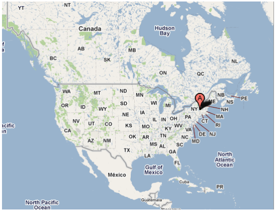
CUDA-Accelerated HD-ODETLAP: Lossy High Dimensional Gridded Data Compression

W. Randolph Franklin, You Li[1], Tsz-Yam Lau, and Peter Fox

Rensselaer Polytechnic Institute
Troy, NY, USA



MAT4GIScience, 18 Sept 2012

Partially supported by NSF grants CMMI-0835762 and IIS-1117277.

[1] PhD, RPI, 2011

The problem: Compress hi-dimensional data

- Large and larger amounts of hi dimensional data...
- in environmental, meteorological, CFD domains.
- Need to compress it.
- Possible because
  - The data is auto-correlated in every direction, and
  - more processing available.
- *Impact:* Store, transmit more data for better climatological & environmental analysis & prediction.

Compression Basics

- Lossy or lossless.
- Lossy is much more compact.
- Limited input data precision $\Rightarrow$ lossy ok.
- Most compression is only 2D.
- Compress higher dimensional data in slices.
- Ignores auto-correlations $\Rightarrow$ suboptimal.

2D ODETLAP –Overdetermined Laplacian Method

Basis for this work

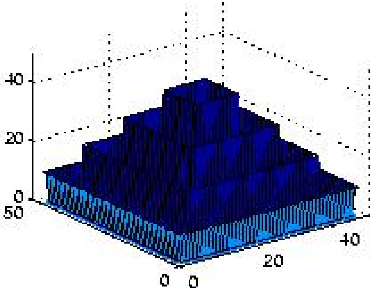
- Small set of posts $\Rightarrow$ complete matrix of posts
- Overdetermined linear system:
  - $z_{ij} = h_{ij}$ for known points,
  - $4z_{ij} = z_{i-1,j} + z_{i+1,j} + z_{i,j-1} + z_{i,j+1}$ for all nonborder points.
  - Emphasize accuracy or smoothness by weighting the two types of equations differently.
- Original goal: fill contours to a grid w/o showing terraces; competing methods have these problems:
  - Information does not flow across contours $\rightarrow$ slopes discontinuous
  - If rays are fired from the test point to the first known point, then method is not conformal etc.

ODETLAP Advantages

Handles

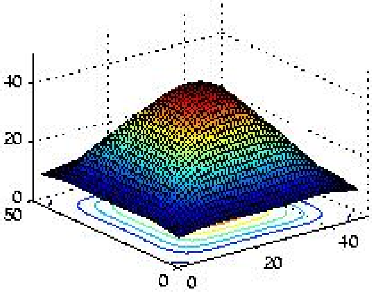
- missing–data holes.
- incomplete contours,
- complete contours,
- kidney–bean contours,
- isolated points,
- inconsistent data.

ODETLAP hard example



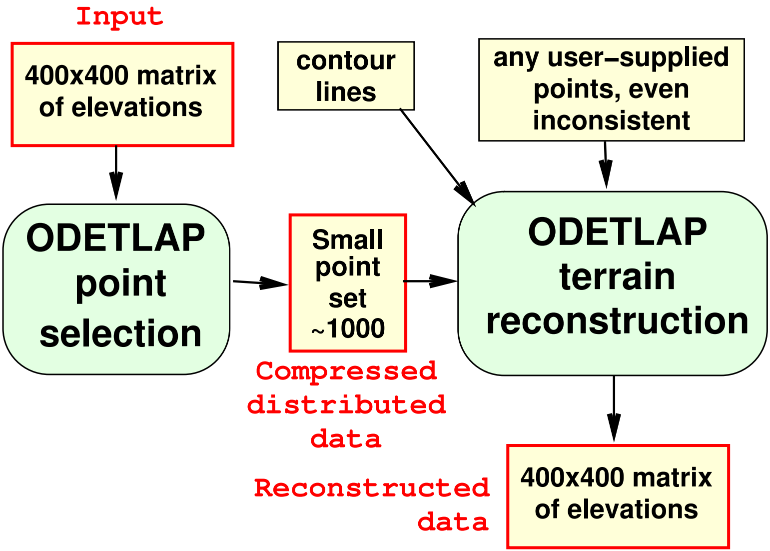
R=1, maxerr=5.5%, mean=0.6%

R=3, maxerr=12%, mean=2.7%

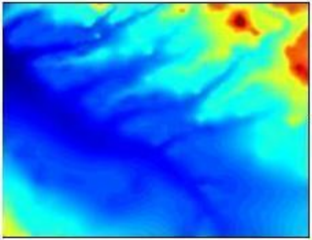


- input: contours with sharp corners
- output: smooth silhouette edges, inferred top

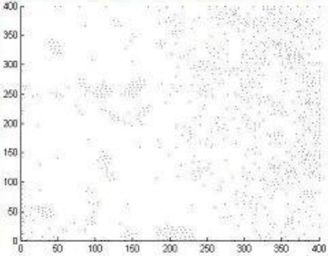
ODETLAP process



ODETLAP summary



Original Surface
(320 KB)



Compressed Surface
(4071 Bytes)

Average Absolute Error = 2.451
Maximum Absolute Error = 25.822

ODETLAP computational requirements

- Depends on number of known points, which
- Affects sparsity of system.
- 400×400 data grid with 1000 points $\Rightarrow$
- Solving 160000×170000 overdetermined sparse system is fast.
- Solving a 104976×104976 data grid: 58 GB main memory, 1.8 hours on workstation with four 2.4GHz processors and 60 GB of main memory running Ubuntu 10.04.2 and 64-bit Matlab R2009a.
- Approximate iterative solution sufficient.

ODETLAP Compressed data format

- 2 components:
  - Set of $\{(x, y)\}$ Locations of selected points, $+$
  - Ordered list of corresponding $\{z\}$.
- Compress locations with lossless bitmap algorithm, e.g., CCITT Group 4 fax.
- Compress heights with standard algorithm, e.g., 7zip.

5D data compression dataset

- Sensors, e.g., in World Ocean Atlas 2005, collecting multiple bands of environmental data –
- temperature, salinity, oxygen concentration,
- producing set of values over 5D grid (x, y, z, t, b).
- *Compress it!*
- little prior art.

Principles:

- Assume one band's large derivative at particular $(x, y, z, t) \Rightarrow$ likely for the other bands,
- Treat the data as one 5-D dataset, and
- Compress lossily since the data is imprecise.

Data compression technique

- extend ODETLAP to 3D, then 4D, 5D.
- *Major challenge:* Everything harder in higher dimensions.
- Use blend of 4D and 5D — separate 4D compressions but using same set of points.
- Compression ratios up to. 100:1 (mean error $< 1.5\%$).

	3D-ODETLAP			3D-SPIHT		
Variable	Mean Err(%)	Max Err(%)	Compr Ratio	Mean Err(%)	Max Err(%)	Compr Ratio
Salinity	0.0532	0.2174	77:1	0.0530	0.4946	11:1
Temperature	0.4993	2.0673	98:1	0.50	17.91	135:1
Dissolved O_2	0.9993	4.4145	100:1	1.002	24.9965	71:1
Apparent O_2 util.	0.9999	4.0170	85:1	0.9991	20.3609	81:1
Percent O_2 satur.	0.9985	4.5672	78:1	0.9969	20.3610	65:1
Phosphate	0.9993	4.5241	86:1	0.9978	15.6922	65:1
Nitrate	1.0242	4.6946	66:1	1.0006	18.5360	59:1
Silicate	0.9996	5.1437	91:1	1.0018	21.6457	81:1

ODETLAP's smaller compression error than SPIHT

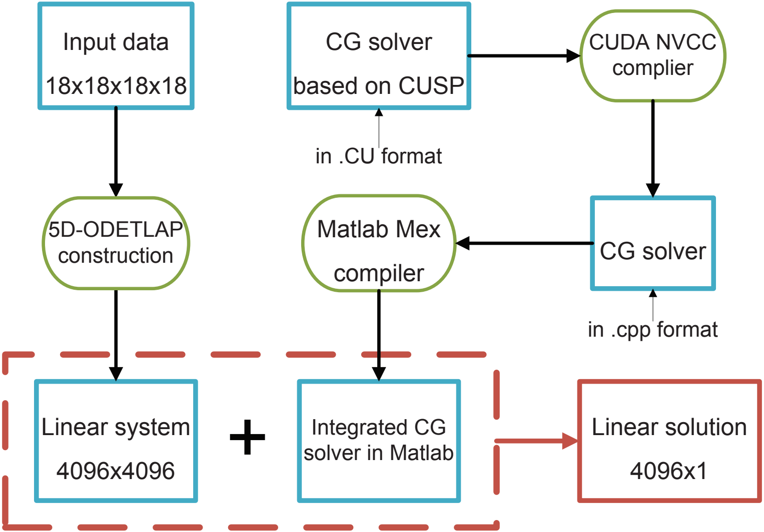
Parallel implementation

- Test data: 234256 $\times$ 234256 linear system from 5D-ODETLAP.
- Matlab CF direct solver: 49237 secs.
- Matlab iterative Congugate Gradient (CG) solver: 5495 secs.
- CUSP: open source sparse linear system library using CUDA.
- CUSP CG solver: 179 secs.
- All but 9 secs was data transfer.

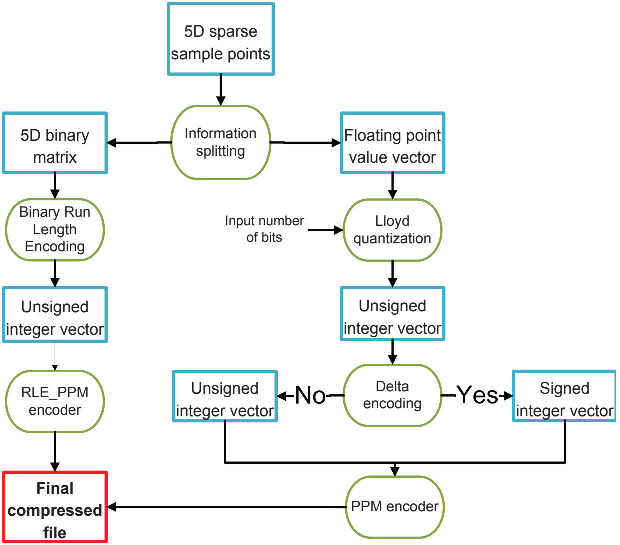
Optimization

- Use a Matlab executable (MEX).
- Write CUSP code in MEX style.
- Compile into C++. (CUDA can be compiled into C++).
- Then compile into MEX.
- Then call from Matlab.
- On 104976^2 system, Matlab direct time: 6488 secs, Matlab CG: 40 secs, CUSP CG: 5 secs.
- Also uses less memory (only 13GB in Matlab, 512MB in GPU).

System diagram



Compressed dataset coding



Comparison to JPEG, SPIHT

Dataset	% Fixed Mean Err	% Max Err, JPEG 2000	% Max Err, 3D-SPIHT	% Max Err, 5D-ODET-LAP	ODET-LAP JPEG Ratio	ODET-LAP SPIHT Ratio
woa05_1	1.42	44.83	66.46	10.41	8.16	2.40
woa05_2	1.48	49.33	59.18	9.35	9.56	3.61
woa05_3	1.47	65.56	80.23	8.94	4.24	1.13
woa05_4	1.56	67.56	74.14	10.81	8.57	2.41
woa09_1	1.46	48.13	68.18	9.02	8.18	2.41
woa09_2	1.49	51.21	62.15	8.77	9.80	3.75
woa09_3	1.54	75.00	79.35	11.13	4.24	1.14
woa09_4	1.58	65.55	71.50	11.58	8.58	2.42

Conclusion and Future

- Lots of main memory useful.
- Matlab useful.
- ODETLAP exploits multi-dimensional autocorrelation better.
- Already competitive for compressing 5D data.
- However time and space intensive.
- CUDA, CUSP make ODETLAP more practical.
- ODETLAP still has many unexplored optimizations.
- Investigate other parallel toolkits.
- To test on other data, e.g., CFD.