Thinlinc: thinlinc.gbar.dtu.dk with DTU (or guest) login

Applications -> DTU -> Mathematics -> Astra Tomography Toolbox Snapshot 20210110...
Additional resources for exercises

• SPOT Toolbox:
  https://www.cs.ubc.ca/labs/scl/spot/

• Two data sets:
  https://bit.ly/2KVZR2a
  • Exercises.zip
  • ExampleDatasets.zip
Rotation matrix

In two dimensions

In two dimensions, the standard rotation matrix has the following form:

\[ R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}. \]

This rotates column vectors by means of the following matrix multiplication,

\[ \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \]

Thus, the new coordinates \((x', y')\) of a point \((x, y)\) after rotation are

\[ x' = x \cos \theta - y \sin \theta \]
\[ y' = x \sin \theta + y \cos \theta. \]

Examples

For example, when the vector \(\hat{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\) is rotated by an angle \(\theta\), its new coordinates are \(\begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}\).

and when the vector \(\hat{y} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}\) is rotated by an angle \(\theta\), its new coordinates are \(\begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix}\).

From wikipedia: Rotation matrix
Geometric magnification and virtual detector

- **Geometric magnification** \( \frac{S+D}{S} \)

- Typical use reconstruction grid same with same number of pixels as detector. In this case, the geometric magnification is the scaling factor between reconstruction pixel size and the detector pixel size.

- **Virtual detector**: Pretend \( S = 0 \).

- Then geometric magnification is 1, and pixels same size.

https://www.medicalconnections.co.uk/kb/Geometric-Magnification
Other tomographic reconstruction software

- **TIGRE**: [https://github.com/CERN/TIGRE](https://github.com/CERN/TIGRE)
- **CIL**: [https://www.ccpi.ac.uk/cil](https://www.ccpi.ac.uk/cil)

Many others, see overview at [https://tomopedia.github.io/software/](https://tomopedia.github.io/software/)
Core Imaging Library (CIL): Modular object-oriented reconstruction framework

- Conventional and multi-channel CT, ...
- "Mix & match" operators, functions, algorithms, ...
- Open-source Python, available from conda.
- www.ccpi.ac.uk/cil

### Optimisation Framework

**Least Squares**
\[
\min_u \|Au - g\|_2^2
\]

**Tikhonov Regularisation**
\[
\min_u \alpha \|\nabla u\|_2^2 + \frac{1}{2}\|Au - g\|_2^2
\]

**Total Variation Regularisation**
\[
\min_{u>0} \alpha \|\nabla u\|_{2,1} + \int Au - g \log(Au)
\]

```python
# Setup and run the CGLS algorithm
2cglsl = CGLS(u_init, A, g)
3cglsl.run(10)

# Setup and run the FISTA algorithm
6f = alpha * Norm2Sq(Gradient)
7g = 0.5 * Norm2Sq(A, g)
8fista = FISTA(u_init, f, g)
9fista.run(2000)

# Setup and run the PDHG algorithm
12operator = BlockOperator(Gradient, A)
13f = BlockFunction( alpha * MixedL2Norm(),
14        KullbackLeibler(g) )
15g = IndicatorBox(lower=0)
16pdhg = PDHG(f, g, operator, tau, sigma)
17pdhg.run(3000)
```
Tomographic imaging of planar samples with laminography

- Planar samples like composite panels and printed circuit boards difficult to scan due to different exposure along views.
- Conventional scan gives limited-angle artifacts, missing edges.
- Laminography allows uniform exposure.
- Non-standard geometry needs dedicated reconstruction – here used CGLS.

Laminography Lego case with TV and nonnegativity in CIL

- Least Squares, unconstrained
- Least squares, nonnegativity

No regu.

TV regu.