Streaming Multiframe Deconvolution of Atmospherically Distorted Images on GPUs

Matthias A Lee & Tamás Budavári
The Johns Hopkins University

Motivation

Atmospheric distortion is an issue in various fields relying on long-distance imaging, especially in astronomy. Earth-based telescopes face the challenge of peering through earth's ever-changing and blur-inducing atmosphere. This can be especially detrimental to observations of small and faint objects. The point spread function (PSF) defining this blur is unknown, hard to predict and varies wildly making recovery more difficult.

Modern telescopes produce high volumes of extremely large images, upwards of 100PB in 10 years, yielding an even more compute intensive and time consuming reconstruction. The current "State-of-the-Art" images are commonly Co-Add images made by overlaying multiple faint observations to create one higher quality image. (See Figure 2) These Co-Add images are very noisy and missing detail. More advanced reconstruction is computationally complex and therefore slow and laborious. We need GPU-accelerated, statistically sound and extensible tools to keep up with, explore, and deblur these images in real time.

Problem

The general problem can be defined as shown in equation 1, where \( y \) is the observed image and the Point Spread Function defining the blur at time \( t \), \( x \) is the underlying "true" image which we assume to be constant across all observed frames. The observed image \( y \) consists of the "true" image \( x \), which has been convolved with an unknown blur \( f \). Both \( x \) and \( f \) are unknown but can be constrained to being at least non-negative. To recover the "true" image we need to estimate the PSF to be able to deconvolve it from the observed image.

In order to get a good estimate of the PSF, we extract information from each of a series of single frames and iteratively apply it to our "true" image estimate.

Methods & Implementation

Our tool currently features two separate iterative methods, a Gaussian-based Multiframe Blind Deconvolution (MFBD) as described by [1] and [2] and a Poisson-based Richardson-Lucy (RL) deconvolution.[3]

Both approaches use an update formula, which is iteratively applied to every observed image \( y_t \) in an attempt to estimate it's PSF. This is done by holding the last iteration's estimate of \( x \) constant and applying the update formula repeatedly while solving for \( f \). Once we have estimated the PSF we take a swing in reverse, this time holding the PSF constant and updating the \( x \) image. Currently we have both the Gaussian-based[2] and Richardson-Lucy-based[3] update formulas implemented, Equation 2 and Equation 3 respectively. We repeat this process for every image in the series of images, therefore incrementally improving our estimate of what the underlying "true" \( x \) image is.

At the moment the majority of the GPU-acceleration has been done by implementation of naive element-wise kernels. This provides the majority of the speedup available from the GPU, but also keeps the path of execution simple and uncluttered. Once the code has reached a more mature state, we will consider optimizing beyond this.

Our GPU accelerated tool allows us to easily experiment with different algorithms and parameters. CPU-based processing of a set of 20 images can take upwards of an 60 minutes depending on the settings and algorithms used. The same task can now be accomplished in under 3 minutes. That is an over 20x speed up. Since parameter adjustments are vital to suit diverse sets of images, it is essential to be able to experiment with image results in near real time.

In the future we aim to add support for super resolution, which will likely yield a better reconstruction. While implementing wavelet filtering option, we found no viable python-based GPU-accelerated wavelet library. Currently, even when running in GPU mode, we compute the wavelet transform on the CPU which causes a major slowdown. To remedy this issue we are working on a python and pyCUDA wrapper for gpuOpen. Our tool is also Open Source under GPLv3 and available at: http://github.com/madmaze/pymfbd

Results

References


Contact us:

Matthias A Lee
Matthias Lee: gtc@matthiaslee.com

Tamás Budavári
Department of Astronomy and Physics
The Johns Hopkins University
3400 N Charles Street
Baltimore, MD 21218
