Ultrafast focus detection using multi-scale histologic features

Maksim Levental
University of Chicago

Ryan Chard
Argonne National Laboratory

Gregg A. Wildenberg
University of Chicago

ABSTRACT

We present a fast out-of-focus detection algorithm for electron microscopy images collected serially. Such images are collected for the purposes of post-processing tasks such as montaging, alignment, and image segmentation. Such an algorithm is necessitated by recent increases in collection rates owing to advances in microscopy technology. Our technique adapts classical computer vision and is based on detecting various fine-grained histologic features. We further exploit the inherent parallelism in the technique by employing GPGPU primitives in order to accelerate characterization. Tests are performed that demonstrate faster than real time detection of out-of-focus conditions. <We also deploy to funcX something something>. We discuss extensions that enable scaling out to support multi-beam microscopes and integration with existing focus systems for purposes of implementing auto-focus.

1 INTRODUCTION

Advancements in the automation of serial scanning electron microscopy (SEM) impose a regime where thousands, if not tens of thousands, of images can now be automatically collected by researchers. TODO: <bio use cases> This puts greater demand on conventional auto-focus algorithms for ensuring each image is in focus, as an alternative to the user manually evaluating each image by eye. Without such algorithms, critical bottlenecks are created where the user is forced to reacquire individual, deficient (out-of-focus), images and manually reinsert them into the sequence of thousands of other images already acquired. This is an onerous task which requires taking into account alignment and boundary overlap. Furthermore, failure to quickly identify and reacquire deficient images negatively impacts the accuracy of downstream, post-processing; for example 2D montaging, 3D alignment, or automatic segmentation pipelines. While many microscopes have builtin auto-focus algorithms, these often fail to achieve acceptable accuracy due to intrinsic mediating factors (e.g. stage drift) and extrinsic mediating factors (e.g. sample artifacts, non-uniformity in the sample).

Auto-focus technology is a critical component of many imaging systems; from consumer cameras (for purposes of convenience) to industrial inspection tools to scientific instrumentation. Such technology is typically either active or passive; active methods exploit some auxiliary device or mechanism to measure the distance of the optics from the scene, while passive methods analyze the definition of sharpness of an image by virtue of some proxy measure. Here we focus on passive methods, as we explicitly aim to augment existing microscopy equipment without the need for costly and complex retrofitting.

Passive proxies for the degree-of-focus (DOF) include the energy of the Laplacian, discrete cosine transform, or weighted histogram of an image; for effecting a high DOF a search can be performed. When used as a component of an auto-focus system (as opposed to OOF detection system) all such passive methods are unsuitable for the purpose of real-time (or even near-real-time) characterization of DOF due to their long scanning times (multiple images need to be collected at potentially different depths). As our method currently aims only to detect OOF events we do not consider or implement any focus search techniques (but do describe plans for such future work).

To overcome these challenges, thereby ensuring that images are faithfully acquired, we propose a method to evaluate image definition based multi-scale histologic feature detection (MHD). By multi-scale histologic feature detection we mean the resolving and characterization of histological structure at multiple length scales; for our particular use-case this means structures ranging from cell walls to whole organelles. The key insight being that the ability to resolve structure across the range of feature scales is highly correlated with a high-definition, i.e. in-focus, image.

Due to limitations of the extensibility of commercial microscopy equipment, we do not aim here to directly implement auto-focusing. Rather than focusing the microscope, as auto-focusing algorithms would, our algorithm operates downstream of collection and reports out-of-focus (OOF) events to the user. This enables the user to intervene and initiate reacquisition protocols (on the microscope) before unknowingly proceeding with collecting the next series of images or proceeding with downstream image processing and analysis. This human-in-the-loop remediation protocol already saves the user much wasted collection time and tedium in triaging defective collection runs.

This rest of this article is organized as follows: section 3 describes our focus detection method in the abstract, section 4 discusses optimizations made in order to achieve real-time performance with our method, section 5 reports results of evaluating our method on sequences of images collected at varying focus depths, section 6 discusses related work and how our work is distinct therefrom, and finally section 7 concludes with a discussion of future research.

2 GREGG SELLS SCIENCE!

TYPE HERE

3 MULTI-SCALE HISTOLOGIC FEATURE DETECTION

We base our multi-scale histologic feature detection on classic on scale-space representations of signals. We give a brief overview (a more comprehensive discussion is available []) and describe our adaptation.

The fundamental principle of scale-space feature detection is that natural images possess structure at multiple scales and that features at a particular scale can be characterized in isolation of features at other scales. Typically characterization is effected by convolution with a filter that satisfies the constraints of non-enhancement of local extrema, scale invariance and rotational invariance (along with some others []). One such filter [] is the symmetric, mean zero,

2D, Gaussian filter

$$G(x, y, \sigma) \coloneqq \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Thus, define the scale-space representation $L(x, y, t)$ of an image $I(x, y)$ to be the convolution of that image with a mean zero Gaussian filter:

$$L(x, y, t) \coloneqq G(x, y, t) * I(x, y)$$

where t is the standard deviation of the Gaussian and determines the *scale* of $L(x, y, t)$. $L(x, y, t)$ has the interpretation that image one-dimensional structures of scale smaller than $\sqrt{t^2} = t$ have been removed due to blurring. This is due to the fact that the variance of the Gaussian filter is t^2 and features of this scale are therefore "beneath the noise floor" of the filter or, in effect, suppressed by filtering procedure. A corollary is that features with length scale t will have maximal response being filtered by $G(x, y, t)$; for $t' < t$ smaller length scale features will dominate the response and for $t'' > t$, as already mentioned, the response will have been suppressed. Hence, at various scales t we can use linear and non-linear combinations of space derivatives ∂_x, ∂_y and derivatives in the scale ∂_t to construct scale-invariant feature detectors; such feature detectors detect features such as corners, edges, and ridges. For example, the zeros in scale of the scale normalized Laplacian

$$\partial_t \nabla^2 L \coloneqq \partial_t \left(\partial_x^2 + \partial_y^2\right) L = 0 \tag{1}$$

correspond to uniform region (otherwise known as blobs) detectors.

4 IMPLEMENTATION

We therefore propose to use a feature detector as a proxy for DOF, reasoning that quantity of features detected is positively correlated with DOF (see figure 1). To this end, we develop a feature detector based on eqn. 1 but optimized for latency (rather than for accuracy). In order to verify our hypothesis we compare the number of histologic features detected as a function of absolute deviation from in-focus ($|f - f'|$ where f' is the correct focal depth) for a series of sections with known focal depth (see figure 2a). We observe a very strong log-linear relationship (see figure 2b). Fitting such a log-linear relationship produces a line with $r = -0.9754$, confirming our hypothesis that quantity of histologic features detected is a good proxy measure for DOF.

We now discuss our implementation[1] of the feature detector, with particular attention paid to optimizations in consideration of inference latency. Eqn. 1 permits a discretization[2] called *Difference of Gaussians* (DoG) (see [])

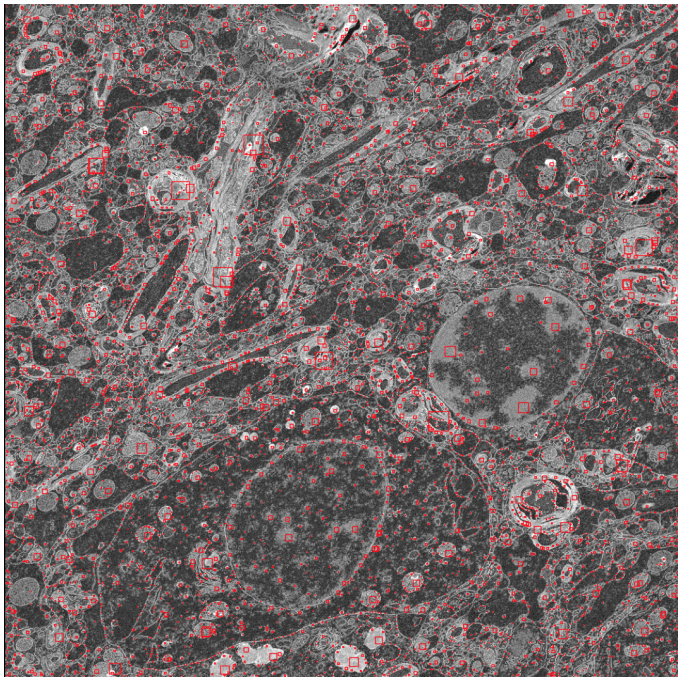
$$t^2 \nabla^2 L \approx t \times (L(x, y, t + \delta t) - L(x, y, t))$$

Therefore, define

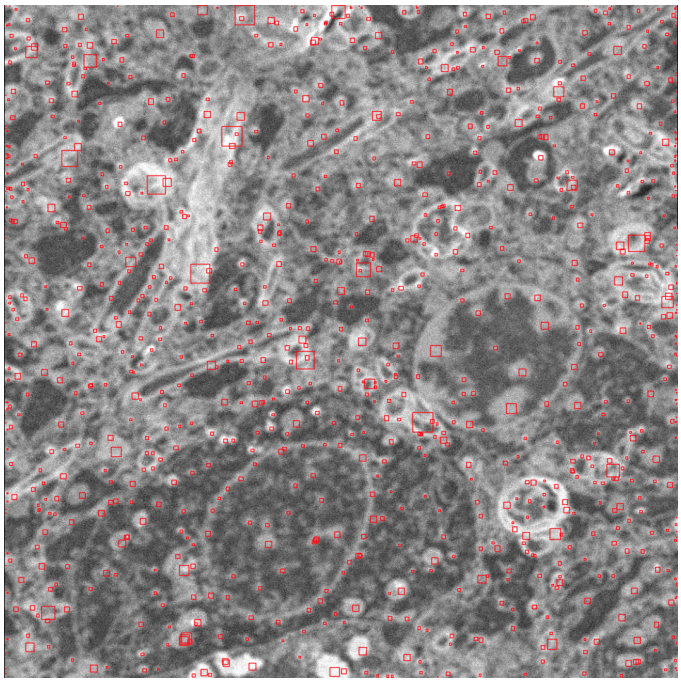
- `n_bin`, which determines the quantity of scales determined
- `min_t`, the minimum scale detected
- `max_t`, the maximum scale detected
- $\delta t \coloneqq$ `(max_t - min_t)/n_bin`
- $t_i \coloneqq$ `min_t` $+ (i - 1) \times \delta t$, the discrete scales detected

[1] https://github.com/makslevental/cuda_blob/

[2] By virtue of G being the Green's function of the heat equation $t\nabla^2 G = \partial_t G$

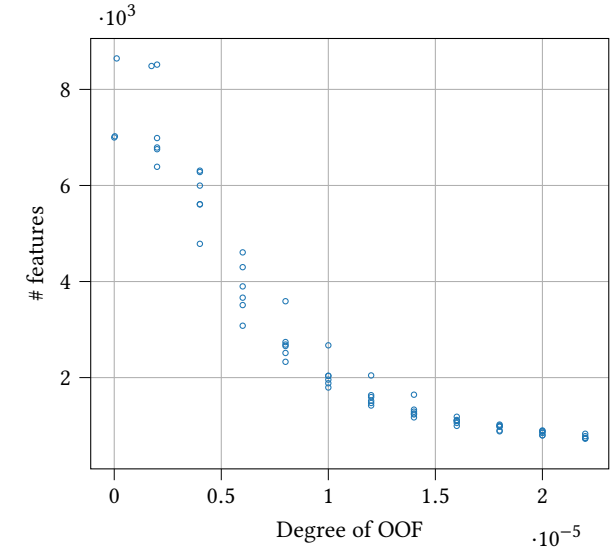


(a) Histologic features of an in-focus section.

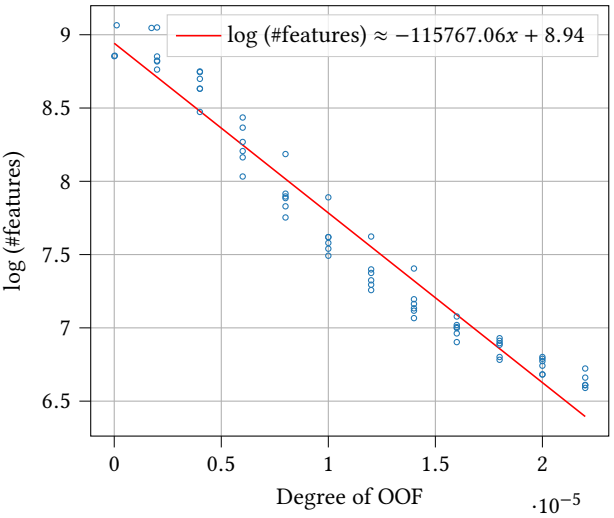


(b) Histologic features of an out-of-focus section.

Figure 1: Comparison of sections with histologic feature recognition as a function of focal depth.



(a) Number of histologic features as a function of absolute deviation from focused ($|f - f'|$ where f' is the correct focal depth).



(b) Log plot and line fit with $r = -0.9754$.

Figure 2: Comparison of histologic feature recognition as a function of focal depth.

and finally the discretized DoG

$$\mathrm{DoG}(x, y, i) \coloneqq t_i \times (L(x, y, t_{i+1}) - L(x, y, t_i)) \tag{2}$$

This produces a stack $\{\mathrm{DoG}(x, y, i)\}$, in the scale dimension, of filtered and scaled images (called a Gaussian pyramid []).

Computing the maxima of $\mathrm{DoG}(x, y, i)$ in the scale dimension (equivalently zeros of eqn. 1) necessarily entails computing local[3] maxima at every scale. We make the heuristic assumption that at each pixel there is a single unique, maximal, response at some scale; this response corresponds to the scale at which the variance of the Gaussian filter G most closely corresponds to the scale of the feature. We therefore search for local maxima in x, y but *global* maxima in the scale dimension

$$\{(\hat{x}_j, \hat{y}_j, \hat{i}_j)\} \coloneqq \operatorname*{argmaxlocal}_{x,y} \operatorname*{argmax}_{i} \mathrm{DoG}(x, y, i) \tag{3}$$

where the subscript j indexes over the features detected.

It is readily apparent that our feature detector is parallelizable; for each scale i we can compute $L(x, y, t_i)$ independently. Naturally, this suggests a GPGPU implementation []. Therefore we develop our histologic feature detector to be maximally parallelizable in order to take advantage of the SIMT [] execution model of the conventional GPU. A further parallelization is possible for the argmax operation since the maximum is computed independently across pixels. In order to make full use of this optimization we first perform the inner argmax in eqn. 3 and then the outer. The inner argmax is "free", as the argmax primitive is implemented in exactly this way on GPUs, and the outer argmaxlocal is implemented using a MaxPool2D(n, n) (with $n = 3$). Employing MaxPool2D in this way has the added benefit of effectively performing non-maximum suppression, since it effectively rejects candidate maxima within a 3×3 neighborhood of a true maximum.

Typically one would compute $L(x, y, t_i)$ in the naive way (by convolving G and I) but prior work has shown [**?**] that performing the convolution in the Fourier domain is much more efficient; namely

$$L(x, y, t_i) = \mathcal{F}^{-1}\left\{\mathcal{F}\{G(x, y, t_i)\} \cdot \mathcal{F}\{I(x, y)\}\right\}$$

This approach has the added benefit that we can make use of highly optimized FFT routines made available by GPU manufacturers. In particular we can take advantage of *distributed* FFT routines; by partition the set of Gaussian filters $\{G(x, y, t_i)\}$ across m nodes we can, in principle reap, a linear increase in efficiency of the FFT. That is to say we actually carry out

$$\{L(x, y, t_i) \mid i \in I_j\} = \left\{\mathcal{F}^{-1}\{\mathcal{F}\{G(x, y, t_i)\} \cdot \mathcal{F}\{I(x, y)\}\} \mid i \in I_j\right\}$$

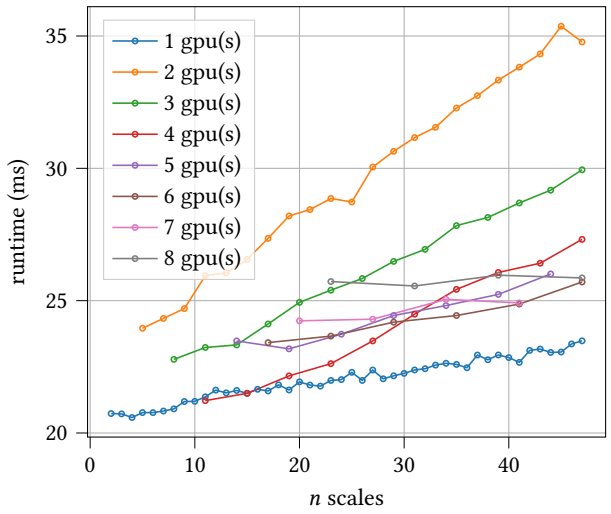
where for $j = 1, \ldots, m$ the set I_j indexes the scales allocated to a node j. In practice FFT time (both forward and inverse) is strongly dominated by I/O but this partitioning is still crucial in instances where our images are too large to fit in the RAM available on a single GPU (see section 5).

One remaining detail is histogram normalization of the images. Due to the dynamic range (i.e. variable bit depth) of the SEM we need to normalize the histogram of pixel values; we do this by saturating .175% of the darkest pixels, saturating .175% of the lightest pixels, and mapping the entire range to $[0, 1]$. We find this gives us consistently robust results with respect to noise and anomalous features. This histogram normalization is also parallelized using GPU primitives.

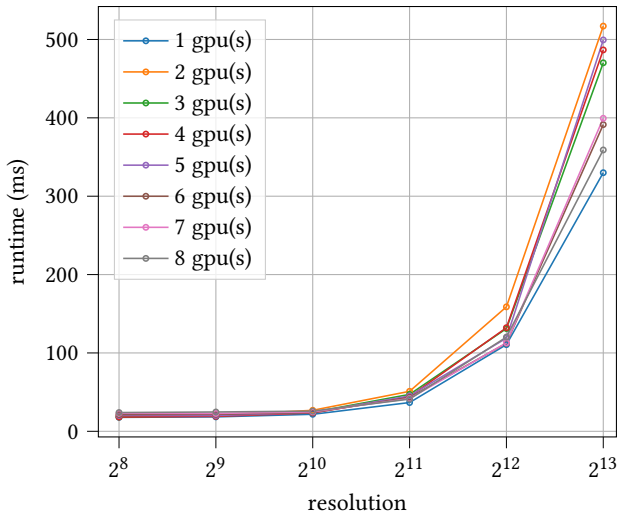
5 EVALUATION

Brains were prepared in the same manner and as previously described []. Briefly, an anesthetized animal was first transcardially perfused with 10ml 0.1 M Sodium Cacodylate (cacodylate) buffer, pH 7.4 (Electron microscopy sciences (EMS) followed by 20 ml of fixative containing 2% paraformaldehyde (EMS), 2.5% glutaraldehyde (EMS) in 0.1 M Sodium Cacodylate (cacodylate) buffer, pH 7.4 (EMS). The brain was removed and placed in fixative for at least 24 hours at 4C. A series of 300 um vibratome sections were prepared and

[3] In a small pixel neighborhood in both space and scale dimensions.



(a) Median runtime as a function of number of feature scales at resolution $= 1024 \times 1024$**.**



(b) Median runtime as a function of section resolution with 16 feature scales.

Figure 3: Scaling experiments for runtime with respect to number of GPUs, resolution, and number of feature scales.

put into fixative for 24 hours at 4C. The primary visual cortex (V1) was identified using areal landmarks and reference atlases. A small piece (2 x 2 mm) containing V1 was cut out and prepared for EM by staining sequentially with 2% osmium tetroxide (EMS) in cacodylate buffer, 2.5% potassium ferrocyanide (Sigma-Aldrich), thiocarbohydrazide, unbuffered 2% osmium tetroxide, 1% uranyl acetate, and 0.66% Aspartic acid buffered Lead (II) Nitrate with extensive rinses between each step with the exception of potassium ferrocyanide. The tissue was then dehydrated in ethanol and propylene oxide and infiltrated with 812 Epon resin (EMS, Mixture: 49% Embed 812, 28% DDSA, 21% NMA, and 2.0% DMP 30). The resin-infiltrated tissue was cured at 60oC for 3 days. Using a commercial ultramicrotome

Table 1: Test platform

CPU	Dual AMD Rome 7742 @ 2.25GHz
GPU	8x NVIDIA A100-40GB
HD	4x 3.84 U.2 NVMe SSD
RAM	1TB
Software	CuPy-8.3.0, CUDA-11.0, NVIDIA-450.51.05

(Powertome, RMC), the cured block was trimmed to a 1.0mm x 1.5 mm rectangle and 2,000, 40nm thick sections were collected on polyimide tape (Kapton) using an automated tape collecting device (ATUM, RMC) and assembled on silicon wafers as previously described (ref??). Images at different focal distances were acquired using backscattered electron detection with a Gemini 300 scanning electron microscope (Carl Zeiss), equipped with ATLAS software for automated imaging. Dwell times for all datasets were 1.0 microsecond.

We perform runtime experiments across a range of parameters of interest (section resolution, number of feature scales). Our test platform is a NVIDIA DGX A100 (see table 1). Experiments consist of computing the DOF of a sample section for a given configuration. All experiments are repeated k times (with $k = 21$) and all metrics reported are in fact median statistics[4].

For a section resolution of 1024×1024 pixels we achieve approximately a 50Hz runtime in the single GPU configuration; this is XXXX faster than real time. We observe that, as expected, runtime grows linearly with the number of feature scales and quadratically with the resolution of the section; naturally, this is owing to the parallel architecture of the GPU. The principle defect of our technique is that it is highly dependent on the available RAM of the GPU it is deployed to. In practice, most GPUs available at the edge, i.e. proximal to microscopy instruments, will have insufficient ram to accommodate large section resolutions and wide feature scale ranges. In fact, even the 40GB of the DGX's A100 is exhausted at resolutions above 4096×4096 for more than ~ 20 feature scales.

Therefore, we further investigate parallelizing MHD across multiple GPUs. Our implementation parallelizes MHD in a straightforward fashion: we partition the set of filters across the GPUs, perform the "lighter" FFT-IFFT pair on each constituent GPU, and then gather the results to the root GPU (arbitrarily chosen). Note that for such multi-GPU configurations the range of feature scales was chosen to be a multiple of the number of GPUs (hence the proportionally increasing sparsity of data in figure 3a). We observe that, as one would expect, runtime is inversely proportional to number of GPUs (see figure 3b) but that for instances where a single GPU configuration is sufficient it is also optimal. More precise timing reveals that parallelization across multiple GPUs incurs high copy costs during the gather phase of parallel MHD (see figure 4). Note that this latency persists even after taking advantage of CUDA IPC [1]. In effect, this is a fairly obvious demonstration of Amdahl's law. Therefore, we emphasize that parallelization across multiple GPUs should only be considered in instances where full resolution section images are of the utmost necessity[5].

[4]We discard the first execution since it is an outlier due to various initializations (e.g. pinning CUDA memory).

[5]For example, when feature scale range are very wide, with detection at the lower end of the scale being critical. In all other cases downsampling by bilinear interpolation in

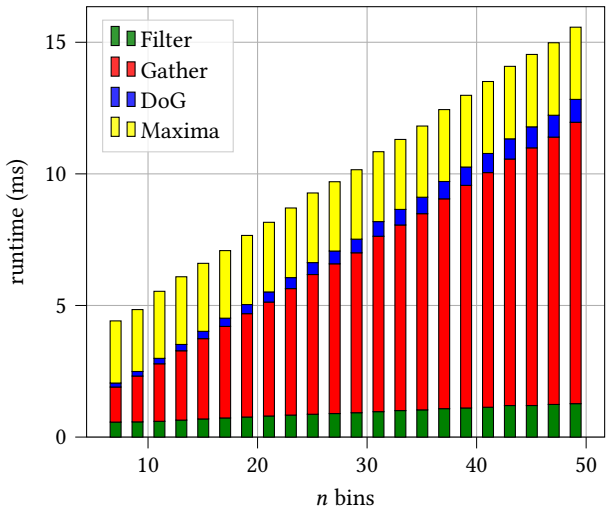


Figure 4: Breakdown of runtime into the four major phases for two GPUs across feature scales at resolution = 1024×1024**.**

6 RELATED WORK

7 CONCLUSION

ACKNOWLEDGMENTS

This work was supported by the U.S. Department of Energy, Office of Science, under contract DE-AC02-06CH11357.

REFERENCES

[1] S. Potluri, H. Wang, D. Bureddy, A. K. Singh, C. Rosales, and D. K. Panda. 2012. Optimizing MPI Communication on Multi-GPU Systems Using CUDA Inter-Process Communication. In *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops PhD Forum*. 1848–1857. https://doi.org/10.1109/IPDPSW.2012.228

order to satisfy GPU RAM constraints yields a more than reasonable tradeoff between accuracy and latency.