



Multilevel Methods for Sparse Representation of Topographical Data

Prashant Shekhar¹, Abani Patra^{1*} and E.R. Stefanescu²

¹State University of New York at Buffalo, New York, U.S.A.

²Mercedes America, U.S.A.

pshekhar@buffalo.edu, abani@buffalo.edu, ramonastef@gmail.com

Abstract

With the onset of the data age, more and more of research is being carried out on construction of efficient algorithms for handling Big Data sets. The current work proposes a multiscale approach for intelligent prediction of missing data on a given DEM. The algorithm utilizes the Gaussian covariance kernel for studying the correlation between the available data points. Dimensionality reduction through parallel pivoted QR makes the approach scalable on large computing clusters. We have also studied the performance of the algorithm in terms of accuracy, scalability and convergence in order to validate the applicability of our approach. Use in generation of sparse representations for memory taxing datasets further establishes the efficiency of the multilevel analysis.

Keywords: Big Data, Topography and DEM, Multilevel Sparse Representation

1 Introduction

A Digital Elevation Model or DEM (Fig. 2) is a way of reconstructing a surface patch with the help of the elevation measurements that are recorded at different locations. These surface patches are represented as regularly spaced elements of definite measurement and height, inferred from the available data. The application of DEMs has been long known and their usage can be very frequently seen in the research areas such as hazard mapping, climate impact studies, geological modeling etc. In this regard, (Elkhrachy, 2015) presented a procedure to map flash flood hazard with the help of DEMs. (Stefanescu et al, 2010) on the other hand have analyzed the effects of uncertainty in the DEMs on the volcanic mass flow hazard analysis.

When we deal with large areas (Helm et al, 2014) such as involving continent wide exploration (frequently required for analysis of interior regions of Antarctica and Greenland), this data grows rapidly making it almost impossible for the standard algorithms to handle it efficiently. Also, the

* Corresponding Author

estimation of elevation at points for which the data is not available becomes a necessity, because given the size of the land area to be explored; the generated data becomes highly sparse. Therefore, the required analysis calls for a technique, which is not only able to efficiently handle extremely large data sets but also intelligently interpolate the elevation values for the unknown regions. Thus the problem to be tackled pertains to a supervised learning scenario, where we have some data points for which the elevation is known and we have to predict the elevation at other unknown regions in the neighborhood. This research also discusses the application of the presented approach for the generation of sparse representations of the given DEMs. The problem at hand may seem straightforward and one may be tempted to use simple predictive algorithms such as a well-tuned structure of neural networks. However, the complexity of the problem lies in the specific domain from which this data is collected. The geographical location greatly influences the nature of the terrain. General logic says that the elevation at a location will be greatly influenced by the elevation at the nearby points as compared to the elevation at distant points. (Bermanis et al, 2013) particularly makes use of this approach while formulating their correlation kernel. However, the standard predictive algorithms are unable to efficiently capture this property and thus the available structural manifold information is not exhaustively exploited. In order to tackle this problem, in this research we are proposing a parallel implementation of a multi-scale approach which is a variant of the algorithm introduced in (Bermanis et al, 2013), to handle the computationally expensive problem of DEM approximation. This algorithm proceeds through a graphical approach where all the known data points are assumed to be the vertices of the graph and the relationship between points is represented through a Gaussian covariance kernel. This approach proceeds in a multi-scale manner where at each scale, data is subsampled based on the conveyed information, and elevation measurement is predicted for the unknown locations. The implementation of the multi-scale approach for DEM generation is motivated by the fact that for large surface areas, available machine learning algorithms or other single step approaches won't be able to handle the computational complexity.

The contribution of the presented research can be summarized very precisely. Firstly, we have proposed a predictive learning algorithm that is capable of handling extremely large data sets by exploiting the multicore architectures of current day computing clusters. Secondly, instead of superficial learning like most of the abstract learning algorithms, our approach is aimed at using the information underlying the structure from which the data has been generated. Finally, the idea of using the multiscale procedure for generation of sparse representation of complex data-structures opens new vistas of analytical procedures for huge data sets.

The rest of the paper proceeds as follows. In the next section we have discussed the available literature for the topics relevant to this research. Section 3 presents a detailed explanation of the multiscale approach along with details regarding the parallel implementation through Elemental (Poulson et al, 2013), which is an optimized C++ library for linear algebra. Section 4 includes the performance analysis of our approach in terms of scalability, accuracy, sensitivity and convergence analysis. The subsequent section guides the application of the algorithm for generation of sparse representations of the DEM. Finally towards the end; we have also suggested the areas in which the current approach could be improved for even better handling of 'Big' data sets.

2 Background and Related work

With the extreme increase in the amount of available data, multilevel methods are getting more attention from the researchers than before. (Floater et al, 1996) introduced a multiscale process for sampling and interpolation. (Larson et al, 2007) presents a multiscale finite element solution procedure. (Abdulle et al, 2012) have presented and discussed the applicability of Heterogeneous

multiscale methods. Their scheme is based on the conventional multigrid methods where the missing data in the macroscopic scale is approximated with the help of the microscopic study. (Bermanis et al, 2013) have presented a multiscale approach, which is in a manner extended by our work through the parallel implementation of the algorithm by utilizing optimized C++ libraries.

The study of DEMs in hazard analysis is also one of the areas that is gaining momentum. (Stefanescu et al, 2012) studied the uncertainty associated with the hazards based on the variation of the input parameters of the DEM. (Schiefer et al, 2007) utilized DEMs for deduction of changes in terrain surfaces. Therefore, the availability and study of larger DEMs calls for better approaches that can effectively analyze and generate low rank approximations to the given data. In this way, the storage requirement would reduce, rendering the analysis easier. (Radenovic' et al, 2015) have presented a methodology where they have constructed a vector representation for a large-scale image. Their approach is based on the dimensionality reduction of multiple vocabularies for data labeling. (Boutsidis et al, 2015) have constructed a technique for dimensionality reduction that is based on optimal feature selection followed by clustering on them in order to identify the patterns in the data. The concept of low rank approximation is also crucial for an exhaustive analysis of large data sets. (Cohen et al, 2015) have put forth a technique for the low rank approximation of a Matrix and they have shown the applicability of the result in the solution procedure for a k-rank approximation problem within a bounded range of error.

3 Multi-scale Approach with distributed implementation

Our approach is based on graphical representation using the mutual distances between the data points as a parameter for the measure of their correlation. The algorithm begins with a graph $G = (V, E)$, where V is the set of vertices $V = \{1, 2, \dots, m\}$ and E represents a set of edges given by $E = \{e_{ij} \mid i, j \in V\}$. Here vertices represent the data points at which elevation measurements are available and weights for the edges are a proxy for the relationship between the points. Let $A = [a_{ij}]$ be the $m \times m$ adjacency matrix such that a_{ij} represents the weight of edge e_{ij} . If there is no edge between vertices i and j , then $a_{ij} = 0$. Thus, in our approach if the points are very far away, then they are assumed to be independent of each other and therefore the weight of the edge between them is negligible.

Many dimensionality reduction methods involve a spectral decomposition of large matrices whose dimensions are proportional to the size of the data. However, the computational cost of these approaches makes them unsuitable for sufficiently large data sets. The n observations f_1, f_2, \dots, f_n for the elevation measurements are considered to be the observed data points. When the covariance of the data points is unknown, an artificial function has to be chosen. The wide variety of literature available on the Gaussian kernel application makes it a simple choice:

$$g_{\epsilon}(x, x') = \exp\left(-\frac{\|x - x'\|^2}{\epsilon}\right), \quad (1)$$

where $\|\dots\|$ constitute a metric on the space. For our case this operator represents the Euclidean distance between the points. The corresponding covariance (affinities) or the edge weights are thus given as

$$[G] = g_{\epsilon}(x_i, x_j), i, j = 1, 2, \dots, n \quad (2)$$

This covariance kernel constitutes the backbone of our approach and at each scale it is modified to incorporate more information regarding the interaction between the data points.

Algorithm: Multi-scale approach

Input: A Data set $D = \{x_1, x_2, \dots, x_n\} \in \mathbb{R}^d, T > 0, P \geq 2, a \text{ new data point } x_* \in \mathbb{R}^d, a \text{ function } f = [f_1, f_2, f_3, \dots, f_n]^T \text{ to be extended and an error parameter } err \geq 0.$
Output: An approximation $F = [F_1, F_2, \dots, F_n]^T$ of f on D and its extension F_* to x_*

Initialization: Set the scale parameter $s = 0, F^{(s-1)} = 0 \in \mathbb{R}^n$ and $F^* = 0$

While $\|f - F^{(s-1)}\| > err$ **do**

1. Generate the Gaussian Kernel $G^{(s)}$ on D with $\epsilon_s = T/P^s$
2. Estimate the numerical rank $l^{(s)}$ of $G^{(s)}$
3. Generate a matrix A whose entries are i.i.d Gaussian random variable of zero mean and unit variance. Dimension of A would be $(l^{(s)} + 8, n)$
4. Apply, pivoted QR on $W \rightarrow WP_R = QR$ and split R and Q such that:

$$W = AG^{(s)}$$

$$R = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix}$$

$$Q = [Q_1 \mid Q_2]$$

5. Calculate the matrix S given as a product $S = Q_1 R_{11}$.
6. Columns of S constitute a subset $(l^{(s)})$ of columns of W . The corresponding columns of $G^{(s)}$ are stored in $B^{(s)}$
7. Calculate the pseudo-inverse of $B^{(s)}$
8. Calculate the coordinates vector of the orthogonal projection of $f^{(s)}$ on the range of $B^{(s)}$ in the basis of $B^{(s)}$'s columns, $c = (B^{(s)})^\dagger f$;
9. Calculate the orthogonal projection of f on the columns of $B^{(s)}$, $f^{(s)} = B^{(s)}c$
10. Form the matrix

$$G_*^{(s)} = [g_{\epsilon}(x_*, x_{s1}) \dots \dots g_{\epsilon}(x_*, x_{sl^{(s)}})]$$

11. Calculate the extension:

$$F_*^{(s)} = G_*^{(s)}c$$

12. $F^{(s)} = f^{(s)}$; $s = s+1$

end

Figure 1: Pseudo code for the multi-scale algorithm

Although, the Nyström method has been vastly used in the literature for out of sample extension, it has several disadvantages focusing mainly on high computational cost due to diagonalization of G . The G matrix also may be ill conditioned due to fast decay of its spectrum. One other significant problem with Nyström method is the lack of clarity regarding the procedure for choosing parameter ϵ . To overcome these limitations a multiscale approach is used, involving a sequence of Gaussian kernel matrices G_s ($s = 0, 1, \dots$), whose entries are $[G] = g_{\epsilon}(x_i, x_j)$. Here, ϵ_s is a positive monotonic decreasing function of s , which tends to zero as the scale parameter s tends to infinity (i.e. $\epsilon_s = T/P^s$, $s = 0, 1, \dots$). The parameter P is one of the major determinants for the performance of the approach and is studied in the results section. Criterion for choosing T is given by Eq. 3.

By the application of a randomized interpolative decomposition (ID) to G_s , a well-conditioned basis is identified for its numerical range. In each scale f is decomposed into a sum of its projections

on this basis and it is extended as $f_* = G_* G^{-1} f$. In addition, selection of the proper columns in G_s is equivalent to data sampling of the associated data points.

This method requires no grid. It automatically generates a sequence of adaptive grids according to the data distribution. It is based on the mutual distances between the data points and on a continuous extension of Gaussian functions. In addition, most of the costly computations are done just once during the process, independently of the number of the extended data points since they depend only on the data and on the given function.

3.1 Algorithmic Details

Figure 1, shows the pseudo-code for the multiscale algorithm proposed for DEM generation. Here, the condition of convergence depends on a user-defined parameter err , which depends on the type of the analysis. The type of norm to be used for convergence condition also depends on the user preference and the application. The algorithm begins by passing the input data as an argument. As the dimension for DEM data is 2 (X and Y coordinate) along with the elevation measurement at the points, therefore the value of d in the above algorithm is 2. The new data point x_* is the extension, out of sample point for which the value has to be predicted. Function f is the functional value (i.e. the elevation measurements) at the different input points.

The Algorithm begins with generation of the Gaussian covariance kernel as given in Eq. 1. The value of ϵ_s is assumed to be equal to T/P^s for scale s . Here T is computed as

$$T = \max\{\text{dist}(x_*, D), K(D)\} \quad (3)$$

where $K(D)$ is computed as

$$K(D) = 2(\text{maximum distance}/2)^2 \quad (4)$$

Here maximum distance refers to the distance between the most distant points in the input dataset D . The term $\text{dist}(x_*, D)$ represents the maximum distance between the input data points and the point for which the prediction has to be made. This particular choice of T ensures that even in the initial scale, x_* is significantly influenced by the input dataset D .

After obtaining the Gaussian Kernel for the current scale, we calculate its numerical rank with the help of singular values through SVD decomposition. The out of sample extension was carried out with the help of the Gaussian kernel vector computed in step 10 and extension carried out in step 11.

3.2 Parallel Implementation

As the Algorithm involves generation of the correlation matrix, therefore distributed memory architecture is more suited for parallel implementation due to the extreme memory requirement. For computationally scalable application of our approach we have used the parallel linear Algebra routines from the Elemental C++ Library. The major contribution of the library can be summarized as follows:

- Distributed Rank calculation in step 2.
- Distributed multiplication of matrices and vectors such as in steps 3, 5 etc.
- Distributed QR decomposition in step 4

- Parallel subsampling of columns in matrix B from the Gaussian kernel.
- Calculation of pseudo-inverse of B through parallel SVD calculation
- Parallel MATVEC kernel for steps 8 and 9

Therefore, with the help of the mentioned computing tasks, we parallelized the entire multiscale process. Elemental was chosen for parallelizing the entire computation process due to its good scalability and flexibility with respect to the matrix configurations, which eased up the transition of calculations from one step to the next.

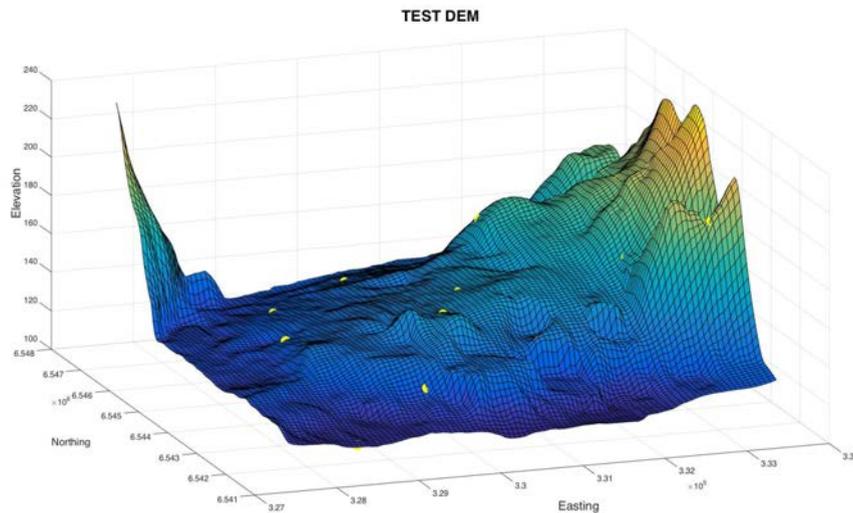


Figure 2: DEM used for testing and analysis with sparse mapping of representative points

4 Results and Analysis

In this section we will present the analysis for the performance and the accuracy of our algorithm. It is a well-known fact that the major problem with the distributed memory paradigm has always been the excessive communication between the processes that makes the application of distributed processing non-scalable. Implementing an algorithm, which utilizes the memory from different nodes in a computing cluster while minimizing the communication between processes has always been an area of research. Therefore, even with the availability of immense computational power, the true benefit of distributed computing cannot be harnessed without proper algorithmic implementation. In order to prove our point, we have presented different forms of analysis on our technique that proves the applicability of our approach. The DEM used for this research have been shown in Figure 2. It consists of 10000 data points and complete analysis of the algorithm has been carried out on sections of this DEM. The extreme variation in the elevation measurement over the space is the major contributing factor to the complexity of the problem at hand. The yellow dots in the figure represent sparse representations of the complete DEM and have been discussed in detail in section 5.

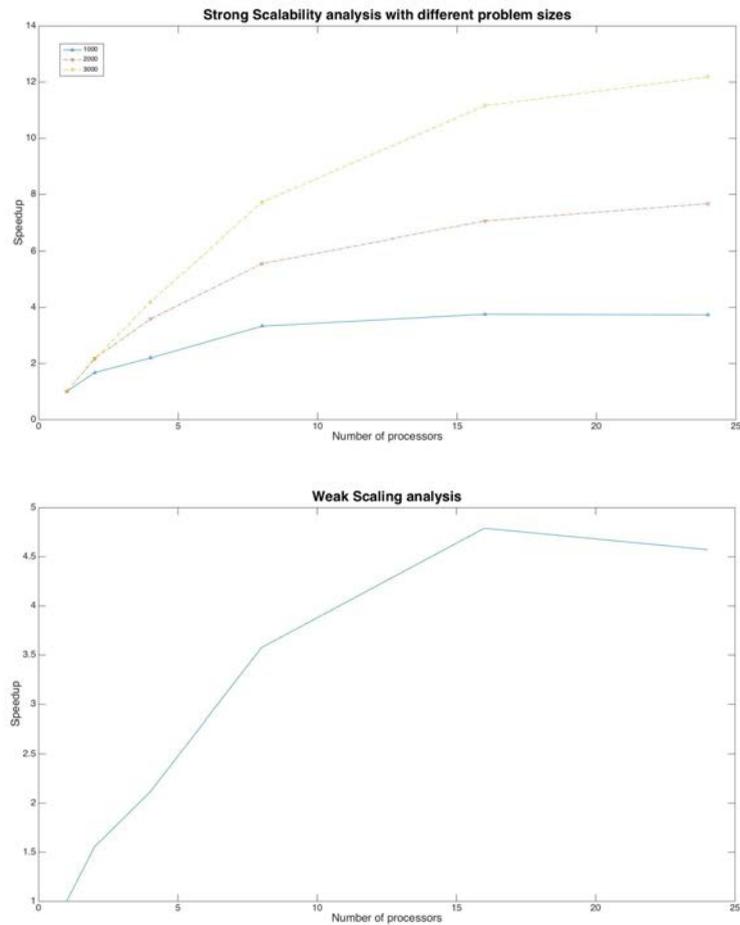


Figure 3: Strong and Weak Scaling analysis

4.1 Scalability

Strong and Weak scaling analysis has always been a major indicator for the performance of parallel implementation of an algorithm. Figure 3 shows the scaling analysis for our implementation of the introduced multi-scale approach. Strong scaling is precisely aimed to find the gain in the performance with increase in the number of processes while keeping the problem size constant. Problems size here refers to the number of data points in the DEM provided to the algorithm for prediction at unknown points. The first figure in Fig. 3 shows the speedup curve for three different problem sizes for strong scaling analysis. The fact that the scalability of our approach is improving with the problem size makes the algorithm valid and suitable for analysis of large data sets.

Weak Scaling on the other hand is very much dependent on the problem on hand. If computational requirement varies non-linearly with the problem size then getting a good weak scalability is really difficult. This analysis again refers to the case where the performance was studied with respect to out of sample extension for a point. The second figure in Fig.2 brings out the fact that the computational

complexity of the algorithm varies non-linearly with the problem size and therefore the computation time doesn't remain same as the problem size is varied in proportions to the number of processors.

4.2 Accuracy and stability

In order to study the accuracy for our approach, we randomly removed 5 distinct data points from the input DEM and predicted the elevation values at those five locations with the help of the remaining data. Since, a very common problem with many predictive algorithms is the loss in accuracy with the increase in the size of the input data set. This is because with increased size, the numerical and precision errors accumulate leading to worsening of the result. Therefore we have studied the accuracy for problem of different sizes similar to the ones studied for the weak scaling case. In order to quantify accuracy, we have calculated the RMS value of the error resulting from the five predictions done. The detailed results are shown in table 1

Problem Size	100	200	400	800	1600	3200
RMS	3.3166e-04	3.3166e-04	3.3166e-04	3.3166e-04	3.3166e-04	3.3166e-04

Table 1: Accuracy Analysis with varying problem sizes

Table 1 clearly reveals two major performance measures. Firstly, it is seen that as we increase the problem size, the RMS value for the error does not change. This result establishes the stability of our algorithm with respect to different problem sizes. Secondly, the RMS value is of the order of negative 4. Therefore, our approach is able to predict the elevation measurements at out of sample points with acceptable accuracy as well as stability.

4.3 Sensitivity and Convergence Analysis

As mentioned in the previous sections, the Algorithm consists of a parameter P . This parameter plays a significant role in the convergence of the algorithm as it determines the amount of information that would be embedded in the Gaussian Kernel at each scale. Table 2 shows the detailed analysis of results.

Parameter (P)	Problem Size					
	100	200	400	800	1600	3200
2	0.149	0.579	1.849	6.474	22.107	96.425
4	0.109	0.400	1.156	2.920	09.518	37.620
8	0.061	0.201	0.631	3.244	12.568	74.335
16	0.076	0.250	0.720	2.067	07.085	44.743

Table 2: Convergence times for variety of problem parameters

The main information conveyed by this analysis can be summarized on the basis of the performance of the algorithm when the algorithmic parameter is varied with the problem size. For smaller problem sizes, as the parameter P is increased, more and more information gets embedded on the starting scales and therefore the algorithm converges quickly, reducing the computational time. However, for the highest scale (16), the computational time actually increases. This can be explained

on the basis of the fact that with such a large factor, the optimal amount of information needed for convergence is exceeded and thus based on communications going on between the processes for the computation of rank and for computation of pseudo-inverse, it converges at a later stage. Now, when we consider larger problem sizes, the size of the Gaussian covariance kernel grows and therefore larger value of parameter actually favors the situation. Therefore, for the problem sizes of 800, 1600 and 3200; the convergence time is actually smaller for the parametric value of 16 as compared to the time for 8. However, here the timings are actually higher for value of 8 as compared to 4. The reason for this is similar to the one mentioned for the peaks at 16 for smaller problem sizes.

5 Application to hazard mapping and analysis

As mentioned in the introduction section, DEM analysis finds application in many research areas. Hazard analysis is one of the areas where the output in the form of hazard maps is highly sensitive to the elevation measurements in the DEM. However, as the coverage area increases, the amount of data to be tackled increases rapidly making the computation highly expensive. In order to deal with the issue, our approach could be used to generate sparse representations of the DEM so that the storage requirement is less and further analysis could be carried without much computational power. The basic idea revolves around using a sampling procedure (such as LHS) to generate representative points in the domain for the DEM. Then, using the multiscale approach to find the elevation at these points. This new DEM with elevation measurements at only critical locations is referred to as sparse representation of the DEM. For testing this, we generated 16 points using LHS over the domain of the DEM and then we have predicted the values at those locations. The yellow dots in Fig. 2 show the elevations at those 16 random points. Therefore, with the help of these sparse representations, the original DEM could be produced any time as per the requirement. A major benefit of this application is the reduction in the memory requirement for the data. Additionally, it could also be used as a representation of the original DEM in further analysis, which reduces the high computational power requirement.

6 Conclusions and Future Works

In this research we proposed a multiscale approach to efficiently handle large volumes of data and make useful predictions for the missing values. The major focus of the approach lies in exploiting the multicore architecture of modern computers in a distributed memory paradigm for handling the issue of high memory requirement as well as computing capabilities. The algorithm was implemented in Elemental, which is an optimized C++ library for linear algebra routines. In order to demonstrate the applicability of our approach, we have shown the performance of our algorithm on a variety of test DEMs where we predicted elevations at out of sample locations. The scalability and accuracy results have justified the applicability of the approach on large datasets.

Although the multiscale approach in itself is highly promising, several areas of improvement can be suggested. Even after large-scale parallel implementation, the application of SVD for computation of the rank of Gaussian kernel as well as for the computation of pseudo-inverse of the sampled dataset, takes a lot of computational performance. Therefore, alternate procedures for the mentioned requirements can be implemented. Also, although Elemental is found to be performing quite well regarding scalability and precision issues, several other linear algebra routines could also be tested for the optimal implementation of the algorithm. As a closing remark, it could be easily concluded that the increase in the available computational power alone will not be able to match up with the recent data

outburst. Therefore, development of efficient algorithms for analytics as well as handling of large datasets should be a major focus of the current research in this area.

References

- [1] Elkhachy, I. Flash Flood Hazard Mapping Using Satellite Images and GIS Tools: A case study of Najran City, Kingdom of Saudi Arabia (KSA). *The Egyptian Journal of Remote Sensing and Space Science*, 18(2), 2015, 261-278.
- [2] Stefanescu, E. R., Bursik, M., Dalbey, K., Jones, M., Patra, A. K., & Pitman, E. B. DEM uncertainty and hazard analysis using a geophysical flow model. In *Proc. of the 2010 Int. Congress on Environmental Modelling and Software, Ottawa, Canada*, 2010, 5-8.
- [3] Helm, V., Humbert, A., & Miller, H. Elevation and elevation change of Greenland and Antarctica derived from CryoSat-2. *The Cryosphere*, 8(4), 2014, 1539-1559.
- [4] Bermanis, A., Averbuch, A., & Coifman, R. R. Multiscale data sampling and function extension. *Applied and Computational Harmonic Analysis*, 34(1), 2013, 15-29.
- [5] Floater, M. S., & Iske, A. Multistep scattered data interpolation using compactly supported radial basis functions. *Journal of Computational and Applied Mathematics*, 73(1), 1996, 65-78.
- [6] Larson, M. G., & Målqvist, A. Adaptive variational multiscale methods based on a posteriori error estimation: energy norm estimates for elliptic problems. *Computer methods in applied mechanics and engineering*, 196(21), 2007, 2313-2324.
- [7] Abdulle, A., Weinan, E., Engquist, B., & Vanden-Eijnden, E. The heterogeneous multiscale method. *Acta Numerica*, 21, 2012, 1-87.
- [8] Stefanescu, E. R., Bursik, M., Cordoba, G., Dalbey, K., Jones, M. D., Patra, A. K., ... & Sheridan, M. F. Digital elevation model uncertainty and hazard analysis using a geophysical flow model. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, The Royal Society, Vol. 468, No. 2142, 2012, 1543-1563.
- [9] Schiefer, E., & Gilbert, R. Reconstructing morphometric change in a proglacial landscape using historical aerial photography and automated DEM generation. *Geomorphology*, 88(1), 2007, 167-178.
- [10] Radenović, F., Jégou, H., & Chum, O. Multiple measurements and joint dimensionality reduction for large scale image search with short vectors. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, ACM, 2015, 587-590.
- [11] Boutsidis, C., Zouzias, A., Mahoney, M. W., & Drineas, P. Randomized dimensionality reduction for k-means clustering. *Information Theory, IEEE Transactions on*, 61(2), 2015, 1045-1062.
- [12] Cohen, M. B., Elder, S., Musco, C., Musco, C., & Persu, M. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, ACM, 2015, 163-172.
- [13] Poulson, J., Marker, B., Van de Geijn, R. A., Hammond, J. R., & Romero, N. A. Elemental: A new framework for distributed memory dense matrix computations. *ACM Transactions on Mathematical Software (TOMS)*, 39(2), 2013, 13.