Ψηφιακή Επεξεργασία Εικόνας (ΨΕΕ) – ΜΥΕ037

Εαρινό εξάμηνο 2023-2024

Image Restoration and Reconstruction (Linear Restoration Methods)

Άγγελος Γιώτης

a.giotis@uoi.gr

Contents

In this lecture we will look at linear image restoration techniques

- Differentiation of matrices and vectors
- Linear space invariant degradation
- Restoration in absence of noise
  - Inverse filter
  - Pseudo-inverse filter
- Restoration in presence of noise
  - Inverse filter
  - Wiener filter
  - Constrained least squares filter

Differentiation of Matrices and Vectors

Notation:

$\mathbf{A}$ is a MxN matrix with elements a_{ij}.

$\mathbf{x}$ is a Nx1 vector with elements x_i.

f($\mathbf{x}$) is a scalar function of vector $\mathbf{x}$.

$\mathbf{g}(\mathbf{x})$ is a Mx1 vector valued function of vector $\mathbf{x}$.

Differentiation of Matrices and Vectors (cont...)

Scalar derivative of a matrix.

$\mathbf{A}$ is a MxN matrix with elements a_{ij}.

$$\frac{\partial \mathbf{A}}{\partial t} = \begin{pmatrix} \frac{\partial a_{11}}{\partial t} & \cdots & \frac{\partial a_{1N}}{\partial t} \\ \vdots & \ddots & \vdots \\ \frac{\partial a_{M1}}{\partial t} & \cdots & \frac{\partial a_{MN}}{\partial t} \end{pmatrix}$$

Differentiation of Matrices and Vectors (cont...)

Vector derivative of a function (gradient).

$\mathbf{x}$ is a Nx1 vector with elements x_i.

$\mathrm{f}(\mathbf{x})$ is a scalar function of vector $\mathbf{x}$.

$$\frac{\partial f}{\partial \mathbf{x}} = \nabla_{\mathbf{x}} f = \begin{pmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_N} \end{pmatrix}^T$$

Differentiation of Matrices and Vectors (cont...)

Vector derivative of a vector (Jacobian):

$\mathbf{x}$ is a Nx1 vector with elements x_i.

$\mathbf{g}(\mathbf{x})$ is a Mx1 vector valued function of vector $\mathbf{x}$.

$$\frac{\partial \mathbf{g}}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_M}{\partial x_1} & \cdots & \frac{\partial g_M}{\partial x_N} \end{pmatrix}$$

Differentiation of Matrices and Vectors (cont...)

Some useful derivatives.

$\mathbf{x}$ is a Nx1 vector with elements x_i.

$\mathbf{b}$ is a Nx1 vector with elements b_i.

$$\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{b}^T \mathbf{x}\right) = \mathbf{b}$$

It is the derivative of the scalar valued function $\mathbf{b}^T\mathbf{x}$ with respect to vector $\mathbf{x}$.

Differentiation of Matrices and Vectors (cont...)

Some useful derivatives.

$\mathbf{x}$ is a Nx1 vector with elements x_i.

$\mathbf{A}$ is a NxN matrix with elements a_{ij}.

$$\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{x}^T \mathbf{A} \mathbf{x}\right) = \left(\mathbf{A} + \mathbf{A}^T\right)\mathbf{x}$$

If A is symmetric:

$$\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{x}^T \mathbf{A} \mathbf{x}\right) = 2\mathbf{A}\mathbf{x}$$

Differentiation of Matrices and Vectors (cont...)

Some useful derivatives.

$\mathbf{x}$ is a Nx1 vector with elements x_i.

$\mathbf{b}$ is a Mx1 vector with elements b_i.

$\mathbf{A}$ is a MxN matrix with elements a_{ij}.

$$\frac{\partial}{\partial \mathbf{x}} \left\| \mathbf{Ax} + \mathbf{b} \right\|^2 = 2\mathbf{A}^T \left(\mathbf{Ax} + \mathbf{b} \right)$$

It may be proved using the previous properties.

Differentiation of Matrices and Vectors (cont...)

$$\frac{\partial}{\partial \mathbf{x}} \left\| \mathbf{A}\mathbf{x} + \mathbf{b} \right\|^2 = 2\mathbf{A}^T \left(\mathbf{A}\mathbf{x} + \mathbf{b} \right)$$

$$\boldsymbol{f}(\boldsymbol{x}) = \|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}\|^2 = (\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})^T(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})$$
$$= (\boldsymbol{A}\boldsymbol{x})^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{A}\boldsymbol{x} + (\boldsymbol{A}\boldsymbol{x})^T\boldsymbol{b} + \boldsymbol{b}^T\boldsymbol{b}$$

Since $(\boldsymbol{A}\boldsymbol{x})^T = \boldsymbol{x}^T\boldsymbol{A}^T$ and scalar quantities are equal to their transpose:

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} + 2\boldsymbol{b}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{b},$$

where $\boldsymbol{b}^T\boldsymbol{b}$ is constant with respect to $\boldsymbol{x}$, so its derivative will be zero.

Differentiation of Matrices and Vectors (cont...)

- Using the result from matrix calculus for $\boldsymbol{x}^T\boldsymbol{C}\boldsymbol{x}$ where $\boldsymbol{C} = \boldsymbol{A}^T\boldsymbol{A}$ is symmetric:

$$\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^T A^T A\mathbf{x}) = 2A^T A\mathbf{x}$$

- Since $\boldsymbol{b}^T\boldsymbol{A}\boldsymbol{x}$ is a scalar, we use the result for the derivative of a linear form $\boldsymbol{c}^T\boldsymbol{x}$ (where $\boldsymbol{c}^T$ is a linear combination of the rows of $\boldsymbol{A}$, defined by $\boldsymbol{b}^T$, i.e., a row vector):

$$\frac{\partial}{\partial \mathbf{x}}(2\mathbf{b}^T A\mathbf{x}) = 2A^T\mathbf{b}.$$

- Combining the derivatives:

$$\frac{\partial f}{\partial \mathbf{x}} = \frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^T A^T A\mathbf{x}) + \frac{\partial}{\partial \mathbf{x}}(2\mathbf{b}^T A\mathbf{x})$$

$$\frac{\partial f}{\partial \mathbf{x}} = 2A^T A\mathbf{x} + 2A^T\mathbf{b}$$

Yields:

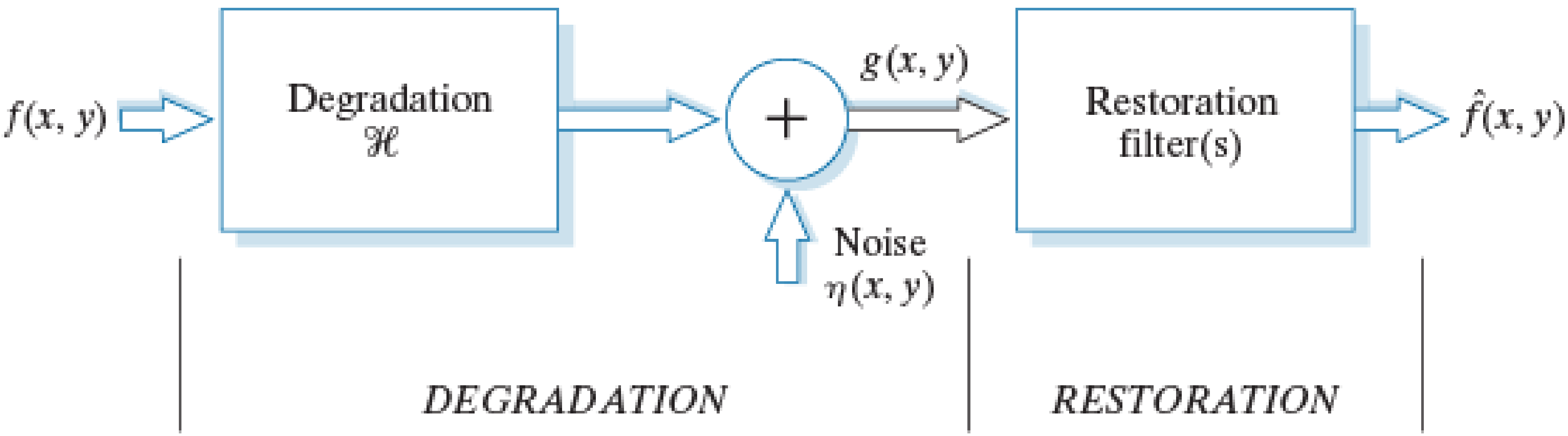
$$\frac{\partial}{\partial \mathbf{x}}\|\mathbf{Ax}+\mathbf{b}\|^2 = 2\mathbf{A}^T(\mathbf{Ax}+\mathbf{b})$$

So far: Standard Additive noise model

- Until now: we considered a noisy image to be modelled as follows:

$$g(x,y) = f(x,y) + \eta(x,y)$$

under the "**white Gaussian noise**" assumption.



Linear, Position-Invariant Degradation

We now consider a degraded image to be modelled by:

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$$

where $h(x, y)$ is the impulse response of the degradation function (i.e. *point spread function* blurring the image).

Linear, Position-Invariant Degradation

$$g(x,y) = h(x,y) * f(x,y) + \eta(x,y)$$

The convolution implies that the degradation mechanism is linear and position invariant (it depends only on image values and not on location).

An operator $\mathcal{H}$ having the input-output relationship $g(x,y) = \mathcal{H}[f(x,y)]$

is said to be *position* (or *space*) *invariant* if

$$\mathcal{H}[f(x-\alpha, y-\beta)] = g(x-\alpha, y-\beta)$$

Linear, Position-Invariant Degradation

Example degraded images-observations $g(x, y)$



Linear, Position-Invariant Degradation (cont...)

In the Fourier domain:

$$G(k,l) = H(k,l)F(k,l) + N(k,l)$$

where multiplication is element-wise.

In matrix-vector form:

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\eta}$$

where $\mathbf{H}$ is a doubly block circulant matrix and $\mathbf{f}$, $\mathbf{g}$, and $\boldsymbol{\eta}$ are vectors (lexicographic ordering).

Linear, Position-Invariant Degradation (cont...)

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$$

$$G(k, l) = H(k, l)F(k, l) + N(k, l)$$

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\eta}$$

If the degradation function is unknown the problem of simultaneously recovering $f(x,y)$ and $h(x,y)$ is called *blind deconvolution*.

Estimating the point spread function

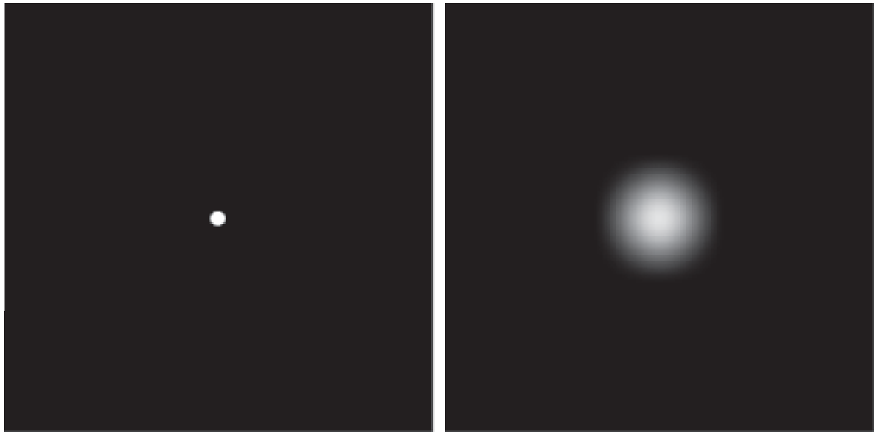
- In what follows, we consider that the degradation function is known.
- If the psf is not known, some **basic** methods to estimate it are:
  - By observation
    - Apply sharpening filters to a sub-image $g_s(m,n)$ where the signal is strong (there is almost no noise) and obtain a visually pleasant result $f_s(m,n)$. The psf may be approximated by $H_s(k,l)= G_s(k,l)/ F_s(k,l)$.
    - The task needs trial and error and may be tedious.
    - Used in special circumstances (e.g. Restoration of old photographs)

Estimating the point spread function (cont.)

- If the psf is not known:
  - By experimentation
    - If the acquisition equipment or a similar one is available an image similar to the degraded may be obtained by varying the system settings.
    - Then obtain the image of an impulse (small dot of light) using the same settings.
    - We estimate $H(k,l) = G(k,l)/A$ (for constant A)

a b

FIGURE 5.24 Estimating a degradation by impulse characterization. (a) An impulse of light (shown magnified). (b) Imaged (degraded) impulse.



Estimating the point spread function (cont.)

- If the psf is not known, some **basic** methods to estimate it are:
  - By **modeling**: **atmospheric turbulence:**

$$H(u,v) = \exp\left(-k\left(u^2+v^2\right)^{5/6}\right)$$

a b
c d

FIGURE 5.25 Modeling turbulence. (a) No visible turbulence. (b) Severe turbulence, $k = 0.0025$. (c) Mild turbulence, $k = 0.001$. (d) Low turbulence, $k = 0.00025$. All images are of size 480×480 pixels. (Original image courtesy of NASA.)

Estimating the point spread function (cont.)

- By modeling: **planar motion**
  - $x_0(t)$ and $y_0(t)$ are the time varying components of motion at each pixel.
  - The total exposure at any pixel is obtained by integrating the instantaneous exposure over the time the shutter is open.
  - Assumption: the shutter opening and closing is instantaneous.
  - If T is the duration of the exposure, the recorded image is expressed by:

$$g(x,y)=\int_0^T f[x-x_0(t),y-y_0(t)]dt$$

Estimating the point spread function (cont.)

$$
\begin{aligned}
G(u,v) &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g(x,y)e^{-j2\pi(ux+vy)}dxdy \\
&= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\left[\int_{0}^{T} f[x-x_0(t),y-y_0(t)]dt\right]e^{-j2\pi(ux+vy)}dxdy \\
&= \int_{0}^{T}\left[\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f[x-x_0(t),y-y_0(t)]e^{-j2\pi(ux+vy)}dxdy\right]dt \\
&= \int_{0}^{T} F(u,v)e^{-j2\pi[ux_0(t)+vy_0(t)]}dt \\
&= F(u,v)\int_{0}^{T} e^{-j2\pi[ux_0(t)+vy_0(t)]}dt \Leftrightarrow \\
H(u,v) &= \int_{0}^{T} e^{-j2\pi[ux_0(t)+vy_0(t)]}dt
\end{aligned}
$$

Estimating the point spread function (cont.)

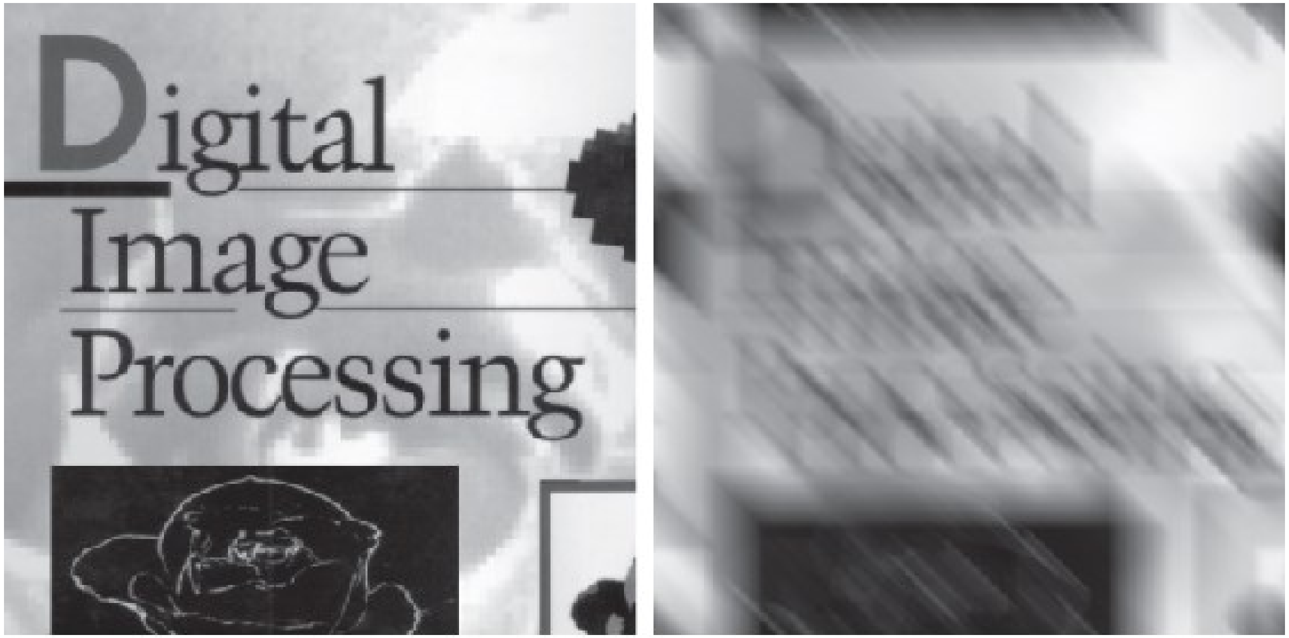
- Considering uniform linear motion:

$$x_0(t) = a\frac{t}{T},\ y_0(t) = b\frac{t}{T}$$

- The psf becomes:

$$\begin{aligned} H(u,v) &= \int_0^T e^{-j2\pi(uat+vbt)/T}\,dt \\ &= \frac{T}{\pi(ua+vb)}\sin\left[\pi(ua+vb)\right]e^{-j\pi(ua+vb)} \end{aligned}$$

Estimating the point spread function (cont.)



- Result of blurring with:

$$x_0(t) = a\frac{t}{T},\ y_0(t) = b\frac{t}{T},\ \ a = b = 0.1,\ T = 1$$

Linear Restoration

Using the imaging system

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\eta}$$

we want to estimate the true image from the degraded observation with **known** degradation $\mathbf{H}$.

A linear method applies an operator (a matrix) $\mathbf{P}$ to the observation $\mathbf{g}$ to estimate the unobserved noise-free image $\mathbf{f}$:

$$\hat{\mathbf{f}} = \mathbf{P}\mathbf{g}$$

Restoration in Absence of Noise
The Inverse Filter

When there is no noise:

$$\mathbf{g} = \mathbf{H}\mathbf{f}$$

an obvious solution would be to use the **inverse filter**:

$$\mathbf{P} = \mathbf{H}^{-1}$$

yielding

$$\hat{\mathbf{f}} = \mathbf{P}\mathbf{g} = \mathbf{H}^{-1}\mathbf{g} = \mathbf{H}^{-1}\mathbf{H}\mathbf{f} = \mathbf{f}$$

Restoration in Absence of Noise
The Inverse Filter (cont...)

$$\hat{\mathbf{f}} = \mathbf{H}^{-1}\mathbf{g}$$

For a NxN image, $\mathbf{H}$ is a N^2xN^2 matrix!

To tackle the problem we transform it to the Fourier domain.

$\mathbf{H}$ is doubly block circulant and therefore it may be diagonalized by the 2D DFT matrix $\mathbf{W}$:

$$\mathbf{H} = \mathbf{W}^{-1}\mathbf{\Lambda}\mathbf{W}$$

Restoration in Absence of Noise
The Inverse Filter (cont...)

$$\mathbf{H} = \mathbf{W}^{-1}\mathbf{\Lambda}\mathbf{W}$$

where

$$\mathbf{\Lambda} = diag\{H(1,1),\ldots,H(N,1),H(1,2),\ldots H(N,N)\}$$

Therefore:

$$\hat{\mathbf{f}} = \mathbf{P}\mathbf{g} \Leftrightarrow \hat{\mathbf{f}} = \mathbf{H}^{-1}\mathbf{g} \Leftrightarrow \hat{\mathbf{f}} = (\mathbf{W}^{-1}\mathbf{\Lambda}\mathbf{W})^{-1}\mathbf{g}$$

$$\Leftrightarrow \hat{\mathbf{f}} = \mathbf{W}^{-1}\mathbf{\Lambda}^{-1}\mathbf{W}\mathbf{g} \Leftrightarrow \mathbf{W}\hat{\mathbf{f}} = \mathbf{W}\mathbf{W}^{-1}\mathbf{\Lambda}^{-1}\mathbf{W}\mathbf{g}$$

$$\Leftrightarrow \hat{\mathbf{F}} = \mathbf{\Lambda}^{-1}\mathbf{G}$$

Restoration in Absence of Noise The Inverse Filter (cont...)

This is the vectorized form of the DFT of the image:

$$\hat{\mathbf{F}} = \mathbf{\Lambda}^{-1}\mathbf{G} \quad \Leftrightarrow \hat{F}(k,l) = \frac{G(k,l)}{H(k,l)}$$

Take the inverse DFT and obtain $f(m,n)$.

Problem: what happens if $H(k,l)$ has zero values?

Cannot perform inverse filtering!

Restoration in Absence of Noise The Pseudo-inverse Filter

A solution is to set:

$$\hat{F}(k,l)=\begin{cases}\dfrac{G(k,l)}{H(k,l)} & , \quad H(k,l)\neq 0\\[2ex] 0 & , \quad H(k,l)=0\end{cases}$$

which is a type of pseudo-inversion.

Notice that the signal cannot be restored at locations where $H(k,l)$=0.

Restoration in Absence of Noise The Pseudo-inverse Filter (cont...)

A pseudo-inverse filter also arises by the unconstrained least squares approach.

Find the image $\mathbf{f}$, that, when it is blurred by $\mathbf{H}$, it will provide an observation as close as possible to $\mathbf{g}$, i.e. It minimizes the distance between $\mathbf{Hf}$ and $\mathbf{g}$.

Restoration in Absence of Noise
The Pseudo-inverse Filter (cont...)

This distance is expressed by the norm:

$$J(\mathbf{f}) = \left\| \mathbf{Hf} - \mathbf{g} \right\|^2$$

$$\min_{\mathbf{f}} \{ J(\mathbf{f}) \} \Leftrightarrow \frac{\partial J}{\partial \mathbf{f}} = 0 \ \Leftrightarrow \frac{\partial}{\partial \mathbf{f}} \left(\left\| \mathbf{Hf} - \mathbf{g} \right\|^2 \right) = 0$$

$$\Leftrightarrow 2\mathbf{H}^T \left(\mathbf{Hf} - \mathbf{g} \right) = 0 \Leftrightarrow 2\mathbf{H}^T \mathbf{Hf} = 2\mathbf{H}^T \mathbf{g}$$

$$\Leftrightarrow \mathbf{f} = \left(\mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{H}^T \mathbf{g}$$