

Inverse problems and Structured Illumination Microscopy

F. Orieux

orieux@l2s.centralesupelec.fr

Laboratoire des Signaux et Systèmes
Univ. Paris-Sud – CNRS – CentraleSupélec – Univ. Paris-Saclay

We address the problem of image reconstruction in structured illumination microscopy (SIM). The forward model is

$$\mathbf{g} = \mathbf{HMR}\mathbf{f} + \mathbf{n}$$

where

- $\mathbf{f} \in \mathbb{R}^N$ is the unknown image,
- $\mathbf{g} \in \mathbb{R}^N$ the collection of modulated data images,
- $\mathbf{n} \in \mathbb{R}^M$ the unknown noise,
- \mathbf{R} the replication matrix, the transpose \mathbf{R}^t being a “sum”.
- \mathbf{M} the block diagonal modulation matrix,
- $\mathbf{H} = \mathbf{F}^\dagger \mathbf{\Lambda}_H \mathbf{F}$ the block circulant convolution matrix.

The directory contains Matlab[®] / Octave codes to resolve the problems.

Codes must be downloaded at <http://research.orieux.fr/files/teaching/codes.zip>

Slides and subject are available <http://research.orieux.fr/teaching.html>

1 Supervised reconstruction

The naive solution (*i.e.*, the least square solution) is ill-posed. We defined the solution as the minimizer of the regularized least square solution

$$\hat{\mathbf{f}} \stackrel{\text{def}}{=} \arg \min_{\mathbf{f}} J(\mathbf{f}) = \|\mathbf{g} - \mathbf{HMR}\mathbf{f}\|^2 + \lambda \|\mathbf{D}\mathbf{f}\|^2.$$

The gradient is

$$\nabla J(\mathbf{f}) = 2\mathbf{R}^t \mathbf{M}^t \mathbf{H}^t \mathbf{g} + 2\mathbf{R}^t \mathbf{M}^t \mathbf{H}^t \mathbf{HMR}\mathbf{f} + 2\lambda \mathbf{D}^t \mathbf{D}\mathbf{f}.$$

The Hessian is

$$\nabla^2 J = 2\mathbf{R}^t \mathbf{M}^t \mathbf{H}^t \mathbf{HMR} + 2\lambda \mathbf{D}^t \mathbf{D}.$$

The solution is known and equals

$$\hat{\mathbf{f}} = (\mathbf{R}^t \mathbf{M}^t \mathbf{H}^t \mathbf{HMR} + 2\lambda \mathbf{D}^t \mathbf{D})^{-1} \mathbf{R}^t \mathbf{M}^t \mathbf{H}^t \mathbf{g}$$

Unfortunately it can't be directly computed. A conjugate gradient optimization algorithm is used to reach it.

- Line 2 computes the gradient, at initial point.
- Line 8 updates the unknown with a descent direction.
- Line 7 computes the optimal step.
- Line 12 computes the new direction conjugate. direction, using the gradient and the previous direction.
- Line 9 computes the gradient with a recursive formula of with the original formula.

Algorithm 1 The conjugate gradient algorithm

- 1: Set $\mathbf{f}^{(0)}$ and $n \leftarrow 0$
 - 2: $\mathbf{r} \leftarrow \mathbf{b} - \mathbf{Q}\mathbf{f}^{(0)}$ $\triangleright = -\nabla J(\mathbf{f}^{(0)})$
 - 3: $\mathbf{d}^{(0)} \leftarrow \mathbf{r}$
 - 4: $\delta^{(0)} \leftarrow \mathbf{r}^t \mathbf{r} = \|\mathbf{r}\|^2$
 - 5: **repeat**
 - 6: $\mathbf{q} \leftarrow \mathbf{Q}\mathbf{d}$
 - 7: $\alpha \leftarrow \delta^{(n)} / \mathbf{d}^t \mathbf{q}$
 - 8: $\mathbf{f}^{(n+1)} \leftarrow \mathbf{f}^{(n)} + \alpha \mathbf{d}$
 - 9: $\mathbf{r} \leftarrow \mathbf{r} - \alpha \mathbf{q}$ \triangleright or $\mathbf{r} \leftarrow \mathbf{b} - \mathbf{Q}\mathbf{f}^{(n)}$ if
 - mod($n, 50$) = 0
 - 10: $\delta^{(n+1)} \leftarrow \mathbf{r}^t \mathbf{r}$
 - 11: $\beta \leftarrow \delta^{(n+1)} / \delta^{(n)}$
 - 12: $\mathbf{d} \leftarrow \mathbf{r} + \beta \mathbf{d}$
 - 13: $n \leftarrow n + 1$
 - 14: **until** Some criterion is met \triangleright Ex : $\|\mathbf{d}^{(n)}\|^2 \leq \epsilon \|\mathbf{d}^{(0)}\|^2$
-

Instructions

1. Look `forward`, `forward_transpose`, and `hessian` functions and compare with equations.
Observe how the code implements implicit operators and not explicit operator : there is no matrix and vectors in the code but application of codes on variables.
2. Run `main.m` script to initialize the problem.
3. Inspect the data \mathbf{g} (`data`), the modulation \mathbf{M} (`grid`), the optical transfer function \mathbf{H} (`otf`).
4. Try to separate D_0 , D_1 and D_2 with 9 modulated images.
5. Run `main_supervised.m` to compute $\hat{\mathbf{f}}$. You can also inspect the conjugate gradient code and look for the different variables values during the iteration.
6. Try different hyper-parameter values for `hypers(2)`.
7. Try different number of iterations.
8. Try different modulation \mathbf{M} (in number, orientation, angles, ...).
9. Compute the criterion values inside the conjugate gradient

$$J(\mathbf{f}) = \|\mathbf{g} - \mathbf{HMR}\mathbf{f}\|^2 + \lambda\|\mathbf{D}\mathbf{f}\|^2$$

and check that criterion value decrease.

2 Unsupervised reconstruction

We want to estimate the regularization parameters. Within a Bayesian approach, the posterior law writes

$$p(\mathbf{f}, \gamma_{\mathbf{f}}, \gamma_{\mathbf{n}} \mid \mathbf{g}) \propto \gamma_{\mathbf{n}}^{\frac{N}{2}-1} \gamma_{\mathbf{f}}^{\frac{N-1}{2}-1} \exp\left(-\frac{\gamma_{\mathbf{n}}}{2}\|\mathbf{g} - \mathbf{HMR}\mathbf{f}\|^2 + \frac{\gamma_{\mathbf{f}}}{2}\|\mathbf{D}\mathbf{f}\|^2\right)$$

where $\lambda = \gamma_{\mathbf{f}}/\gamma_{\mathbf{n}}$ in the supervised framework. With the Gibbs sampler

1. $\mathbf{x}^{(k)} \sim p(\mathbf{f} \mid \gamma_{\mathbf{f}}^{(k-1)}, \gamma_{\mathbf{n}}^{(k-1)}, \mathbf{g})$,
2. $\gamma_{\mathbf{f}}^{(k)} \sim p(\gamma_{\mathbf{f}} \mid \mathbf{f}^{(k)}, \gamma_{\mathbf{n}}^{(k-1)}, \mathbf{g})$,
3. $\gamma_{\mathbf{n}}^{(k)} \sim p(\gamma_{\mathbf{n}} \mid \mathbf{f}^{(k)}, \gamma_{\mathbf{f}}^{(k)}, \mathbf{g})$,
4. $k \leftarrow k + 1$ and return in 1,

to simulate samples, the posterior mean estimator is approximated as

$$\hat{\mathbf{x}} \approx \frac{1}{K} \sum_{k=0}^{K-1} \mathbf{x}^{(k)}.$$

Instructions

9. Run the `main_unsupervised.m` script.
10. Inspect the function and identify each part of the Gibbs sampler.
11. Identify the equation and conditional posterior law.
12. After running the script, look at the image and the posterior standard deviation.

13. Look at the hyper-parameters chains for the noise \mathbf{n} and the image \mathbf{f} .
14. Identify the burn-in period in chains.
15. Compute the posterior mean of hyper-parameters given the chains and compare the value you found with supervised algorithm in point 6 of Sec. 1.
16. Estimate the posterior standard deviation of hyper-parameters from their chains.
17. Try different values for the number of iterations and burn-in period.