Integral and Discrete Transforms in Image Processing

Sampling & Resampling

David Svoboda

email: svoboda@fi.muni.cz
Centre for Biomedical Image Analysis
Faculty of Informatics, Masaryk University, Brno, CZ

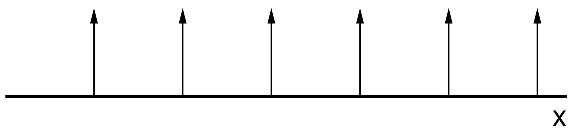
October 14, 2022

Outline

Comb function

In 1D space

$$\text{III}(x) = \sum_{n=-\infty}^{\infty} \delta(x - n)$$



Notice: "III" is pronouced as *shah* (Cyrilic character).

Comb function

Some properties

- $\text{III}(-x) = \text{III}(x)$
- $\text{III}(x + n) = \text{III}(x)$
- $\text{III}(x - \frac{1}{2}) = \text{III}(x + \frac{1}{2})$
- $\text{III}(x) = 0 \quad x \neq n$
- $\text{III}(ax) = \frac{1}{|a|} \sum \delta(x - \frac{n}{a})$
- $\text{III}(\frac{x}{\tau}) \supset \tau \text{III}(\tau\omega)$

Sampling

Sampling in 1D:

- a process of converting a continuous signal into a discrete sequence.
- a multiplication with comb function:

$$\text{III}(x)f(x) = \sum_{n=-\infty}^{\infty} f(n)\delta(x-n)$$

Question: What happens in frequency (Fourier) domain?

Sampling

Image domain *versus* Fourier domain

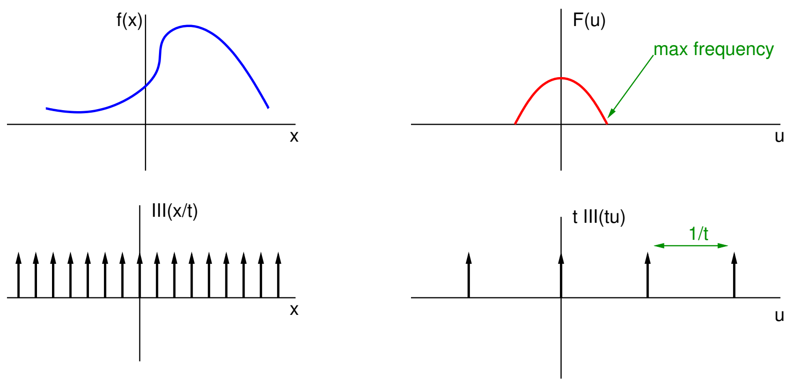
Image/Time domain:

- multiplication of the function f and III
- sampling

Fourier domain:

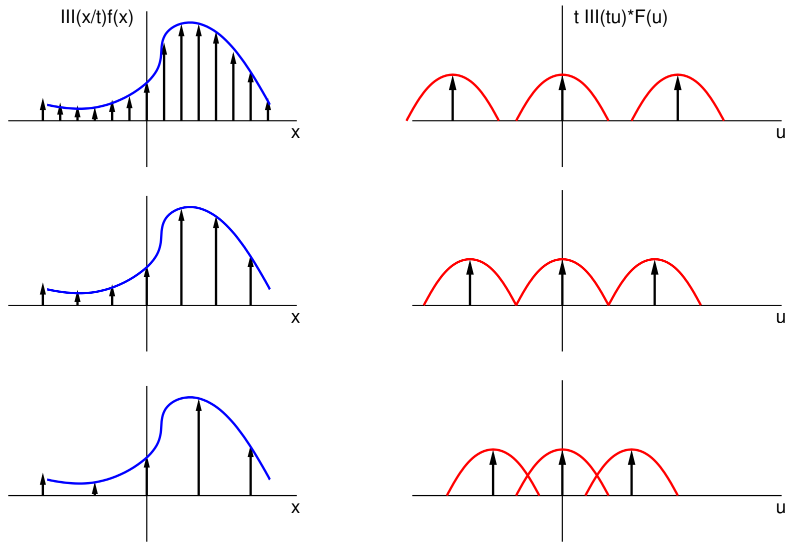
- convolution of the function $FT(f)$ and $FT(\text{III})$
- convolution with Dirac impulses $\rightarrow$ replication of $FT(f)$
- scaling property is also valid for III function

Sampling



Notice: The comb function density must be high enough to guarantee proper sampling

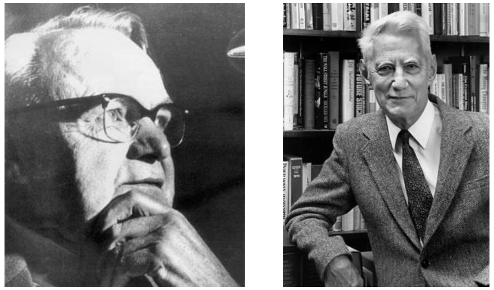
Sampling



Sampling

Nyquist-Shannon theorem

Exact reconstruction of a continuous signal from its samples is possible if the signal is bandlimited and the sampling frequency is greater than twice the signal maximal frequency



Harry Nyquist (1889 – 1976) & Claude Elwood Shannon (1916 – 2001)

Question: How to use N-S theorem, if the original signal is unlimited?

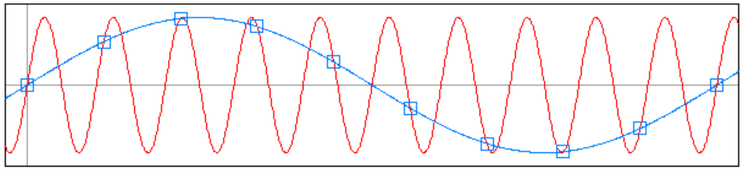
Sampling

Common problems – aliasing

The cause of aliasing:

when Nyquist-Shannon condition is broken, i.e.

- sampling frequency is not high enough or (time alias – wagon wheel effect)
- the signal in not bandlimited (PC games – far horizon)



Sampling

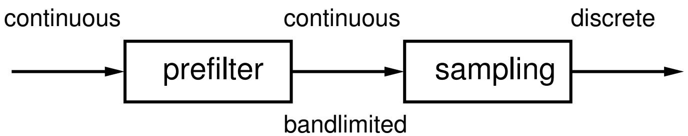
Common problems – aliasing

How to eliminate aliasing?

- sampling at higher frequency
  - does it help if the signal is not band limited?
  - expensive for memory and time

OR

- prefiltering
  - before sampling the input signal is "prefiltered" by lowpass filter



Sampling

Common problems – aliasing

Some lowpass filters

- Gaussian filter

$$f_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$

- Sinc filter

$$f(x) = \frac{sin(x)}{x}$$

- B-spline filter

$$b_1(x) = \begin{cases} 1 & |x| \leq 1/2 \\ 0 & |x| > 1/2 \end{cases}$$

$$b_n(x) = b_1(x) * b_1(x) * \cdots * b_1(x) \quad \text{n-times}$$

Reconstruction/Interpolation

Inverse process to sampling

The purpose: reconstruction of the original continuous signal from the sampled sequence.

Reconstruction $\equiv$ convolution with a *low-pass* filter.

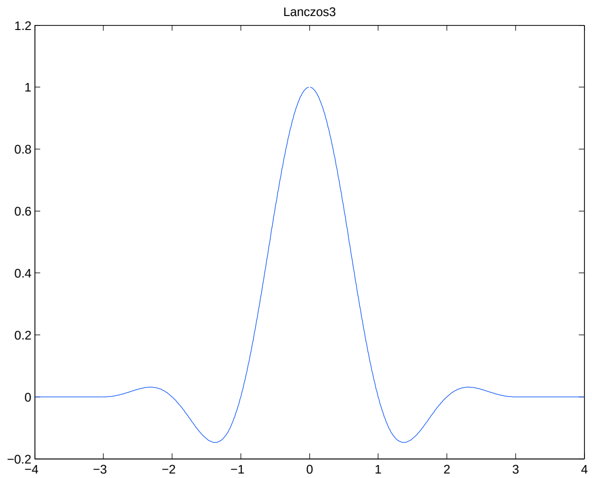
Common reconstruction filters:

- *box* (nearest neighbour)
- *tent* (linear interpolation)
- cubic B-spline (cubic polynomial interpolation)
- Gaussian
- *sinc* function
- Lanczos (windowed sinc function)

Notice: The unit area under the curve.

Reconstruction/Interpolation

Lanczos filter

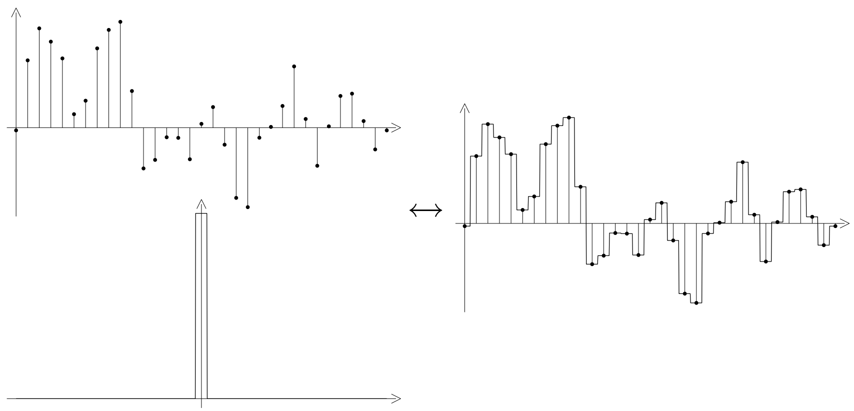


$$lanczos3(x) = \begin{cases} \frac{\sin \pi x}{\pi x} \frac{\sin \pi \frac{x}{3}}{\pi \frac{x}{3}} & |x| < 3 \\ 0 & |x| \geq 3 \end{cases}$$

Reconstruction/Interpolation

Example

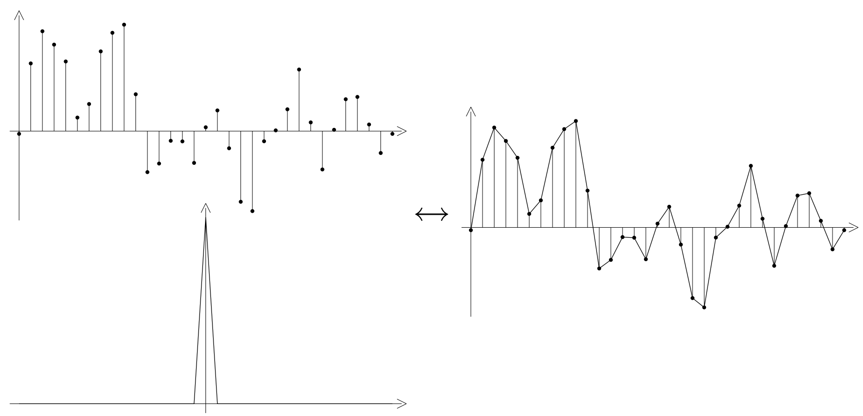
Box filter



Reconstruction/Interpolation

Example

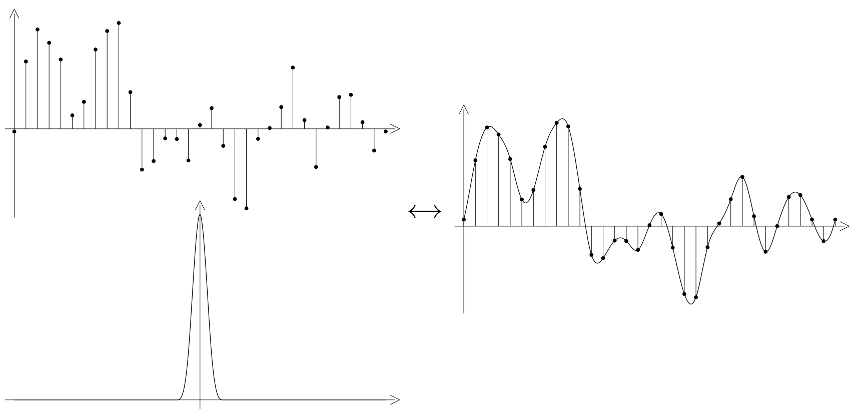
Tent filter



Reconstruction/Interpolation

Example

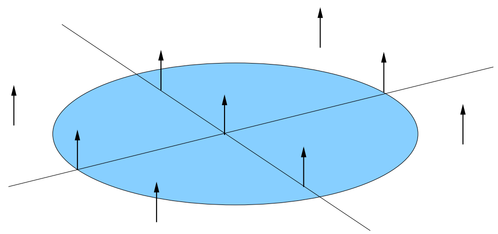
Cubic B-spline filter



Comb function

In 2D space

$$^2\text{III}(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(x-m, y-n)$$



Separability of delta function implies:

$$^2\text{III}(x,y) = \text{III}(x)\text{III}(y)$$

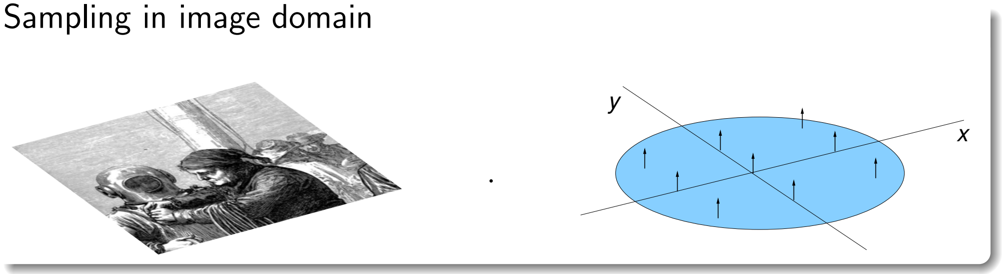
Sampling

Sampling in 2D:

- a process of converting a continuous 2D function into a discrete grid
- a multiplictation with comb function:

$$^2\text{III}(x,y)f(x,y) = \sum_m \sum_n f(m,n)\delta(x-m, y-n)$$

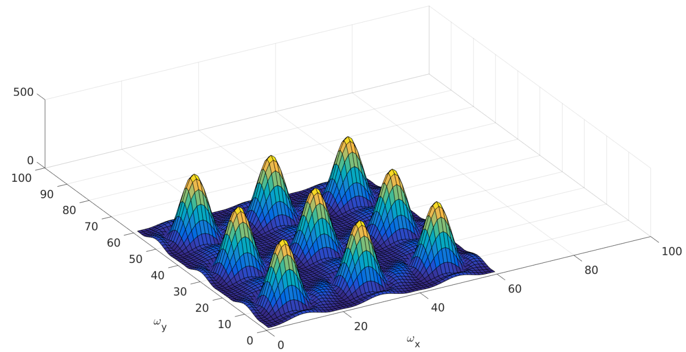
Sampling in image domain



Sampling

Common problems – aliasing

Insufficient sampling

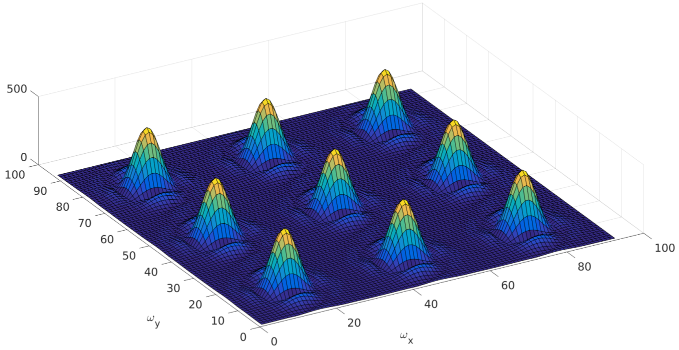


reconstructed image

Sampling

Common problems – aliasing

Sufficient sampling

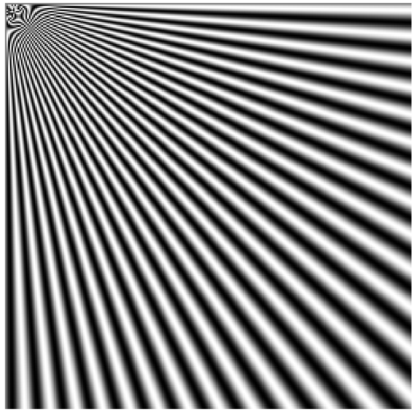


reconstructed image

Sampling

Common problems – aliasing

An example



Sampling

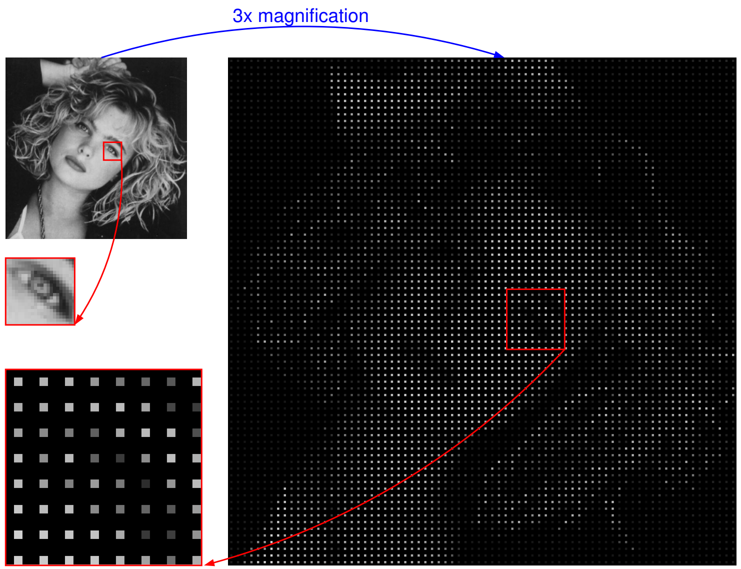
Common problems – aliasing

An example

Reconstruction/Interpolation

Task to solve

Problem of missing pixels when zooming into the digital image

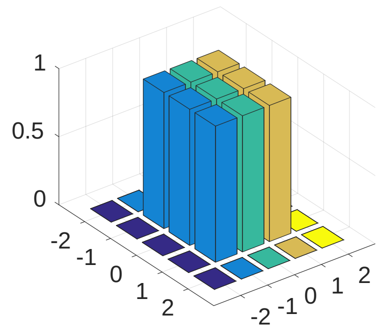


Reconstruction/Interpolation

Solution

Nearest neighbour interpolation

$$Kernel_{3\times3} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



Note: $Kernel_{3\times3} = [1\ 1\ 1]^T\ [1\ 1\ 1]$

Reconstruction/Interpolation

Solution

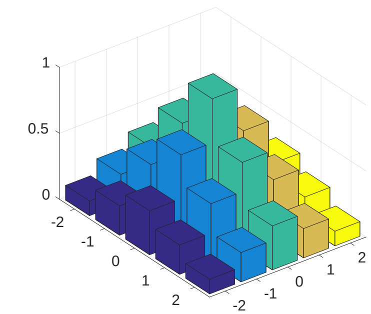
Completion of missing pixels (nearest neighbour)

Reconstruction/Interpolation

Solution

Bilinear interpolation

$$Kernel_{5\times5} = \frac{1}{9}\begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 4 & 6 & 4 & 2 \\ 3 & 6 & 9 & 6 & 3 \\ 2 & 4 & 6 & 4 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix}$$



Note: $Kernel_{5\times5} = \left(\frac{1}{3}[1\ 2\ 3\ 2\ 1]^T\right)\left(\frac{1}{3}[1\ 2\ 3\ 2\ 1]\right)$

Reconstruction/Interpolation

Solution

Completion of missing pixels (bilinear)

Resampling in 1D

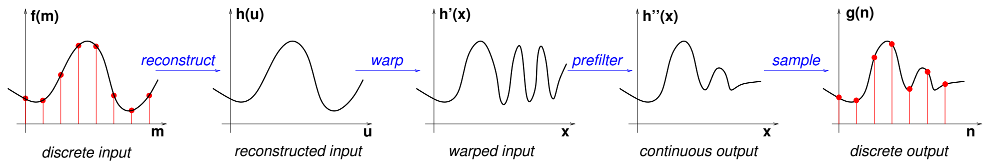
Let us design a 1D resampling filter

- The filter should be easy to implement and fast for computation.
- The filter should solve the alias problem.

Resampling in 1D

Design of the resampling filter

1. reconstruct the continuous signal from the discrete one
2. warp the domain of the continuous signal
3. prefilter the warped, continuous signal
4. sample this signal to produce the discrete output signal



Resampling in 1D

Important (implementation) notes

- During the resampling process we actually never construct a continuous signal $h(u)$, $h'(x)$ or $h''(x)$.
- We pick up the individual positions in the resampled image $g(n)$ and look for their corresponding positions and their neighbourhood in the original image $f(m)$.
- As the computation is inverted, we never use γ function. We use only γ^{-1}.

Resampling in 1D

Derivation of an ideal resampling filter

Computation of one sample point

$$\begin{aligned} g(n) &= h''(n) = \int h'(t)p(n-t)dt \\ &= \int h(\gamma^{-1}(t))p(n-t)dt \\ &= \int p(n-t)\sum_k f(k)r(\gamma^{-1}(t)-k)dt = \sum_k f(k)\rho(n,k) \end{aligned}$$

where

$$\rho(n,k) = \int p(n-t)r(\gamma^{-1}(t)-k)dt$$

- $\rho(n,k)$ is called a resampling filter.
- If γ is affine, we can derive: $\rho(n,k) = p(\gamma^{-1}(n)-k) * r(\gamma^{-1}(n)-k)$.

Resampling in 1D

Practical problems

- If the mapping γ is not affine, the filter $\rho(n, k)$ is space variant.

Solution (*postfiltering/supersampling*)

1. Reconstruct the continuous signal from the discrete input signal.
2. Warp the domain of the input signal.
3. Sample the warped signal at *very high resolution* to avoid alias.
4. Postfilter the signal to produce a lower resolution output signal.

Notice: The convolution is employed in the very end of this algorithm, i.e. it is discrete and space invariant.

Resampling in 2D

Task

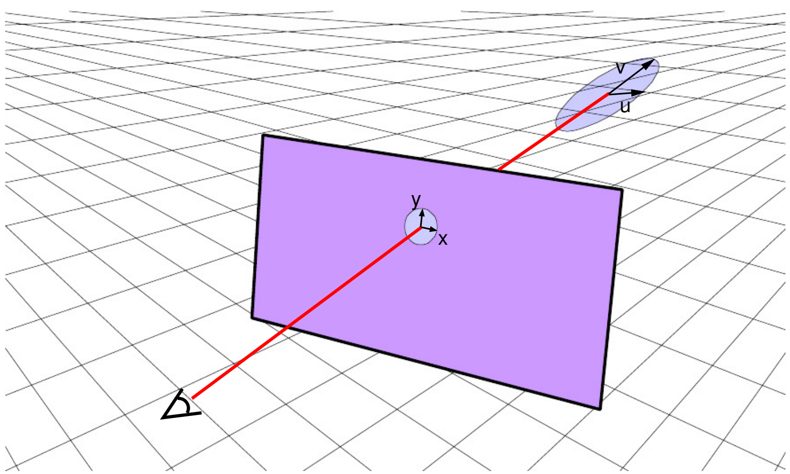
Design a 2D resampling filter

- Obey the rules that are valid for 1D resampling filter.
- The filter maps texels from texture space to screen space.
- The filter might be anisotropic.
- The filter should work fast.

Resampling in 2D

γ-mapping

γ^{-1} converts coordinates *from* screen space $\mathbf{x} = (x, y)$ *to* texture space $\mathbf{u} = (u, v)$



Resampling in 2D

γ-mapping

γ is projective

- circular neighbourhood of one pixel is transformed into an elliptical neighbourhood.
- can be locally approximated by linear mapping – Jacobian matrix J:

$$J^{-1} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix} = \begin{bmatrix} U_x & U_y \\ V_x & V_y \end{bmatrix}$$

$$\gamma : \mathbf{x} = J\mathbf{u}$$
$$\gamma^{-1} : \mathbf{u} = J^{-1}\mathbf{x}$$

Resampling in 2D

γ-mapping

Gaussian warped by γ gives Gaussian:

$$
\begin{aligned}
g_I(\mathbf{u}) &= g_I(J^{-1}\mathbf{x}) \\
&= \frac{1}{2\pi} e^{-\frac{1}{2}(J^{-1}\mathbf{x})^T J^{-1}\mathbf{x}} = \\
&= \frac{1}{2\pi} e^{\frac{1}{2}\mathbf{x}^T (J^{-1T} J^{-1})\mathbf{x}} = \qquad /V^{-1} = J^{-1T} J^{-1}/ \\
&= |V|^{1/2} \cdot \frac{1}{2\pi |V|^{1/2}} e^{\frac{1}{2}\mathbf{x}^T V^{-1}\mathbf{x}} = \qquad /|V|^{1/2} = |J|/ \\
&= |J| \cdot g_V(\mathbf{x})
\end{aligned}
$$

where

- $J^{-1T} J^{-1}$ ... variance matrix (positive definite)
- standard multivariate Gaussian:

$$g_\Sigma(\mathbf{u}) = \frac{1}{2\pi|\Sigma|^{1/2}} e^{-\frac{1}{2}\mathbf{u}^T \Sigma^{-1}\mathbf{u}}$$

Resampling in 2D

γ-mapping

Formation of Gaussian regions

Variance matrix V defines a conic:

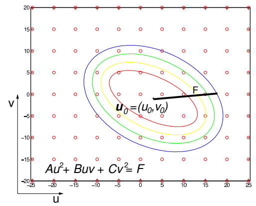
$$V^{-1} = \begin{pmatrix} A & B/2 \\ B/2 & C \end{pmatrix}^{-1} = J^{-1T} J^{-1}$$

from which the shape of zero-centered ellipse can be derived:

$$Q(u,v) = Au^2 + Buv + Cv^2 < F$$

Here

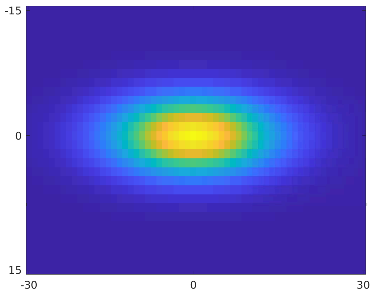
$$
\begin{aligned}
A &= V_x^2 + V_y^2 + 1 \\
B &= -2\,(U_x V_x + U_y V_y) \\
C &= U_x^2 + U_y^2 + 1 \\
F &= AC - \tfrac{B^2}{4}
\end{aligned}
$$



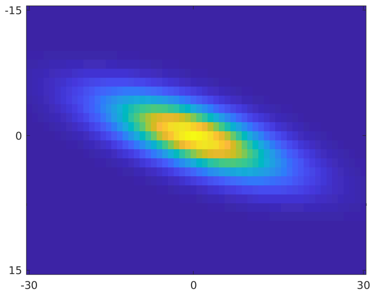
Resampling in 2D

γ-mapping

Formation of Gaussian regions



$$\Sigma = \begin{bmatrix} 10^2 & 0 \\ 0 & 3^2 \end{bmatrix}$$



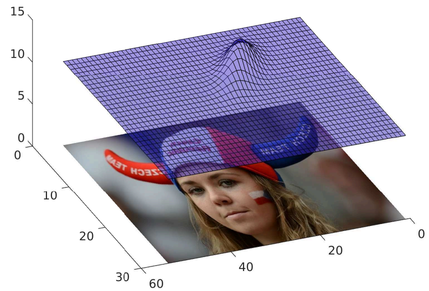
$$\Sigma = \begin{bmatrix} 10^2 & 20 \\ 20 & 3^2 \end{bmatrix}$$

Resampling in 2D

γ-mapping

BoundingBox for elliptical regions

$$
\begin{aligned}
u &= \pm 2\sqrt{\frac{CF}{4AC - B^2}} \\
v &= \pm 2\sqrt{\frac{AF}{4AC - B^2}}
\end{aligned}
$$



Weight for a particular texture pixel

$$
\begin{aligned}
\mathrm{weight}(u,v) &= /\rho((x,y),(u,v))/ \\
&= e^{-\alpha Q(u,v)}
\end{aligned}
$$

Notice: α ... user defined constant (e.g. $\alpha = 2$)

Resampling in 2D

γ-mapping

Image Pyramids (MIP map)

- Size of ellipse determines level of detail in MIP map pyramid that should be fetched from the memory.
- MIP map pyramid is precomputed to avoid (pointless) repetitive downsampling.

Resampling in 2D

Elliptical Weighted Average (EWA)

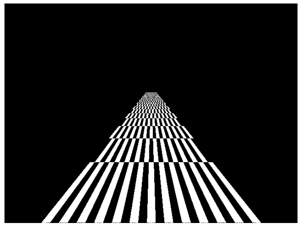
Implementation Notes

For each image pixel (x,y) from screen space:

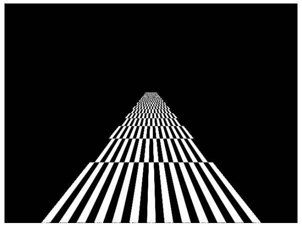
1. Find corresponding point (u,v) in texture space.
2. Define the local affine transform γ.
3. Compute Jacobian J of this mapping.
4. Delineate the ellipse in texture space (find bounding box)
5. Using the ellipse size choose the two nearest MIP map levels
6. Perform the following steps in each MIP map level and combine the results
   1. Build the Gaussian over the texture.
   2. Evaluate direct convolution of texture image with Gaussian.
   3. Get one value as a result.
7. Store the result (one value) in the screen pixel (x,y).

Resampling in 2D

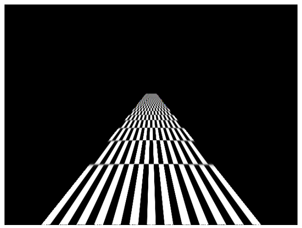
EWA – An Example



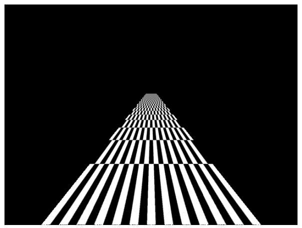
Nearest neighbour



Linear interpolation



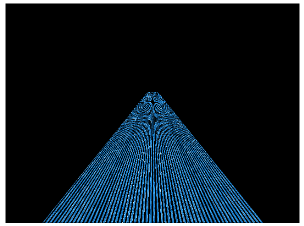
Linear interpolation + mipMAP



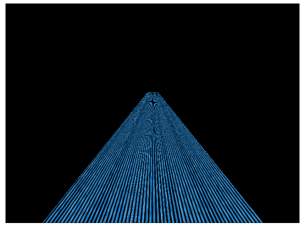
EWA + mipMAP

Resampling in 2D

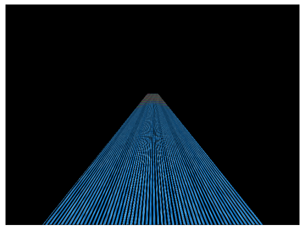
EWA – An Example



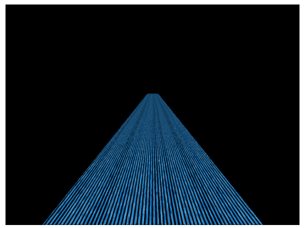
Nearest neighbour



Linear interpolation



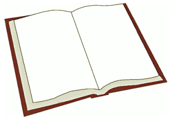
Linear interpolation + mipMAP



EWA + mipMAP

Bibliography

1. Paul Heckbert, Fundamentals of Texture Mapping and Image Warping, Master's thesis, UCB/CSD 89/516, CS Division, U.C. Berkeley, June 1989, 86 pp
2. McCormack, J., Perry, R.N., Farkas, K.I.; Jouppi, N.P. "Feline: Fast Elliptical Lines for Anisotropic Texture Mapping", ACM SIGGRAPH, pp 243-250, August 1999
3. Pharr, M., Humphreys, G. Physically Based Rendering: From Theory to Implementation. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004, https://www.pbr-book.org/



You should know the answers . . .

- Show that the 2D DFT is separable transform.
- Derive the complexity of 2D discrete FFT.
- Explain the reciprocity of wide and narrow shapes in time and frequency domain, respectively.
- Derive (dot not formulate) the Nyquist-Shannon theorem for 2D image data.
- Show an example of the aliasing effect.
- What is a prefilter?
- What is the difference between a screen space and texture space?
- Give an example of γ warping function both for 1D and 2D case.
- What is the difference between projective and affine mappings?
- Describe individual steps of EWA filter.