INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING
ICPP/2021/CHICAGO/USA
acm In-Cooperation
sighpc
AUGUST 9-12, 2021

GPU Accelerated SL0 for Multidimensional Signals

Gabriel G. Baravdish, Jonas Unger, Ehsan Miandji
Department of Science and Technology, Linköping University, Sweden

Problem: Find sparse representation

- We seek a sparse representation **s** for measurements **p**

$$\begin{aligned} \min \quad & ||\mathbf{s}||_0 \\ \text{s.t.} \quad & ||\mathbf{p} - \Phi\mathbf{D}\mathbf{s}||_2 \leq \epsilon \end{aligned}$$

- The Smoothed L0-algorithm (SL0) solves this problem for 1D-case.
  - *H. Mohimani, et al. "A Fast Approach for Over-complete Sparse Decomposition Based on Smoothed ℓ0-Norm" , IEEE Transactionson Signal Processing, 2009*

Our Approach: Multidimensional SL0 on GPU

- Extension to multidimensional signals – tensors.

$$\mathbf{A}^{(j)} = \mathbf{\Phi}^{(j)}\mathbf{D}^{(j)}$$

$$\boldsymbol{\mathcal{S}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \ldots \times_n \mathbf{A}^{(n)} = \boldsymbol{\mathcal{P}}$$

- Cholesky-factorization for computing pseudo-inverses.

- Parallel batchwise computation of the tensors.

Our Approach: Multidimensional SL0 on GPU

Algorithm 2 Parallel multidimensional SL0 algorithm

Require: Input tensors $\mathcal{P}_P$, input matrices $\mathrm{A}_P^{(1\ldots n)}$, limit parameter σ_{min}, decreasing factor σ_f, iteration L, step μ

Ensure: $\mathcal{S}_P$, P sparse solutions to

$\mathcal{S}_i \times_{1\ldots n} \mathrm{A}_i^{(1\ldots n)} = \mathcal{P}_i$

$\forall i = 1 \ldots P$ computed in parallel

$\mathcal{S}_i \leftarrow \mathcal{P}_i \times_{1\ldots n} \mathrm{A}_i^{(1\ldots n)\dagger}$ {initial solution}

$\sigma \leftarrow$ largest absolute element of $\mathcal{P}_P$

while $\sigma > \sigma_{min}$ **do**

  for $k = 1 \ldots L$ **do**

    $\Delta\mathcal{S}_i = -\frac{\mathcal{S}_i}{\sigma^2} \circ exp\left(-\frac{\mathcal{S}_i \circ \mathcal{S}_i}{2\sigma^2}\right)$

    $\hat{\mathcal{S}}_i \leftarrow \mu \cdot \Delta\mathcal{S}_i$ {Steepest ascent step}

    $\hat{\mathcal{S}}_i \leftarrow \hat{\mathcal{S}}_i - (\hat{\mathcal{S}}_i \times_{1\ldots n} \mathrm{A}_i^{(1\ldots n)} - \mathcal{P}_i) \times_{1\ldots n} \mathrm{A}_i^{(1\ldots n)\dagger}$

  end for

  $\mathcal{S}_i \leftarrow \hat{\mathcal{S}}_i$

  $\sigma = \sigma_f \cdot \sigma$

end while

Parallel Cholesky factorization

Batchwise multidimensional computation on GPU

Arithmetic Intensity – An analysis

- The arithmetic intensity:

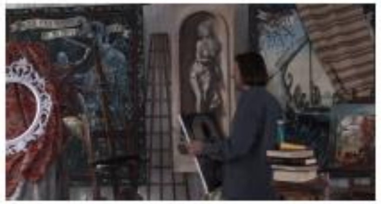
$$\frac{12S + nS\left(\sum_{i=1}^{n} M_i K_i^2\right) + P + nP\left(\sum_{j=1}^{n} K_j M_j^2\right) + S}{4\left(2\sum_{k=1}^{n} M_k K_k + P + 5S\right)}$$

- We get arithmetic lower bound at: $S \geq 8$ and $P \geq 8$ for tensors with at least 3 dimensions or higher.

Results – Light fields

(a) Bracelet.

(b) Tarot Cards.



(c) Painter.

(d) Train.

Results – 5D Light fields

Data set	# data points	Time nD SL0 GPU	Time nD SL0 CPU	Speedup	Sampled size	Sample ratio
Bracelet	26 240	54	1022	18.9x	4x4x3x4x4	16%
Bracelet	26 240	72	1195	16.6x	4x5x3x4x5	25%
Bracelet	26 240	88	1811	20.6x	4x4x3x7x7	49%
Bracelet	26 240	110	2123	19.3x	4x5x3x8x8	80%
2xBracelet	52 480	111	1931	17.4x	4x4x3x4x4	16%
2xBracelet	52 480	132	2219	16.8x	4x5x3x4x5	25%
2xBracelet	52 480	164	3486	21.3x	4x4x3x7x7	49%
2xBracelet	52 480	234	3788	16.2x	4x5x3x8x8	80%
3xBracelet	78 720	159	2935	18.5x	4x4x3x4x4	16%
3xBracelet	78 720	171	3351	19.6x	4x5x3x4x5	25%
3xBracelet	78 720	221	5148	23.9x	4x4x3x7x7	49%
3xBracelet	78 720	308	6344	20.6x	4x5x3x8x8	80%
4xBracelet	104 960	202	3911	19.4x	4x4x3x4x4	16%
4xBracelet	104 960	245	4429	18.1x	4x5x3x4x5	25%
4xBracelet	104 960	298	6823	22.9x	4x4x3x7x7	49%
4xBracelet	104 960	457	7311	16x	4x5x3x8x8	80%
Tarot Cards	42 025	117	1590	13.6x	4x4x3x4x4	16%
Tarot Cards	42 025	128	1782	13.9x	4x5x3x4x5	25%
Tarot Cards	42 025	151	2597	17.2x	4x4x3x7x7	49%
Tarot Cards	42 025	217	2812	12.9x	4x5x3x8x8	80%

Results – 6D Light field video

# data points	T_{avg} - nD SL0 GPU	T_{avg} - nD SL0 CPU	Speedup	Size	Signal length
89 380	92	1196	13x	5x5x3x4x4x2	4800
45 708	112	1325	11.8x	7x7x3x4x4x2	9408
27 588	116	1440	12.4x	9x9x3x4x4x2	15 552
18 513	103	1579	15.3x	11x11x3x4x4x2	23 232
13 272	104	1707	16.4x	13x13x3x4x4x2	32 448
10 001	89	1869	21x	15x15x3x4x4x2	43 200
5665	90	2168	24x	20x20x3x4x4x2	76 800
3608	95	2404	25x	25x25x3x4x4x2	120 000
2553	98	2664	27x	30x30x3x4x4x2	172 800

Results – 6D Light field video

# data points	T_{avg} - nD SL0 GPU	T_{avg} - nD SL0 CPU	Speedup	Sampled size	Sample ratio
22 345	41	651	15.8x	5x4x3x4x2x2	5%
22 345	61	753	12.4x	5x4x3x4x4x2	10%
22 345	91	1186	13.1x	5x5x3x4x4x4	25%
22 345	105	1571	14.9x	10x5x3x4x4x4	50%
22 345	132	2059	15.9x	10x10x3x3x4x4	75%

Conclusion

- Convergence analysis of the Multidimensional SL0-algorithm.
- Multidimensional SL0 on GPU for different types of discrete signals.
- Evaluations show that data are reconstructed within practical time.
- Analysis of the arithmetic intensity of the algorithm.

Thank you