[Lewcid.com](#) (by Subject) (by Time) (by Folder) (by Lewey Geselowitz)

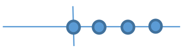
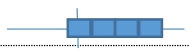
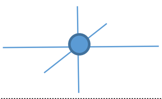
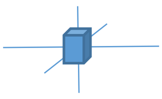
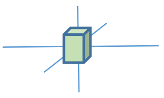
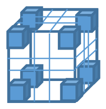
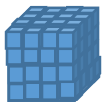
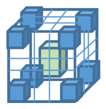
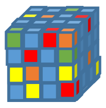
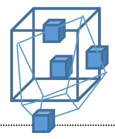
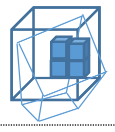
[THIS WORK IS IN PROGRESS AND NOT REVIEW READY]

The Holographic Interval

Most information is only ever known to a certain measure accuracy or precision (9 ± 0.5, 8.5~9.5, 9.0 ± 0.05, etc.), the range of that accuracy is what we refer to as the interval. These intervals or ranges utilize the holographic principal to represent numbers or other kinds of information by encoding only their boundary.

For use in computer graphics, I utilize the following notation to simplify the process of working with intervals, which has implications reaching from robotic control theory to time-and-pixel aware motion-blur in movies and video game rendering:

Interval and Cubic Operators

Notation	***Definition***	***Term***	***Icon***	***Explanation***
a	$= a$	point		*single precision point*
$\tilde{a}$	$= [\, b, c\,)$ $= \{x \mid b \le x < c\}$	range		*The range from B to almost C*
$\ddot{a}$	$= \{\, \tilde{b}\, , \ldots , \tilde{c}\, \}$	tiling		*tiling defined from range B to range C*
$\overset{\frown}{a}$	$= [\, a_0, \ldots, a_n]$	array		*Array of values*
$\overset{\frown}{\ddot{a}}$	$= [\, \tilde{b}, \ldots, \tilde{c}\,]$	tile array		*Array of tiles*
$\vec{a}$	$= \{\, x, y, z\, \}$	cubic point		*A 3-dimensional point*
$\vec{\tilde{a}}$	$= \{\, \tilde{x}, \tilde{y}, \tilde{z}\, \}$	cubic range		*A 3-dimensional range*
f	$= f(\, \vec{\tilde{a}}\,)$	cubic function		*A function which takes a cubic range and returns a value*
$\vec{\ddot{a}}$	$= \{\, \ddot{x}, \ddot{y}, \ddot{z}, \ddot{w}\}$	cubic tiling		*The intersection of 3 tiled dimensions*
$\overset{\frown}{\vec{\ddot{a}}}$	$= [\vec{\tilde{x}}_\iota \mid \vec{\tilde{x}}_\iota \in \vec{\ddot{a}}]$	cubic tile array		*A 3-dimensional array of tiles*
v_i	$= \left(\overset{\frown}{\vec{\ddot{a}}}\right)_i$	volume element "voxel"		*An tile within the 3-dimensional array of tiles*
V	$= \overset{\frown}{f(v_i)}$	cubic buffer "data cube"		*An cubic array of the values, each associated with a tile*
P	$= \vec{\ddot{a}}$	cubic projection		*The projection or dimensional rotation*
$P(\vec{\tilde{x}})$	$= (\vec{\tilde{x}} * P) \cap \overset{\frown}{\vec{\ddot{a}}}$	projected voxels covered by *x*		*The voxels covered by a point x*

-Lewey Geselowitz