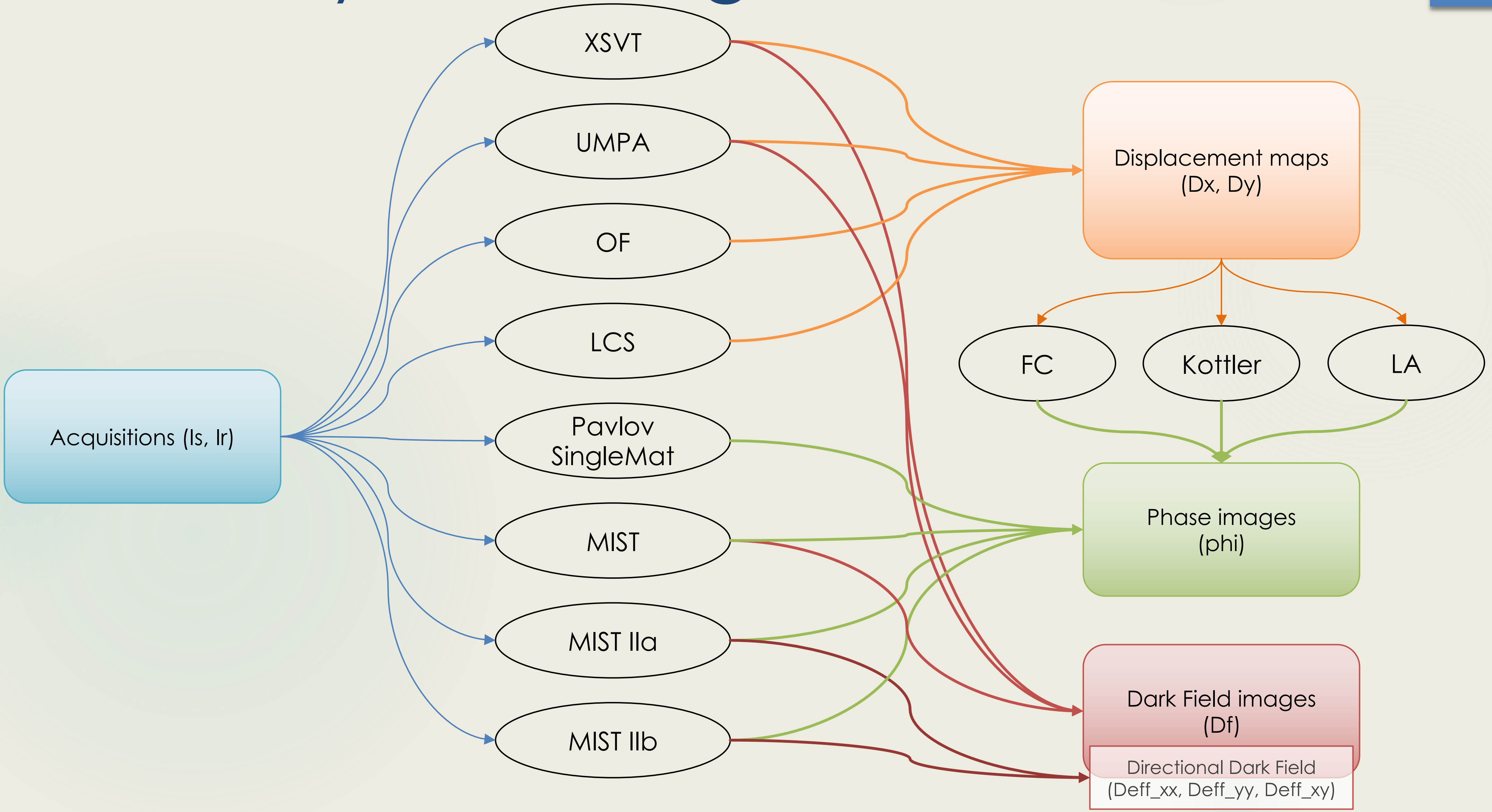
Phase retrieval algorithms

DATA SHEET AND USER GUIDE

Summary of the algorithms



Algorithms hypothesis and limitations

	Hypothesis	Limitations	Min number of acquisitions
UMPA	None	Resolution of the system	Several
XSVT	None	Resolution of the system	Several
OF	Phase object = NO absorption	Absorption	1
LCS	Low coherence but works also for quasi coherent systems	Noise, source blurring and PSF	3
Pavlov Single Material	Single material (AND phase object?)		1
MIST I	Phase object	Detector resolution/absorption	2
MIST IIa	Phase object	Detector resolution?	4
MIST IIb	Single material	Detector resolution?	4

Papers:

	Reference	Year
UMPA	Zdora, M. C., Thibault, P., Zhou, T., Koch, F. J., Romell, J., Sala, S., ... & Zanette, I. (2017). X-ray phase-contrast imaging and metrology through unified modulated pattern analysis. Physical review letters, 118(20), 203903.	2017
XSVT	Berujon, S., & Ziegler, E. (2015). Near-field speckle-scanning-based x-ray imaging. Physical Review A, 92(1), 013837.	2015
OF	Paganin, D. M., Labriet, H., Brun, E., & Berujon, S. (2018). Single-image geometric-flow x-ray speckle tracking. Physical Review A, 98(5), 053813.	2018
LCS	To be published	
Pavlov Single Material	Pavlov, K. M., Li, H. T., Paganin, D. M., Berujon, S., Rougé-Labriet, H., & Brun, E. (2020). Single-shot x-ray speckle-based imaging of a single-material object. Physical Review Applied, 13(5), 054023.	2020
MIST I	Pavlov, K. M., Paganin, D. M., Li, H. T., Berujon, S., Rougé-Labriet, H., & Brun, E. (2020). X-ray multi-modal intrinsic-speckle-tracking. Journal of Optics, 22(12), 125604.	2020
MIST IIa	To be published	
MIST IIb	To be published	

User guide

- 3 files to modify:
  - ExperimentParameters.xml
  - AlgorithmParameter.xml
  - allPhaseRetrievalMethods.py

The first two contain all the parameters of the experiment and the ones to use for the phase retrieval.

The .py allows to chose which phase retrieval algorithms to use ant to launch the calculations.

User guide

ExperimentParameters.xml

Start by creating a new **`experiment`** and enter all the correct values

```
<experiment>
    <experiment_name>MoucheSimapAout2017</experiment_name>
    <exp_folder>/Data/SIMAP/MoucheSimapAout2017/</exp_folder>
    <output_folder>/PhaseRetrieval/Results/MoucheSimapAout2017/</output_folder>
    <energy unit="keV">27</energy>
    <pixel unit="m">48e-6</pixel>
    <dist_object_detector unit="m">0.6385</dist_object_detector>
    <dist_source_object unit="m">0.0316</dist_source_object>
    <delta>3.49E-07</delta>
    <beta>3.57E-10</beta>
    <source_size unit="um">2</source_size>
    <detector_PSF unit="pix">1.2</detector_PSF>
    <crop_on>False</crop_on>
    <cropDebX unit="pix">0</cropDebX>
    <cropDebY unit="pix">0</cropDebY>
    <cropEndX unit="pix">-1</cropEndX>
    <cropEndY unit="pix">-1</cropEndY>

</experiment>
```

/!\The experiment name must be consistent in all 3 files

Folder containing the data
/!\ the folders must be called ‘ref’ and ‘sample’

Folder that will contain the computed images (must already exist)

Detector pixel size

Refraction and absorption indices (can be found there: http://ts-imaging.science.unimelb.edu.au/Services/Simple/)

FHWM (not actually useful for now…)

Value used for de-blurring of the acquisitions.

To choose a region of interest

/!\ all units given in the .xml files are only here to help the user fill the file, they must not be changed! They are not read in the program.

User guide

AlgorithmParameters.xml

Start by creating a new **`experiment`** and enter all the correct values

```xml
<experiment>
    <experiment_name>MoucheSimapAout2017</experiment_name>
    <nb_of_point>20</nb_of_point>
    <pad_size>0</pad_size>
    <pad_type>reflect</pad_type>
    <do_deconvolution>False</do_deconvolution>
    <deconvolution_type>unsupervised_wiener</deconvolution_type>
    <absorption_correction_sigma>15</absorption_correction_sigma>
    <max_shift>1</max_shift>
    <sigma_regularization>1</sigma_regularization>
    <LCS_median_filter>3</LCS_median_filter>
    <umpaNw>5</umpaNw>
    <MIST_median_filter>3</MIST_median_filter>
</experiment>
```

/!\The experiment name must be consistent in all 3 files

Number of measurement {Is, Ir} to use

Padding acquisitions to reduce edge effects in some methods (may be usefull for OF, Pvlov et MISTx)

De-blurring of the acquisitions (uses the PSF value) greatly improves results in some cases

In some cases we assume no absorption, it can be corrected using the ratio Is/Ir with a small blurring of this size. This correction is not great in most cases because it induces a edge enhancement that will be confused with phase.

Limit of displacement in pixel that should be measured (usefull for filtering outliers)

Filtering very low frequencies for methods using Fourier transforms (OF, Pavlov, MISTx)

Filtering of shot noise in displacement maps retrieved with LCS

Size of the cross correlation windows for UMPA: window=(2*umpaNw+1)x(2*umpaNw+1)

Median filter used on the retrieved directional dark field images

/!\ all units given in the .xml files are only here to help the user fill the file, they must not be changed! They are not read in the program.

User guide

allPhaseRetrievalMethods.py

Set the value of “studied_case”

studied_case = 'MoucheSimapAout2017’ same experiment name as in the other files

Choose the phase retrieval methods you want to use by setting them to “True” (“False” otherwise)

Run the script

Code architecture

- allPhaseRetrievalMethods.py
  - Main():
    - Receives all interesting parameters
    - Calls readStudiedCase()
    - Calls one by one each processers of phase retrieval algorithms « process… ()»

  - readStudiedCase(): contains and reads the data specific to an experiment,
    - opens the acquisitions and normalizes
    - Arguments:
      - sCase [string]: the experiment name
      - nbImages [int]: the number of pairs of acquisitions to take into account
      - machinePrefix [string]: the name of the machine you are working on
    - Outputs:
      - Is [numpy array]: contains the nbImages sample images
      - Ir [numpy array]: contains the nbImages reference images

Code architecture

- allPhaseRetrievalMethods.py
  - ProcessXXX(): calls the XXX() phase retrieval function and saves all the calculated images

- OpticalFlow2020.py, UMPA.py, LCS.py
  - processProjectionXXX():
    - calls XXX()phase derivative calculators
    - Converts displacements to phase derivative
    - Calls, frankotchellappa(), kottler() and LarkinAnissonSheppard() integration functions to calculate the phase 3 different ways.
    - Returns displacement maps and phases images

Calculations – XSVT

- **XSVT() /!\ matlab implementation**

  → Crossed correllation

$$D_{\perp}(x_0, y_0) = \underset{\delta_x,\delta_y}{\operatorname{argm}ax} \iint \widehat{I_s}\,(P_x, P_y, x, y)\widehat{I_R}\,\left(P_x, P_y, x - \delta_x, y - \delta_y\right)dxdy$$

Calculations - UMPA

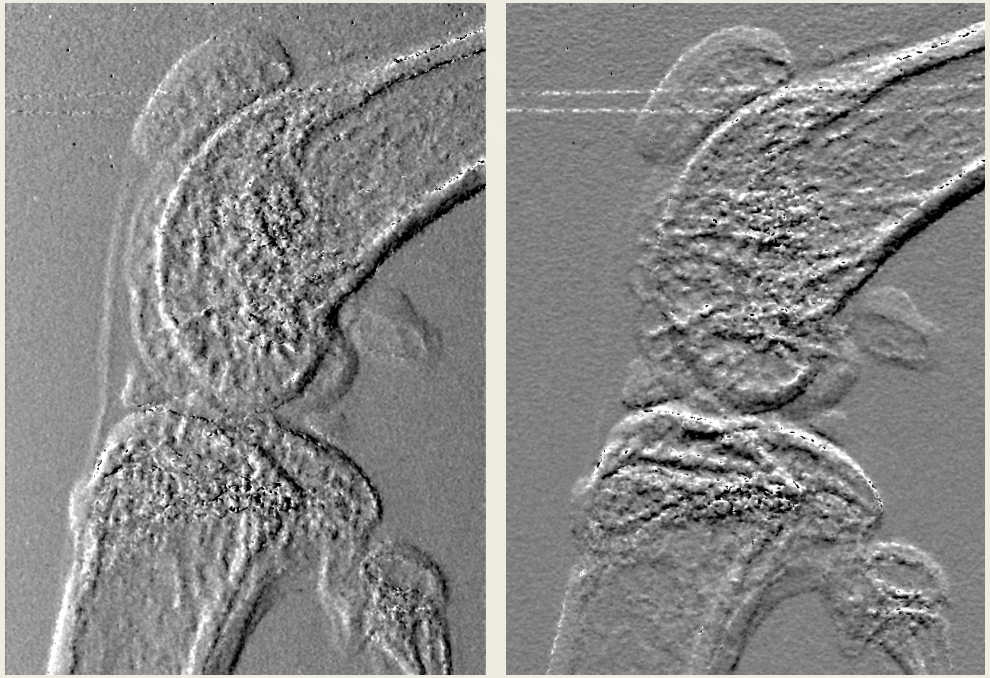
- **UMPA()**

  Calculates the displacement images of a pure phase object from sample and reference images.

  - Returns:
    - Dx [numpy array]: the displacements along x axis
    - Dy [numpy array]: the displacements along y axis

→ Minimization of the mean squared error:

$$D_{\perp}(x_0, y_0) = \underset{\delta_x, \delta_y}{\operatorname{argmin}} \iint w(x - x_0, y - y_0)\left[TI_r(x - \delta_x, y - \delta_y) - I_s(x, y)\right]^2 dxdy$$



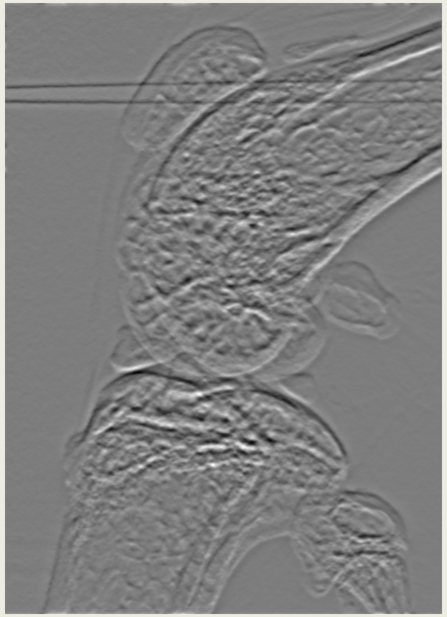
Rq: also gives DarkField and thickness but the code downloaded from github does not work very well, I still have to figure out why.

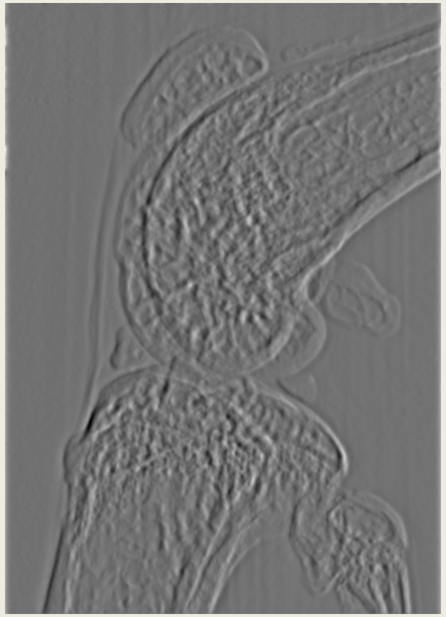
Calculations - OF

- **derivativesByOpticalflow()**

  Calculates the displacement images of a pure phase object from sample and reference images.

  - Returns:
    - Dx [numpy array]: the displacements along x axis
    - Dy [numpy array]: the displacements along y axis





Transport of intensity equation: $I_s(x,y) - I_r(x,y) = \nabla_\perp \cdot [I_r(x,y) D_\perp(x,y)]$

Numerical resolution method: $D_\perp(x,y) = \frac{i}{I_r(x,y)} \mathcal{F}^{-1}\left\{(k_x, k_y)\left[\frac{\mathcal{F}[I_s(x,y) - I_r(x,y)]}{k_x^2 + k_y^2}\right]\right\}$

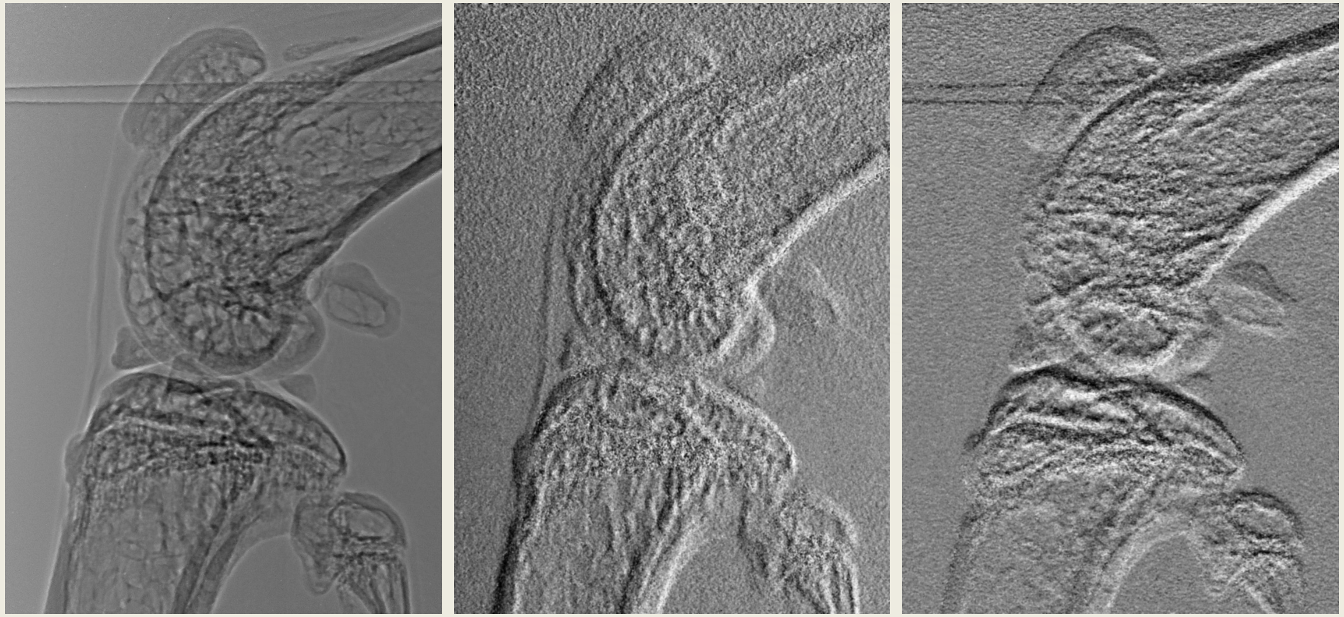
Calculation - LCS

- **LCS()**

  → Calculates the displacement images from sample and reference images using the LCS system

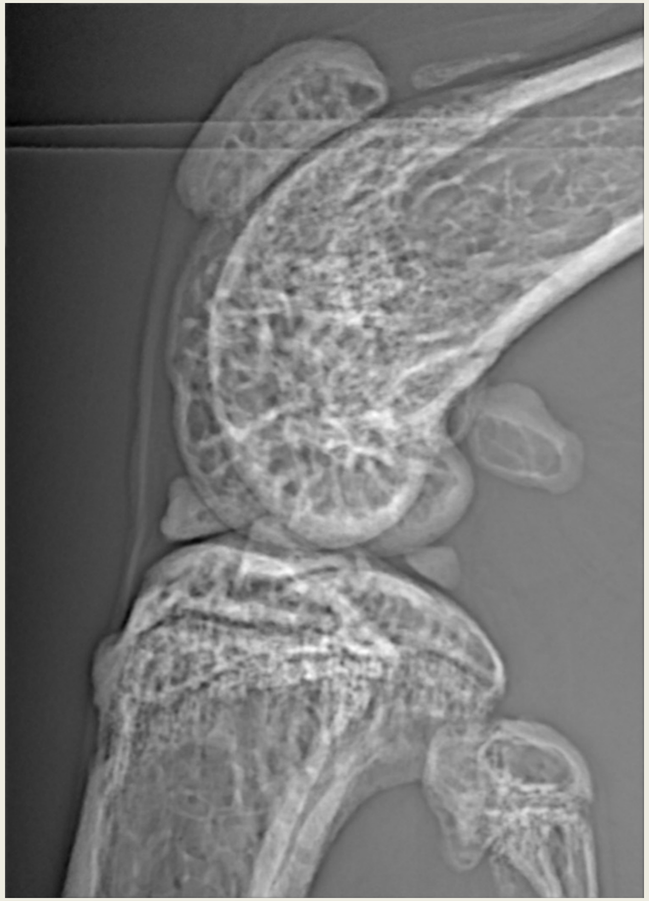
  - Returns:
    - Dx [numpy array]: the displacements along x axis
    - Dy [numpy array]: the displacements along y axis
    - absoprtion [numpy array]: the absorption
  - System to solve (Variables are in red)

$$\begin{cases} I_r^{(1)}(x,y) = \frac{1}{I_{abs}(x,y)} I_s^{(1)}(x,y) - D_x(x,y)\frac{\partial I_r^{(1)}(x,y)}{\partial x} + D_y(x,y)\frac{\partial I_r^{(1)}(x,y)}{\partial y} \\ I_r^{(2)}(x,y) = \frac{1}{I_{abs}(x,y)} I_s^{(2)}(x,y) - D_x(x,y)\frac{\partial I_r^{(2)}(x,y)}{\partial x} + D_y(x,y)\frac{\partial I_r^{(2)}(x,y)}{\partial y} \\ I_r^{(3)}(x,y) = \frac{1}{I_{abs}(x,y)} I_s^{(3)}(x,y) - D_x(x,y)\frac{\partial I_r^{(3)}(x,y)}{\partial x} + D_y(x,y)\frac{\partial I_r^{(3)}(x,y)}{\partial y} \end{cases}$$



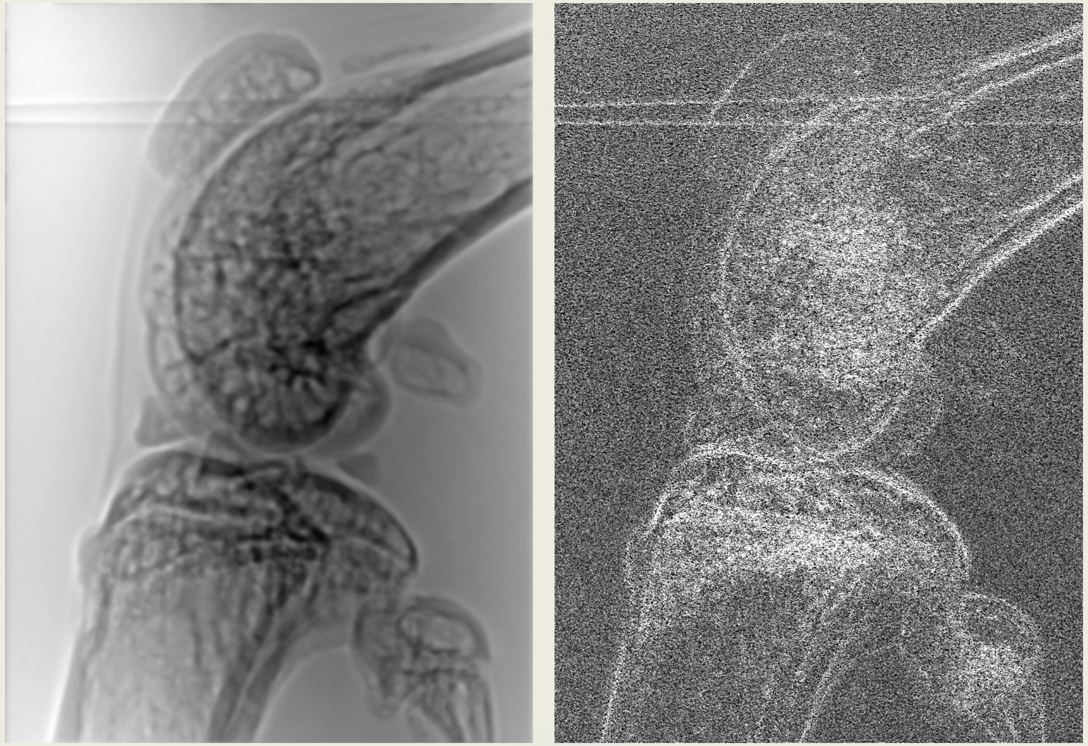
Calculation – Pavlov Single Material

- **tie_Pavlovetal2020()**
- → Calculates the thickness (equivalent to phase in this case) of a single material object form sample and reference images (starting from the TIE).
  - $t(x, y) = \frac{1}{\mu}\mathcal{F}^{-1}\left\{LFF\left[\frac{\mathcal{F}\{1-I_s(x,y)/I_r(x,y)\}}{1+\pi\gamma L\lambda\left(k_x^2+k_y^2\right)}\right]\right\}$
  - $\phi(x, y) = -k\delta t(x, y)$ (Proportionality between phase and thickness of a single material object)



Calculations – MISTI

- **MISTI():** calculates the dark field and the phase of a single-material object from sample and reference images.
  - $Deff$ [numpy arrays]: the Dark-Field
  - phi [numpy array]: the phase of the object



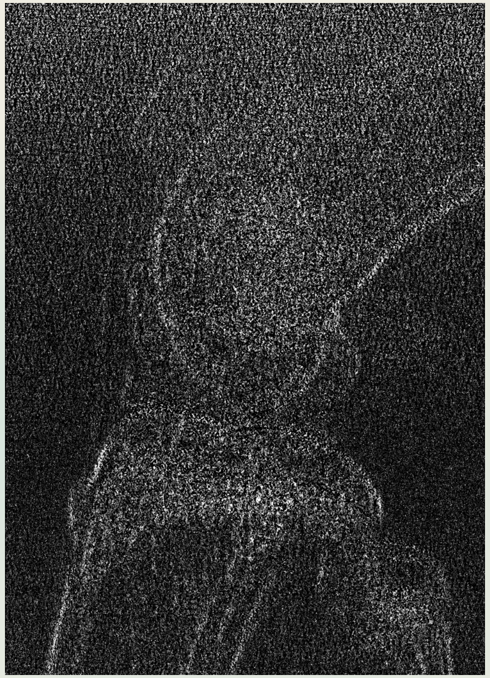
Calculations – MISTII2

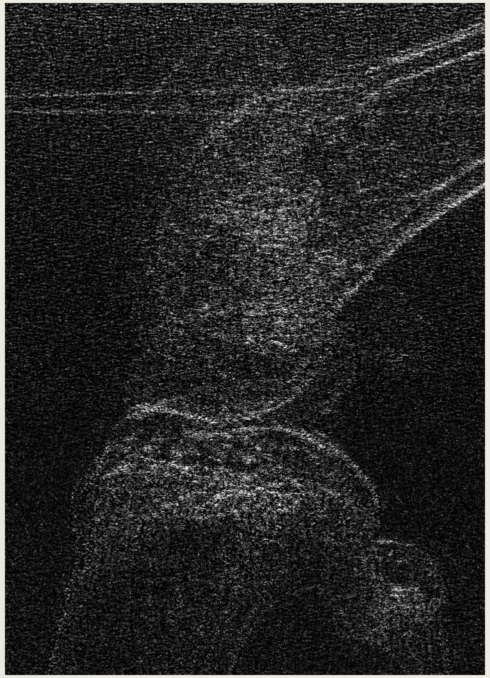
- MISTII2() : Calculates the directionnal dark field of a single material object from sample and reference images
- Returns:
  - $D_{eff}^{(xx)}, D_{eff}^{(yy)}, D_{eff}^{(xy)}$ [numpy arrays]: the tensors of the directional Dark-Field
  - thickness [numpy array]: the thickness of the object
- Solves the system:

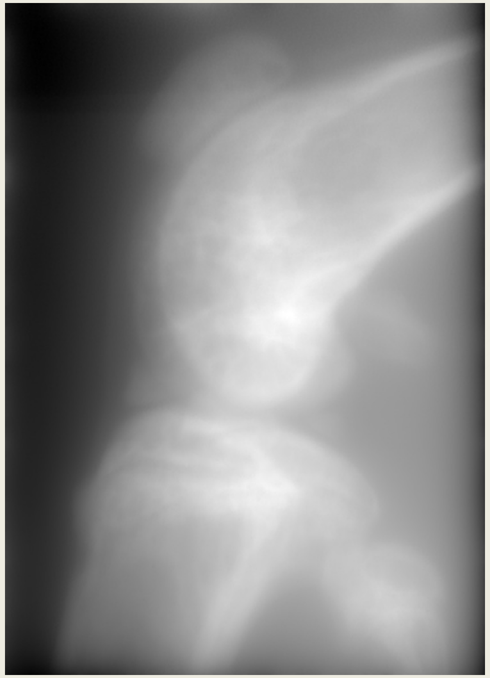
$$\frac{I_s^{(i)}(x,y)}{I_r^{(i)}(x,y)} = \left(1 - \frac{\gamma\Delta}{2k}\nabla_\perp^2\right) I_{ob}(x,y) + \Delta D_{eff}^{(xx)} \frac{\partial^2}{\partial x^2} I_{ob}(x,y) + \Delta D_{eff}^{(yy)} \frac{\partial^2}{\partial y^2} I_{ob}(x,y) + \Delta D_{eff}^{(xy)} \frac{\partial^2}{\partial x \partial y} I_{ob}(x,y) + \Delta D_{eff}^{(xx)} I_{ob}(x,y) \frac{\frac{\partial^2}{\partial x^2} I_r^{(i)}(x,y)}{I_r^{(i)}(x,y)} + \Delta D_{eff}^{(yy)} I_{ob}(x,y) \frac{\frac{\partial^2}{\partial y^2} I_r^{(i)}(x,y)}{I_r^{(i)}(x,y)} + \Delta D_{eff}^{(xy)} I_{ob}(x,y) \frac{\frac{\partial^2}{\partial x \partial y} I_r^{(i)}(x,y)}{I_r^{(i)}(x,y)}$$

  - $\nabla_\perp^2$: orthogonal Laplacien (along x and y)
  - Δ distance object detector
  - $k = 2\pi E \frac{e}{hc}$ wavenumber

Calculations – MISTII2







Calculations – MISTII1

- **MISTII1** : Calculates the directionnal dark field of a pure phase object from sample and reference images
- Returns:
  - $D_{eff}^{(xx)}, D_{eff}^{(yy)}, D_{eff}^{(xy)}$ [numpy arrays]: the tensors of the directional Dark-Field
  - thickness [numpy array]: the thickness of the object

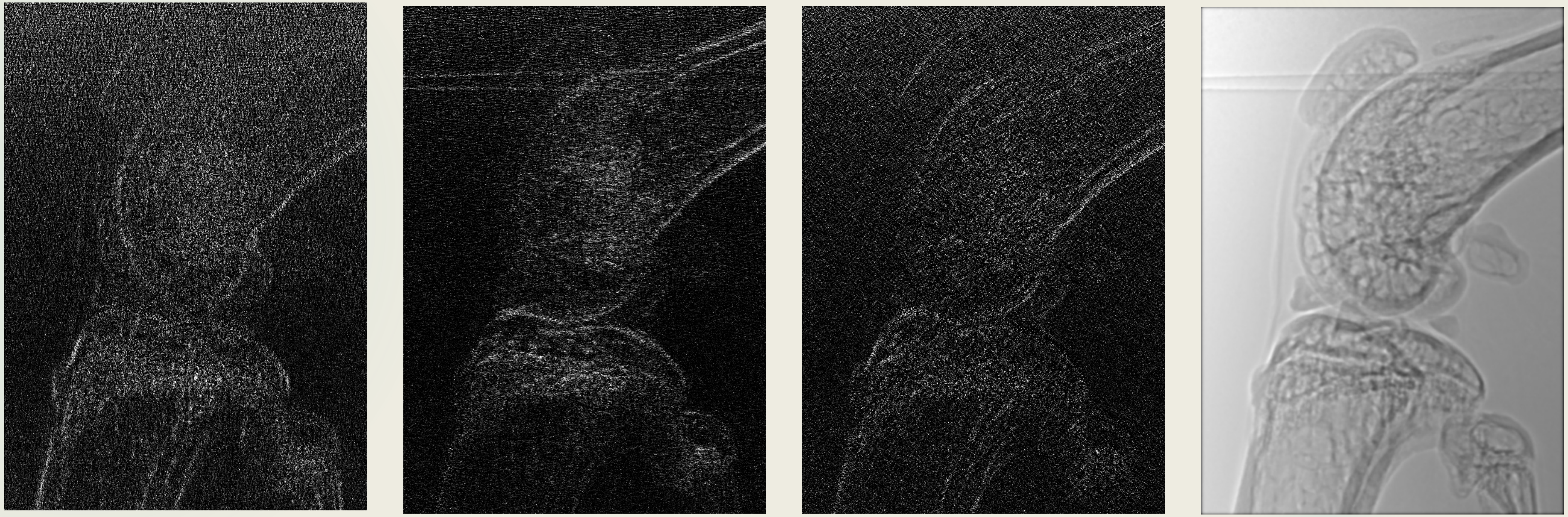
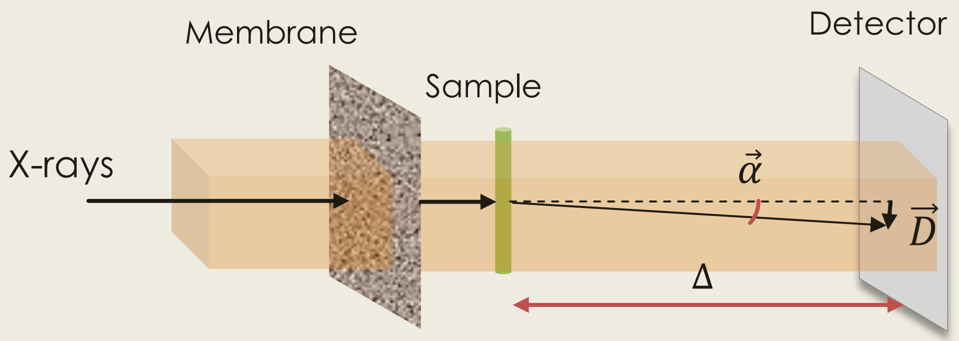


Image integration

$$\frac{d\phi}{dx} = \frac{2\pi\alpha_x}{\lambda} = \frac{2\pi D_x}{\lambda\Delta}$$

$$\frac{d\phi}{dy} = \frac{2\pi\alpha_y}{\lambda} = \frac{2\pi D_y}{\lambda\Delta}$$



Algorithm	Reference	Year
FC	R. T. Frankot and R. Chellappa, "A Method for Enforcing Integrability in Shape from Shading Algorithms," 1988.	1988
LA	M. R. Arnison, K. G. Larkin, C. J. R. Sheppard, N. I. Smith, and C. J. Cogswell, "Linear phase imaging using differential interference contrast microscopy," *J. Microsc.*, vol. 214, no. 1, pp. 7–12, Apr. 2004.	2004
Kottler	C. Kottler, C. David, F. Pfeiffer, and O. Bunk, "A two-directional approach for grating based differential phase contrast imaging using hard x-rays," 2007.	2007

Image integration - FC

- frankoChellappa.py
  - frankotchellappa()

$$\phi(x,y) = \mathcal{F}^{-1}\left\{\frac{1}{k_x^2 + k_y^2}\left(-ik_x\mathcal{F}\left\{\frac{\partial\phi}{\partial x}(x,y)\right\} - ik_y\mathcal{F}\left\{\frac{\partial\phi}{\partial y}(x,y)\right\}\right)\right\}$$

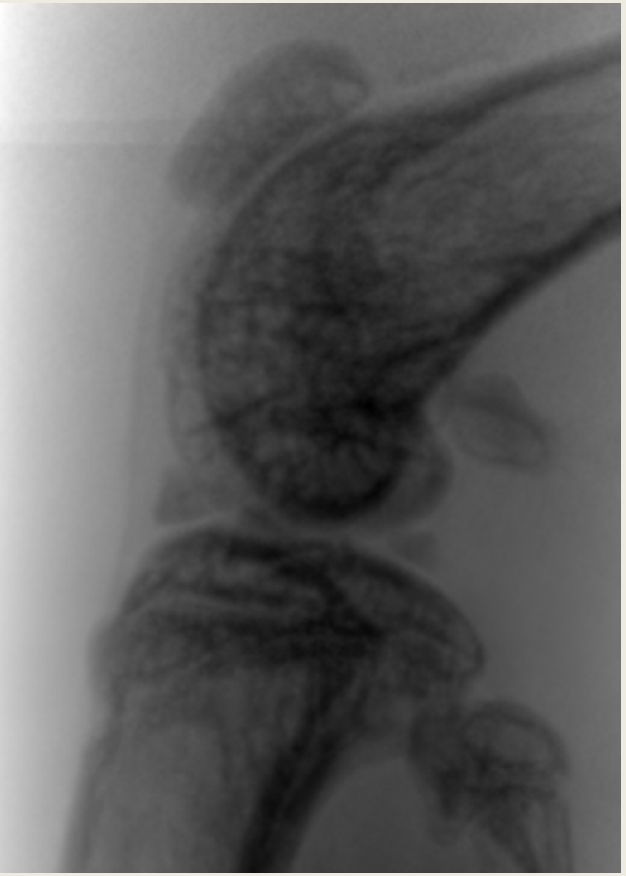


Image integration - LA

- phaseIntegration.py
  - LarkinAnissonSheppard()

$$\phi(x,y) = \mathcal{F}^{-1}\left\{\frac{1}{ik_x - k_y}\mathcal{F}\left\{\frac{\partial\phi}{\partial x}(x,y) + i\frac{\partial\phi}{\partial y}(x,y)\right\}\right\}$$

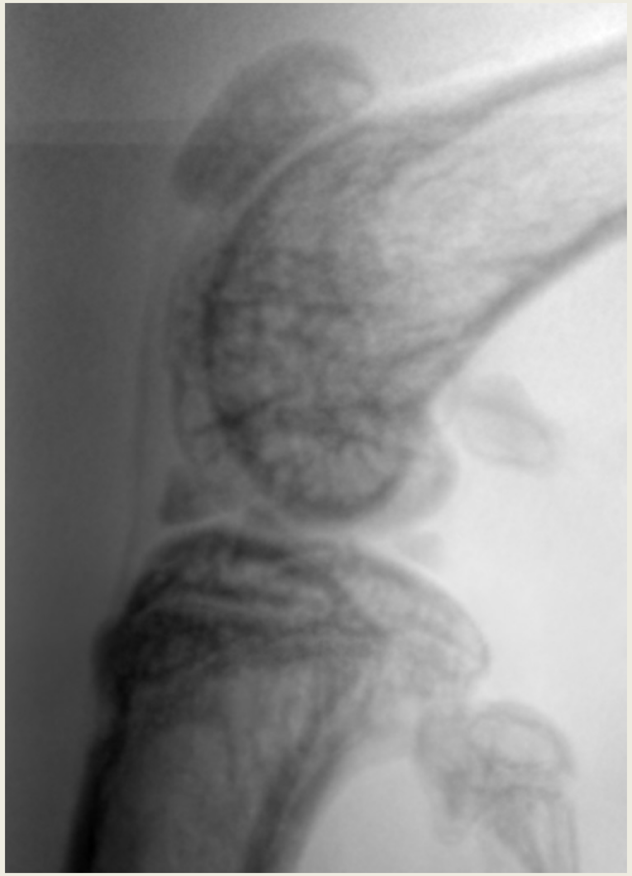


Image integration - kottler

- phaseIntegration.py
  - kottler()

$$\phi(x,y) = \mathcal{F}^{-1}\left\{\mathcal{F}\left\{\frac{\partial\phi}{\partial x}(x,y) + i\frac{\partial\phi}{\partial y}(x,y)\right\} * e^{i\tan^{-1}(k_x/k_y)}\right\}$$