c}{d-c}$$

$$\hat{f}(x,y) = (1-\Delta_x)(1-\Delta_y)\, f(a,c) + (1-\Delta_x)\, \Delta_y\, f(a,d)$$

$$+ \Delta_x (1-\Delta_y)\, f(b,c) + \Delta_x \Delta_y\, f(b,d)$$

Equivalent to interpolation with a pyramid function

$$\text{pyr} = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

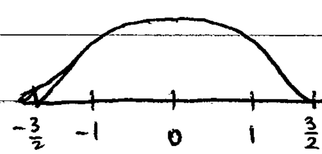
Called bilinear interpolation because it is linear as a fct of x when y is held fixed, and also linear as a function of y when x is held fixed.

$$\begin{matrix} 10 & 12 \\ 20 & 24 \end{matrix} \rightarrow \begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 12 \\ 0 & 0 & 0 & 0 \\ 0 & 20 & 0 & 24 \end{matrix} \xrightarrow[\text{w/ pyr}]{\text{convolve}} \begin{matrix} 10 & 11 & 12 \\ 15 & 16.5 & 18 \\ 20 & 22 & 24 \end{matrix}$$

upsample with zeros

$h_3(x) = h_1(x) * h_1(x) * h_1(x)$

$\boxed{\text{Quadratics}}$ Convolve 3 rect functions:



$$h_3(x)\ \cancel{R_2(x)} = \begin{cases} \frac{1}{2}\left(x+\frac{3}{2}\right)^2 & -\frac{3}{2} \leq x \leq -\frac{1}{2} \\ \frac{3}{4} - x^2 & -\frac{1}{2} < x \leq \frac{1}{2} \\ \frac{1}{2}\left(x-\frac{3}{2}\right)^2 & \frac{1}{2} < x \leq \frac{3}{2} \end{cases}$$

Does not satisfy $R_2(0)=1$ nor $R_2(n)=0 \quad n \neq 0$

Quadratic Approximator

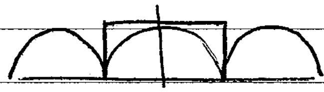
Quadratic Interpolator : $\tilde{R}_2(x) = \begin{cases} 1 - x^2 & -\frac{1}{2} < x \leq \frac{1}{2} \\ \frac{3}{2} + |x|^2 - \frac{5}{2}|x| & \frac{1}{2} \leq |x| < \frac{3}{2} \end{cases}$

In general, moving towards lower % interp. error, higher % resolution error (more blurriness)

$$h_N(x) = \underbrace{h_1(x) * h_1(x) * \cdots h_1(x)}_{N-1 \text{ times}}$$

rect fct

How to evaluate?



Non-ideal reconstruction

① Attenuates signal within Nyquist limits

② Lets in some of the freqs beyond Nyquist limits

Define:

$$E_R = \int_{-\frac{\omega_{xs}}{2}}^{+\frac{\omega_{xs}}{2}} \int_{-\frac{\omega_{xs}}{2}}^{+\frac{\omega_{xs}}{2}} \mathcal{W}_{F_I}(\omega_x,\omega_y)\,|\mathcal{H}(\omega_x,\omega_y)|^2\, d\omega_x\, d\omega_y$$

power spectral density of the ideal image

$$E_{RM} = \int_{-\frac{\omega_{xs}}{2}}^{+\frac{\omega_{xs}}{2}} \int_{-\frac{\omega_{ys}}{2}}^{+\frac{\omega_{ys}}{2}} \mathcal{W}_{F_I}(\omega_x,\omega_y)\, d\omega_x\, d\omega_y$$

E_{RM} is the ~~actual~~ ideal interpolated image energy within the Nyquist sampling band limits

E_R is the actual interp. image energy, attenuated by $|\mathcal{H}|^2$ within those limits

$E_{RM} - E_R$ is the energy loss

$$\boxed{\%\ \text{Resolution Error} = \frac{E_{RM} - E_R}{E_{RM}}}$$

Define:

$$E_T = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathcal{W}_{F_I}(\omega_x,\omega_y)\,|\mathcal{H}(\omega_x,\omega_y)|^2\, d\omega_x\, d\omega_y$$

is the total energy of the interpolated image.

$E_T - E_R$ is that portion of the interp. image energy lying outside the Nyquist band limits

$$\boxed{\%\ \text{Interpolation Error} = \frac{E_T - E_R}{E_T}}$$

(attributable to high spatial freq artifacts)

zero-order hold has very high interp. error