Inverse Problems in Imaging
Lecture 11

Simon R. Arridge[1]

[1] Department of Computer Science, University College London, UK

UCL, Term 2

Outline

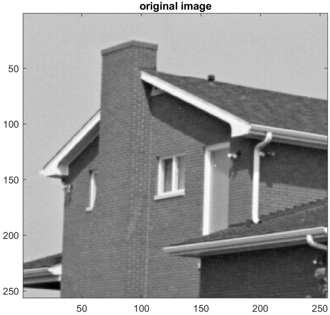
1. Worked Example : In-Painting
   - Inpainting as an inverse problem
   - PDE methods for Inpainting
   - Dictionary Approach to Inpainting

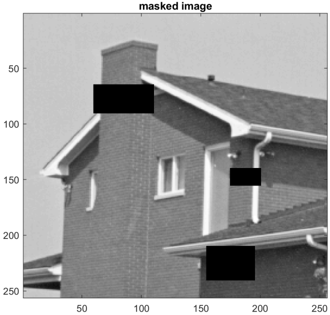
Outline

1. Worked Example : In-Painting
   - Inpainting as an inverse problem
   - PDE methods for Inpainting
   - Dictionary Approach to Inpainting

Image Inpainting
Introduction

Inpainting is defined as the recovery of missing pixels values.





We want to put this into an inverse Problems framework
Define this as

$$\boldsymbol{f}_{\mathrm{recon}} = \arg\min_{\boldsymbol{f}} \left[\Phi(\boldsymbol{f}, \boldsymbol{g}) = \frac{1}{2} \left\| \boldsymbol{g} - \mathrm{I}_{\Omega} \boldsymbol{f} \right\|^2 + \alpha \Psi(\boldsymbol{f}) \right] \tag{11.1}$$

We want to put this into an inverse Problems framework
Define this as

$$\boldsymbol{f}_{\text{recon}} = \arg\min_{\boldsymbol{f}} \left[\Phi(\boldsymbol{f}, \boldsymbol{g}) = \frac{1}{2} \|\boldsymbol{g} - \mathrm{I}_\Omega \boldsymbol{f}\|^2 + \alpha \Psi(\boldsymbol{f}) \right] \tag{11.1}$$

Suppose that the original image has n_{pix} pixel and there are n_{missing} missing pixels.
Let us define Ω as the set of $M = n_{\text{pix}} - n_{\text{missing}}$ pixels that are present and $\overline{\Omega}$ as the set of missing pixels, so that $\Xi = \Omega \cup \overline{\Omega}$ is the set representing the complete image.
Then I_Ω is a $M \times n_{\text{pix}}$ matrix got by deleting rows corresponding to missing pixels from the identity matrix.

Image Inpainting

Forward Model

We want to put this into an inverse Problems framework
Define this as

$$\boldsymbol{f}_{\text{recon}} = \arg\min_{\boldsymbol{f}} \left[\Phi(\boldsymbol{f}, \boldsymbol{g}) = \frac{1}{2} \|\boldsymbol{g} - \mathrm{I}_{\Omega} \boldsymbol{f}\|^2 + \alpha \Psi(\boldsymbol{f}) \right] \tag{11.1}$$

Suppose that the original image has n_{pix} pixel and there are n_{missing} missing pixels.
Let us define Ω as the set of $M = n_{\text{pix}} - n_{\text{missing}}$ pixels that are present and $\overline{\Omega}$ as the set of missing pixels, so that $\Xi = \Omega \cup \overline{\Omega}$ is the set representing the complete image.
Then I_{Ω} is a $M \times n_{\text{pix}}$ matrix got by deleting rows corresponding to missing pixels from the identity matrix.

$$\mathrm{I}_{\Omega} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Example for $n_{\text{pix}} = 6$ and $M = 4$. The missing pixels are $i_{\text{missing}} = [3, 4]$.

We want to put this into an inverse Problems framework
Define this as

$$\boldsymbol{f}_{\mathrm{recon}} = \arg\min_{\boldsymbol{f}} \left[\Phi(\boldsymbol{f}, \boldsymbol{g}) = \frac{1}{2} \|\boldsymbol{g} - \mathrm{I}_{\Omega} \boldsymbol{f}\|^2 + \alpha \Psi(\boldsymbol{f}) \right] \tag{11.1}$$

Suppose that the original image has n_{pix} pixel and there are n_{missing} missing pixels.
Let us define Ω as the set of $M = n_{\mathrm{pix}} - n_{\mathrm{missing}}$ pixels that are present and $\overline{\Omega}$ as the set of missing pixels, so that $\Xi = \Omega \cup \overline{\Omega}$ is the set representing the complete image.
Then I_{Ω} is a $M \times n_{\mathrm{pix}}$ matrix got by deleting rows corresponding to missing pixels from the identity matrix.

$$\mathrm{I}_{\Omega} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Example for $n_{\mathrm{pix}} = 6$ and $M = 4$. The missing pixels are $i_{\mathrm{missing}} = [3, 4]$.
Question : What is the SVD of this matrix ?

Image Inpainting

Inverse Solution

Let's first assume we will use a quadratic prior $\Psi(\boldsymbol{f}) = \frac{1}{2}\boldsymbol{f}^{\mathrm{T}}\Gamma\boldsymbol{f}$
The normal equations for eq. 11.1 become

$$\left(\mathsf{I}_\Omega^{\mathrm{T}}\mathsf{I}_\Omega + \alpha\Gamma\right)\boldsymbol{f} = \mathsf{I}_\Omega^{\mathrm{T}}\boldsymbol{g}$$

The Hessian $\mathsf{I}_\Omega^{\mathrm{T}}\mathsf{I}_\Omega$ is an $n_{\mathrm{pix}} \times n_{\mathrm{pix}}$ identity matrix with zeros on the diagonal corresponding to the missing pixels.

Let's first assume we will use a quadratic prior $\Psi(\boldsymbol{f}) = \frac{1}{2}\boldsymbol{f}^{\mathrm{T}}\Gamma\boldsymbol{f}$
The normal equations for eq. 11.1 become

$$\left(\mathsf{I}_\Omega^{\mathrm{T}}\mathsf{I}_\Omega + \alpha\Gamma\right)\boldsymbol{f} = \mathsf{I}_\Omega^{\mathrm{T}}\boldsymbol{g}$$

The Hessian $\mathsf{I}_\Omega^{\mathrm{T}}\mathsf{I}_\Omega$ is an $n_{\mathrm{pix}} \times n_{\mathrm{pix}}$ identity matrix with zeros on the diagonal corresponding to the missing pixels. For the above example it will be

$$\mathsf{I}_\Omega^{\mathrm{T}}\mathsf{I}_\Omega = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Let's first assume we will use a quadratic prior $\Psi(\boldsymbol{f}) = \frac{1}{2}\boldsymbol{f}^{\mathrm{T}}\Gamma\boldsymbol{f}$
The normal equations for eq. 11.1 become

$$\left(\mathsf{I}_\Omega^{\mathrm{T}}\mathsf{I}_\Omega + \alpha\Gamma\right)\boldsymbol{f} = \mathsf{I}_\Omega^{\mathrm{T}}\boldsymbol{g}$$

The Hessian $\mathsf{I}_\Omega^{\mathrm{T}}\mathsf{I}_\Omega$ is an $n_{\mathrm{pix}} \times n_{\mathrm{pix}}$ identity matrix with zeros on the diagonal corresponding to the missing pixels. For the above example it will be

$$\mathsf{I}_\Omega^{\mathrm{T}}\mathsf{I}_\Omega = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The "backprojection operator" $\mathsf{I}_\Omega^{\mathrm{T}}\boldsymbol{g}$ simply takes the M-length vector $\boldsymbol{g}$ and produces a vector of length n_{pix} by adding zeros at the missing values.

Let's first assume we will use a quadratic prior $\Psi(\boldsymbol{f}) = \frac{1}{2}\boldsymbol{f}^{\mathrm{T}}\Gamma\boldsymbol{f}$
The normal equations for eq. 11.1 become

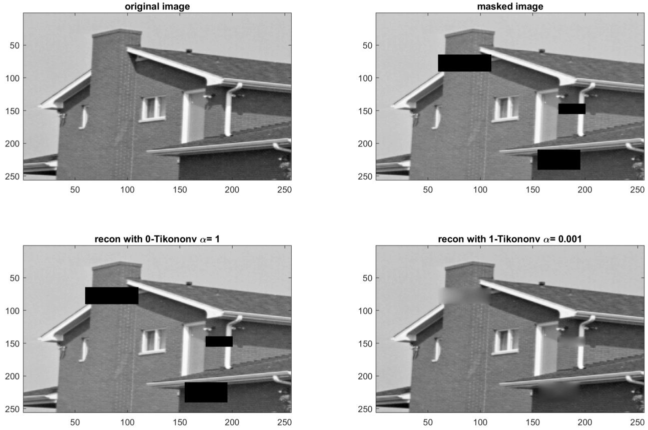
$$\left(\mathsf{I}_\Omega^{\mathrm{T}}\mathsf{I}_\Omega + \alpha\Gamma\right)\boldsymbol{f} = \mathsf{I}_\Omega^{\mathrm{T}}\boldsymbol{g}$$

The Hessian $\mathsf{I}_\Omega^{\mathrm{T}}\mathsf{I}_\Omega$ is an $n_{\mathrm{pix}} \times n_{\mathrm{pix}}$ identity matrix with zeros on the diagonal corresponding to the missing pixels. For the above example it will be

$$\mathsf{I}_\Omega^{\mathrm{T}}\mathsf{I}_\Omega = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The "backprojection operator" $\mathsf{I}_\Omega^{\mathrm{T}}\boldsymbol{g}$ simply takes the M-length vector $\boldsymbol{g}$ and produces a vector of length n_{pix} by adding zeros at the missing values.
Clearly, if Γ is diagonal, it cannot generate any values in the missing pixels!

Results for zero and first order Tikhonov regularisation



Code: `InpaintPrior/inpaint1`

Recall that the 1^{st}-order Tikhonov prior corresponds to $\Gamma = -\nabla^2$.
The normal equations correspond to solving a PDE

$$\left(\mathsf{I}_\Omega^{\mathrm{T}} \mathsf{I}_\Omega - \alpha \nabla^2\right) \boldsymbol{f} = \mathsf{I}_\Omega^{\mathrm{T}} \boldsymbol{g} \tag{11.2}$$

$$\equiv \qquad \nabla^2 \boldsymbol{f} = \frac{1}{\alpha} \boldsymbol{f} \tag{11.3}$$

$$\text{subject to } \boldsymbol{f} = \boldsymbol{g} \in \Omega$$

Recall that the 1^{st}-order Tikhonov prior corresponds to $\Gamma = -\nabla^2$.
The normal equations correspond to solving a PDE

$$\left(\mathsf{I}_\Omega^{\mathrm{T}}\mathsf{I}_\Omega - \alpha\nabla^2\right)\boldsymbol{f} = \mathsf{I}_\Omega^{\mathrm{T}}\boldsymbol{g} \tag{11.2}$$

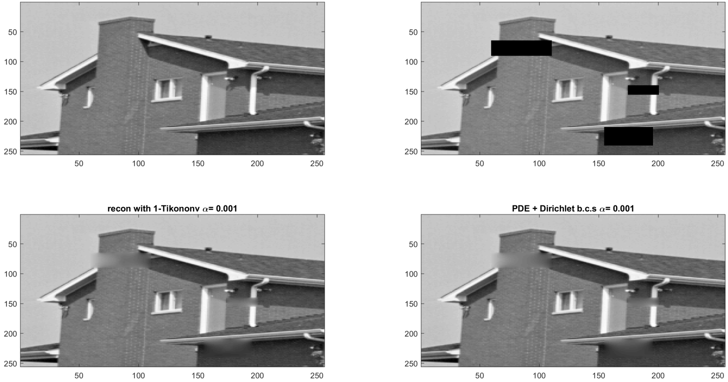
$$\equiv \quad \nabla^2\boldsymbol{f} = \frac{1}{\alpha}\boldsymbol{f} \tag{11.3}$$

$$\text{subject to } \boldsymbol{f} = \boldsymbol{g} \in \Omega$$

Since we know $\boldsymbol{f}$ in Ω, it makes sense to state this problem only for the missing pixels. We can state this as

$$\begin{aligned} \nabla^2\boldsymbol{f} &= \frac{1}{\alpha}\boldsymbol{f}, \qquad \boldsymbol{f} \in \overline{\Omega} \\ \text{subject to } \boldsymbol{f} &= \boldsymbol{g} \qquad \text{on } \partial\Omega \end{aligned} \tag{11.4}$$

This is a diffusion equation with Dirichlet boundary conditions.



Code `InpaintPrior/inpaint1`

The problem in eq. 11.4 is very similar to the denoising problem.

The problem in eq. 11.4 is very similar to the denoising problem.
In that case, we also considered non-linear priors, and solved the PDE as a time-evolution problem. If we apply this here we get

$$\begin{aligned} \frac{\partial \boldsymbol{f}}{\partial t} &= -\alpha \mathcal{L}(\boldsymbol{f})\boldsymbol{f}, \qquad \boldsymbol{f} \in \overline{\Omega} \\ \text{subject to } \boldsymbol{f} &= \boldsymbol{g} \qquad \text{on } \partial\Omega \end{aligned} \tag{11.5}$$

The problem in eq. 11.4 is very similar to the denoising problem.
In that case, we also considered non-linear priors, and solved the PDE as a time-evolution problem. If we apply this here we get

$$\begin{aligned} \frac{\partial \boldsymbol{f}}{\partial t} &= -\alpha \mathcal{L}(\boldsymbol{f})\boldsymbol{f}, \qquad \boldsymbol{f} \in \overline{\Omega} \\ \text{subject to } \boldsymbol{f} &= \boldsymbol{g} \qquad \text{on } \partial\Omega \end{aligned} \tag{11.5}$$

Recall that $\mathcal{L}(\boldsymbol{f})$ becomes a second order differential operator for the regularisers we have considered so far, e.g

$$\mathcal{L}(\boldsymbol{f}) = -\nabla \cdot (\kappa(\boldsymbol{f})\nabla \boldsymbol{f})$$

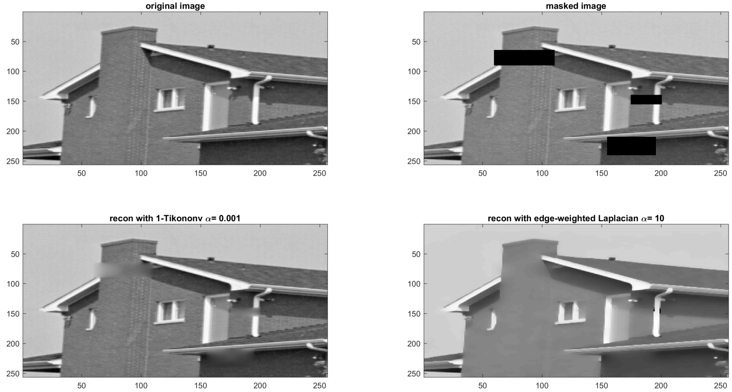
The problem in eq. 11.4 is very similar to the denoising problem.
In that case, we also considered non-linear priors, and solved the PDE as a time-evolution problem. If we apply this here we get

$$\begin{aligned} \frac{\partial \boldsymbol{f}}{\partial t} &= -\alpha \mathcal{L}(\boldsymbol{f})\boldsymbol{f}, \qquad \boldsymbol{f} \in \overline{\Omega} \\ \text{subject to } \boldsymbol{f} &= \boldsymbol{g} \qquad \text{on } \partial\Omega \end{aligned} \tag{11.5}$$

Recall that $\mathcal{L}(\boldsymbol{f})$ becomes a second order differential operator for the regularisers we have considered so far, e.g

$$\mathcal{L}(\boldsymbol{f}) = -\nabla \cdot (\kappa(\boldsymbol{f})\nabla \boldsymbol{f})$$

Other forms of PDE have been considered, e.g. higher order derivatives; see e.g R. D. Adam, P. Peter, J. Weickert: "Denoising by inpainting" *In* F. Lauze, Y. Dong, A. B. Dahl (Eds.): Scale Space and Variational Methods in Computer Vision. Lecture Notes in Computer Science, Vol. 10302, 121-132, Springer, Cham, 2017.



Code `InpaintPrior/inpaint1`

Suppose we have access to a set of example images $\{\boldsymbol{f}_i; i = 1 \ldots n_{\text{samples}}\}$.

Suppose we have access to a set of example images $\{\boldsymbol{f}_i; i = 1 \ldots n_{\mathrm{samples}}\}$.

One simple approach is to assume that the samples are from a Normal distribution $\boldsymbol{f}_i \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{f}}, \mathrm{C}_{\boldsymbol{f}})$ and to take the dictionary as the set of eigenfunctions of $\{\boldsymbol{v}_k; k = 1 \ldots n_v\}$ of $\mathrm{C}_{\boldsymbol{f}}$ up to a chosen cutoff n_v.

Suppose we have access to a set of example images $\{\boldsymbol{f}_i; i = 1 \dots n_{\text{samples}}\}$.

One simple approach is to assume that the samples are from a Normal distribution $\boldsymbol{f}_i \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{f}}, \mathrm{C}_{\boldsymbol{f}})$ and to take the dictionary as the set of eigenfunctions of $\{\boldsymbol{v}_k; k = 1 \dots n_v\}$ of $\mathrm{C}_{\boldsymbol{f}}$ up to a chosen cutoff n_v.

We estimate the mean $\boldsymbol{\mu}_{\boldsymbol{f}}$ and covariance $\mathrm{C}_{\boldsymbol{f}}$ from the sample set

$$\boldsymbol{\mu}_{\boldsymbol{f}} \simeq \frac{1}{n_{\text{samples}}} \sum_{i=1}^{n_{\text{samples}}} \boldsymbol{f}_i, \qquad \mathrm{C}_{\boldsymbol{f}} \simeq \frac{1}{n_{\text{samples}}} \sum_{i=1}^{n_{\text{samples}}} (\boldsymbol{f}_i - \boldsymbol{\mu}_{\boldsymbol{f}})(\boldsymbol{f}_i - \boldsymbol{\mu}_{\boldsymbol{f}})^{\mathrm{T}}$$

A problem dealing with images is that the size of the covariance ($n_{\text{pix}} \times n_{\text{pix}}$) is very large.
Statistics tells us that we need about 10 samples per matrix element to get a reliable estimate of the covariance.
For a $n_{\text{pix}} = 256^2$ image that gives us $n_{\text{samples}} = 2^{33} \simeq 10^{10}$!

A problem dealing with images is that the size of the covariance ($n_{\mathrm{pix}} \times n_{\mathrm{pix}}$) is very large.
Statistics tells us that we need about 10 samples per matrix element to get a reliable estimate of the covariance.
For a $n_{\mathrm{pix}} = 256^2$ image that gives us $n_{\mathrm{samples}} = 2^{33} \simeq 10^{10}$!
Fortunately, we assume that images are correlated and so $\mathrm{C}_{\boldsymbol{f}}$ is low-rank, so it can be approximated by its *principle components*

$$\mathrm{C}_{\boldsymbol{f}} \simeq \tilde{\mathrm{C}}_{\boldsymbol{f}} = \sum_{k=1}^{n_v} \sigma_k^2 \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{T}}$$

A problem dealing with images is that the size of the covariance ($n_{\rm pix} \times n_{\rm pix}$) is very large.
Statistics tells us that we need about 10 samples per matrix element to get a reliable estimate of the covariance.
For a $n_{\rm pix} = 256^2$ image that gives us $n_{\rm samples} = 2^{33} \simeq 10^{10}$!
Fortunately, we assume that images are correlated and so $\mathsf{C}_{\boldsymbol{f}}$ is low-rank, so it can be approximated by its *principle components*

$$\mathsf{C}_{\boldsymbol{f}} \simeq \tilde{\mathsf{C}}_{\boldsymbol{f}} = \sum_{k=1}^{n_v} \sigma_k^2 \boldsymbol{v}_k \boldsymbol{v}_k^{\rm T}$$

We can find the components by taking the SVD (to limited order) of the $n_{\rm pix} \times n_{\rm samples}$ data matrix $\tilde{\mathsf{F}} = \mathsf{F} - \boldsymbol{\mu}_{\boldsymbol{f}} \mathbf{1}^{\rm T}$ where the $i^{\rm th}$ column of F is $\boldsymbol{f}_i$ and $\mathbf{1}$ is a vector of 1's of length $n_{\rm samples}$.

The problem to solve now is

$$\boldsymbol{f}_{\mathrm{recon}} = \arg\min_{\boldsymbol{f}} \left[\Phi(\boldsymbol{f}, \boldsymbol{g}) = \frac{1}{2} \left\| \boldsymbol{g} - \mathsf{I}_{\Omega} \boldsymbol{f} \right\|^2 + \alpha \frac{1}{2} (\boldsymbol{f} - \boldsymbol{\mu}_{\boldsymbol{f}}) \tilde{\mathsf{C}}_{\boldsymbol{f}}^{-1} (\boldsymbol{f} - \boldsymbol{\mu}_{\boldsymbol{f}})^{\mathrm{T}} \right] \quad (11.6)$$

which leads to the normal equations

$$\left(\mathsf{I}_{\Omega}^{\mathrm{T}} \mathsf{I}_{\Omega} + \alpha \tilde{\mathsf{C}}_{\boldsymbol{f}}^{-1} \right) \boldsymbol{f} = \mathsf{I}_{\Omega}^{\mathrm{T}} \boldsymbol{g} - \alpha \tilde{\mathsf{C}}_{\boldsymbol{f}}^{-1} \boldsymbol{\mu}_{\boldsymbol{f}}$$

The problem to solve now is

$$\boldsymbol{f}_{\rm recon} = \arg\min_{\boldsymbol{f}} \left[\Phi(\boldsymbol{f}, \boldsymbol{g}) = \frac{1}{2} \left\| \boldsymbol{g} - \mathsf{I}_\Omega \boldsymbol{f} \right\|^2 + \alpha \frac{1}{2} (\boldsymbol{f} - \boldsymbol{\mu}_{\boldsymbol{f}}) \tilde{\mathsf{C}}_{\boldsymbol{f}}^{-1} (\boldsymbol{f} - \boldsymbol{\mu}_{\boldsymbol{f}})^{\rm T} \right] \quad (11.6)$$

which leads to the normal equations

$$\left(\mathsf{I}_\Omega^{\rm T} \mathsf{I}_\Omega + \alpha \tilde{\mathsf{C}}_{\boldsymbol{f}}^{-1} \right) \boldsymbol{f} = \mathsf{I}_\Omega^{\rm T} \boldsymbol{g} - \alpha \tilde{\mathsf{C}}_{\boldsymbol{f}}^{-1} \boldsymbol{\mu}_{\boldsymbol{f}}$$

Since $\tilde{\mathsf{C}}_{\boldsymbol{f}}^{-1}$ is low rank (by design) we replace it's inverse by the pseudo-inverse

$$\tilde{\mathsf{C}}_{\boldsymbol{f}}^{-1} \to \tilde{\mathsf{C}}_{\boldsymbol{f}}^{\dagger} = \sum_{k=1}^{n_v} \frac{1}{\sigma_k^2} \boldsymbol{v}_k \boldsymbol{v}_k^{\rm T}$$

The problem to solve now is

$$\boldsymbol{f}_{\mathrm{recon}} = \arg\min_{\boldsymbol{f}} \left[\Phi(\boldsymbol{f}, \boldsymbol{g}) = \frac{1}{2} \|\boldsymbol{g} - \mathsf{I}_\Omega \boldsymbol{f}\|^2 + \alpha \frac{1}{2} (\boldsymbol{f} - \boldsymbol{\mu}_{\boldsymbol{f}}) \tilde{\mathsf{C}}_{\boldsymbol{f}}^{-1} (\boldsymbol{f} - \boldsymbol{\mu}_{\boldsymbol{f}})^{\mathrm{T}} \right] \quad (11.6)$$

which leads to the normal equations

$$\left(\mathsf{I}_\Omega^{\mathrm{T}} \mathsf{I}_\Omega + \alpha \tilde{\mathsf{C}}_{\boldsymbol{f}}^{-1} \right) \boldsymbol{f} = \mathsf{I}_\Omega^{\mathrm{T}} \boldsymbol{g} - \alpha \tilde{\mathsf{C}}_{\boldsymbol{f}}^{-1} \boldsymbol{\mu}_{\boldsymbol{f}}$$

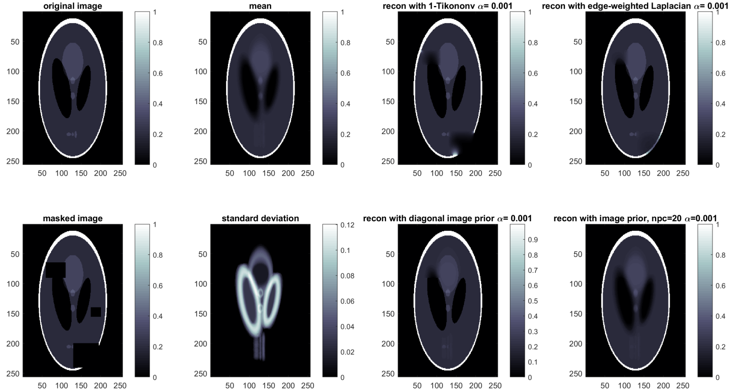
Since $\tilde{\mathsf{C}}_{\boldsymbol{f}}^{-1}$ is low rank (by design) we replace it's inverse by the pseudo-inverse

$$\tilde{\mathsf{C}}_{\boldsymbol{f}}^{-1} \to \tilde{\mathsf{C}}_{\boldsymbol{f}}^{\dagger} = \sum_{k=1}^{n_v} \frac{1}{\sigma_k^2} \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{T}}$$

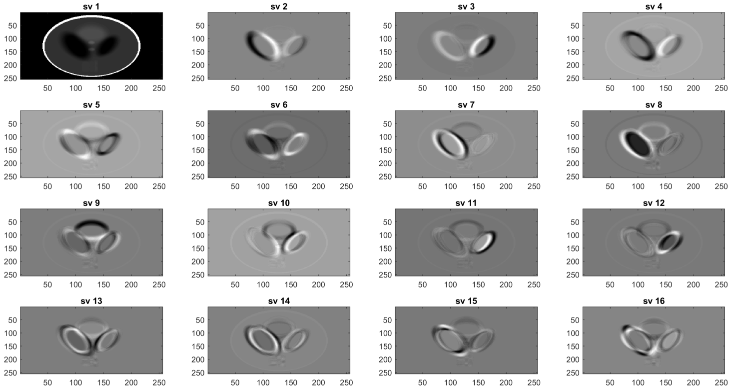
We can now solve the normal equations using a matrix-free solved like `pcg` where the Krylov iterations require only the masking operation $\mathsf{I}_\Omega^{\mathrm{T}} \mathsf{I}_\Omega$, and the projection onto the singular vectors (i.e. the dictionary) up to the order n_v.

Image Inpainting

Covariance Dictionary results



Code: `InpaintPrior/InpaintSL`



Code: `InpaintPrior/InpaintSL`