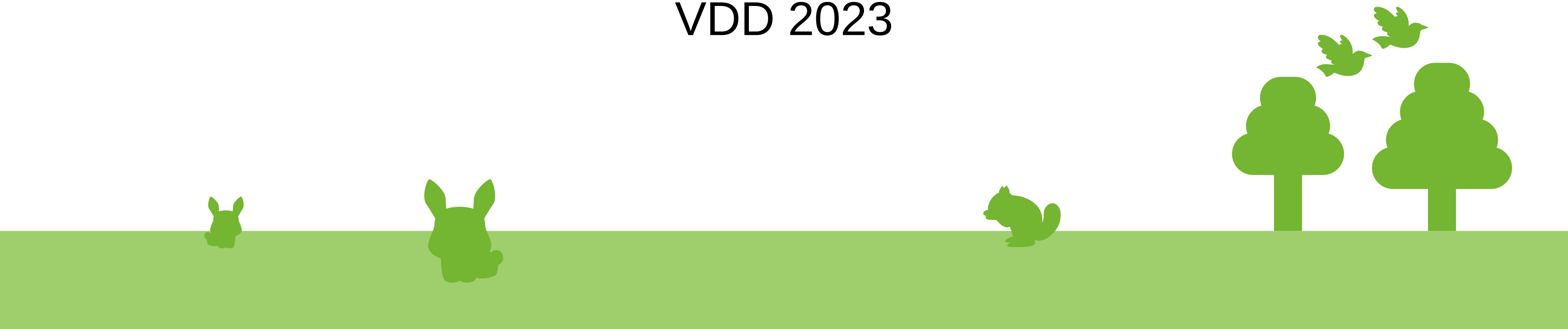
Faster (and better) GPU (down)scaling

in **libplacebo**

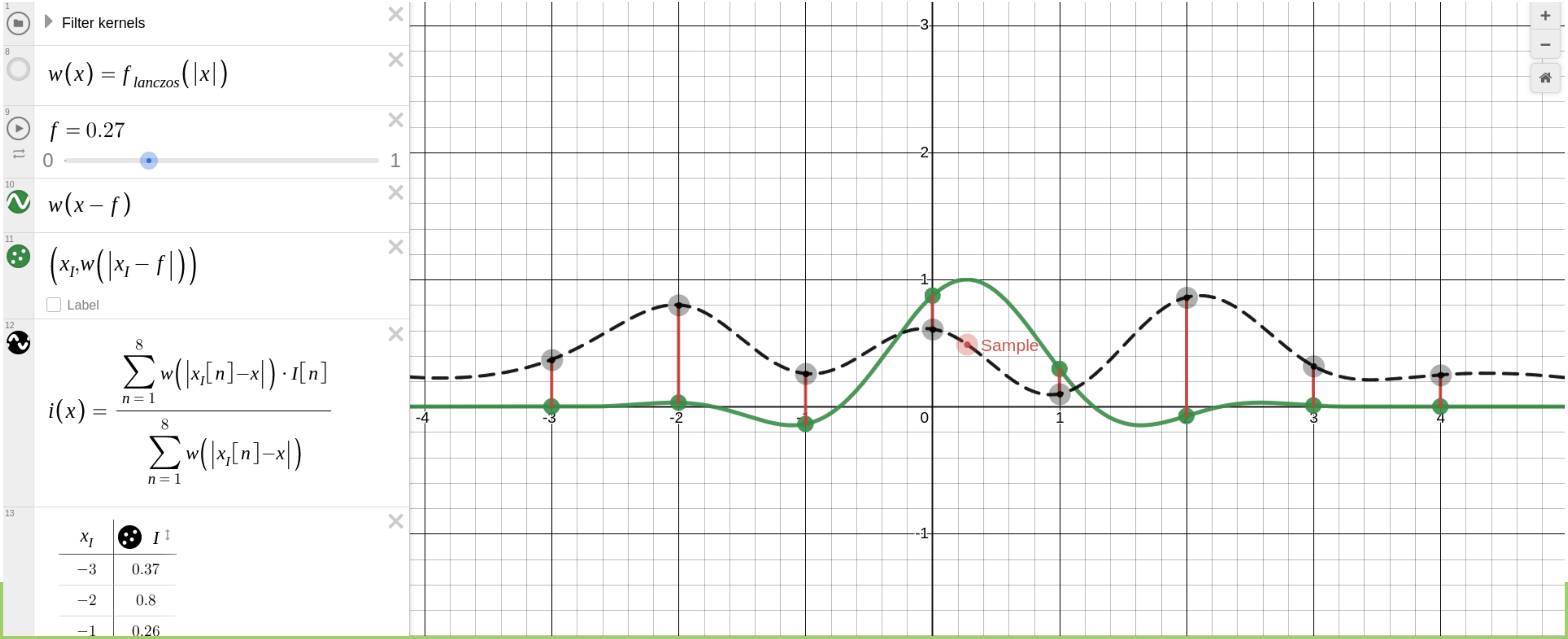
Niklas Haas (haasn)

VDD 2023



Signal reconstruction

https://www.desmos.com/calculator/bh0pwcjfns



1 Filter kernels

8 $w(x) = f_{lanczos}(|x|)$

9 $f = 0.27$

0 1

10 $w(x - f)$

11 $(x_I, w(|x_I - f|))$

□ Label

12 $$i(x) = \frac{\sum_{n=1}^{8} w(|x_I[n] - x|) \cdot I[n]}{\sum_{n=1}^{8} w(|x_I[n] - x|)}$$

13

x_I	I
-3	0.37
-2	0.8
-1	0.26

Signal reconstruction

https://www.desmos.com/calculator/bh0pwcjfns

- $I(x) = \mathbf{w_0}I_0 + \mathbf{w_1}I_1+\mathbf{w_2}I_2+\mathbf{w_3}I_3 + \dots$
- Computation of w slow, nontrivial → **cache in LUT**
- Weights only depend on subpixel offset (x – floor(x))
- Pre-compute:
- LUT(**0.0**) = {w(**-1.0**), w(**0.0**), w(**1.0**), w(**2.0**)}
- LUT(**0.1**) = {w(**-0.9**), w(**0.1**), w(**1.1**), w(**2.1**)}
- LUT(**0.2**) = {w(**-0.8**}, w(**0.2**), w(**1.2**), w(**2.2**)}
- …
- LUT(**1.0**) = {w(**0.0**}, w(**1.0**), w(**2.0**), w(**3.0**)}

Signal reconstruction

https://www.desmos.com/calculator/bh0pwcjfns

- $\mathrm{I}(x) = w_0\mathbf{I_0} + w_1\mathbf{I_1}+w_2\mathbf{I_2}+w_3\mathbf{I_3} + \ldots$
- Requires one **texture fetch per input pixel**
- → often bottleneck

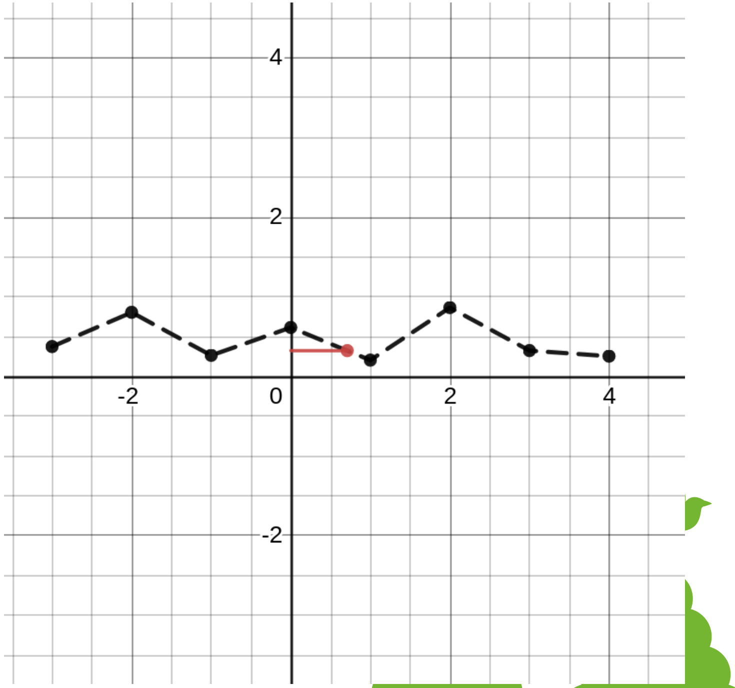
… if only there was a better way?

Linear interpolation

https://www.desmos.com/calculator/y7ryqqn6xe

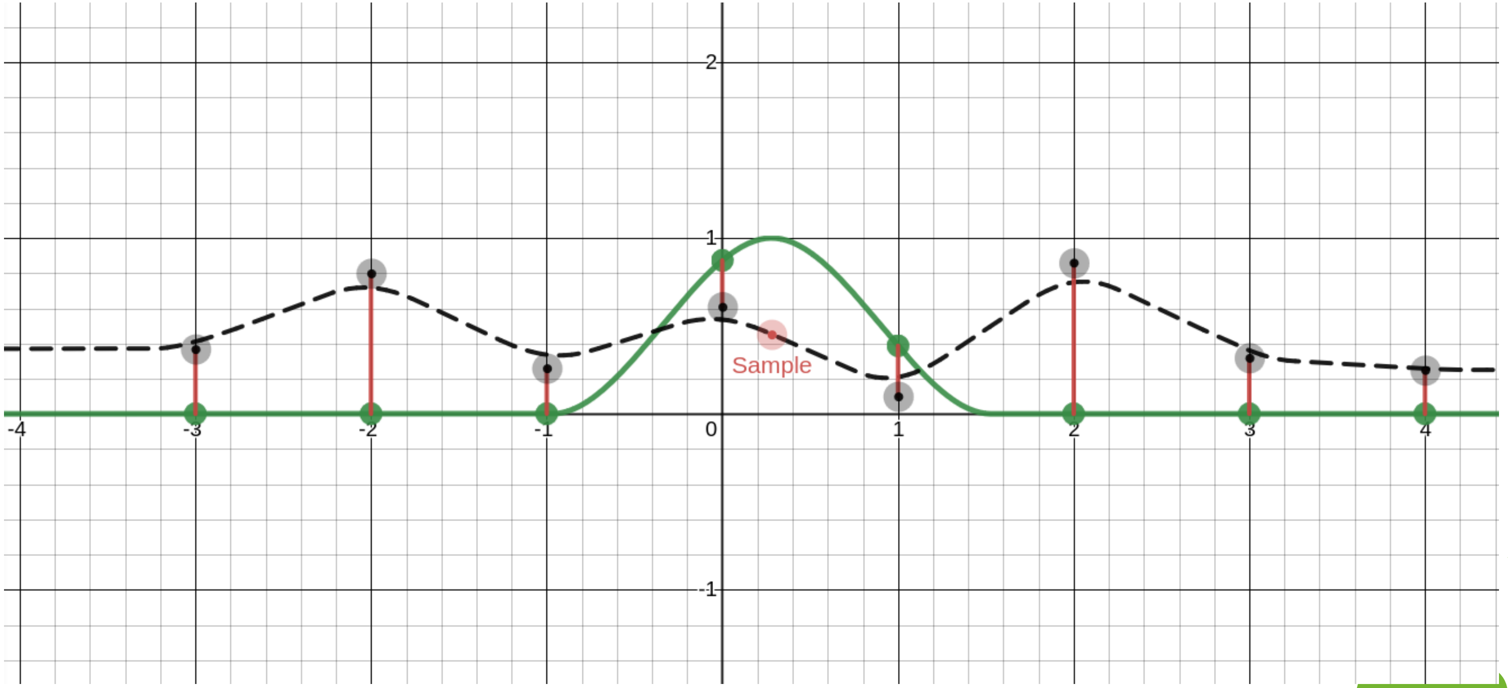
- GPUs very good at linear sampling
- We get: $I(n+\mathbf{f}) = (1 - \mathbf{f})\cdot I(n) + \mathbf{f}\cdot I(n+1)$
- We want: $\mathbf{w_0}\cdot I(n) + \mathbf{w_1}\cdot I(n+1)$
- Solve: $(\mathbf{w_0+w_1})\cdot I(n+\mathbf{w_1/(w_0+w_1)})$

- **Constraint: $\mathbf{sign(w_0) = sign(w_1)}$!!**

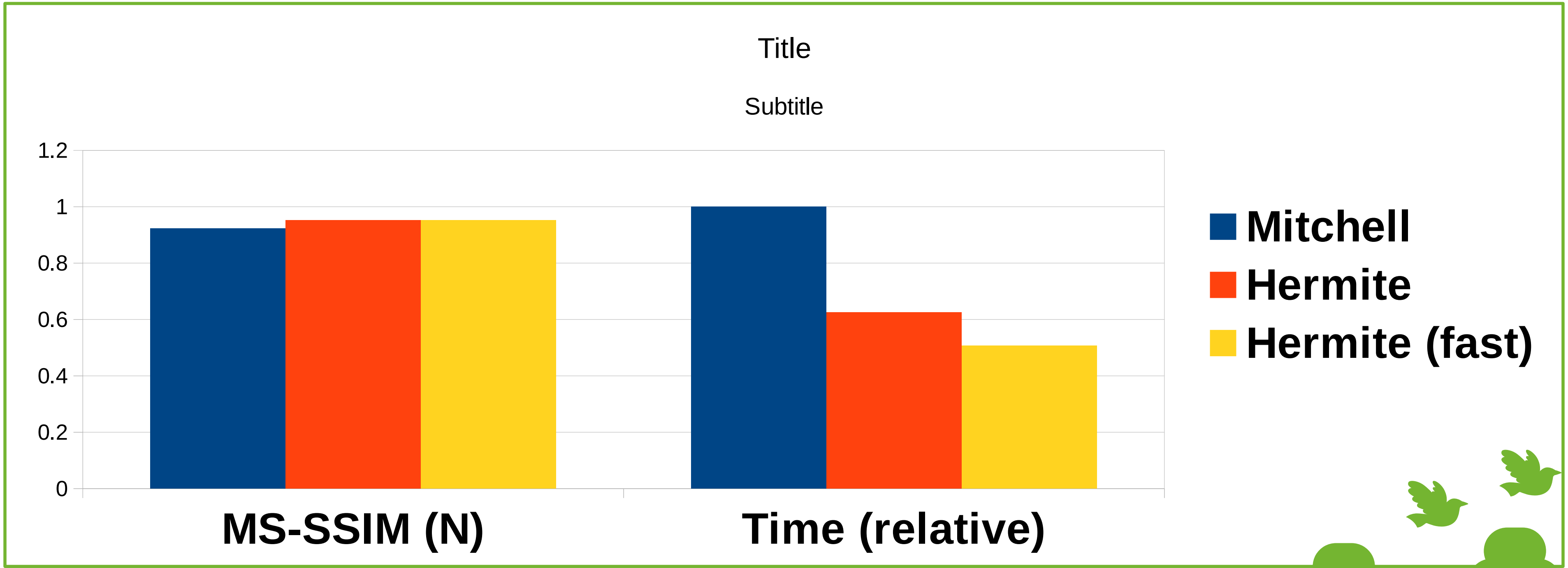


Hermite is love, Hermite is life

https://www.desmos.com/calculator/yksrz8lbyn



Hermite vs Mitchell (downscaling 8K)



→ **Now default in libplacebo v6.337**

More topics

- Cylindrical/Polar sampling tricks
- Compute shaders, loop unrolling, conditional texture reads
- **Novel anti-ringing technique**, based on PowerMean
- … general GPU development

→ **Ask me! (@haasn)**