Rewriting the DCT pre-processing pipeline

João P C Bertoldo

15th April 2021

COPIL BIGMECA April 2021

Materials Center @ Mines Paristech - PSL University
ID11 @ The European Synchrotron Radiation Facility (ESRF)

What is DCT?

DCT

X-ray Diffraction Contrast Tomography

- Acquire transmitted and diffracted beam
- Richer local information: individual grains and their plane directions

← click

Figure 1: DCT setup schematic. Multiple 2D images acquired from different angles relative to the specimen. Credits: Wolfgang Ludwig.

How is it done?

From real world to 3D crystal structure

Main steps of the DCT reconstruction

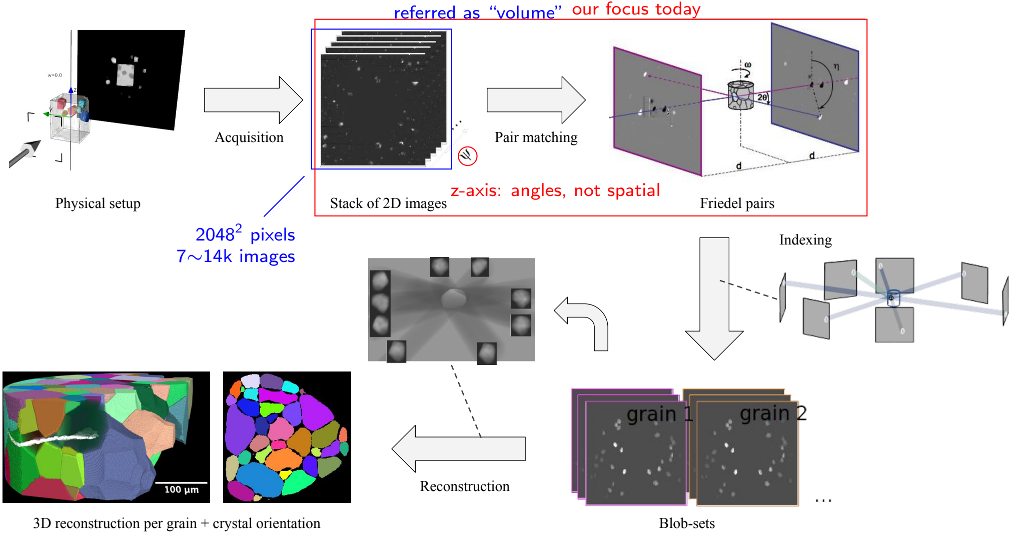

Figure 3: DCT reconstruction pipeline. Credits: Wolfgang Ludwig (modified).

Spotting the spots

From raw signal to Friedel (unpaired) 2D-blobs

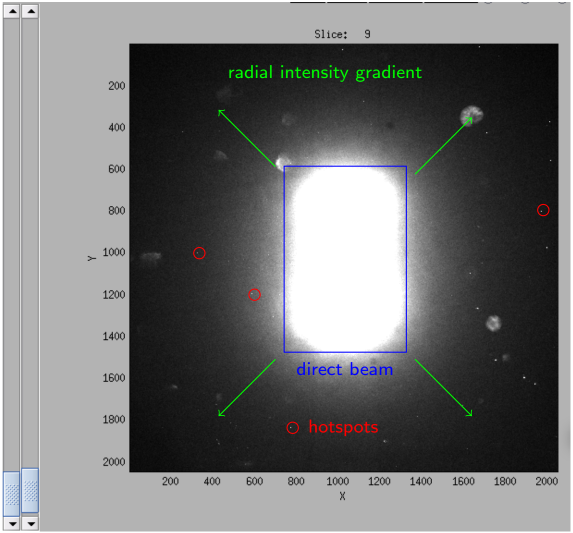

Figure 4: Raw DCT images.

- Pre-process the image
- Segment & extract blobs

Spotting the spots

From raw signal to Friedel (unpaired) 2D-blobs

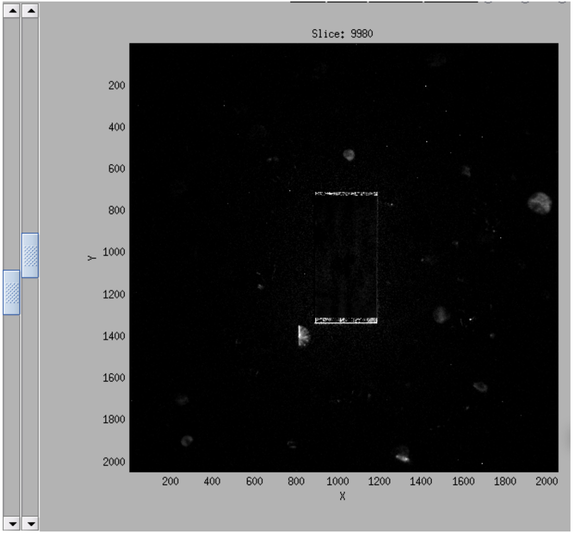

Figure 5: Preprocessed DCT image.

- Pre-process the image
- Segment & extract blobs

Spotting the spots

From raw signal to Friedel (unpaired) 2D-blobs

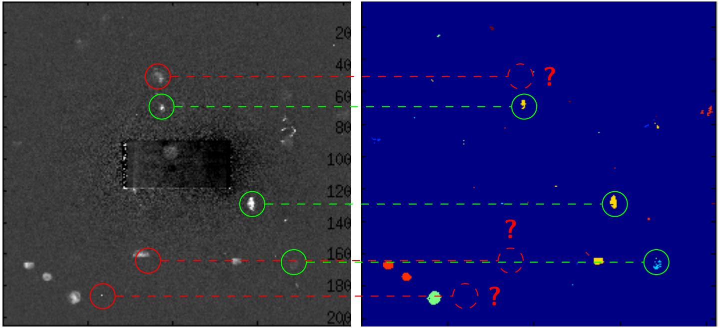

Figure 6: Blobs segmentation.

- Pre-process the image
- Segment & extract blobs

Zooming in

So far

- overall process
- conceptual steps of the preprocessing/segmentation pipeline
- image characteristics

Now

- how things are *actually* done today
- main concrete operations

Then

- why it works, pros and cons
- limitations and opportunities: where we're going

Preprocessing

Direct beam removal

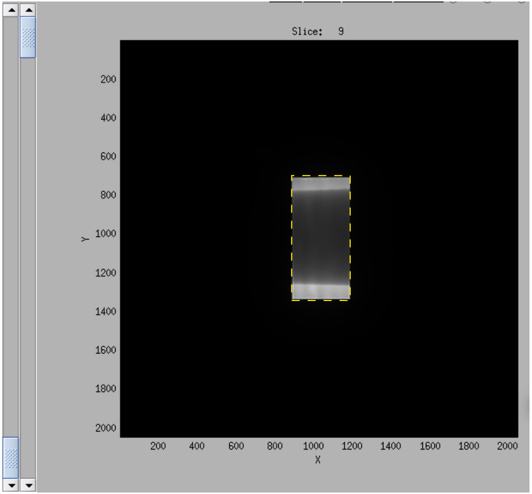

Figure 7: Direct beam selection.

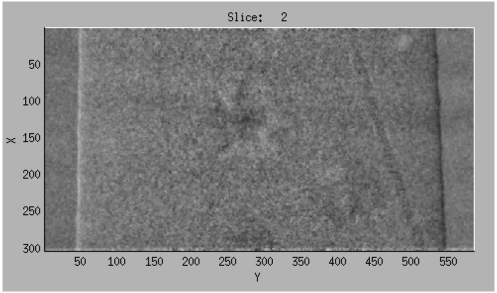

Figure 8: Direct beam preprocessed.

Direct beam is pre-processed separately:

- manual selection (Fig. 7)
- sensor offset removal (image without sample)
- spatial median removal
- normalization

Preprocessing

Background

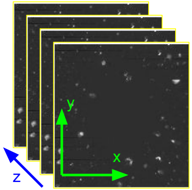

Figure 9: Stack of 2D images.

Volume Image

- W: width, along x, indexed by $i \in 1 \dots W$
- H: height, along y, indexed by $j \in 1 \dots H$
- D: depth[a], along z, indexed by $k \in 1 \dots D$
- I : image, a 3D grid of shape (W, H, D), where $I_{i,j,k} \in [0, V]$

[a] number of rotation steps = number of images

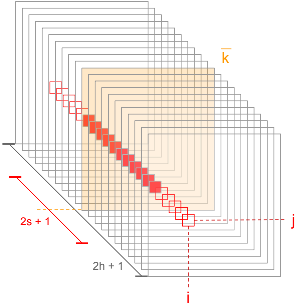

Figure 10: Background.

$$\text{range}(k, w) = \{k\prime \in 1 \dots D \mid max(0, k - w) \leq k\prime \leq min(V, k + w)\}$$

$$\text{Background:} \quad B^h_{i,j,\overline{k}} = \operatorname*{median}_{k' \in range(\overline{k}, h)} I_{i,j,k'}$$

$$\text{Image without background:} \quad I^*_{i,j,k\prime} = I_{i,j,k\prime} - B^h_{i,j,\overline{k}} \quad \text{for all } k\prime \in range(\overline{k}, s)$$

Parameters: $h = 250$ and $s = 25$

$W = H \approx 2.000 \rightarrow 4.000.000$ pixels/background

$D \approx 10.000 \rightarrow 10.000/2s = 10.000/50 = 200$ backgrounds

$\therefore$ $4.000.000 \times 200 = 8 \times 10^8$ medians of 250 values each...

Preprocessing

Background and hotspot removal

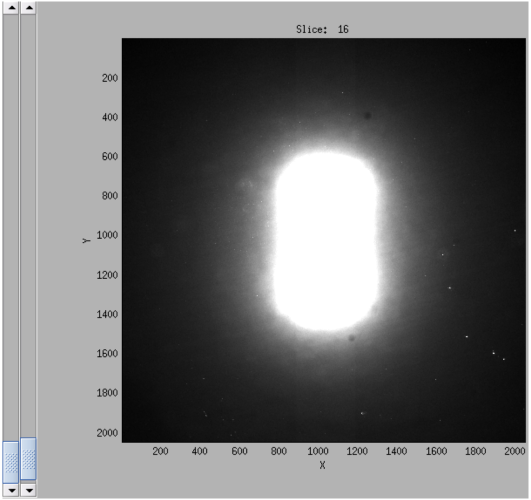

Figure 11: Background image.

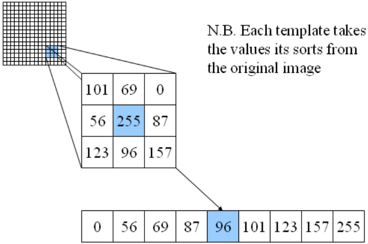

Figure 12: Median filter (*illustration*). Source link. Credits: Larry Bank.

- substract background (Fig. 11)
- applied 2D median filter (Fig. 12) on each slice

Blob segmentation

Double threshold approach + extra filters

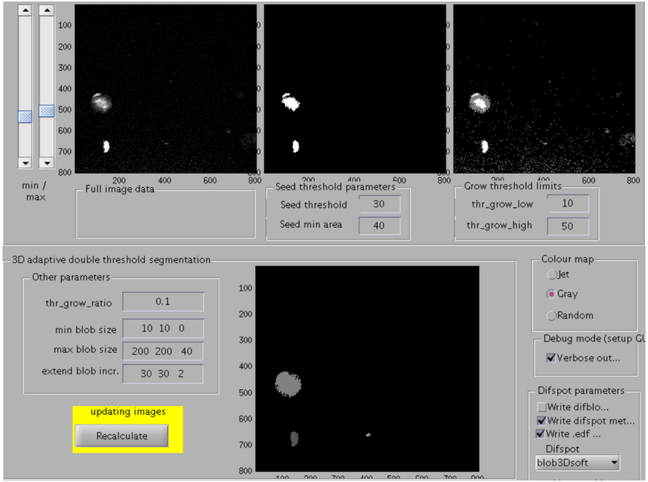

Figure 13: Double threshold + extra blob filters.

- choose a minimum threshold v_{min}
- pick all voxels $I_{i,j,k} \geq v_{min}$
- choose a tolerance $\tau \in [0, 1]$
- choose clipping limits c_{min} and c_{max}
- for each blob b
  - find its maximum intensity v^b_{max}
  - also pick connected voxels s.t.
    $$I_{i,j,k} \geq \tau v^b_{max}$$
  - retain those that
    $c_{min} \leq I_{i,j,k} \leq c_{max}$
- ignore blobs that are
  - too small or too big in XY
  - too long in Z to the direct beam

State of the play

Pre-processing

- Why is this (simple) background removal well adapted?
  - Link with the physical problem
  - Spots will pop up and go way, but the background is always there.
- Pains?
  - Manual direct beam selection
  - Messy parametrization
  - A lot of computation: billion-order medians – i.e. sort operations
  - *Why not a a single or spatially-batched median?* Because of the radial glow.
  - *Why not a single median per pixel position?* Beam intensity variation.
- Opportunities
  - An unexploited redundancy: overlay of the pixel-wise medians
  - Goal: speed up the process – ideally 30∼40 images/sec
  - Leads?
    - SLURM
    - Numba
    - GPU-based computation? (e.g. TensorFlow)
  - Alternative, better solutions?

State of the play

Segmentation (double threshold)

- Nice features
  - Notion of locality
  - Adapted to the heterogeneity inter/intra-crystal
- Pains
  - Manual, time-consuming parametrization
  - Needs filtering (XY-tiny and z-long blobs) $=$ +parameters
  - Slow
    - A region growth per blob ($\approx$ 100 thousand)
    - A few hours of processing on $\leq$ 10 jobs
    - (recall) My previous deep learning project: $\approx$ 50 min on 2 GPUs on a similar scale
    - Further acceleration (of a deep learning method): RepVGG (Ding et al., 2021)

Thank you for your attention!

Questions?

João P C Bertoldo, 15th April 2021

COPIL BIGMECA April 2021