GPU-based quasi-real-time Track Recognition in Imaging Devices: from raw Data to Particle Tracks

Cristiano Bozza[1], *Umut Kose*[2], *Chiara De Sio*[1], *Simona Maria Stellacci*[1]

[1]Department of Physics University of Salerno, Via Giovanni Paolo II 132, 84084 Fisciano, Italy
[2]CERN, 1211 Geneve 23, Switzerland

DOI: http://dx.doi.org/10.3204/DESY-PROC-2014-05/2

Nuclear emulsions as tracking devices have been used by recent experiments thanks to fast automatic microscopes for emulsion readout. Automatic systems are evolving towards GPU-based solutions. Real-time imaging is needed to drive the motion of the microscope axes and 3D track recognition occurs quasi-online in local GPU clusters. The algorithms implemented in the Quick Scanning System are sketched. Most of them are very general and might turn out useful for other detectors.

1 Nuclear emulsions as tracking detectors

Nuclear emulsions have a long history in high-energy physics and recently experienced revived interest in the CHORUS[1], DONUT, PEANUT[2] and OPERA[3] experiments. They provide the best spatial resolution currently available, of the order of 0.1 μm. On the other hand, they have no time resolution, recording all charged tracks since the time of production until photographic development. In common setups, a detector is made up of one or more stacks of films, placed approximately orthogonally to the direction of the incoming particles. Each film has two layers of emulsion coating a transparent plastic base (Fig. 1). Typical dimensions are 50 μm for the thickness of emulsion layers and 200 μm for the base. A nuclear emulsion contains AgBr crystals in a gel matrix. Charged particles sensitise the crystals by ionisation, producing a *latent image*. Development of the film produces metallic Ag grains in place of the latent image, resulting in aligned sequences of grains (*microtracks*), typically 0.3$\sim$1 μm in diameter (Fig. 2). In an optical transmission microscope, grains appear as dark spots on light background. In white light, the average alignment residuals of grains with respect to the straight line fit is of the order of 50 nm. The depth of

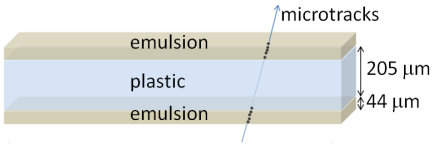

Figure 1: Nuclear emulsion film.

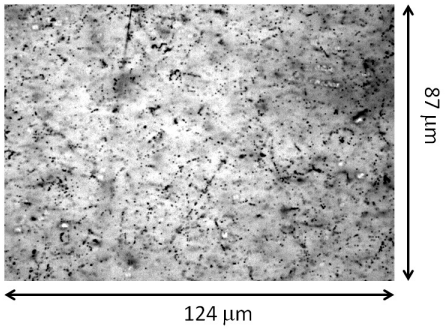

Figure 2: Nuclear emulsion image. High-energy tracks are orthogonal to plane.

field of the optics is usually below 5 μm. The full thickness of the emulsion layer is spanned by moving the optical axis, thus producing optical tomographies with a set of equally spaced planes (15∼31 usually). The image of a particle track in a focal plane is a single grain, not necessarily present in each plane (ionisation is a stochastic process). Because the chemical process of development is just faster for sensitised grains, but it occurs in general for every crystal in a random fashion, many grains appear without having been touched by any particle. Such so-called *fog* grains are overwhelming in number: as a reference, the ratio of *fog* grains to microtrack grains ranges from 10^3 through 10^5. Only 3D alignment is able to single out the few microtrack grains, but also many fake microtracks due to random alignments survive in a single film. Stacking several films allows using coincidence methods to improve background rejection.

It is worth noticing that this resembles the situation of an electronic detector in which background hits due to noise or pile-up overwhelm track hits. Normally, electronic detectors use a time trigger to reduce combinatorial complexity, but in emulsion this is not possible. It is reasonable to think that a tracking algorithm working in emulsion finds an even easier environment if fed with data from other detectors, such as cloud chambers or planes of silicon pads.

2 Data from nuclear emulsion

A nuclear emulsion film has typically a rectangular shape, spanning several tens of cm in both directions. The whole surface is scanned by acquiring optical tomographies in several fields of view with a repetitive motion. An electronic shutter ensures that images are only negligibly affected by motion blur. In the ESS (*European Scanning System* [4], [5], [6], [7], [8]), developed in the first decade of the 21st century, the XY microscope axes hold steady while the Z axis (optical axis) moves and images are grabbed on the fly and sent to vision processors. Its evolution, named QSS (*Quick Scanning System*), moves the X axis continuously. Hence, each *view* (an optical tomography) of ESS is a cuboid, whereas those of QSS are skewed prisms. Images are acquired by fast monochromatic sensors (CMOS) mounted on the optical axis, capable of 200∼400 frames per second, each frame being 1∼4 MPixel (or more for future sensors) using 8 bits/pixel. The resulting data rate ranges from 0.5 GB/s to 2 GB/s. The linear size of the image stored in a pixel, with common optics, is of the order of 0.3 μm. The full size of the field of view is 400×320 μm^2 for ESS, 770×550 μm^2 for QSS.

2D image processing used to be shared in the case of ESS by an industrial vision processor (Matrox Odyssey) based on an FPGA and by the host workstation CPU. It consists of several substeps:

1. grey level histogram computation and stretching – used to compensate for varying light yield of the lamp;
2. FIR (*Finite Impulse Response*) processing with a 5×5 kernel and comparison of filter output to a threshold – selects dark pixels on light background, producing a binary image;
3. reduction of sequences of pixels on horizontal scan lines to segments;
4. reduction of segments in contact on consecutive scan lines to clusters of dark pixels – produces grains to be used for tracking.

In the ESS, steps 1 and 2 are performed on the FPGA-based device. Steps 3 and 4 are executed by the host workstation CPU. For the same task, QSS uses a GPU board hosted in

the workstation that controls the microscope: common choices are double-GPU boards such as NVidia GeForce GTX 590 or GTX 690. A single board can do everything without intervention of the host workstation CPU. The first 3 steps are quite natural applications for GPU's. One single GPU can be up to 7 times faster than the Odyssey, reducing the price by an order of magnitude. Steps 4 and 5 require special attention in code development, because they are reduction operations with an *a priori* unknown number of output objects. In step 4 each thread takes care of a single scan line of the image. In step 5 a recursive implementation has been chosen: at iteration n, the segments on scan line $i \times 2^n$ are compared to those on line $i \times 2^n - 1$ and the related dark clusters are merged together. Indeed, steps 4 and 5 are the slowest, and they define the total processing speed, which is 400 MB/s for GTX 590. The system is scalable, and more GPU boards or more powerful ones can be added if needed. The output of this step is, for each view, a set of clusters of dark pixels on light background, each one being encoded with its X, Y, Z coordinates and size in pixels. Automatic readout uses the distribution of clusters to continuously adjust the Z axis drive of the tomography. 60∼124 MB of image data are encoded to 8∼16 MB cluster block, ready for local storage or to be transmitted over the network for further processing. In the latter case, a RAM disk area is used as a buffer to distribute data to a cluster of GPU's. Tracking servers and the workload manager provide a command-line interface for administration and have an embedded lightweight Web server, originally developed for the SySal project ([7]), that provides a graphical interface for human access and is the backbone for an HTTP-based network protocol for control messages needed for workload assignment.

Particle tracks can cross the boundary of a view, and tracking must be able to do the same. The alignment tolerance to recognise real tracks and discard background cannot exceed 0.5 μm and is usually smaller. The motion control electronics is capable of position feedback, but triaxial vibrations due to combined motion arise, well beyond 1 μm in a single image. Corrections to raw data are needed before they can be used to recognise microtracks. Some such corrections, sketched in Fig. 3, depend on optical aberrations and are systematic effects that can be computed off-line from small data-sets and then applied on each incoming cluster block (view):

1. Spherical aberrations: XY and Z curvature;
2. trapezium distortions: dependence of magnification factor on X and Y;
3. magnification-vs.-Z dependence;
4. camera tilt: rotation in the XY plane;
5. optical axis slant: X-Z and Y-Z couplings due to inclined optical axis.

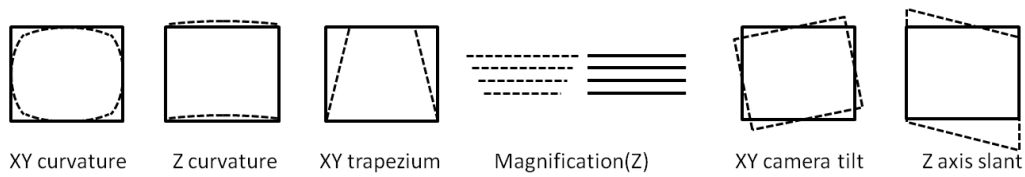

Figure 3: Optical distortion of emulsion image.

Most of the computing power needed for data correction is taken by other effects that are purely stochastic: vibrations due to motion, increased by mechanical wear and aging are unpredictable and spoil the alignment of images within the same view and between consecutive views. Because the depth-of-field usually exceeds the Z pitch between two images in the same sequence taken while the Z axis moves, a sizable fraction of the dark clusters in one image will appear also in the next. Pattern matching works out the relative shift between the images, usually within 1 μm. This procedure requires scanning a square region of possible values of the plane shift vector. Combinatorial complexity is reduced by arranging clusters in a 2D grid of cells and considering only pair matching within each cell. The shift vector that maximizes the number of combinations is the best approximation of the misalignment between the images. Likewise, despite position feedback of all axes, a whole tomography is misaligned with respect to the next. Film scanning is performed so as to leave 30∼40 μm overlap between adjacent views. The dark clusters found in the overlap region are used to perform 3D pattern matching, in the same way as for 2D pattern matching. The standard deviation of the distribution of residuals is of the order of 150 nm (Fig. 4) for X and Y, and 2.6 μm for Z.

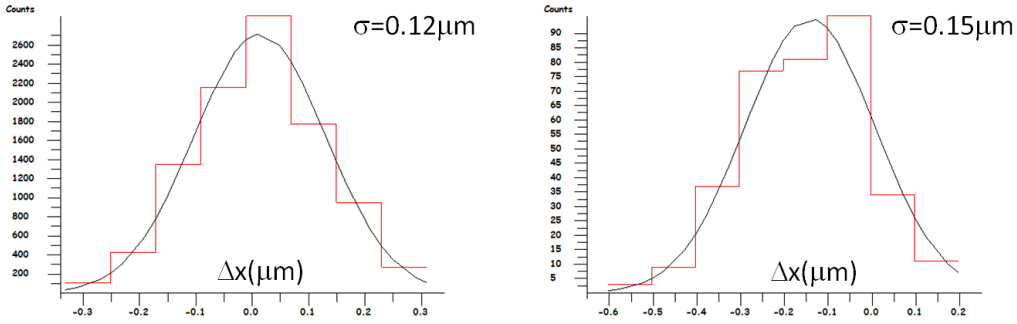

Figure 4: Left: precision of image-to-image alignment in the same tomographic sequence. Right: precision of relative alignment of two tomographic sequences.

3 Track reconstruction

Reconstructing 3D tracks from a set of primitive objects such as emulsion grain images or electronic detector hits is a common task in high-energy physics. The method depicted in the following would work in all cases of straight tracks, i.e. absent or weak magnetic field and scattering effects negligible with respect to the granularity of the detector. Because ionisation is a stochastic process, the algorithm does not require that a microtrack has a dark cluster in every image; furthermore, the notion of sampling planes is deliberately discarded to keep the algorithm general. It just needs to work on 3D pointlike objects, each having a weight, which corresponds to the number of pixels of the dark cluster in this case (e.g. it may be the deposited energy in a silicon counter). Each pair of dark clusters defines a *seed*, i.e. a straight line in 3D space; other clusters are expected to be aligned with it within proper tolerance (Fig. 5-left). In thin emulsion layers, microtracks are formed with 6∼40 clusters, depending on the local sensitivity, on statistical fluctuations and on track angle (the wider the inclination with respect to the

thickness dimension, the longer the path length). Furthermore, high-energy particles ionise less than slow ones. Such reasons suggest not to filter out too many possible pairs of clusters as track seeds, considering all those that are within geometrical acceptance. Physics or experimental conditions constrain the angular acceptance to a region of interest. Optimum thread allocation to computing cores demands that constraints be enforced constructively instead of generating all seeds and discarding invalid ones. Dark clusters are arranged in a 2D grid of prisms, each one spanning the whole emulsion thickness (Fig. 5-right). The angular region of interest is scanned in angular steps. At each step the prisms are generated skewed with the central slope of the step. This ensures that seeds that are very far from the acceptance window will not even be built and followed.

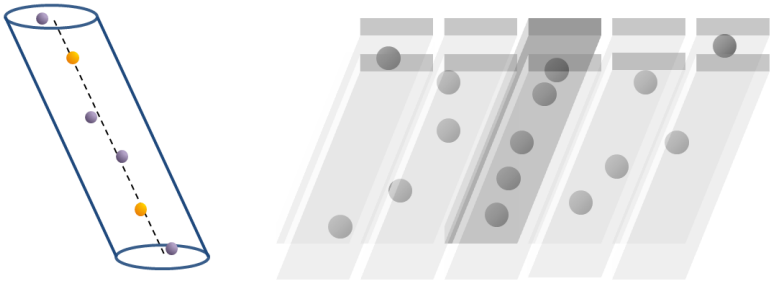

Figure 5: Left: microtrack seeds and grains. Right: 2D grid of prisms to build seeds. A prism containing a track is shown darker.

With common operational parameters, the number of useful combinations ranges within 10^7 and 10^9 per tomographic sequence, depending on the amount of random alignments and *fog*. For each seed, one thread scans the containing prism to find aligned clusters and build the microtrack. This procedure naturally produces clones of the same track, normally in the range 2∼4:1. They are discarded by comparing the geometrical parameters of all neighbor microtracks, neighborhood being defined by means of a 2D spatial grid.

4 Performances and outlook

Performances in terms of processing time vs. grain or microtrack density has been estimated using several NVidia GPU's. The results are shown in Figures 6, 7, 8 and 9.

Denoting the grain density (grains per tomography) with N, the number of seed combinations is of order N^2, and the search for clusters to attach to the seed is of order N^3. Results show that processing steps of high computational complexity are not overwhelming for typical operational conditions.

While one would expect the processing time to scale inversely with the number of available cores, more recent GPU's perform proportionally worse than older ones. The reason for that is to be sought mostly in branch divergence, which affects more the multiprocessors with larger number of cores. In some cases the divergence is due to threads exiting while others keep running. This could be eliminated, but it's not clear whether the additional complications of code would pay off. In other cases the divergence is due to the specific coding style used in

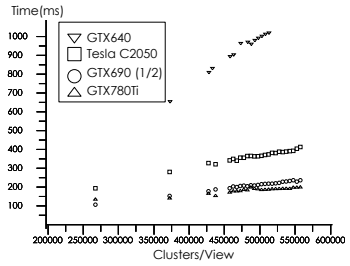

Figure 6: Absolute time (ms) for cluster correction and alignment. For GTX690, only one GPU is considered.

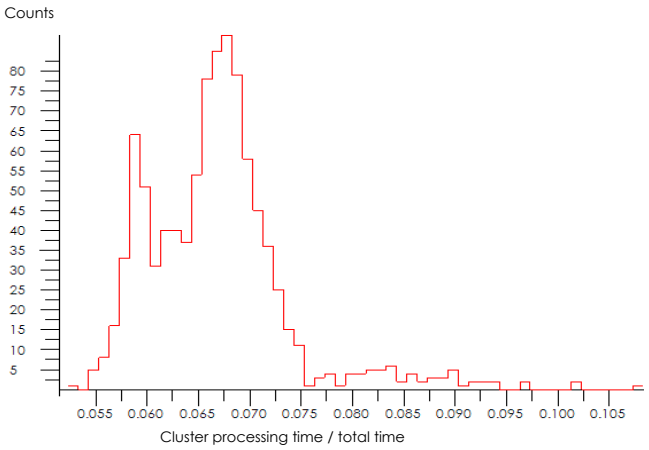

Figure 7: Fraction of total time used for cluster correction and alignment.

the implementation, and could be reduced by fine-tuning the logic: e.g, the kernel that merges track clones takes about 1/3 of the total time, and removing or taming its divergence offers good chances of overall speed-up. The relative improvements of a GPU-based system over traditional technologies can effectively be estimated by the cost of hardware. For QSS, taking data at 40∼90 cm^2/h with 1 GTX 590 for cluster recognition + 6 GTX 690 for alignment and tracking costs about 5.5 times less than the hardware of ESS taking data at 20 cm^2/h on the same emulsion films. The power consumption is similar, although it is worth noticing the data taking speed increase. The GPU-based system is also more modular and scalable.

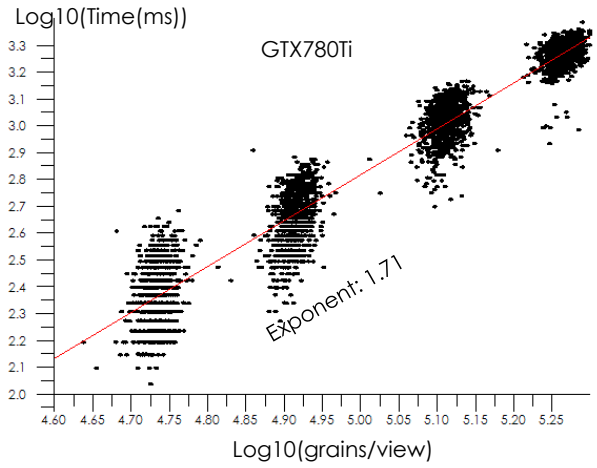

Figure 8: Dependency of tracking time (GTX780Ti) on the number of grain clusters/view (dark clusters with minimum size constraint).

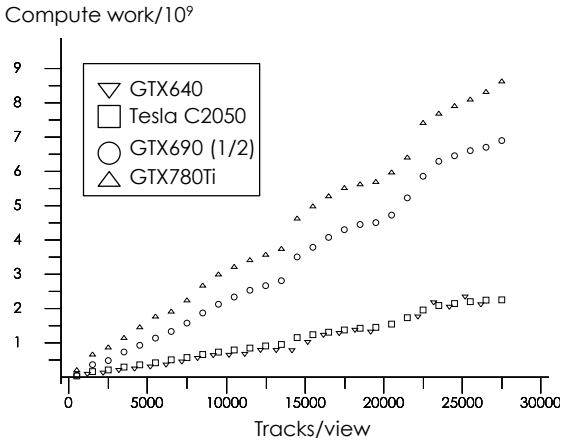

Figure 9: Compute work ($Cores \times Clock \times Time$, arbitrary units) for several boards. For GTX690, only one GPU is considered.

References

[1] E. Eskut *et al.*, Nucl. Phys. B **793** 326 (2008).

[2] A. Aoki *et al.*, New. J. Phys. **12** 113028 (2010).

[3] R. Acquafredda *et al.*, J. Inst. **4** P04018 (2009).

[4] N. Armenise *et al.*, Nucl. Inst. Meth. A **552** 261 (2005).

[5] L. Arrabito *et al.*, Nucl. Inst. Meth. A **568** 578 (2007).

[6] L. Arrabito *et al.*, J. Inst. **2** P05004 (2005).

[7] C. Bozza *et al.*, Nucl. Inst. Meth. A. **703** 204 (2013).

[8] M. De Serio *et al.*, Nucl. Inst. Meth. A **554** 247 (2005).