Global Static Indexing for Real-time Exploration of Very Large Regular Grids

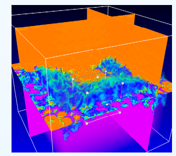
Valerio Pascucci and Randall J. Frank

pascucci,rjfrank@llnl.gov

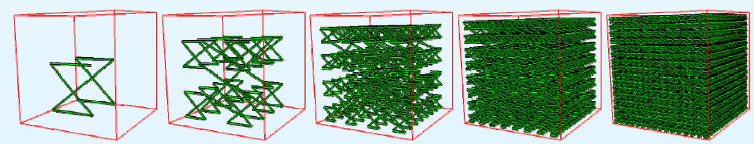
Lawrence Livermore National Laboratory

Outline

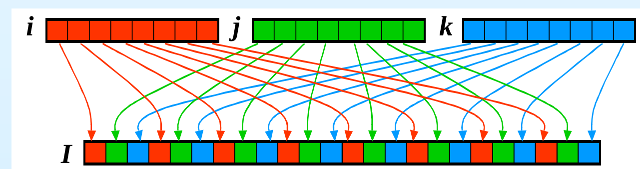
- **Motivation**



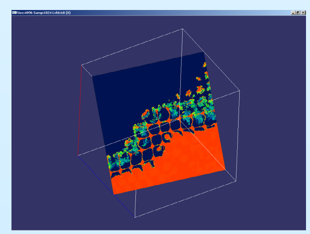
- **Previous work** (bla bla[vis`01], bla bla bla [sc`00] ……)
- **Data layout**



- **2^n tree indexing**



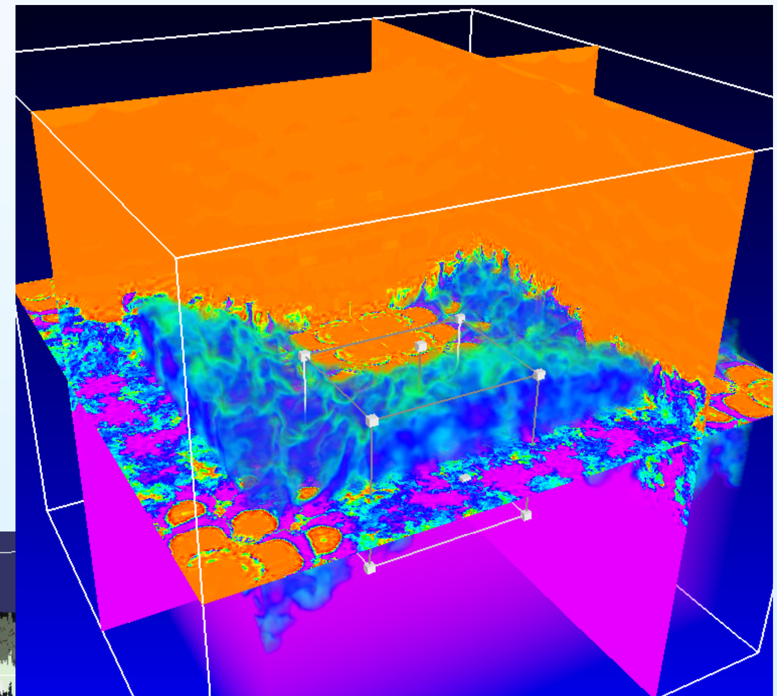
- **Performance for slicing large grids**



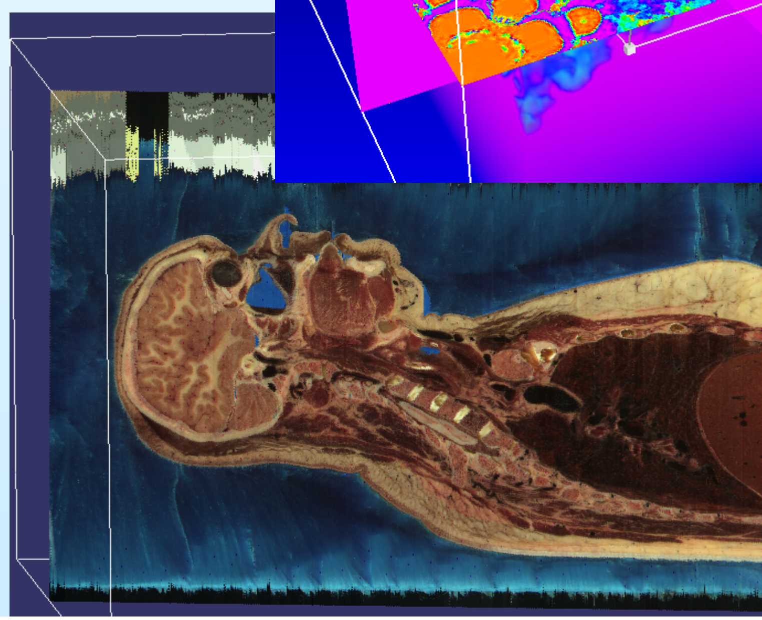
- **Conclusions and future work**

We must achieve real-time interaction with large datasets on a wide variety of platforms.

The problem



- **Extremely large datasets 0.5TB/timestep (8k,8k,8k,(time)).**
- **Interactive rendering for real-time data exploration.**
- **Target platforms: desktop, parallel server, cluster.**



Previous Work

- **Out-of-core geometric algorithms**

  [Goodrich, Tsay, Vengroff, Vitter ‘93]

  [Vitter ‘00][Matias, Segal, Vitter ‘00]

  [Asano, Ranjan, Roos, Welzl ’95][Arge Miltersen ’99]

- **Out-of-core visualization**

  [Chiang, Silva ‘97][Sutton, Hansen ‘99]

  [Livnat,Shen,Johnson ’96][El-Sana, Chiang’00]

  [Bajaj, Pascucci, Thompson, Zhang ‘99]

- **Space filling curves**
  (image processing, multidimensional databse, geomtric datastructure …)

  [Bandou, Kamata.’99][Balmelli, Kovacevic, Vetterli ’99]

  [Parashar, Browne, Edwards, Klimkowski ’97]

  [Niedermeier, Reinhardt, sanders ‘97][wise’00]

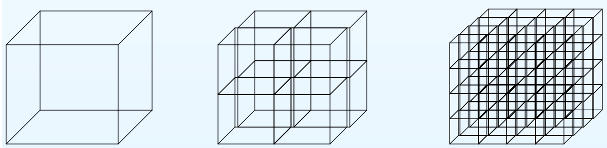
  [Hans Sagan ’94] [Lawder king ’00][Griebel Zumbusch ‘99]

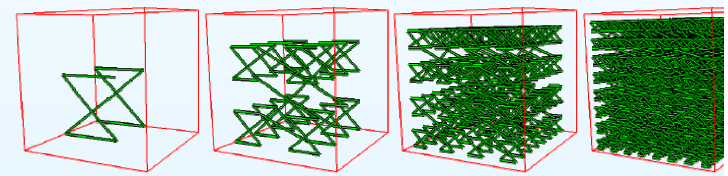
We apply three fundamental techniques to the visualization of large simulation data.

Our approach

- Multi-resolution geometric representation:
  - adaptive view-dependent refinement;
  - minimal geometric output for selected error tolerance.
- Cache oblivious external memory data layouts:
  - exploit spatial and resolution coherency;
  - no need for complicated paging techniques.
- Progressive processing:
  - continuously improved rendering;
  - scalability with the resources without budgeting.

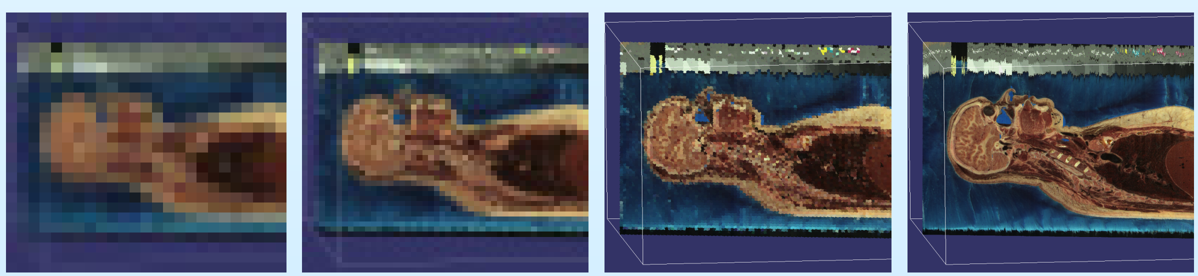
We focus on the progressive computation of slices (any orientation) of large 3D rectilinear grids.





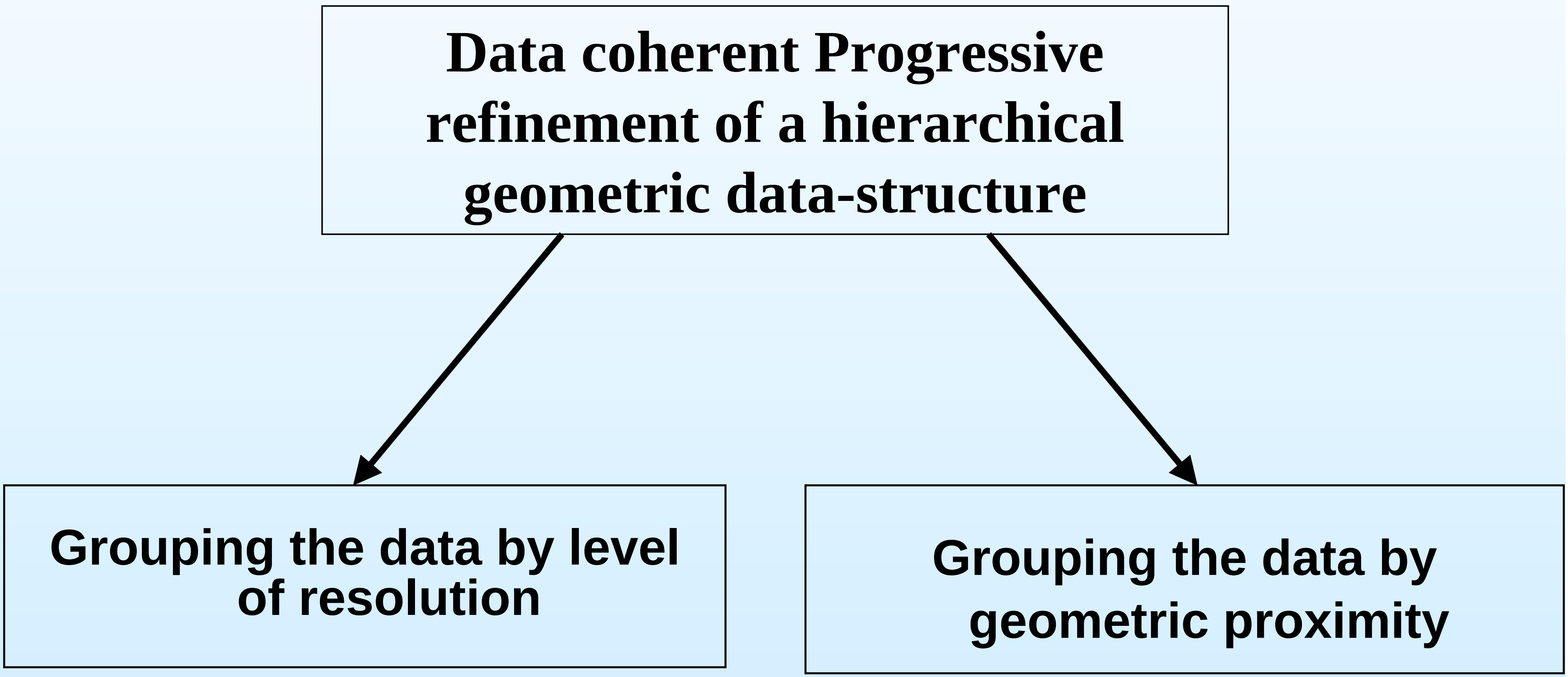
- Rectilinear grid
- Sub-sampling octtree

- 1D order = hierarchical 3D Z-order curve

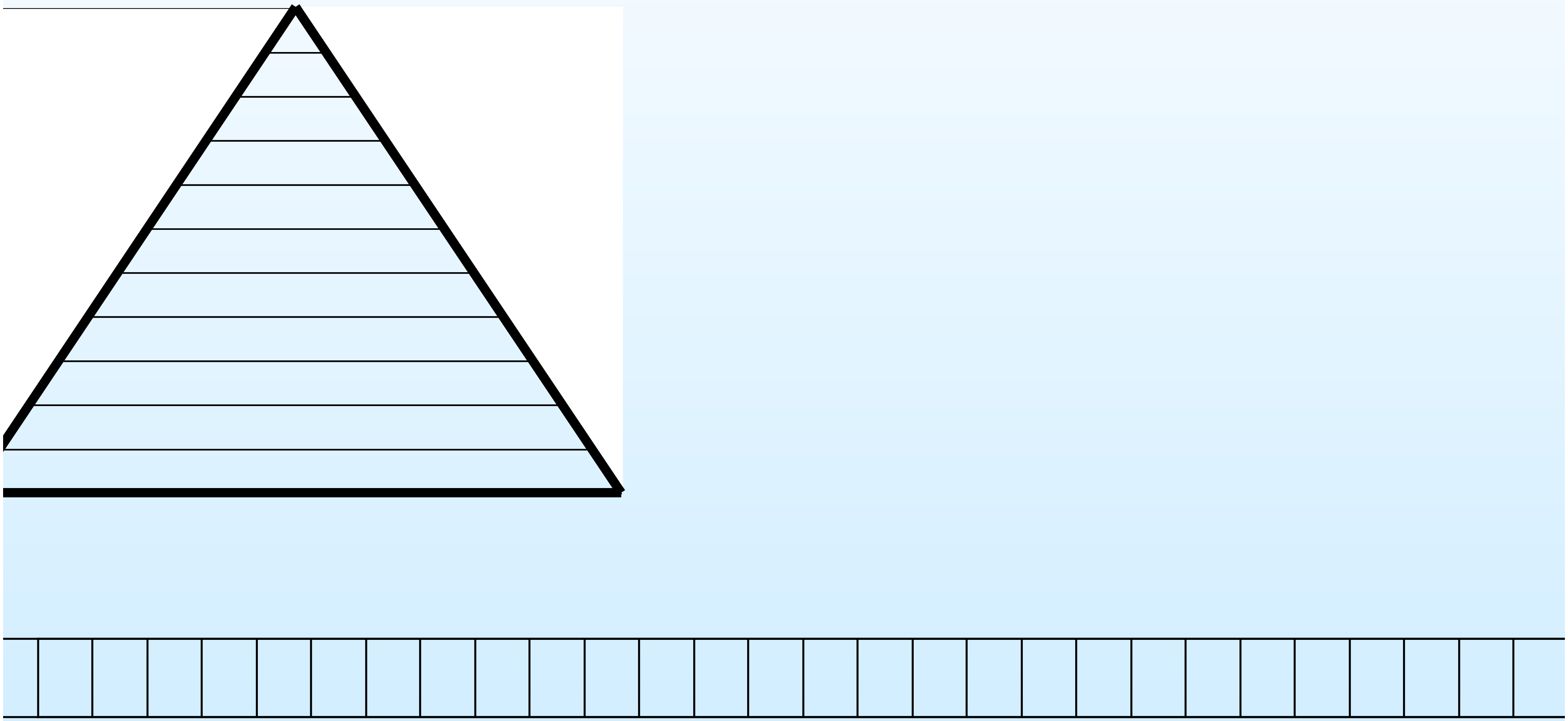


- Coarse to fine slice refinement

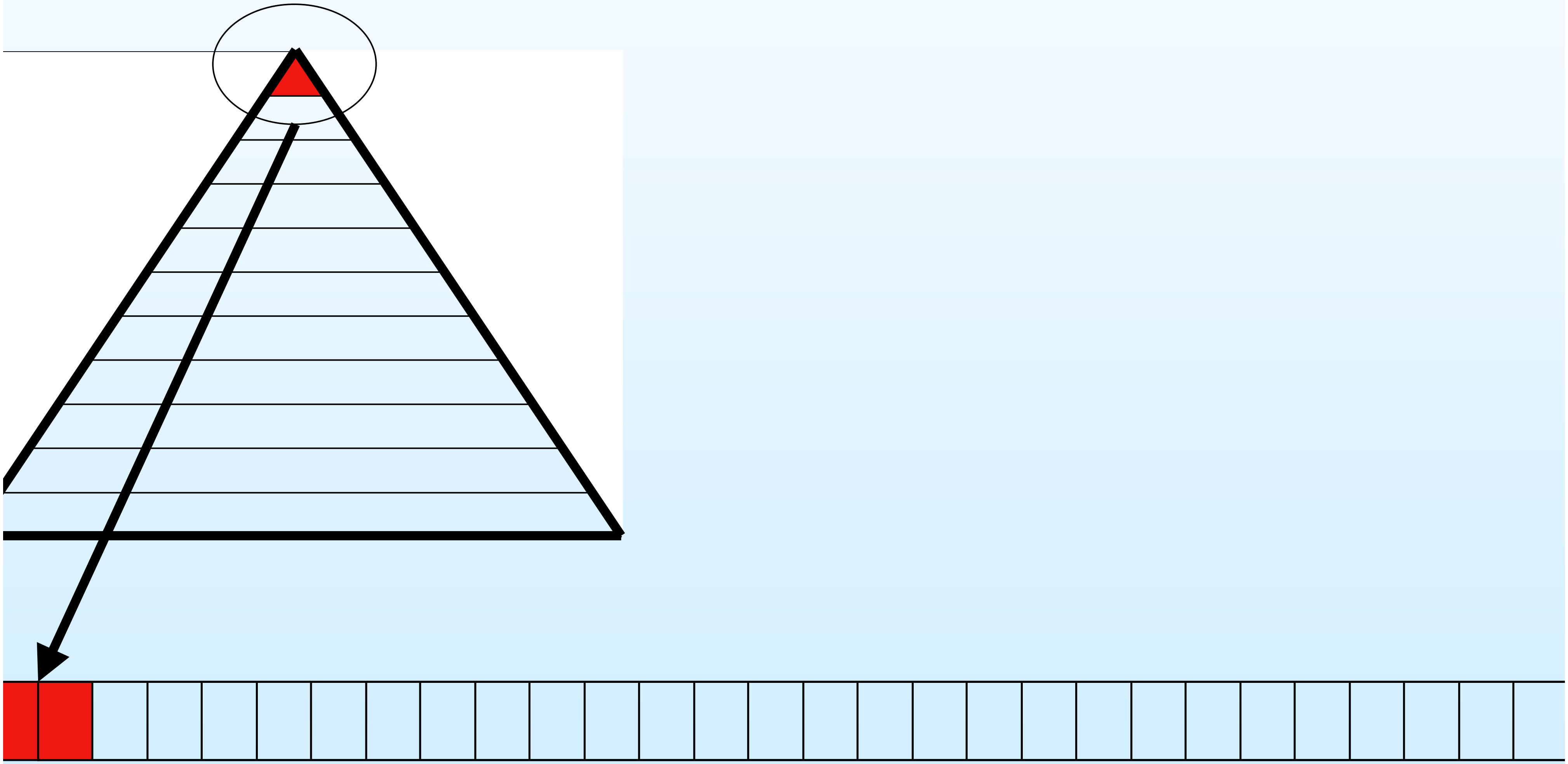
General Data Layout



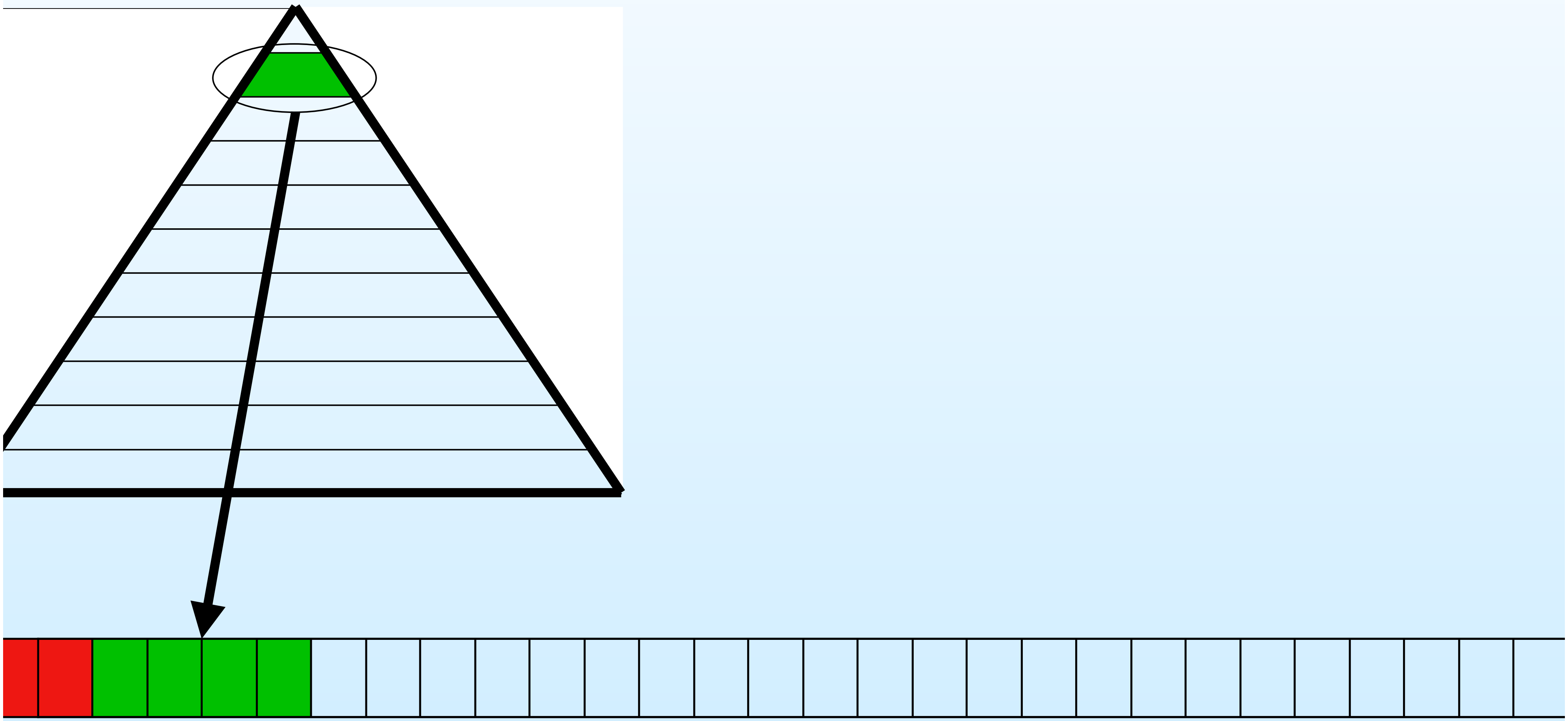
General Data Layout



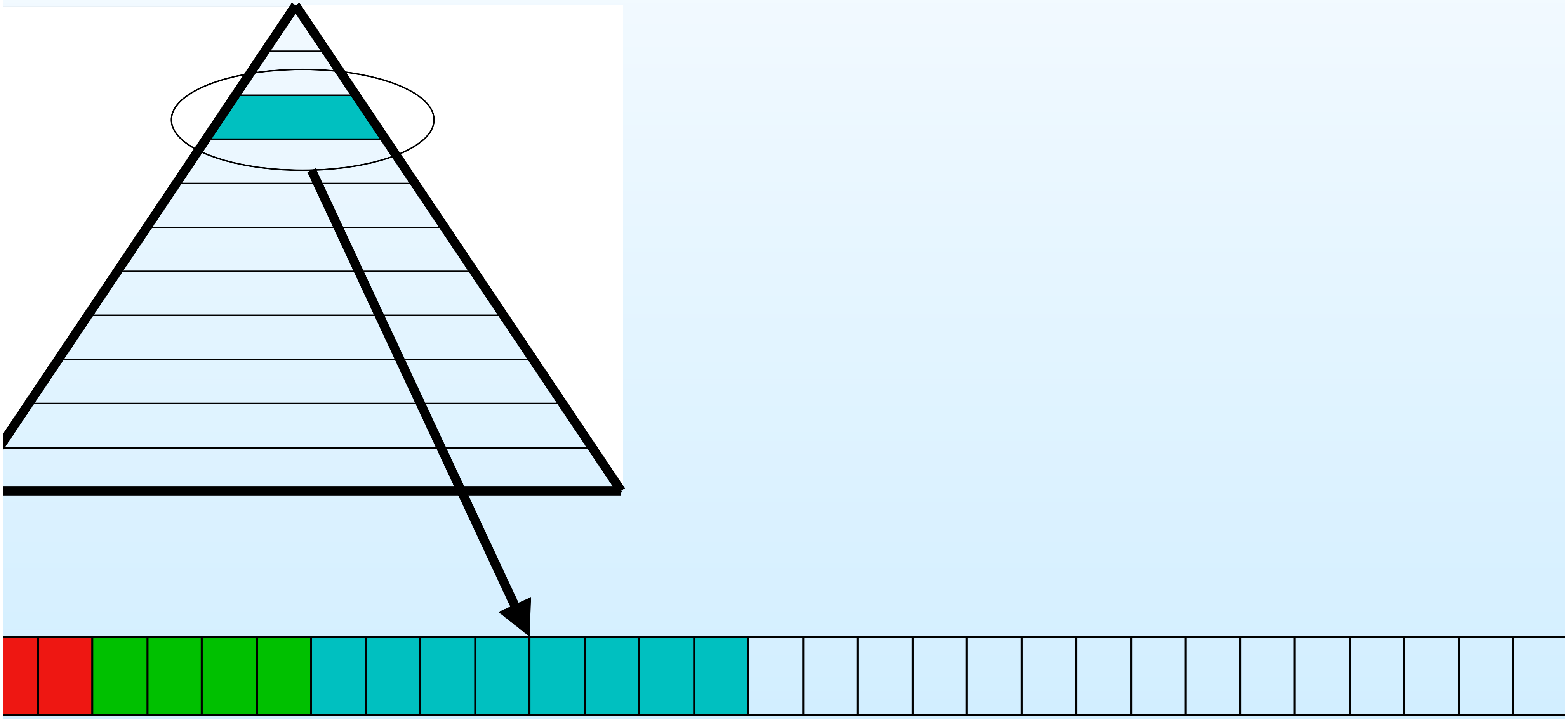
General Data Layout



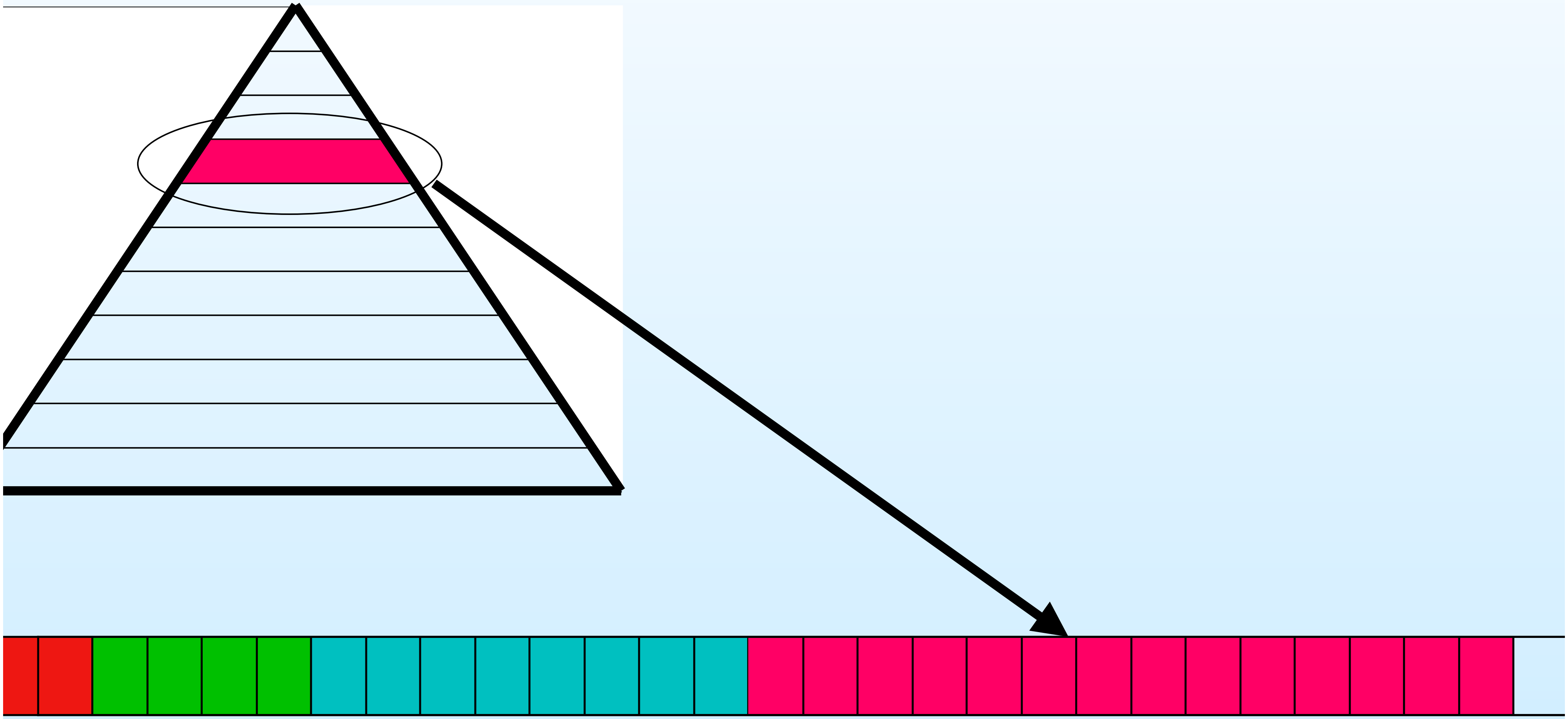
General Data Layout



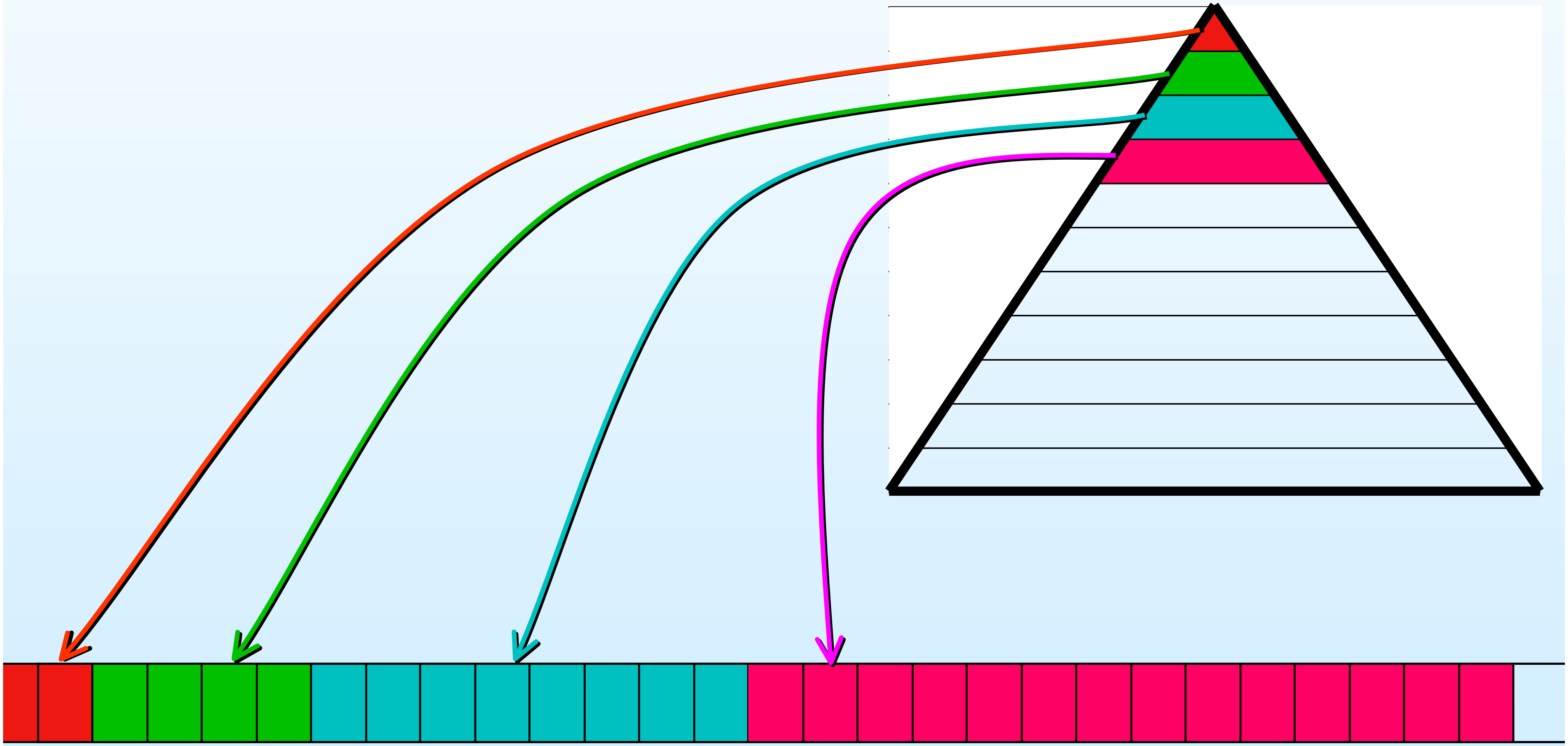
General Data Layout



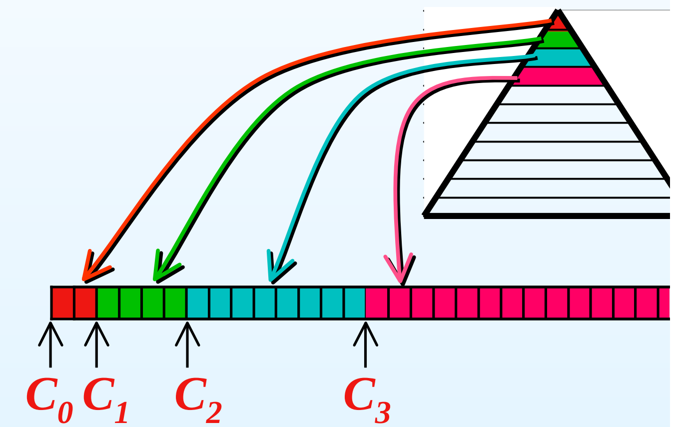
General Data Layout



General Data Layout



General Data Layout



nD to 1D mapping:

$$I \rightarrow I^*$$

$I \rightarrow l$ find the level of resolution l

C_l (pre)compute the number of elements in the levels coarser than l

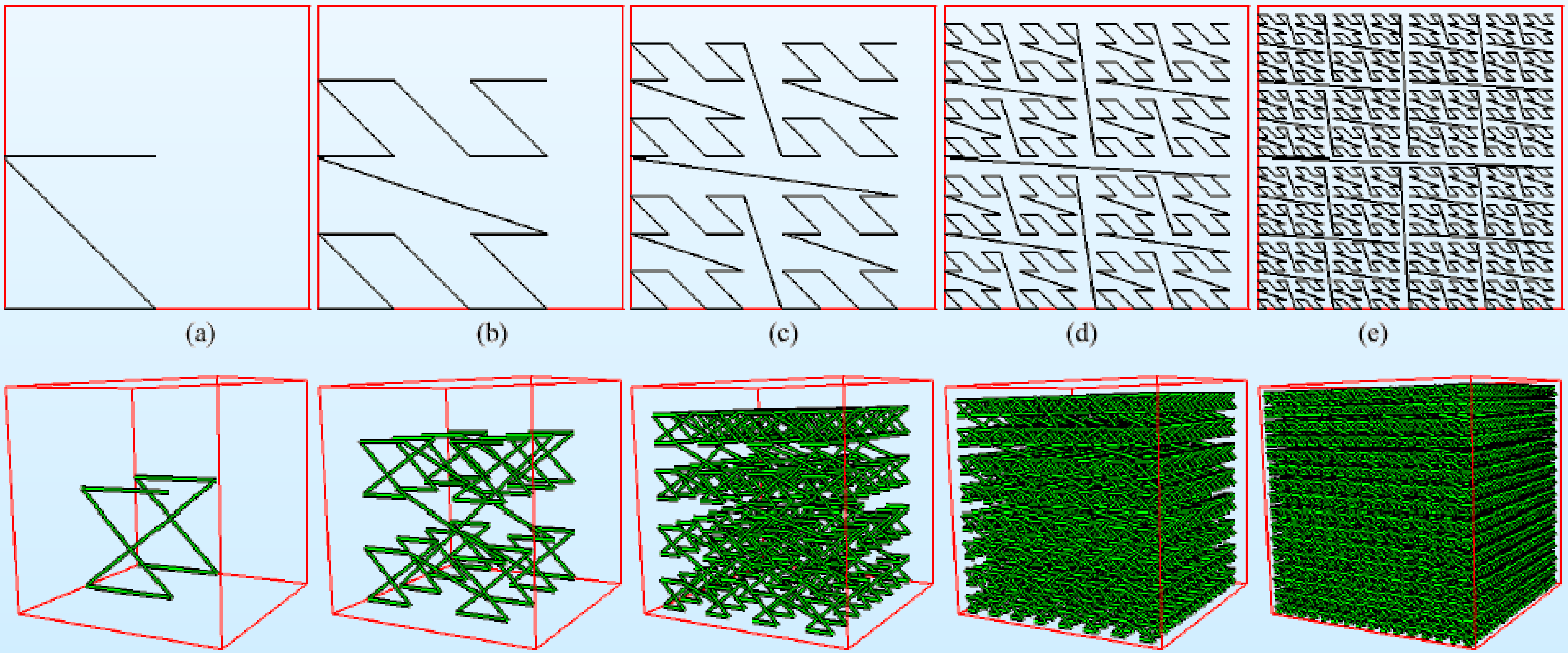
$I \rightarrow I'$ index of the element within its level of resolution

$$I^* = C_l + I'$$

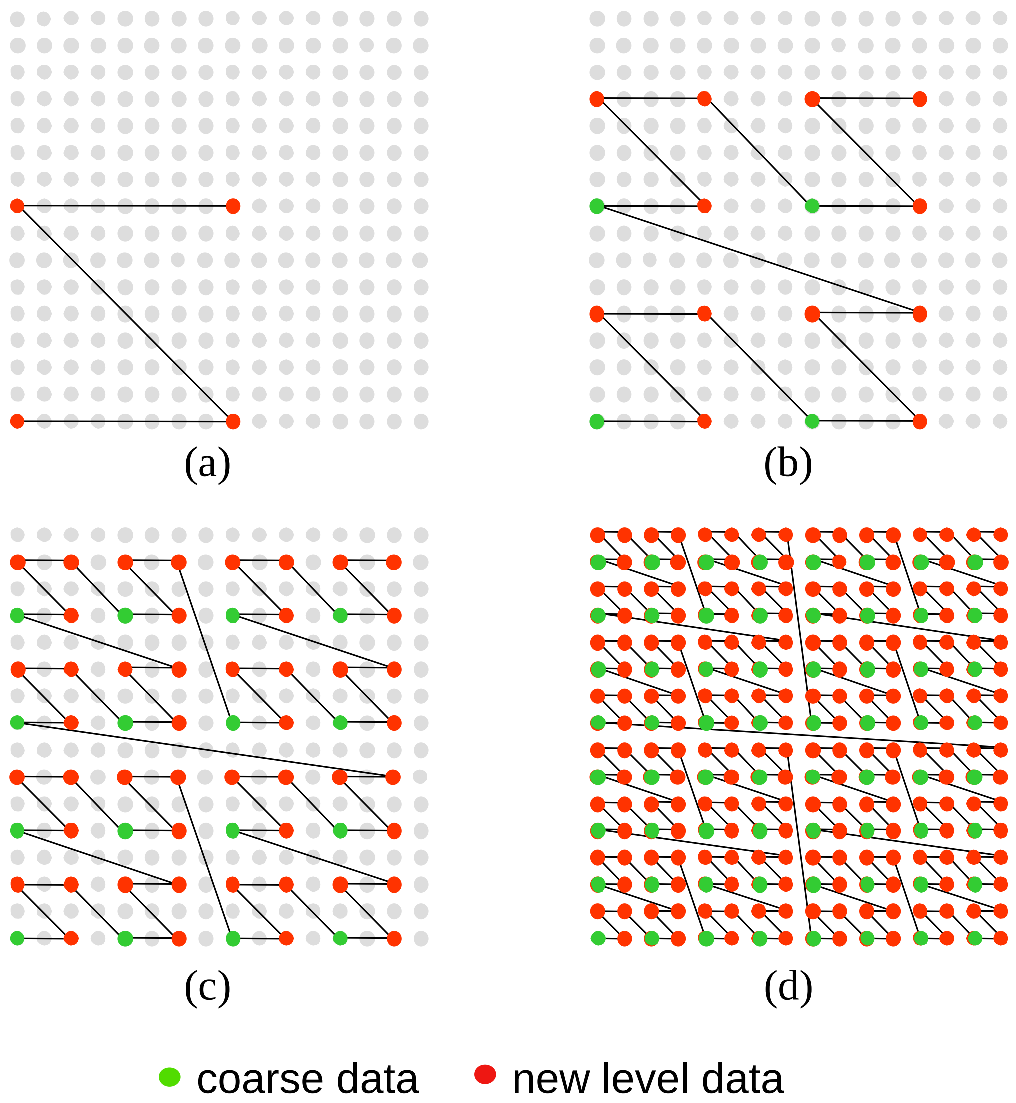
We exploit the correlation of bin/quad/oct-trees with the Lebesgue space-filling curves.

The Lebesgue curve is also known as Z-order, Morton, …. Curve.

Special case of the general definition introduced by Guiseppe Peano in 1890.

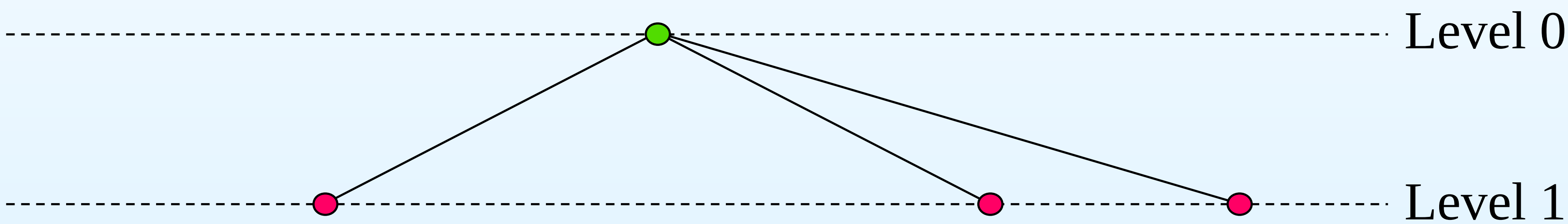


We turn the recursive definition of the Z-order curve into a hierarchical subsampling scheme.



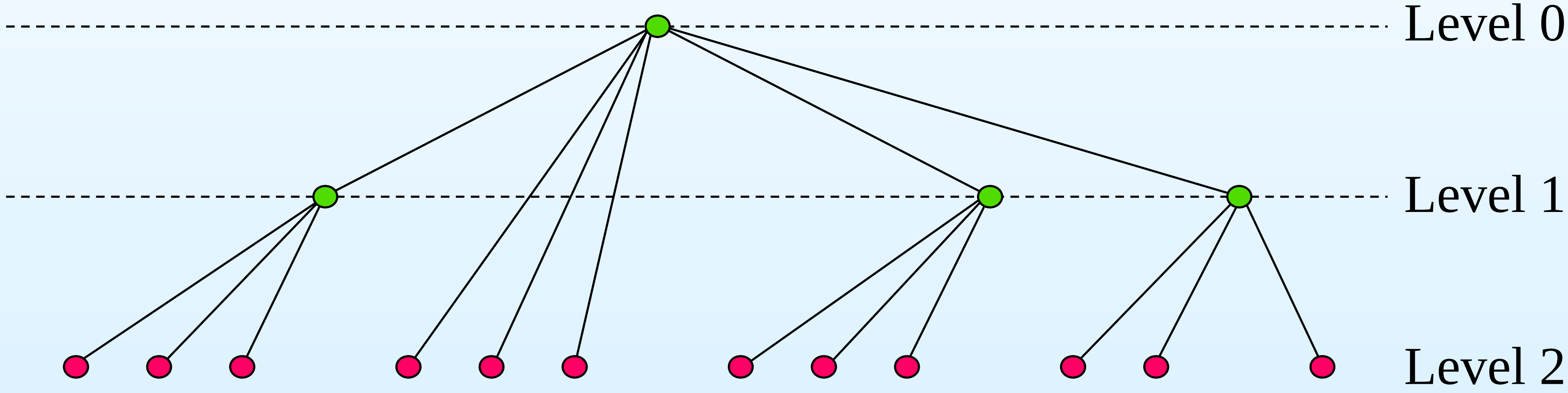
We obtain a multi-resolution hierarchical representation which is not exactly a 2^n-tree.

- **Not exactly a quad-tree ……**



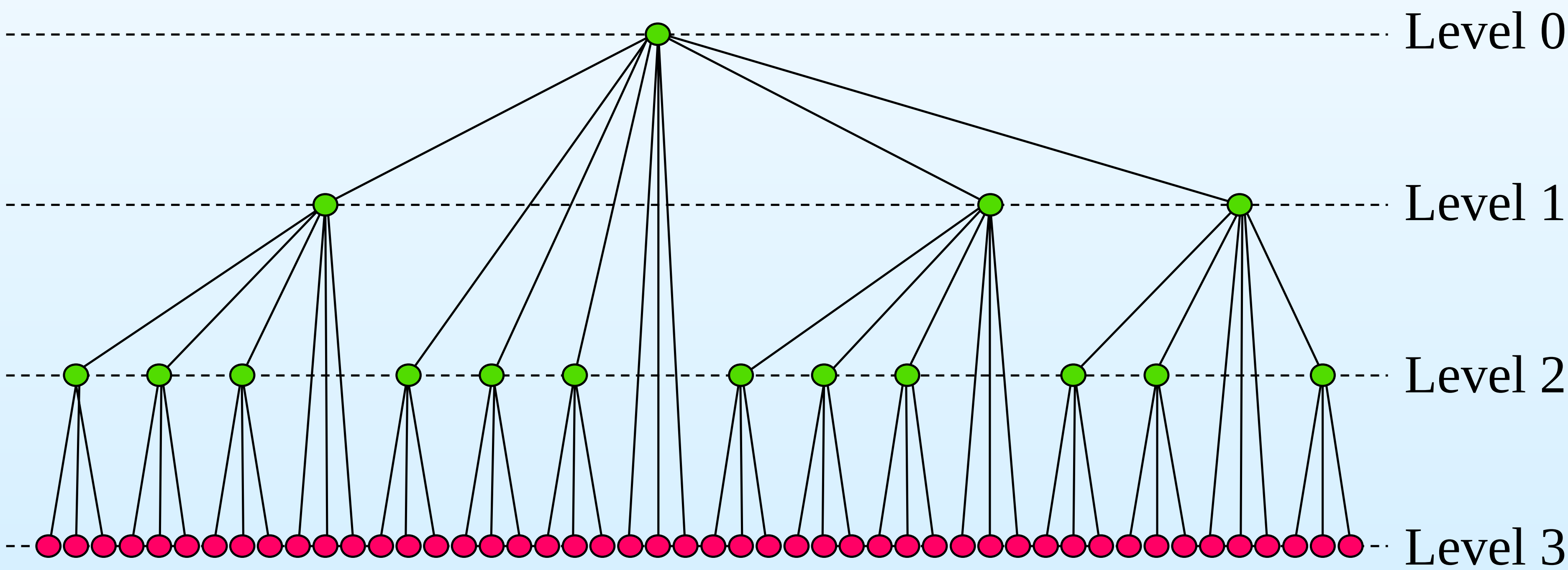
We obtain a multi-resolution hierarchical representation which is not exactly a 2^n-tree.

- **Not exactly a quad-tree ……**

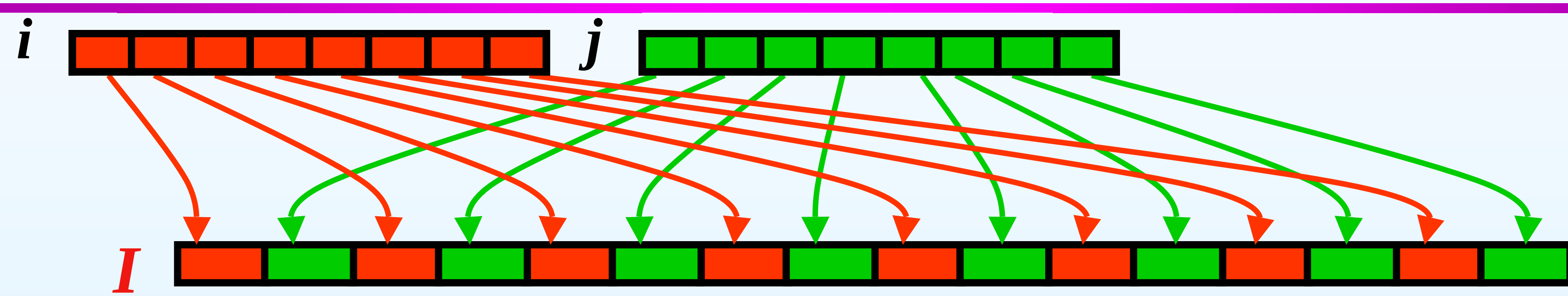


We obtain a multi-resolution hierarchical representation which is not exactly a 2^n-tree.

- **Not exactly a quad-tree ……**



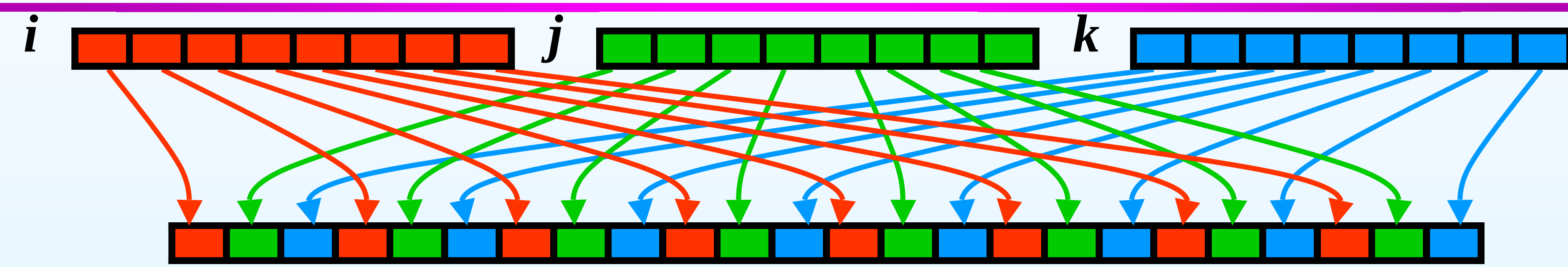
The 1D index I^* can be computed in a simple and efficient way in any dimension.

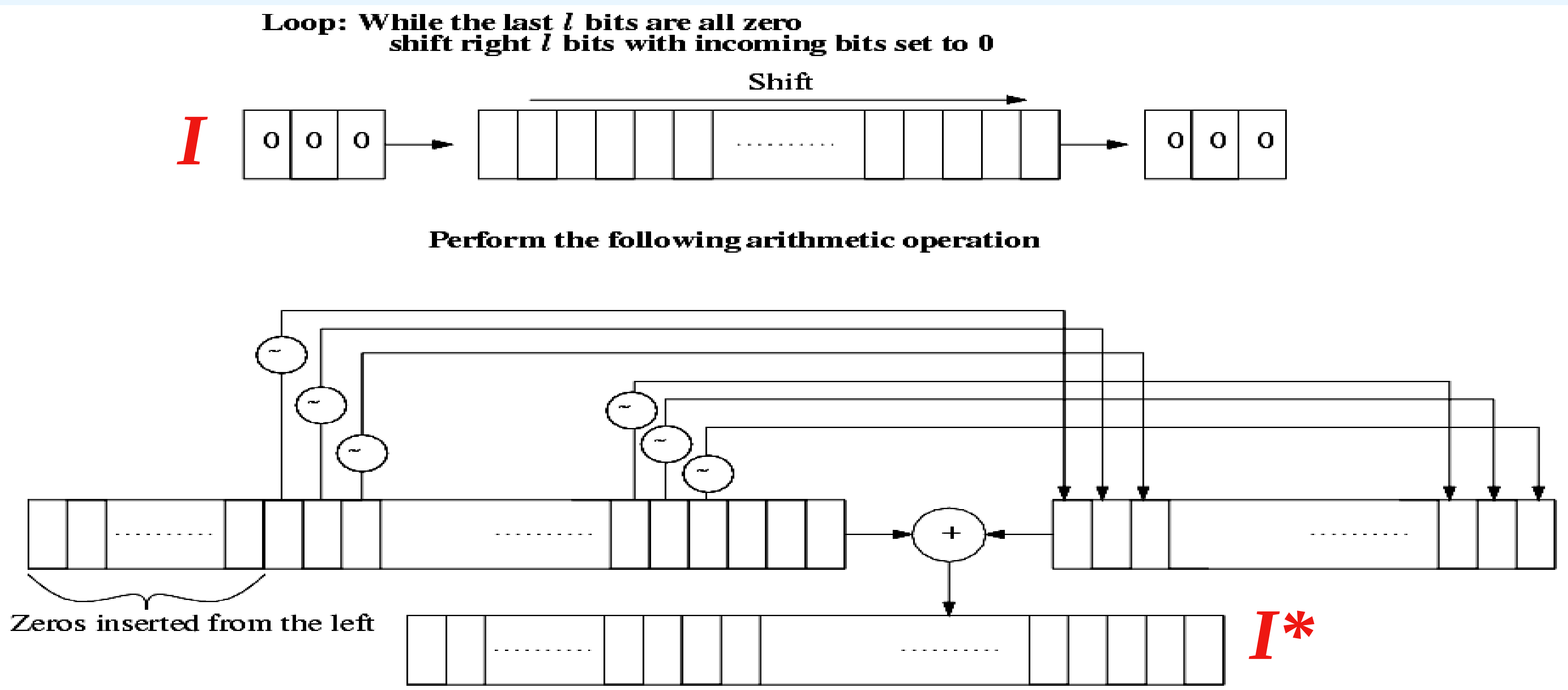


		0	1			2											
I	Z-order 0	0															
	Z-order 2	0	1	2	3												
	Z-order 4	0	4	8	12	1	2	3	5	6	7	9	10	11	13	14	15
*I**		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

$$I^* = C_l + I' \qquad C_l = 2^{2(l-1)} \qquad I' = \left\lfloor \frac{I}{2^{2l}} \right\rfloor - \left\lfloor \frac{I}{2^{2(l+1)}} \right\rfloor - 1$$

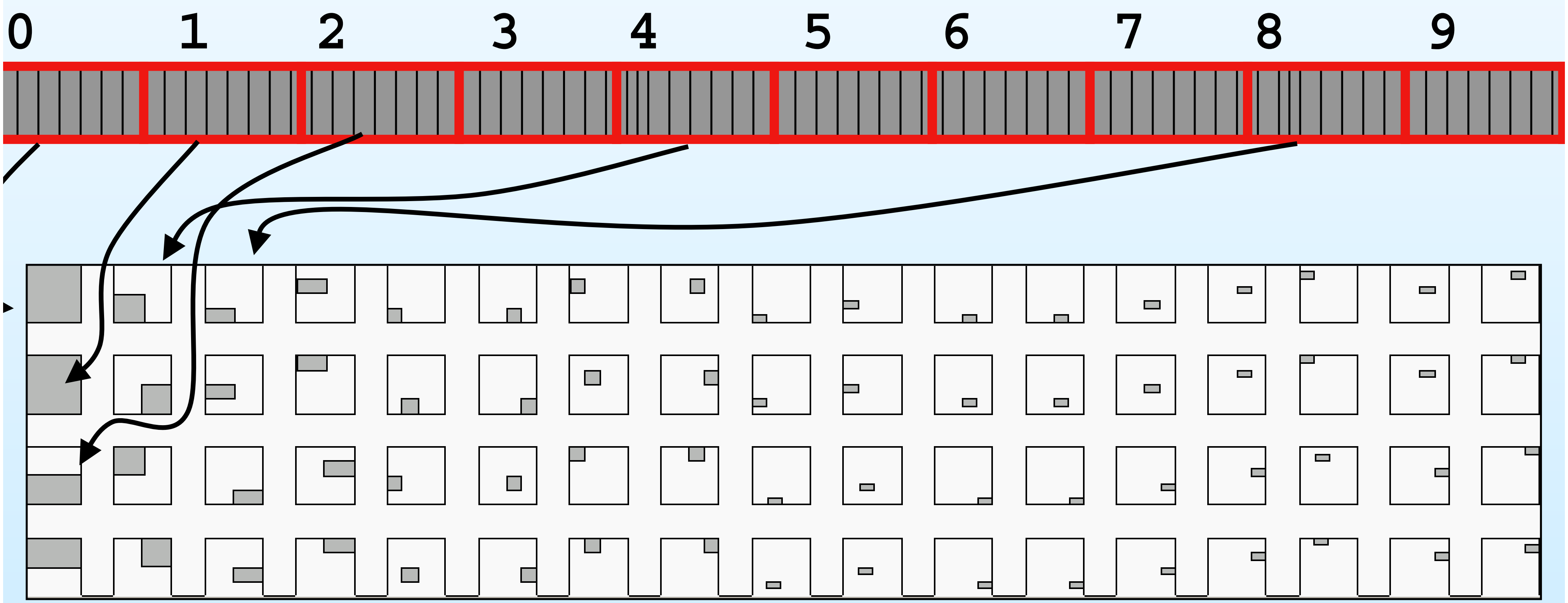
The 1D index *I** can be computed in a simple and efficient way in any dimension.





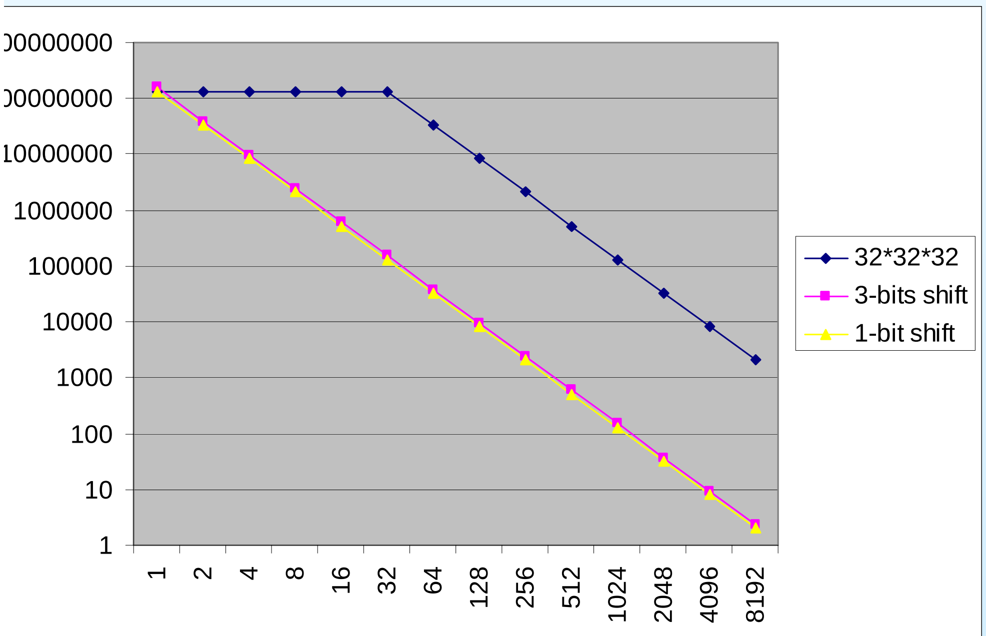
Overall the hierarchical Z-order yields a cache oblivious hierarchical data layout.

Distribution in the grid of each constant size block of data

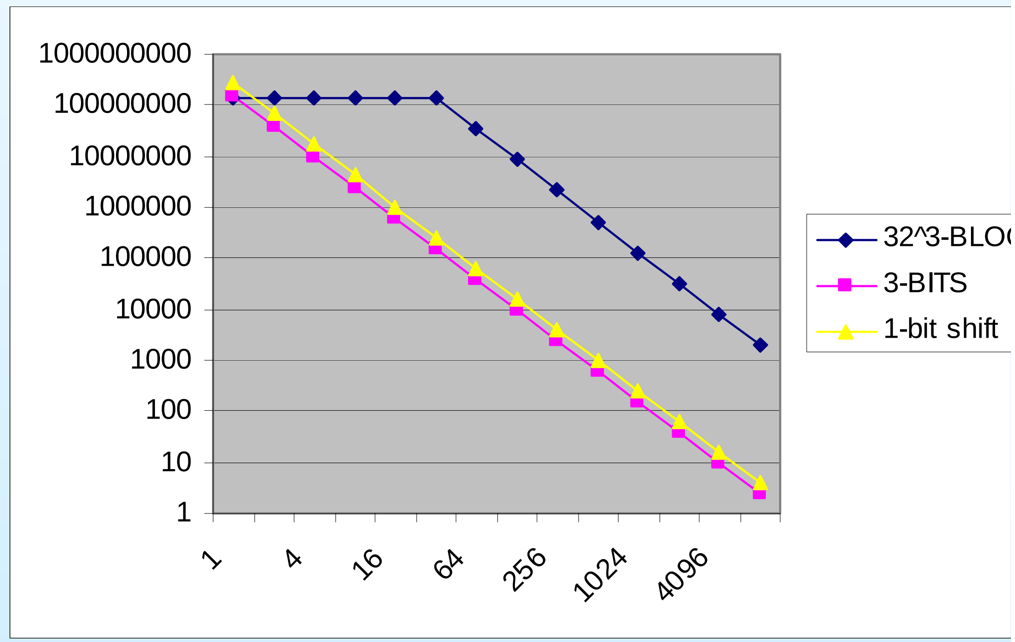


Theoretical analysis shows a gain of orders of magnitude independently of the block size.

Cache oblivious !!!



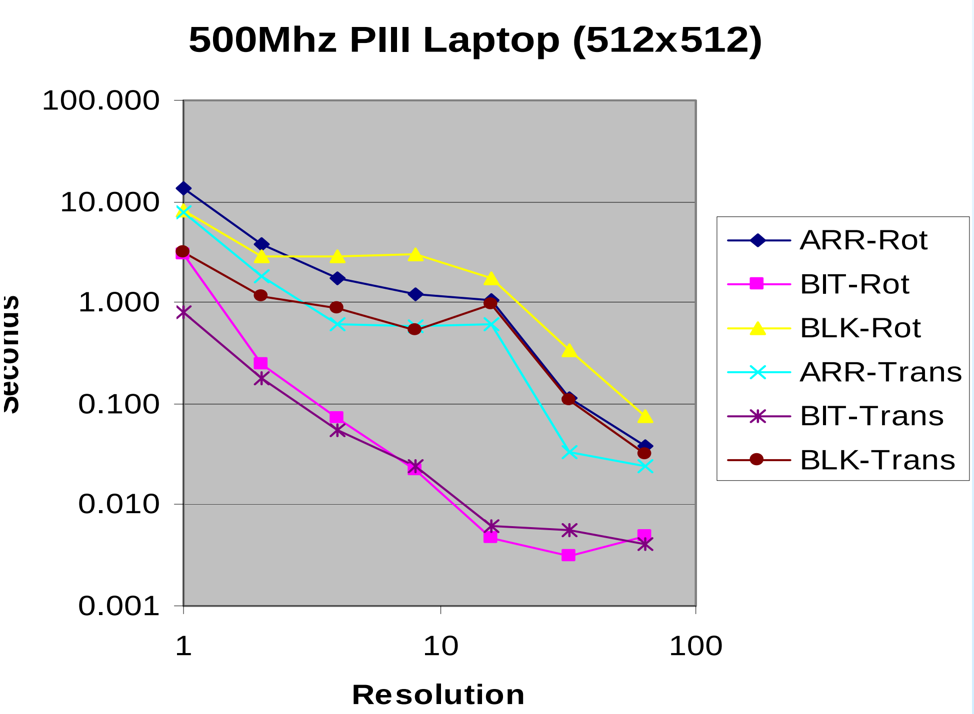
32K blocks

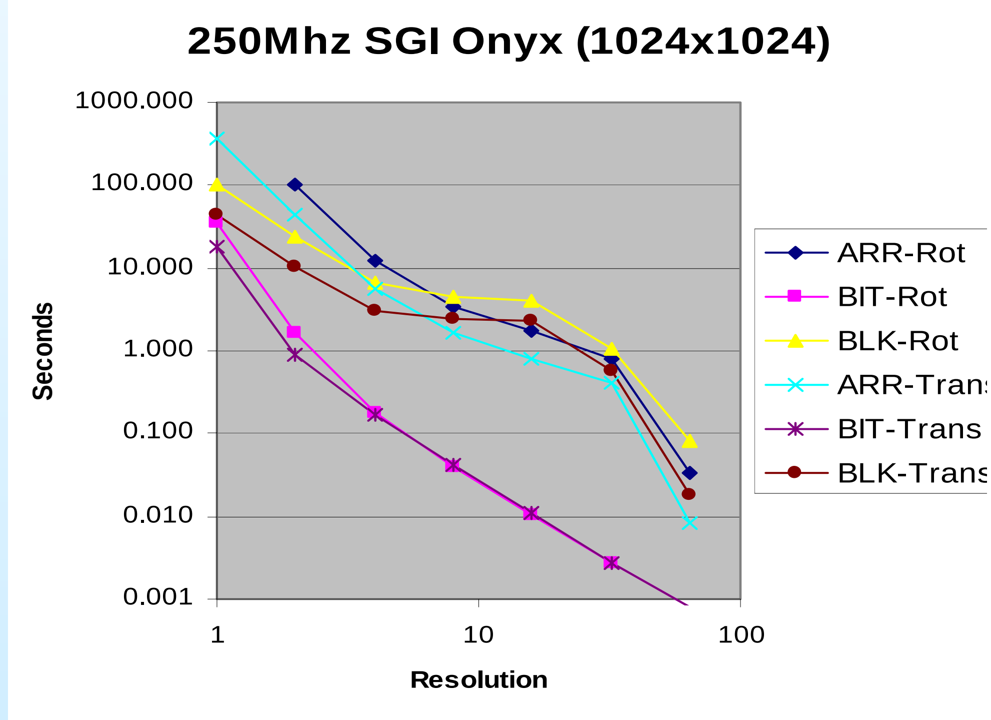


64K blocks

Real speedup matches theoretical expectations: more than 10x improvement, platform scalable.

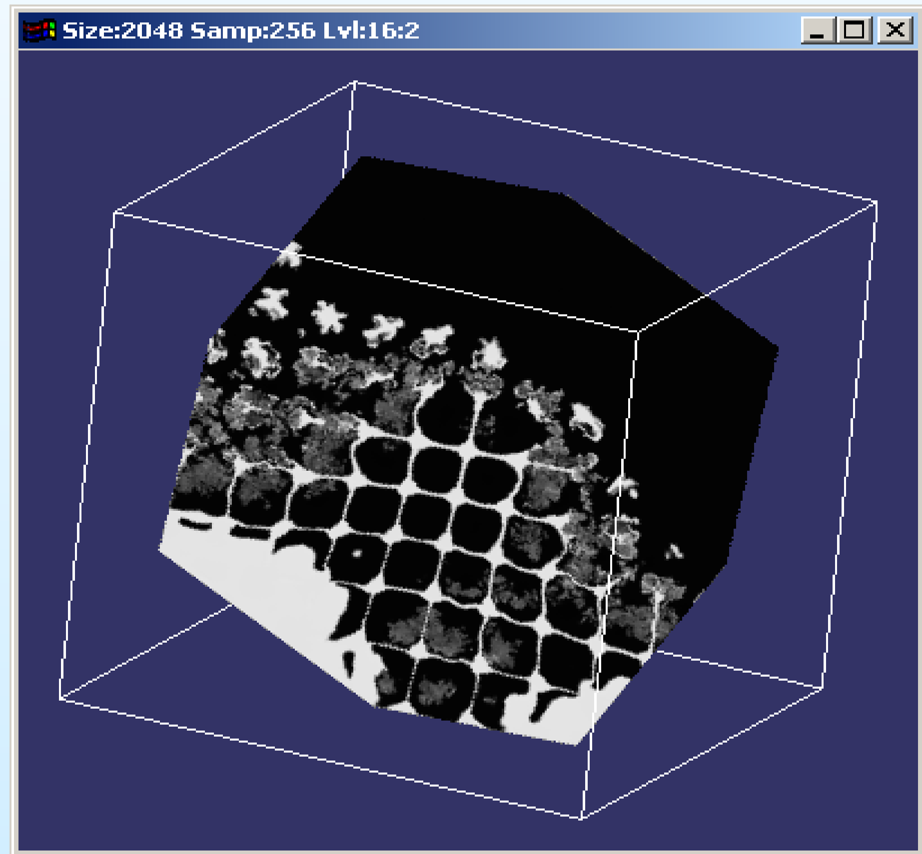
- **2048x2048x1920 dataset (we have run up to 8192x8192x7680)**
- **20MB memory cache**
- **Translation and rotation tests (average over 3 primary axis)**





Conclusions and Future Work

- **Results: an implicit scheme for coherent spatial, multi-resolution regular grid data access**
  - Simple address remapping
  - Read/write access
  - No additional data overhead
  - Supports progressive access
- **Near term applications**
  - Volume rendering
  - Time critical iso-contouring
- **Future work**
  - "View dependent" parameterization
  - Unstructured/temporal hierarchies
  - Improved interpolation
  - Distributed implementation



UCRL-VG-143542

This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48.