Dykstra's projection algorithm in multiresolution analysis

Lutz Künneke, Jan Lebert

CUDA Lab Course 2014, Institut für Numerische und Angewandte Mathematik

Introduction

In modern image analysis multiresolution methods have gained increasing interest in the last years. Their application lies in the statistical inversion of

$$I = A * x + \epsilon$$

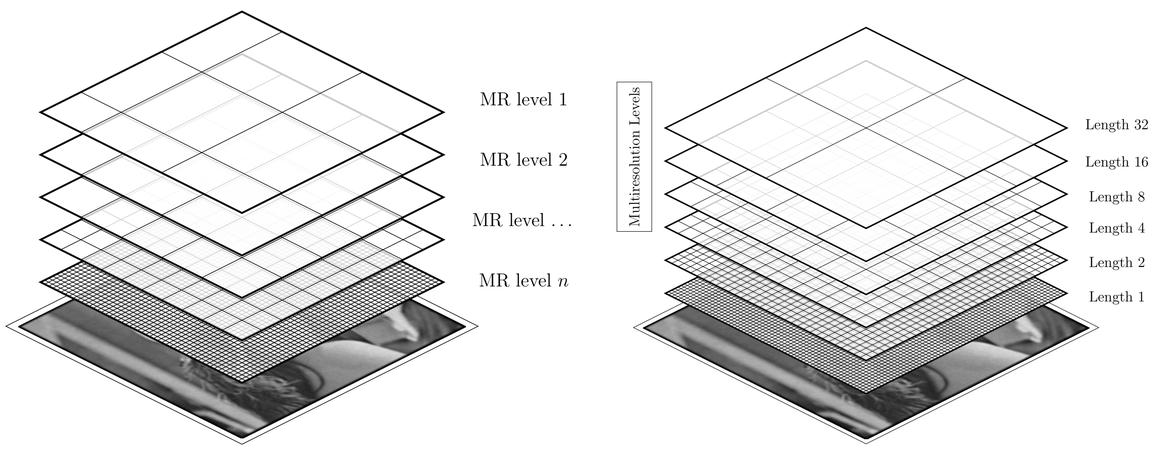
where $A : \mathbb{R}^n \to \mathbb{R}^m$ is linear operator and ϵ is a vector represententing noise.
A multiresolution estimator performs hypothesis tests on subsets of the estimated noise $\hat{\epsilon}$ in order to ensure normality on all scales. The resulting estimate is the underlying true signal of the noisy measurement in each subset with high probability.

Implementation

The hypothesis tests on subsets are calculated using Dykstra's Algorithm. Dykstra's Algorithm provides the projection on the intersection of convex sets $s \in \Omega$ given the projection p_s on each set. Initialize $q_0^s = 0 \; \forall s \in \Omega$, then one iteration of Dykstra's Algorithm consists of $\forall s \in \Omega$

$$x_{r+1} = p_s\left(x_r - q_r^s\right)$$
$$q_{r+1}^s = x_{r+1} - x_r$$

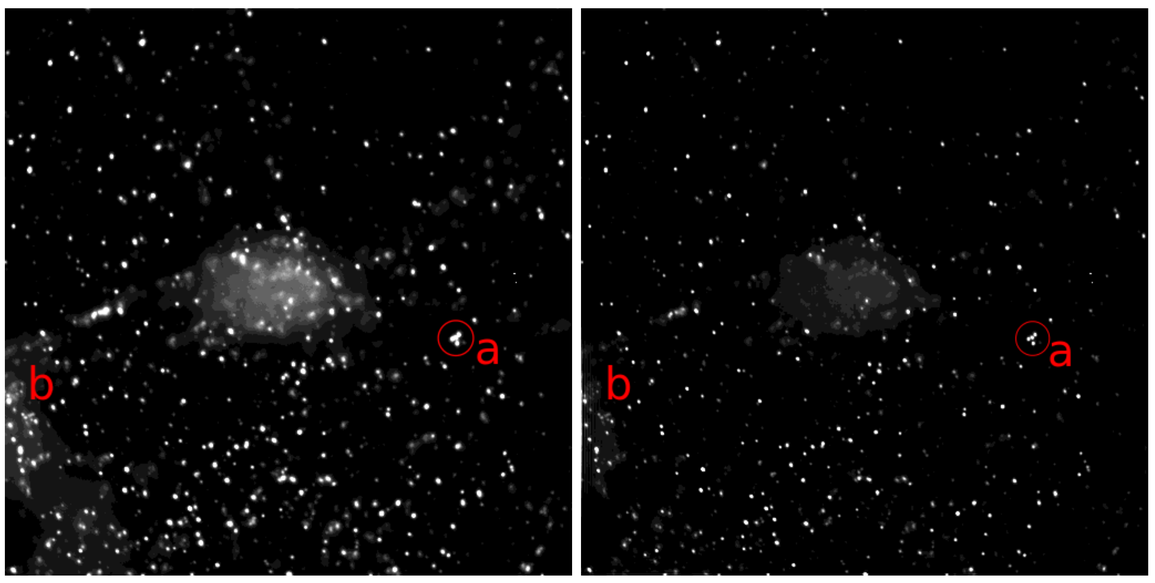
the algorithm is shown to converge towards the projection on the intersection of the $s \in \Omega$ given some x_0.
The order in which the projections are performed is crucial for the convergence of the algorithm, therefore only non-overlapping subsequent projections can be calculated in parallel. Due to this restriction we've developed an approximative version which uses subsets which are suitable for parallel computation with CUDA.
The performance of the exact method is limited by the transfer rate of the PCIe bus.



subset selection of exact **(left)** and approximative **(right)** method

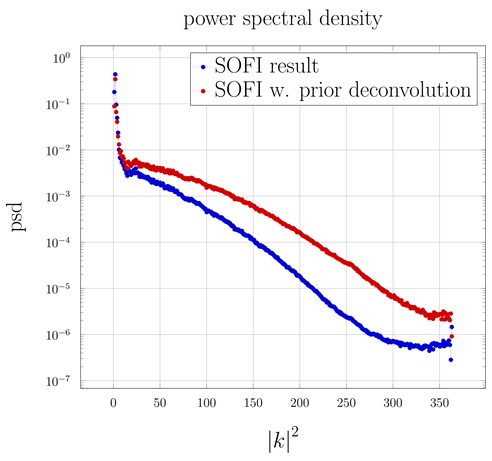
SOFI

real data



SOFI result without **(left)** and with **(right)** prior deconvolution

In order to apply the SOFI algorithm [1] blinking nanobeads are attached to some structure of interest. The nanobeads show uncorrelated blinking. By calculating the spatial and temporal correlation between pixels one can improve the overall resolution.
We analysed a movie with 3500 frames and a duration of 10ms per frame using the approximative method. To produce the right hand image we applied a statistical multiresolution estimator on each image before application of SOFI. The resulting image shows smaller point spread functions **(a)**, but our algorithm introduces some artefacts at the boundaries **(b)**.
The power spectral density **(right)** indicates a favourable resolution in our result.



Lena

simulated data

The Dykstra Algorithm was applied within the estimator

$$\hat{x} = \text{argmin}_{x,e} G(e) + R(x) \text{ s.t. } I = A * x + e$$

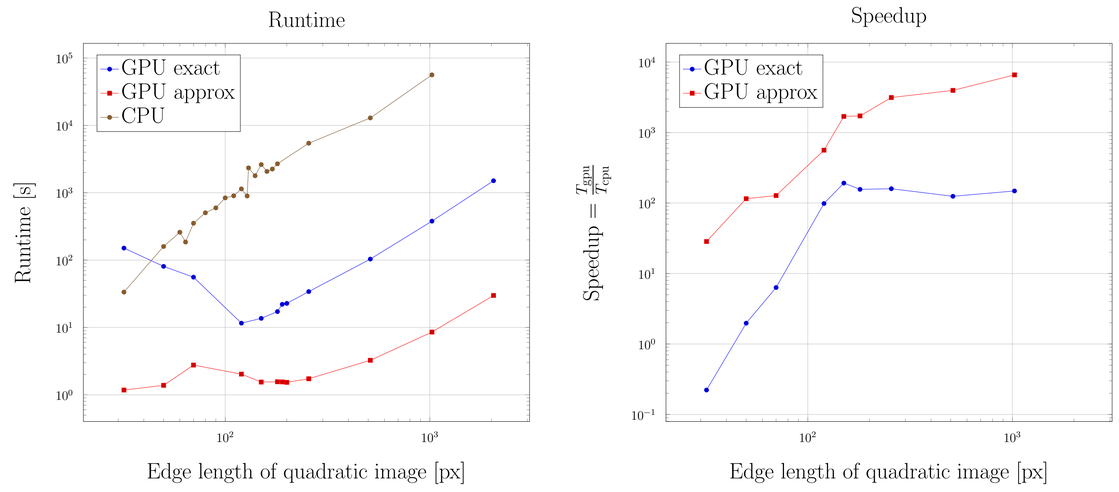
with noise $e \sim N\left(0, \sigma^2\right)$, blur $A : \mathbb{R}^n \to \mathbb{R}^m$ and signal x. R is a regularization and G the characteristic function of the multiresolution.

original image **(left)**, test image with simulated blur and gaussian noise **(right)**

Statistical Multiresolution Estimate using exact Dykstra **(left)** and approximative Dykstra **(right)**

The images show results of the estimator on simulated data. The estimator reduces the blur and noise to a certain extent. There is no visible difference between exact and approximative projection algorithm.

Speedup



The left image compares runtime for three implementations of the projection algorithm, the right images shows the speedup of the CUDA implementations compared to the OpenMP parallelized CPU implementation.
Notice that the approximative Dykstra uses a different algorithm than the CPU implementation, the exact method achieves a speedup of about 100.

References

[1] T. Dertinger, R. Colyer, G. Iyer, S. Weiss, and J. Enderlein. Fast, background-free, 3D super-resolution optical fluctuation imaging (SOFI). *Proceedings of the National Academy of Sciences*, 106(52), 2009.

[2] Klaus Frick, Philipp Marnitz, and Axel Munk. Statistical multiresolution dantzig estimation in imaging: Fundamental concepts and algorithmic framework. *Electronic Journal of Statistics*, 6:231–268, 2012.

This work was supported by the NVIDIA Teaching Center Göttingen, the source code is available as part of SciPAL as *step-35*. We thank Johannes Hagemann and Stephan Kramer for their help.