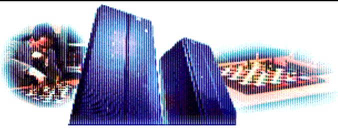
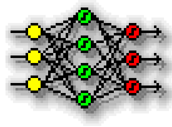




Sistemi Intelligenti
Stima MAP

Alberto Borghese

Università degli Studi di Milano
Laboratory of Applied Intelligent Systems (AIS-Lab)
Dipartimento di Informatica
alberto.borghese@unimi.it

Overview

Filtering images

MAP and Regularization

Poisson noise

A-priori and Markov Random Fields

Cost function minimization

Teorema di Bayes

P(X,Y) = P (Y|X)P(X) = P(X|Y)P(Y)

$$P(X|Y) = \boxed{\frac{P(Y\mid X)P(X)}{P(Y)}}$$

X = causa Y = effetto

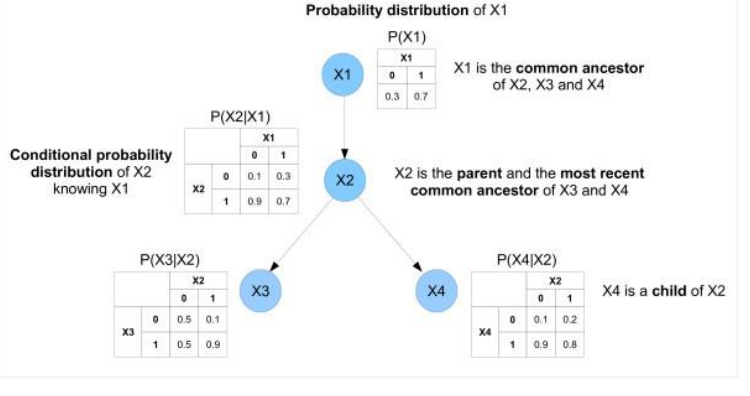
$$P(\text{causa}|\text{effetto}) = \frac{P(Effetto \mid Causa)\,P(Causa)}{P(Effetto\,)}$$

We usually do not know the statistics of the cause, but we can measure the effect and , through frequency, build the statistics of the effect or we know it in advance.

A doctor knows P(Symptons|Causa) and wants to determine P(Causa|Symptoms)

Graphical models

A **graphical model** o **modello probabilistico su grafo (PGM)** è un modello probabilistico che evidenzia le dipendenze tra le variabili randomiche (può evolvere eventualmente in un albero). Viene utilizzato nell'inferenza statistica.



Variabili continue

Caso discreto: prescrizione della probabilità per ognuno dei finiti valori che la variabile X può assumere: P(X).

Caso continuo: i valori che X può assumere sono infiniti. Devo trovare un modo per definirne la probabilità. Descrizione **analitica** mediante la funzione densità di probabilità.

Valgono le stesse relazioni del caso discreto, dove alla somma si sostituisce l'integrale.

$$P(X = x \in [\bar{x}, \bar{x} + \Delta x]) \int_{\bar{x}}^{\bar{x}+\Delta x} \int_{-\infty}^{+\infty} p(x,y)dxdy$$

$$p(x,y) = p(y|x)\, p(x) = p(x\,|\,y)\, p(y)$$

Teorema di Bayes

$$p(x\,|\,y) = \frac{p(y|x)\, p(x)}{p(y)}$$

Problema Inverso

x = causa
y = effetto

Images are corrupted by noise…

i) When measurement of some physical parameter is performed, noise corruption cannot be avoided.

ii) Each pixel of a digital image measures a number of photons.

Therefore, from i) and ii)…

…Images are corrupted by noise!

How to go from noisy image to the true one? It is an inverse problem.

A general framework

- $x = \{x_1, x_2, \dots, x_M\}$, $x_k \in R^M$ e.g. Pixel true luminance
- $y = \{y_1, y_2, \dots, y_M\}$ $y_k \in R^N$ e.g. Pixel measured luminance

- $y = A\,x + h + n$ -> determining x is a **deblurring problem** (the measuring device transforms the image: scale + offset)

x, y are vectors. The image is represented as a vector with columns (rows) one after the other.

Role of A:

- Diagonal elements take into account that the measured signal is the original signal, scaled.
- Off-diagonal elements take into account the effect of the measure of one pixel to the measure of other pixels (blurring). *Blurring can be introduced by lenses/sensor in photographic / video images.* ***Point Spread Function of the sensor.***

Role of h: offset: background radiation.

Role of n: measurement noise.

It is a general framework for measurement systems. It is a linear framework.

Filtering (denoising)

- $x = \{x_1, x_2, \dots, x_M\}$, $x_k \in R^M$ e.g. Pixel true luminance
- $y = \{y_1, y_2, \dots, y_M\}$ $y_k \in R^N$ e.g. Pixel measured luminance

- $y = I\,x + n$ -> determining x is a **denoising problem** (the measuring device introduces only measurment error)

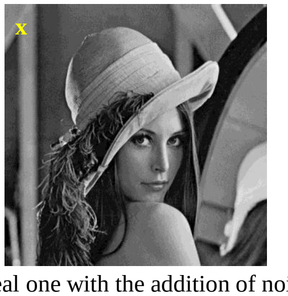
Role of I:

- Identity matrix. Reproduces the input image, x, in the output y.

Role of h: offset: background radiation has been compensated by calibration.

Role of n: measurement noise.

- $y = I\,x + n$



Determining x is a denoising problem (image is a copy of the real one with the addition of noise)

Gaussian noise and likelihood

- Images are composed by a set of pixels, **x** (**x** is a vector!)
- Let us assume that the noise is Gaussian and that its mean and variance is equal for all pixels;
- Let $y_{\mathbf{n},i}$ be the measured value for the i-th pixel (n = noise);
- Let and x_i be the true (noiseless) value for the i-th pixel;

- How can we quantify the probability to measure the image **x**, given the probability density function for each pixel?
- Being the pixels independent, the total probability can be written in terms of product of independent conditional probabilities (conditional likelihood function) $L(\mathbf{y_n} \mid \mathbf{x})$:

$$L(\mathbf{y_n} \mid \mathbf{x}) = \prod_{i=1}^{N} n_i = \prod_{i=1}^{N} p(y_{n,i} \mid x_i) = \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y_{n,i} - x_i}{\sigma} \right)^2 \right]$$

- $L(\mathbf{y_n} \mid \mathbf{x})$ describes the probability to measure the image $\mathbf{y_n}$, given the noise free value for each pixel, **x**. But we do not know these values….

Do we get anywhere?

L is the likelihood function of Y_n, given the object X.

$$L(y_n \mid x) = \prod_{i=1}^{N} p(y_{n,i} \mid x_i)$$

Determine $\{x_i\}$ such that L(.) is maximized. Negative log-likelihood is usually considered to deal with sums: $f(.) = -\log(L(.)) = -\sum_{i=1}^{N} \ln\left(p(y_{n,i} \mid x_i)\right)$

$$f(y_{n,1}, y_{n,2} .... y_{n,N}; x_{n,1}, x_{n,2} .... x_{n,N}; 0, \sigma^2) = -\sum_{i=1}^{N} \ln\left\{ \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left[-\frac{1}{2}\left(\frac{y_{n,i} - x_i}{\sigma} \right)^2 \right] \right\} =>$$

$$\min_{\{x_i\}}(f(.)) = \min\left\{ -\sum_{i=1}^{N} \ln\left(\frac{1}{\sqrt{2\pi}\sigma} \right)_{\{f_i\}} + \frac{1}{2\sigma^2}(y_n - Ax)^2 \right\}$$

$x = (A^T A)^{-1} A^T y_n$

if A = I

$x = y_n$

The system has a single solution, that is good. The solution is $x_i = y_{n,i}$, not a great result….

Can we do any better?

A better approach

$$L(y_n \mid x) = \prod_{i=1}^{N} p(y_{n,i} \mid x_i)$$

We have N pixels, for each pixel we get one measurement.

Let us analyze the probability for each pixel: $p(y_{n,i} \mid x_i)$. If we have more measurements for each pixel, we can write:

$$p(y_{n,i,1}; p_{n,i,2}; p_{n,i,3}; \ldots\ldots p_{n,i,M} \mid x_i) = \prod_{k=1}^{M} p(y_{n,k,i} \mid x_i)$$

If noise is independent, Gaussian, zero mean, the best estimate of x_i is the samples average, this converges through the distribution mean of the measurement taken in the position i.

But, what happens if we do not have such multiple samples or a few samples?

Overview

Filtering images

MAP and Regularization

Poisson noise

A-priori and Markov Random Fields

Cost function minimization

The Bayesian framework

We assume that the object x is a realization of the "abstract" object X that can be characterized statistically as a density probability on X. x is extracted randomly from X (a bit Platonic).

The probability $p(y_n| x)$ becomes a conditional probability: $J_0 = p(y_n| x = x^*)$

Under this condition, the probability of observing y_n can be written as the joint probability of observing both y_n and x. This is equal to the product of the conditional probability $p(y_n | x)$ by a-priori probability on x, p_x:

$$p(y_n, x) = p(y_n | x)\, p(x)$$

As we are interested in determining x, inverse problem, we have to write the conditional probability of x, having observed (measured) y_n : $p(x | y_n)$. We apply Bayes theorem:

$$p(x | y_n) = \frac{p(y_n | x)\, p(x)}{p(y_n)} = J_0(y_n | x) \frac{p(x)}{p(y_n)}$$

A-priori types

- Any statistical information on the distribution of x.
- It can be the structure defined in terms of variations (gradients)
- It can be the amplitude of the signal defined in terms of power.
- It can be a morphable model
- …..

MAP estimate with logarithms

$$p(x \mid y_n) = \frac{p(y_n \mid x)\, p(x)}{p(y_n)} = L(y_n \mid x) \frac{p(x)}{p(y_n)}$$

Logarithms help:

$$-\ln\left(p(x \mid y_n)\right) = -\left\{ \frac{\ln\left(p(y_n \mid x)\, p(x)\right)}{p(y_n)} \right\} = -\left\{ \ln\left(p(y_n \mid x)\right) + \ln\left(p(x)\right) - \ln\left(p(y_n)\right) \right\}$$

We maximize the MAP of x | y_n, by minimizing:

$$\arg\min_x -\left\{ \ln\left(\frac{p(y_n \mid x)\, p(x)}{p(y_n)} \right) \right\} = \arg\min_x -\left\{ \ln\left(p(y_n \mid x)\right) + \ln\left(p(x)\right) - \cancel{\ln\left(p(y_n)\right)} \right\}$$

We explicitly observe that the marginal distribution of y_n is not dependent on x. It does not affect the minimization and it can be neglected. It represents the statistical distribution of the measurements alone.

MAP estimate with logarithms

We maximize the MAP of x | y_n, by minimizing:

$$\arg\min_x -\left\{ \ln\left(p(y_n \mid x)\, p(x)\right) \right\} = \arg\min_x -\left\{ \ln\left(p(y_n \mid x)\right) + \ln\left(p(x)\right) \right\}$$

$J_0\left(y_{n,i} \mid x\right)$

Likelihood =
adherence to the data

A-priori
$J_R(x)$

Depending on the shape of the noise (inside the likelihood) and the a-priori distribution of x(.): $J_R(x)$, we get different solutions.

Gaussian noise on samples

$$x = \arg\min_x -\{\ln(p(y_n \mid x)p(x))\} = \arg\min_x -\{\ln(p(y_n \mid x)) + \ln(p(x))\} =$$

$$\arg\min_x \{J_0(y_n \mid x) + J_R(x)\} =$$

Gaussian noise on the data
Zero mean
Pixels are independent
All measurements have the same variance, σ^2

$$-\log(p(y_n \mid x)) = \quad J_0(y_n \mid x) = \cos\tan te + \left(\frac{1}{\sigma^2}\right)\left(\sum_i \left\| y_{n,i} - Ax_i \right\|^2\right)$$

Mean squared error

What about $J_R(x) = -\log(p(x))$?

Gibb's priors

We often define the a-priori term, $J_R(x)$, as Gibb's prior:

$$p_x = \frac{1}{Z}\left\{ e^{\left(-\frac{1}{\beta}U(x)\right)} \right\} \qquad Z = \int_{-\infty}^{+\infty} e^{-\frac{1}{\beta}U(x)}\, dx$$

$$J_R(x) = -\ln(p_x) = +\ln(Z) + \frac{1}{\beta}U(x)$$

U(x) is also termed potential => $J_R(x)$ is a linear function of the potential U(x).

$1/\beta$ describes how fast $J_R(x)$ decreases with U(x).

Ridge regression

We choose as a-priori term the squared norm of the function x, weighted by P: $U(x) = \|Px^2\|$

$$p(x) = \frac{1}{Z}\left\{ e^{\left(-\frac{1}{\beta}\|Px\|^2\right)} \right\} \qquad J_R(x) = -\log(p(x)) = k + \frac{1}{\beta}\left\|Px^2\right\|$$

Nel caso del filtraggio: P = I, peso tutti i pixel dell'immagine allo stesso modo

$$J_R(x) = k + \frac{1}{\beta}\left\|x^2\right\|$$

Non voglio pixel che "sparino"

Map estimate with U(x) = ||x²||

$$x = \arg\min_x \left(\sum_i \left\|y_{n,i} - Ax_i\right\|^2 + \frac{1}{\beta}\sum_i \left\|p_{ii}x_i\right\|^2 \right)$$ Funzione costo quadratica

Derivo rispetto a x per calcolare il minimo:

$$x : A^T y_n - A^T A x + \lambda P^T P x = 0 \quad => \quad A^T y_n = \left(A^T A + \lambda P^T P\right)x$$

Without $\lambda P^T P$ large values of x are obtained where A^TA is small. These are reduced by $\lambda P^T P$

What happens when we have a filtering problem?

P, A = I $\qquad x : y_n = \left(I + \lambda I\right)x$

Do we get anywhere? $\qquad x_k = y_{nk}\,(1+\lambda)$ per ogni k

Which is the most adequate p(x) for images?

We usually ask to images to be smooth (we look at differential properties)

We look at the local gradient of the image: ∇x.

One possibility is to use the square of the l-2 norm of the gradient: $|| \nabla x ||^2$

This is another form of Tikhonov regularization.

Definition of priors on groups of elements

- **Clique** is a subset of vertices of an undirected graph such that every two distinct vertices in the clique are adjacent; that is, its induced subgraph is complete (Wikipedia).
  - It induces a neighbouring system among elements.
  - For an image, the neighbouring system is trivial: adjacent pixels.
- Markov Random Fields.
  - Definition of probabilities on cliques.
  - Definition of probability of an element as a function of all elements inside a clique.

Differential Gibbs prior

$$p_x = \frac{1}{Z}\left\{ e^{\left(-\frac{1}{\beta}U(x)\right)} \right\} \qquad Z = \int_{-\infty}^{+\infty} e^{-\frac{1}{\beta}U(x)}\, dx$$

U(x) = $\| \nabla x \|^2$

$$\arg\min_x \left\{ \left\| (Ax - y_n)^2 \right\| + \lambda \|\nabla x\|^2 \right\}$$

$$x: \ \left\{ 2A^T (Ax - y_n) + 2\lambda \nabla x \right\} = 0$$

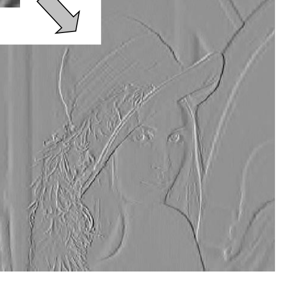
If we apporximate ∇x with the finite differences, we get a qudratic cost function in x.

$$\| \nabla x \|^2 = (x_k - x_{k+1})^2$$

A priori term – image gradients (no noise)

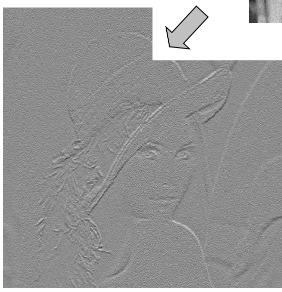
p_x = p(i,j) – p(i-1,j)

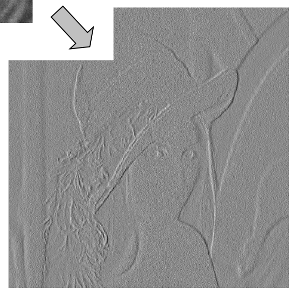
p_y = p(i,j) – p(i,j-1)



A priori term – image gradients (noise)

$$\Delta x_{row} = \frac{x_{i+1,j} - x_{i-1,j}}{2} \qquad \Delta x_{col} = \frac{x_{i,j+1} - x_{i,j-1}}{2}$$

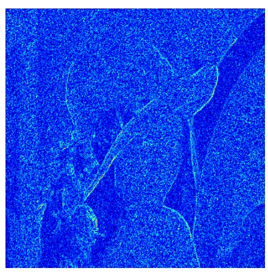




A priori term – norm of image gradient

No noise

Noise



In the real image, most of the areas are characterized by an (almost) null gradient norm;

We can for instance suppose that ||∇x|| is a random variable with Gaussian distribution, zero mean and variance equal to β^2.

[Note that, in the noisy image, the norm of the gradient assume higher values → low ||**∇x**|| means low noise!]

MAP with A-priori on the derivatives

$$\arg\min_x -\{\ln(p(y_n|x)p(x))\} = \arg\min_x -\{\ln(p(y_n|x)) + \ln(p(x))\}$$

$$\boxed{J(f) = J_o(f) + \lambda J_R(f)} \qquad J_R(x) = \mathit{funzione}\left(\|\nabla x\|^2\right)$$

$$\arg\min_x \left\{\left\|(Ax - y_n)^2\right\| + \lambda\|\nabla x\|^2\right\}$$

$$x: \left\{2A^T(Ax - y_n) + 2\lambda\nabla x\right\} = 0$$

If we apporximate ∇x with the finite differences: we get a linear system.

Tikhonov regularization

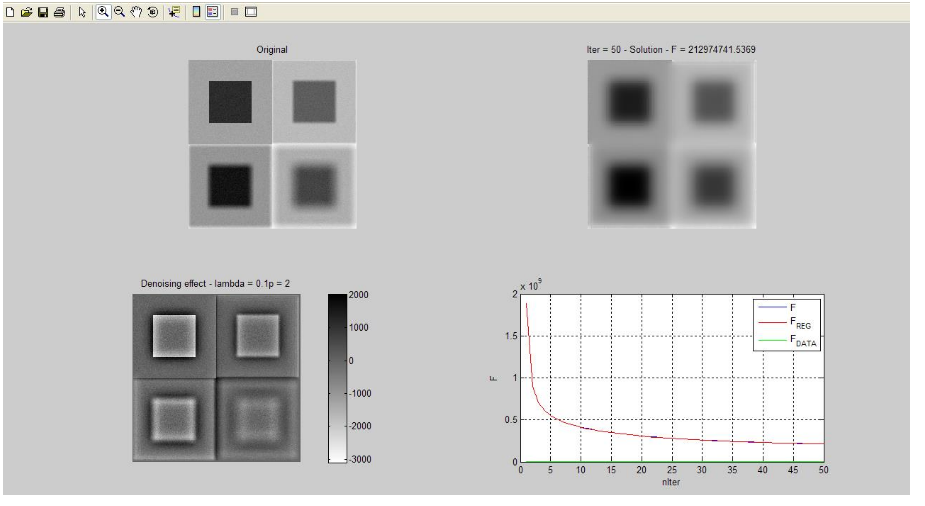
$$x = \arg\min_x \left(\sum_i \|y_{n,i} - Ax_i\|^2 + \lambda\sum_i \|Px_i\|^2\right)$$ (cf. Ridge regression)

$$x = \arg\min_x \left(\sum_i \|y_{n,i} - Ax_i\|^2 + \lambda\sum_i \|\nabla x_i\|^2\right)$$

It is a quadratic cost function. We find *x* minimizing with respect to x the cost function.

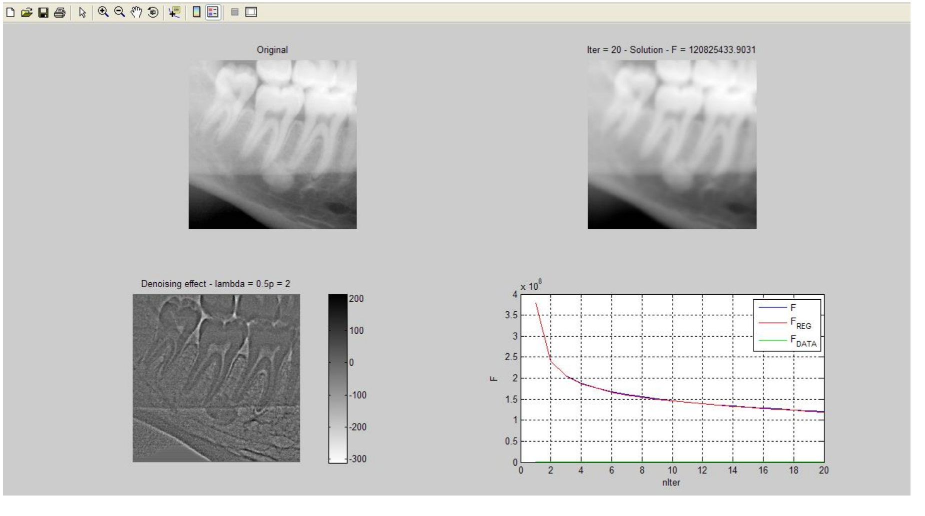
This approach is derived in the domain of mathematics. It leads to the same cost function of the MAP approach.

Tikhonov regularization - simulations



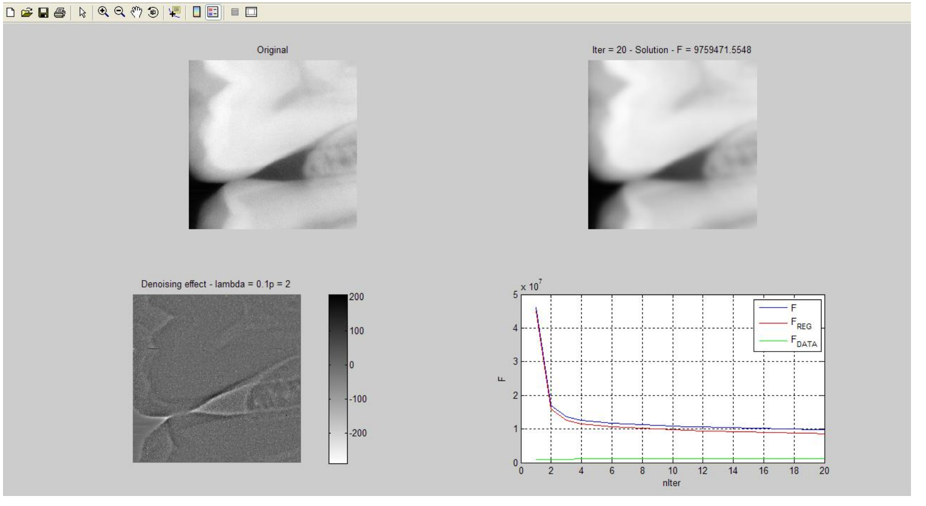
Edge smoothing effect with Tikhonov-like regularization
Poisson noise model – λ = 0.5
P is the gradient operator

Tikhonov regularization – panoramic images



Edge smoothing effect with Tikhonov-like regularization
Poisson noise model - λ = 0.5
P is the gradient operator

Tikhonov regularization - endo-oral images



Edge smoothing effect with Tikhonov-like regularization
Poisson noise model - $\lambda = 0.1$
P is the gradient operator

Gibbs priors and Regularization

$$\arg\min_f -\{\ln(p(g_n \mid f)p_f)\} = \arg\min_f -\{\ln(p(g_n \mid f)) + \ln(p_f)\}$$

Likelihood =
adherence to the data

A-priori

Gaussian $K(\sigma)\sum_i \left\| g_{n,i} - Af_i \right\|^2$

$$-\ln\left\{\frac{1}{Z}e^{\left\{-\frac{1}{\beta}U(\mathbf{f})\right\}}\right\}$$

$$\boxed{J(f) = J_o(f) + \lambda J_R(f)}$$

$J_R(f) = U(f)$

Non-quadratic a-priori: total variation

$$f = \arg\min_f -\{\ln(p(g_n \mid f)p_f)\} = \arg\min_f -\{\ln(p(g_n \mid f)) + \ln(p_f)\}$$

Gaussian $K(\sigma)\sum_i \left\| g_{n,i} - Af_i \right\|^2$

The a-priori term is a gradient and it is expressed in l_2 norm

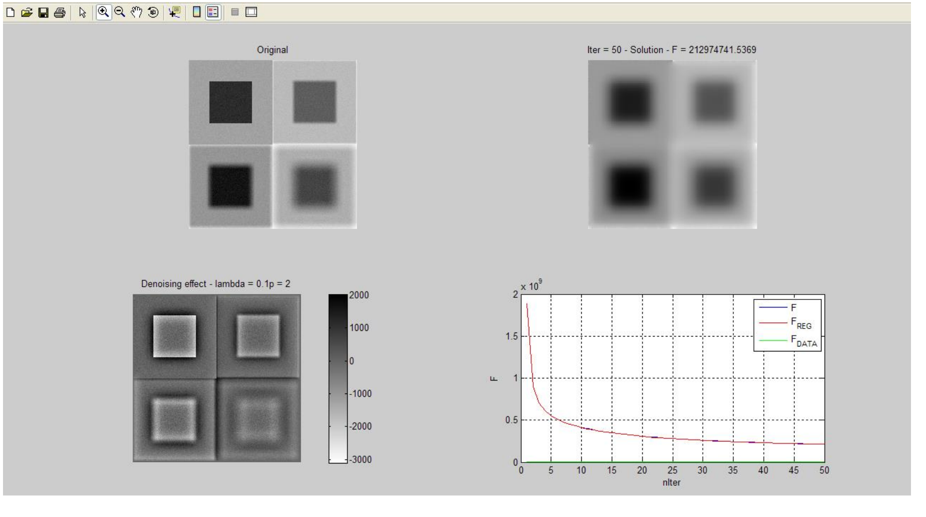
$$\sum_i \sqrt{\sum_p^P f_{p,i}^{\;2}} \qquad \sum_i \sqrt{\left(f_{x,i}^{\;2} + f_{y,i}^{\;2} + f_{z,i}^{\;2}\right)}$$

$$f = \arg\min_f \sum_i \left(\left\| g_n - Af \right\|^2 + \lambda \sqrt{\sum_p^P f_{p,i}^{\;2}} \right)$$

Total variation

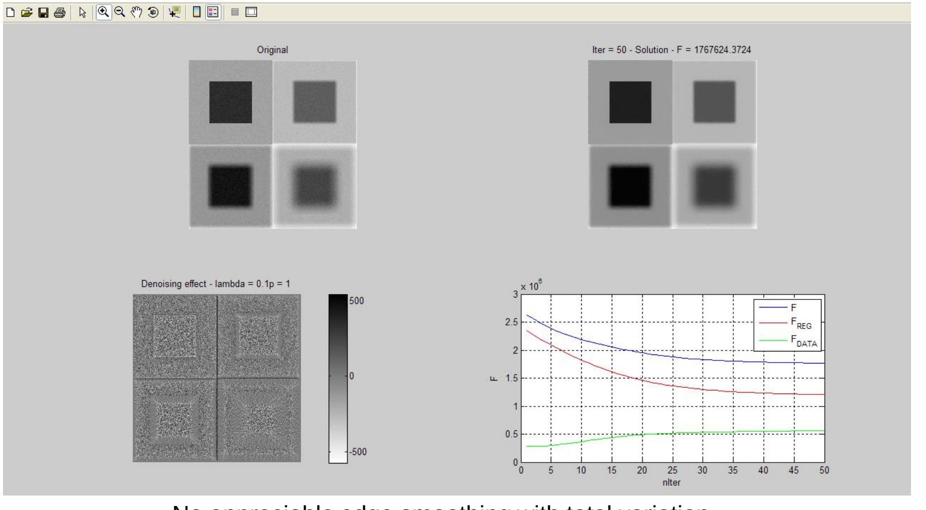
The derivative is not linear anymore because of the square root.

Tikhonov regularization - simulations



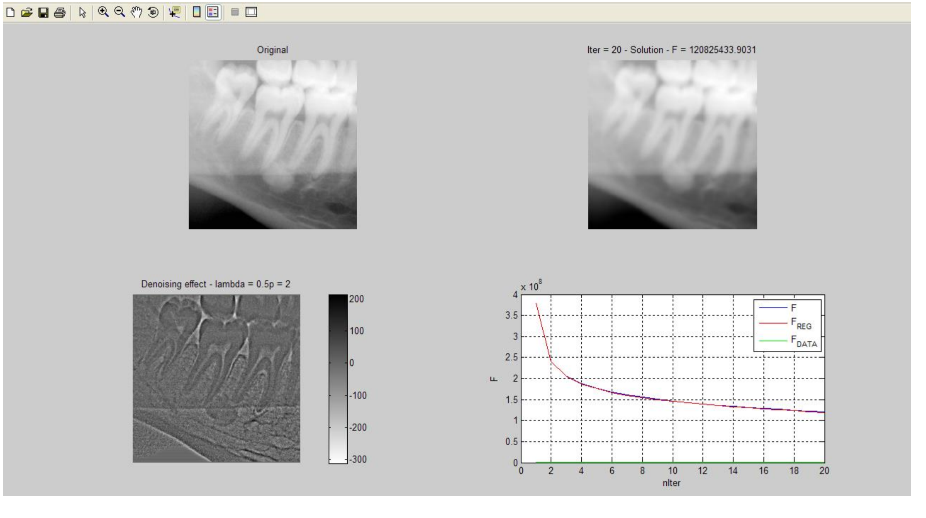
Edge smoothing effect with Tikhonov-like regularization
Poisson noise model – λ = 0.5
P is the gradient operator

Total variation regularization - simulations



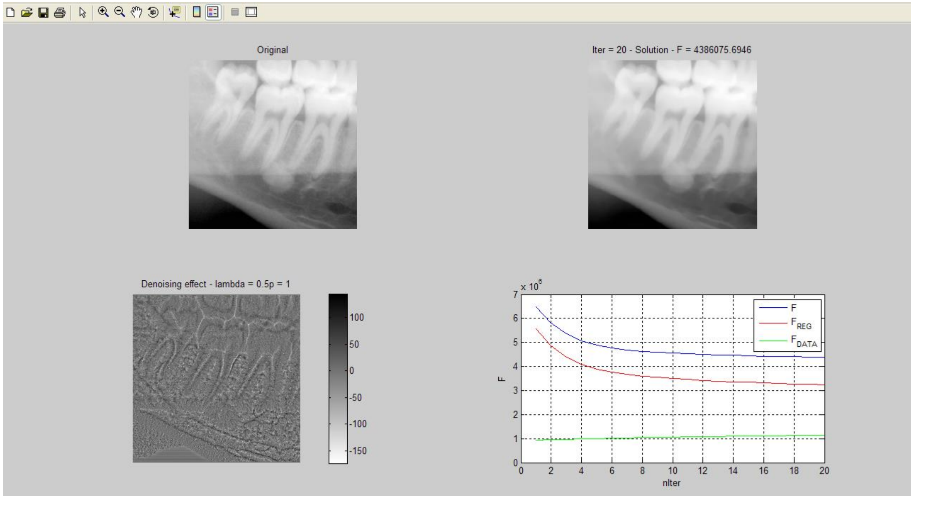
No appreciable edge smoothing with total variation
Poisson noise model - $\lambda = 0.5$
P is the gradient operator

Tikhonov regularization – panoramic images



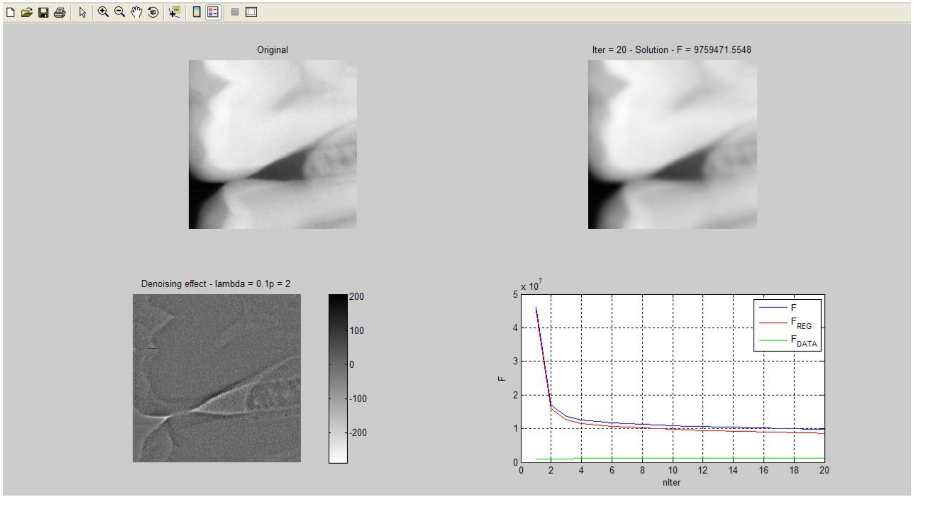
Edge smoothing effect with Tikhonov-like regularization
Poisson noise model - $\lambda = 0.5$
P is the gradient operator

Total variation regularization – panoramic images



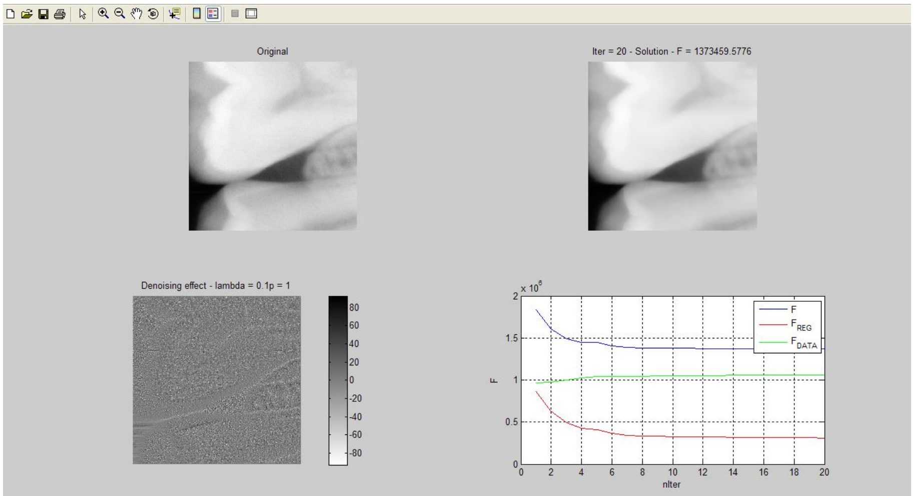
No appreciable edge smoothing with total variation
Poisson noise model - $\lambda = 0.5$
P is the gradient operator

Tikhonov regularization - endo-oral images



Edge smoothing effect with Tikhonov-like regularization
Poisson noise model - $\lambda = 0.1$
P is the gradient operator

Total variation – endo-oral images



No appreciable edge smoothing with total variation
Poisson noise model - λ = 0.1
P is the gradient operator

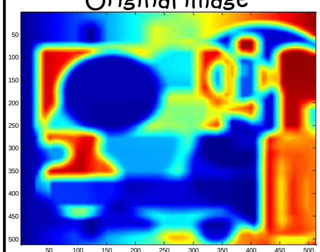
Tikhonov vs. TV (preview)

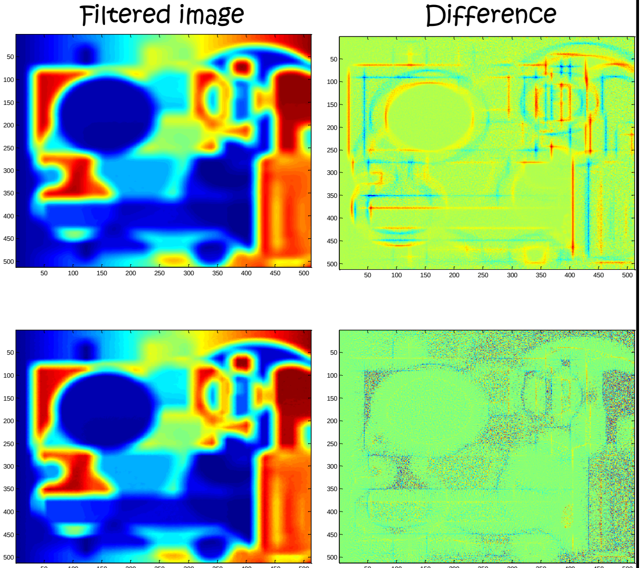
Filtered image

Difference

Tikhonov =>

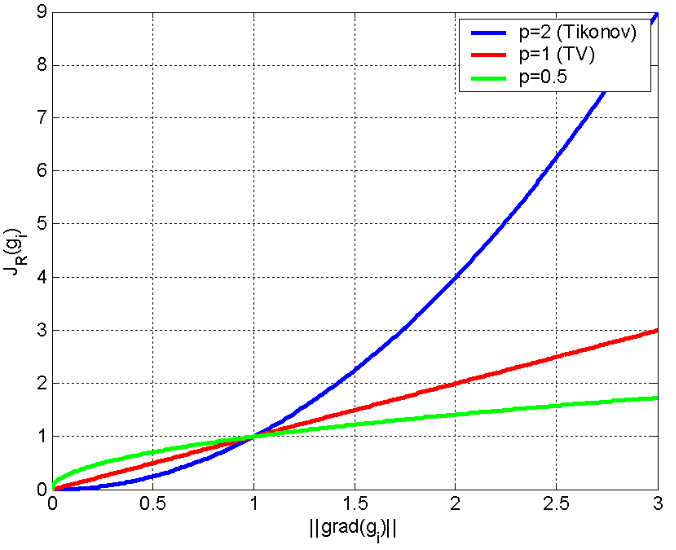
Original image





TV =>

Cost introduced by the regularzation term



Cost increases quadratically with the local gradient in Tikhonov

Open problems

- Better images, but:

Non linear cost functions (non quadratic)

$$f = \arg\min_f \sum_i \left(\|g_n - Af\|^2 + \lambda \sqrt{\sum_p^P f_{p,i}^{\ 2}} \right)$$

Minimization does not lead to a function linear in f (because of the square root) ➔ It requires non-linear iterative minimization.

Singularity in f = 0

Role of λ

$$K(\sigma)\sum_i \left\| g_{n,i} - Af_i \right\|^2 \qquad\qquad -\ln\left\{ \frac{1}{Z} e^{\left\{-\frac{1}{\beta}U(\mathbf{f})\right\}} \right\}$$

$$\boxed{J(f) = J_o(f) + \lambda J_R(f)}$$

λ incorporates different elements here:
- the standard deviation of the noise in the likelihood
- the “temperature”, that is the decrease in the energy of the configurations with their cost (β)
- the normalized constant Z.

λ has been investigated in the classical regularization theory (Engl et al., 1996), but not as deep in the Bayesian framework ➔ λ is set experimentally through cross-validation.

How to set the regularization parameter

Analysis of the residual after the estimate = **Af** – **g**
- The residual should be equal to the noise distribution

Gaussian case:
- λ is increased until (r_i, r_j) = Σ^2 ($||r||^2 = \sigma^2$)
- Sample covariance is equal to distribution covariance

Overview

Filtering images

MAP, Tikhonov and Poisson model of the noise

Poisson noise

A-priori and Markov Random Fields

Cost function minimization

Gibbs priors and Regularization

$$\arg\min_f -\{\ln(p(g_n \mid f)p_f)\} = \arg\min_f -\{\ln(p(g_n \mid f)) + \ln(p_f)\}$$

Likelihood =
adherence to the data

A-priori

Gaussian $K(\sigma)\sum_i \left\| g_{n,i} - Af_i \right\|^2$

Poisson $\sum_i g_{n,i} \ln\left(\frac{g_{n,i}}{Af} + Af_i - g_{n,i}\right)$

$$-\ln\left\{\frac{1}{Z} e^{\left\{-\frac{1}{\beta} U(\mathbf{f})\right\}}\right\}$$

$$\boxed{J(f) = J_o(f) + \lambda J_R(f)}$$

$J_R(f) = U(f)$

Non-quadratic a-priori: total variation

$$f = \arg\min_f -\{\ln(p(g_n \mid f)p_f)\} = \arg\min_f -\{\ln(p(g_n \mid f)) + \ln(p_f)\}$$

Poisson noise model

$$\sum_i g_{n,i} \ln\left(\frac{g_{n,i}}{Af} + Af_i - g_{n,i}\right)$$

The a-priori term is a gradient and it is expressed in l_2 norm

$$\sum_i \sqrt{\sum_p^P f_{p,i}^{\,2}} \qquad \sum_i \sqrt{\left(f_{x,i}^{\,2} + f_{y,i}^{\,2} + f_{z,i}^{\,2}\right)}$$

$$f = \arg\min_f \sum_i \left(\|g_n - Af\|^2 + \lambda \sqrt{\sum_p^P f_{p,i}^{\,2}} \right)$$

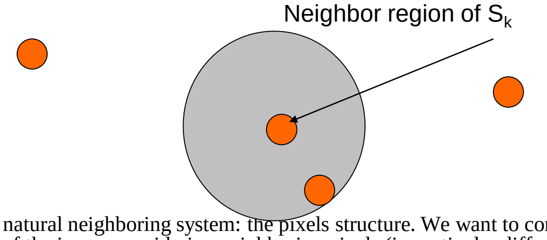
Total variation

The derivative is not linear anymore because of the square root.

A-priori

We can insert in the a-priori term all the desirable characteristic of the image: local smoothness, edges, piece-wise constancy,….

The idea of defining a neighboring system is a natural one:



Images have a natural neighboring system: the pixels structure. We want to consider the local properties of the image considering neighboring pixels (in particular differential properties - our vision system is particularly tuning to gradients both spatial and temporal). Ideas have been borrowed from physics.

Neighboring System

Let P be the set of pixels of the image: $P = \{p_1, p_2, \dots p_P\}$

The neighboring system defined over P, S, is defined as $H = \{\mathcal{N}_p \mid p, \forall p \in P\}$, that has the following properties:

An element is not a neighbor of itself: $p_k \notin \mathcal{N}_{pk}$

Mutuality of the neighboring relationship: $p_k \in \mathcal{N}_{pj} \leftrightarrow p_j \in \mathcal{N}_{pk}$

(S, P) constitute a graph where P contains the nodes of the graph and S the links. An image can be seen also as a graph.

Depending on the distance from p, different neighboring systems can be defined:

	o	
o	x	o
	o	

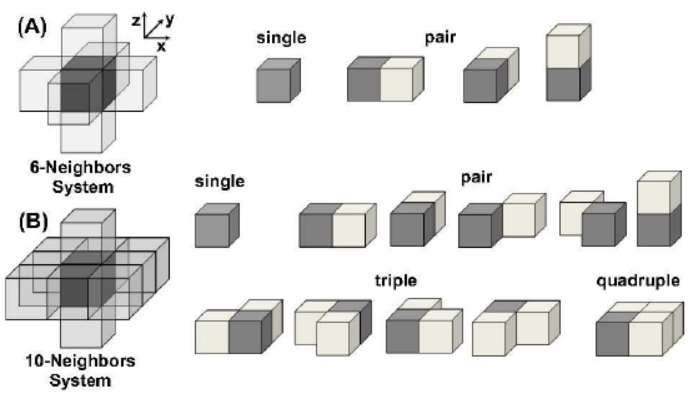
First order neighboring System
4-neighboring System

o	o	o
o	x	o
o	o	o

Second order neighboring System
8-neighboring System

Clique

Borrowed from phisics.



A clique *C*, for (S, P), is defined as a subset of vertices of S, an undirected graph, such that every two vertices in the subset are connected by an edge.

I can consider ordered sets of voxels, that are connected to p through S.
Types of cliques: single-site, pairs of neighboring sites, triples of neighboring sites,… up to the cardinality of $\mathcal{N}_p$

Markov Random Field

Given (S, P) we can define a set of random values, $\{f_k(m)\}$ for each element defined by S, that is in $\mathcal{N}_p$. Therefore we define a **random field** , $\mathcal{F}$, over S:

$\mathcal{F}(\mathcal{N}_p) = \{f_k(m) \mid m \in \mathcal{N}_p\} \ \forall p$

Under the Markovian hypotheses:

$P(f(p)) \geq 0 \ \forall p$ — Positivity

$P(f(p) \mid g(P-\{p\}) = P(f(p) \mid g(\mathcal{N}_p)\}$ — Markovianity

2 expresses the fact that the probability of p assuming a certain value, f (e.g. a certain gradient), is the same considering in p all the pixel of P but p, or only the neighbor pixels, that is the value of f depends only on the value of the pixels in $\mathcal{N}_p$ and not in p.

the random field $\mathcal{F}$ is named **Markov Random Field**.

Energy in a Markov Random Field

A "potential" function, $\phi(f)$, can be defined for a MRF. This is a scalar value that is a function of the random value associated to the pixels for all the possible elements of a clique:

$$\phi_c(f) = \sum_{j \in c} f(p_j)$$

If we consider all the possible cliques defined for each element p, we can define a potential energy function associated to the MRF:

$$U(f) = \sum_{c \in C} \phi_c(f)$$

The higher is the potential energy, the lower is the probability that the set of random values of the elements of the cliques is realized, that is the higher is the penalization for the associated configuration.

We want to go towards minimum energy.

Gibbs prior

If we consider all the possible cliques defined for each element p, we can define a potential energy function associated to the MRF:

U(**f**) = $\sum_{c \in C} \phi_c(\mathbf{f})$

The higher is the potential energy, the lower is the probability that the set of random values of the elements of the cliques is realized, that is the higher is the penalization for the associated configuration.

This is well captured by the Gibbs distribution, that describes the probability of a certain configuration to occur. It is a function exponentially decreasing of U:

P(**f**) = $\frac{1}{Z} e^{\left\{-\frac{1}{\beta} U(\mathbf{f})\right\}}$

P(f) is a Gibbs random field, Hammersley-Clifford theorem (1971). β regulates the decrease in probability and it is associated with temperature in physics. Z is a normalization constant. NB to define Gibbs random fields, P(f) > 0, P(f) → 0 U(f) → ∞: there are not configurations with 0 probability.

Gibbs priors and Regularization

$$\arg\min_f -\left\{\ln\left(p(g_n \mid f)\, p_f\right)\right\} = \arg\min_f -\left\{\ln\left(p(g_n \mid f)\right) + \ln\left(p_f\right)\right\}$$

Likelihood =
adherence to the data

A-priori

Gaussian $K(\sigma)\sum_i \left\| g_{n,i} - Af_i \right\|^2$

Poisson $\sum_i g_{n,i} \ln\left(\frac{g_{n,i}}{Af} + Af_i - g_{n,i}\right)$

$$-\ln\left\{\frac{1}{Z} e^{\left\{-\frac{1}{\beta} U(\mathbf{f})\right\}}\right\}$$

$$\boxed{J(f) = J_o(f) + \lambda J_R(f)}$$

J_R(f) = U(f)

Role of λ

$$K(\sigma)\sum_i \left\| g_{n,i} - Af_i \right\|^2$$

$$-\ln\left\{ \frac{1}{Z} e^{\left\{ -\frac{1}{\beta} U(\mathbf{f}) \right\}} \right\}$$

$$\boxed{J(f) = J_o(f) + \lambda J_R(f)}$$

λ incorporates different elements here:
- the standard deviation of the noise in the likelihood
- the "temperature", that is the decrease in the energy of the configurations with their cost (β)
- the normalized constant Z.

λ has been investigated in the classical regularization theory (Engl et al., 1996), but not as deep in the Bayesian framework ➔ λ is set experimentally through cross-validation.

How to set the regularization parameter

Analysis of the residual after the estimate = **Af** – **g**
- The residual should be equal to the noise distribution

Gaussian case:
- λ is increased until $(r_i, r_j) = \Sigma^2$ $(||r||^2 = \sigma^2)$
- Sample covariance is equal to distribution covariance

Poisson case:
- r_i tends to be larger, the larger is g_i.
- λ is increased until $|r|^2$ / g -> 1

Choice of the Gibbs priors

We choosed || λPf||^2 as a quadratic functional, but not specified P.

P is ofted chosen as a smoothing operator. The rationale is that the noise added to the image is often white (both Gaussian and Poisson) over the image as there is no correlation between adjacent pixels. Therefore its spatial content is unform and with a larger bandwidth that the signal.

As a smoothing operator P is often a differential operator, which penalizes edges.

$$J_R(\mathbf{f}) = \sum_{c \in C} \phi_c(\mathrm{d}^k{}_c \mathbf{f})$$

k is the order of the derivative
ϕ_c can be l_2 norm (total variation), squared (Tikhonov)

k = 2 difference of gradients → piecewise linear areas.
k = 3 difference of Hessian → piecewise squared.
Neighbor of order higher than 2.

Quadratic Priors with k = 0

k = 0 – No derivative, the same gray level – single site cliques.

$$J_R(\mathbf{f}) = \sum_{c \in C} \phi(\mathrm{d}^k{}_c \mathbf{f}) = \sum_{c \in C} (\mathrm{d}^0{}_c \mathbf{f})^2 = \sum_{p \in P} \mathbf{f}(\mathrm{p})^2$$

It has been applied to both Poisson and Gaussian noise models

Reduces bright spots and biases the solution to low intensity values.

Quadratic Priors with k = 1

k = 1 – First order derivatives – pair-sites cliques.

$$J_R(\mathbf{f}) = \sum_{c \in C} \phi(\mathrm{d}^1{}_c \mathbf{f}) = \sum_{p \in P} \sum_{\mathrm{m} \in \mathcal{N}_p} \phi(\mathrm{d}^0{}_c \mathbf{f})^2 = \sum_{p \in P} \sum_{\mathrm{m} \in \mathcal{N}_p} \phi\left(\frac{\mathrm{f(p)} - \mathrm{f(m)}}{\mathrm{d(p,m)}} \right)$$

d(p,m) takes into account anisotropies in computing the distance.

If we consider $\phi(.)$ a squared function, we have another form of Tikhonov regularization:

$$J_R(\mathbf{f}) = \sum_{p \in P} \sum_{\mathrm{m} \in \mathcal{N}_p} \left(\frac{\mathrm{f(p)} - \mathrm{f(m)}}{\mathrm{d(p,m)}} \right)^2$$

Quadratic Priors with k = 1

k = 1 – First order derivatives – pair-sites cliques.

$$J_R(\mathbf{f}) = \sum_{p \in P} \sum_{\mathrm{m} \in \mathcal{N}_p} \left(\frac{\mathrm{f(p)} - \mathrm{f(m)}}{\mathrm{d(p,m)}} \right)^2$$

If we consider $\phi(.)$ a squared function, we have another form of Tikhonov regularization:

$\| \mathrm{P}\mathbf{f} \|^2$

P is the convolution with the Laplacian operator:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

First order neighboring System
4-neighboring System

$$\begin{bmatrix} -\frac{\sqrt{2}}{2} & -1 & -\frac{\sqrt{2}}{2} \\ -1 & 4+2\sqrt{2} & -1 \\ -\frac{\sqrt{2}}{2} & -1 & -\frac{\sqrt{2}}{2} \end{bmatrix}$$

Second order neighboring System
8-neighboring System

Non-quadratic potential functions, k = 1

Quadratic functions priors imposes smoothness everywhere. Large true gradients of the solution are therefore penalized → smoothing sharp edges.

In imaging objects tend to be piecewise smooth, but different pieces of objects are separated by more or less sharp edges. We want to smooth inside the object but not the edge. A parallel worthwhile to be investigated is with anisotropic diffusion (Koenderink, 1987; Perona&Malik, 1990).

We search different potential functions (Geman&McClure, 85; Charbonnier et al., 1994, 1997; Hebert&Lehay, 1989).

Non-quadratic potentials (Charbonier et al., 1997)

1. $\phi(t) \geq = 0 \quad \forall t \qquad \phi(0) = 0$ — Derives from the definition of potential
2. $\Phi'(t) \geq = 0 \quad \forall t$ — Semi-monotone derivatives
3. $\phi(t) = \phi(-t)$ — Positive and negative gradients are equally considered
4. $\phi(t) \in C^1$ — This is to avoid instability.

Up to now quadratic potentials are OK

5. $\dfrac{\varphi'(t)}{2t}$ — The potential increase rate should decrease with t.
6. $\lim_{t \to \infty} \dfrac{\varphi'(t)}{2t} = 0$ — The potential increase rate should decrease for all t (at least for large values of t)
7. $\lim_{t \to 0} \dfrac{\varphi'(t)}{2t} = \cos t > 0$ — The potential increases at least linearly for t = 0.

Few non-quadratic functions (Vicedomini 2008)

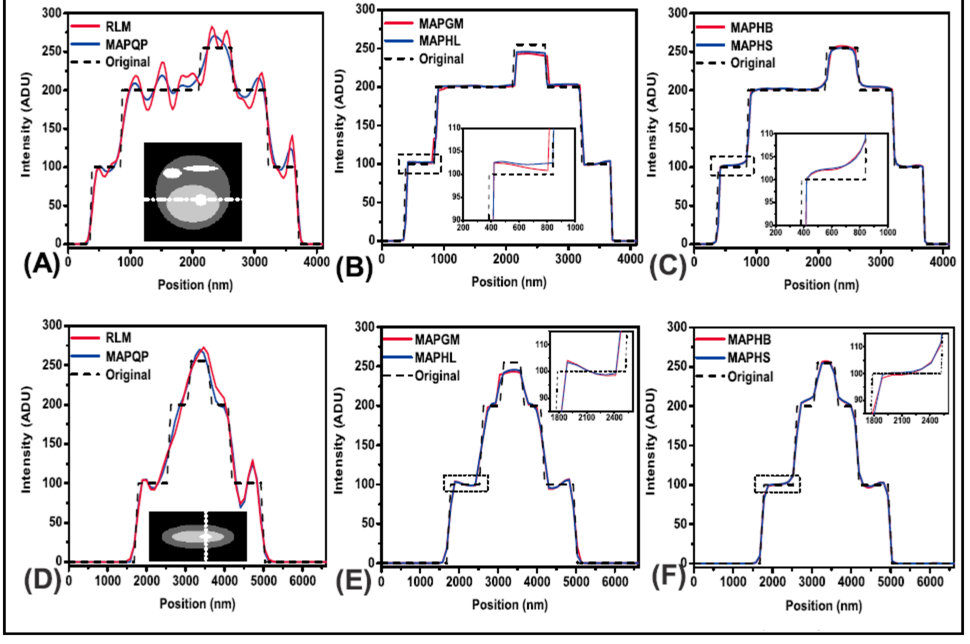
Regularization name	Potential function	Expression of $\varphi(t)$	Expression of $\psi(t) = \varphi'(t)/2t$	Convex
Quadratic-Potential	φ_{QP}	t^2	1	yes
Geman-McClure	φ_{GM}	$\frac{t^2}{1+t^2}$	$\frac{1}{(1+t^2)^2}$	no
Hebert-Leahy	φ_{HL}	$\log(1+t^2)$	$\frac{1}{1+t^2}$	no
Huber	φ_{HB}	$\begin{cases} t^2, & \lvert t\rvert \le 1 \\ 2\lvert t\rvert - 1, & \lvert t\rvert > 1 \end{cases}$	$\begin{cases} 1, & \lvert t\rvert \le 1 \\ 1/\lvert t\rvert, & \lvert t\rvert > 1 \end{cases}$	yes
Hyper-Surface	φ_{HS}	$2\sqrt{1+t^2} - 2$	$\frac{1}{\sqrt{1+t^2}}$	yes

Asymptotic linear behavior

Asymptotic log-like behavior

Why not simply $\sqrt{t^2}$?

Results



Summary

MAP estimate can be seen as a statistical version of regularization.

The regularization term can be derived from the potential energy associated to an adequate neighbor system defined over the object (e.g. over the image).

Under this hypothesis the value assumed by the elements of the object to be reconstructed (e.g. restored or filtered image) represent a MRF.

Different neighbor systems and different potential functions allow defining different properties of the object.

For quadratic potential functions, Tikhonov regularizer are derived.

The discrepancy term for the data represents the likelihood and can accommodate different statistical models: Poison, Gaussian or even mixture models.

Overview

Filtering images

MAP and Regularization

Poisson noise

A-priori and Markov Random Fields

Cost function minimization

Poisson case

$\text{Noise}_i = \|A\ f - g_{ni}\ \|$

We know the statistical distribution of the noise -> we now the statistical distribution of the second term. In case of Poisson noise we have:

For one pixel: $p(g_{ni}, f_i) = \left\{ \frac{e^{-Af_i}(Af_i)^{g_{n_i}}}{g_{n_i}!} \right\}$

$$-\ln(L(g_n; f)) = -\ln\left(\prod_{i=1}^{N} p(g_{n,i}; f_i)\right) = -\sum_{i=1}^{N}\left(-Af_i + g_{n,i}\ln(Af_i) - \ln(g_{n,i}!)\right)$$

To eliminate the factorial term, we normalize the likelihood by $L(g_n, g_n)$:

$$-\ln\left(\frac{L(g_n, f)}{L(g_{n,} g_n)}\right) = -\sum_{i=1}^{N}(g_n \ln(Af) - \ln(g_n) + g_n - Af) = KL\, divergence$$

$$= \sum_i g_n \ln\left(\frac{g_n}{Af} + Af - g_n\right)$$

It is not a distance!
It is not linear

Gibbs priors and Regularization

$$\arg\min_f\ -\{\ln(p(g_n \mid f)p_f)\} = \arg\min_f -\{\ln(p(g_n \mid f)) + \ln(p_f)\}$$

Likelihood = adherence to the data

A-priori

Gaussian $K(\sigma)\sum_i \left\|g_{n,i} - Af_i\right\|^2$

Poisson $\sum_i g_{n,i} \ln\left(\frac{g_{n,i}}{Af} + Af_i - g_{n,i}\right)$

$$-\ln\left\{\frac{1}{Z} e^{\left\{-\frac{1}{\beta}U(\mathbf{f})\right\}}\right\}$$

$$\boxed{J(f) = J_o(f) + \lambda J_R(f)}$$

$J_R(f) = U(f)$

What happens if noise is Poisson?

$$f = \arg\min_f -\{\ln(p(g_n \mid f)p_f)\} = \arg\min_f -\{\ln(p(g_n \mid f)) + \ln(p_f)\}$$

Poisson noise model
Squared shape for the a-priori term

$$\sum_i g_{n,i} \ln\left(\frac{g_{n,i}}{Af} + Af_i - g_{n,i}\right)$$

$|| \lambda Pf||^2$

$$f = \arg\min_f \sum_i g_{n,i} \ln\left(\frac{g_{n,i}}{Af} + Af_i - g_{n,i}\right) + \lambda\|Pf\|^2$$

Regularization

No analytical solution

Overview

Filtering images

MAP, Tikhonov and Poisson model of the noise

Poisson case

A-priori and Markov Random Fields

Cost function minimization

Regularization term

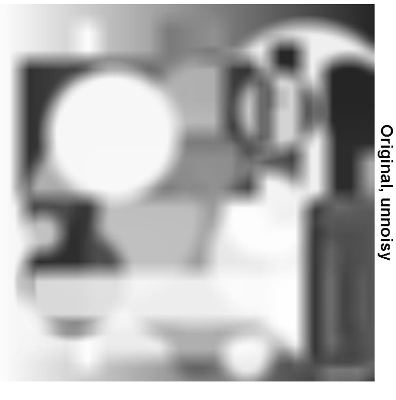
$$J_{REG}(f) = \left\| \nabla f \right\|_2^q$$

For q = 1, it has a singularity in the origin for which its derivative cannot be computed. Solution is one of the potentials functions above, or a numerical solution:

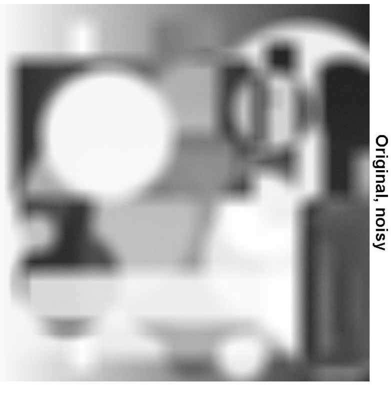
$$J_{REG}(f_i) = \sqrt{\frac{df_i}{dx} + \frac{df_i}{dy} + ... + \varepsilon}$$

$\varepsilon = 2.22 \times 10^{-16}$

Simulated images



Original, unnoisy



Original, noisy

Gradient Descendent is slow

Algorithm

Set $u^{(0)} = \{g\}$

Compute $\nabla J = \left[\frac{\partial}{\partial u_1} J, ..., \frac{\partial}{\partial u_N} J \right]^T$

Update $u^{(k+1)} = u^{(k)} - \eta \nabla J$

η is a scalar parameter (damping factor), optimized at each iteration, such as it is guaranteed that J decreases (line search).

- Time expensive: ~ 210s (with Matlab) on 500x500 images
- We can improve the algorithm and / or the gradient computation

One-step late EM (Green, 1990)

We derive it with fixed point optimization. Let us consider the cost function for Poisson noise:

$$J(g_{n,i} \mid g_i) = -\sum_{i=1}^{N} \{ g_{n,i} \ln(g_i) - g_i \} + \lambda \sum_{i=1}^{N} \| \nabla g_i \|_2^2$$

We suppose all the pixel constant and the variation of each pixel are accumulated and applied to the next step (one-step late).

$$\frac{\partial J(g_{n,k} \mid g_k)}{\partial g_k} = \frac{\partial}{\partial g_k} \{ -[g_{n,k} \ln(g_k) - g_k] \} + \lambda \cdot \frac{\partial}{\partial g_k} J_R(g_k) = -\frac{g_{n,k}}{g_k} + 1 + \lambda \cdot \frac{\partial}{\partial g_k} J_R(g_k) = 0$$

This cannot be solved directly, but it can be solved using fixed point iteration:

$$-\frac{g_{n,k}}{g_k} + 1 + \lambda \cdot \frac{\partial}{\partial g_k} J_R(g_k) = 0 \Rightarrow \frac{g_{n,k}}{g_k} = 1 + \lambda \cdot \frac{\partial}{\partial g_k} J_R(g_k) \Rightarrow g_k = \frac{g_{n,k}}{1 + \lambda \cdot \frac{\partial}{\partial g_k} J_R(g_k)}$$

Expectation Maximization

From emission Tomography (Green, 1990; Panin et al., 1999)

$$u_i^{(new)} = \frac{u_i^{(old)}}{\sum_j h_{i,j} + \lambda \frac{\partial}{\partial u_i} J_{REG}\left(u^{(old)}\right)} \sum_j \frac{h_{i,j} z_j}{\sum_k h_{k,j} u_k^{(old)}}$$

In our case

$$H = \left[h_{i,j}\right] = I$$

The previous formula becomes

$$u_i^{(new)} = \frac{z_i}{1 + \lambda \frac{\partial}{\partial u_i} J_{REG}\left(u^{(old)}\right)}$$

Observations

Semi-convergence properties.

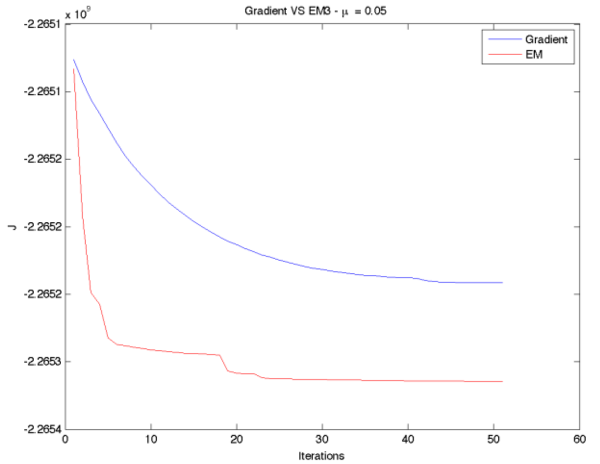
Damping of the solution is required.

- Damped EM, $x^{k+1}=(1-t)x^k+t*EM(x^k)$ (damping, relaxation, reduction of the step lengt

Solutions have been recently proposed for PET images (Mair&Zahnen, 2006).

Large increase in speed has been registered.

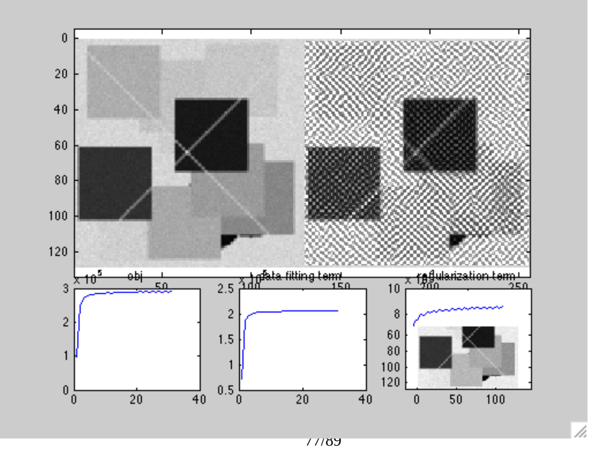
Sensitive to number of steps.



Centered gradient is bad

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & +1 \\ 0 & +1 & 0 \end{bmatrix}$$

If used centered gradient to computer the a-priori, we obtain a checkerboard effect



Different gradient possibilities

We consider only two gradients: North-Center + West-Center

$$\left\| \nabla g(x_i, y_i) \right\|_2 = \sqrt{g_x(x_i, y_i)^2 + g_y(x_i, y_i)^2 + \varepsilon} =$$

$$= \sqrt{[g(x_i, y_i) - g(x_i - 1, y_i)]^2 + [g(x_i, y_i) - g(x_i, y_i - 1)]^2 + \varepsilon}$$

4 neighbors gradient

8 neighbors gradient

Why not to change the norm?

We consider only two gradients: North-Center + West-Center

$$\|\nabla g(x_i,y_i)\|_1 = |g_x(x_i,y_i)| + |g_y(x_i,y_i)| = |g(x_i,y_i)-g(x_i-1,y_i)| + |g(x_i,y_i)-g(x_i,y_i-1)|$$

$$\frac{\partial J_R(\mathbf{g})}{\partial g_k} = \frac{\partial \sum_{i=1}^{N} \|\nabla g(x_i,y_i)\|_1}{\partial g_k} = \frac{\partial\left[\|\nabla g(x_k,y_k)\|_1 + \|\nabla g(x_k+1,y_k)\|_1 + \|\nabla g(x_k,y_k+1)\|_1\right]}{\partial g_k} =$$

$$= \frac{\partial}{\partial g_k}\left[|g(x_k,y_k)-g(x_k-1,y_k)| + |g(x_k,y_k)-g(x_k,y_k-1)|\right] +$$

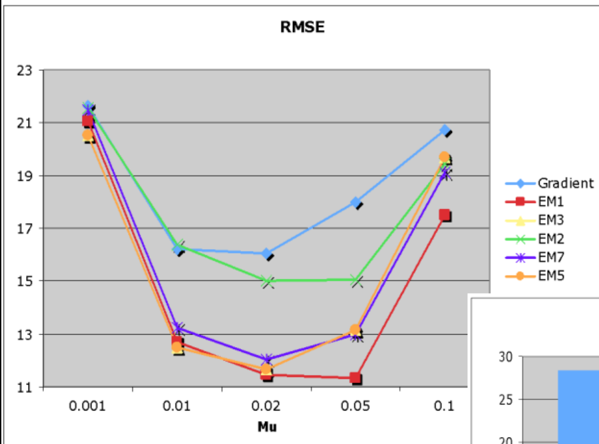
$$\frac{\partial}{\partial g_k}\left[|g(x_k+1,y_k)-g(x_k,y_k)| + |g(x_k+1,y_k)-g(x_k+1,y_k)|\right] +$$

$$\frac{\partial}{\partial g_k}\left[|g(x_k,y_k+1)-g(x_k-1,y_k+1)| + |g(x_k,y_k+1)-g(x_k,y_k)|\right] =$$

$$= \boxed{sign[g_x(x_k,y_k)] + sign[g_y(x_k,y_k)] - sign[g_x(x_k+1,y_k)] - sign[g_y(x_k,y_k+1)]}$$

We do not need ε anymore but we do not have continuity in the origin. May be we can relax Charbonnier et al. conditions….

Experimental results



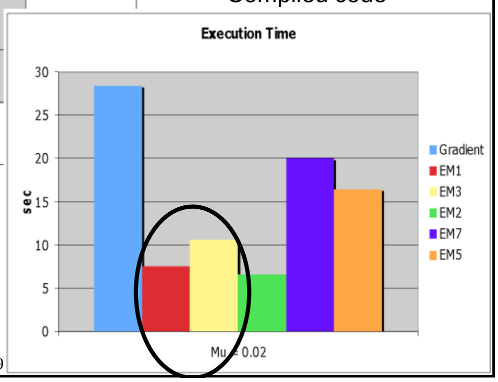
||·||$_2$ EM1

|| ||$_1$ EM3

||·||$_2$ EM7

|| ||$_1$ EM5

Compiled code



||·||$_2$ EM2 – centered gradient

Increase in speed of ≈ 5x

Beyond EM

$$J\left(g_{n,i} \mid g_i\right) = -\sum_{i=1}^{N}\left\{g_{n,i}\ln\left(g_i\right) - g_i\right\} + \lambda\sum_{i=1}^{N}\left\|\nabla g_i\right\|_2^q$$

is an optimization problem, in which g has two interesting properties:

g(p) ≥ 0

$\sum_p g(p) = \cos t$ Flux conservation (preservation of the intensity of the image)

Moreover, J(.) is supposed convex. Under these hypotheses, the so Called Kuhn-Tucker condition for the (unique) minimum should hold:

$g^* \nabla J(g^*; g_n) = 0$

$g^* \geq 0 \qquad \nabla J(g^*; g_n) \geq 0$

Split gradient (Lanteri, 2002)

$$J\left(g_{n,i} \mid g_i\right) = -\sum_{i=1}^{N}\left\{g_{n,i}\ln\left(g_i\right) - g_i\right\} + \lambda\sum_{i=1}^{N}\left\|\nabla g_i\right\|_2^q$$

Singularity when gradient is 0 and q < 2.
The idea is to obtain a term > 0 strictly at the denominator.

∇J(g; gn) = U(g; gn) + V(g; gn) with U(g; gn) $\geq$ 0; V(g; gn) > 0

Kuhn-Tucker condition becomes:

g*∇J(g*; gn) = 0 ➔ g*(U(g; gn) + V(g; gn)) = 0

We can write fixed point iteration and obtain:

$g^{(t+1)}$ = g(t) U(g; gn) / V(g; gn))

Split-gradient Algorithm

Inizialization. Choose $g^{(0)}$, that can be coincident with g_n and compute the flux, that is the $c = \Sigma g_{n,i}$.

Iteration in two steps: update + normalization.

Update:
$$\hat{g}^{(t+1)} = g^{(t)} + a^{(t)} g^{(t)} \left(\frac{U(g;g_n) - V(g;g_n)}{V(g;g_n)} \right)$$

$$c^{(t+1)} = \sum_p g^{(t+1)}(p)$$

Normalization through flux conservation:

$$g^{(t+1)}(p) = \frac{c}{c^{(t+1)}} \hat{g}^{(t+1)}(p)$$

Relaxed Split-gradient Algorithm (α = 1)

Inizialization. Choose $g^{(0)}$, that can be coincident with g_n and compute the flux, that is the $c = \Sigma g_{n,i}$.

Iteration in two steps: update + normalization.

Update:
$$\hat{g}^{(t+1)} = g^{(t)} + a^{(t)} g^{(t)} \left(\frac{U(g;g_n) - V(g;g_n)}{V(g;g_n)} \right) = g^{(t)} \left(\frac{U(g;g_n)}{V(g;g_n)} \right)$$

$$c^{(t+1)} = \sum_p g^{(t+1)}(p)$$

Normalization through flux conservation:

$$g^{(t+1)}(p) = \frac{c}{c^{(t+1)}} \hat{g}^{(t+1)}(p)$$

that has a very attractive multiplicative factor. This is also a Scaled gradient algorithm (Bertero et al., 2008)

Determination of U(.) and V(.)

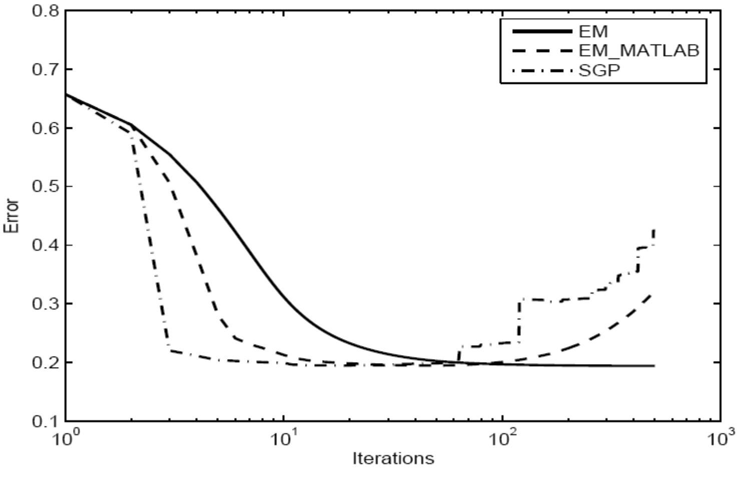
$$J\left(g_{n,i} \mid g_i\right) = -\sum_{i=1}^{N}\left\{g_{n,i}\ln\left(g_i\right) - g_i\right\} + \lambda\sum_{i=1}^{N}\left\|\nabla g_i\right\|_2^q = J_o + \lambda J_R$$

For the likelihood term: ∇J_0

	U	V
Gaussian case	$2g_n$	$2g$
	$2A^T g_n$	$2(A^T Ag + b)$
Poisson case	g_n / g	1
	$A^T g_n / (Ag + b)$	

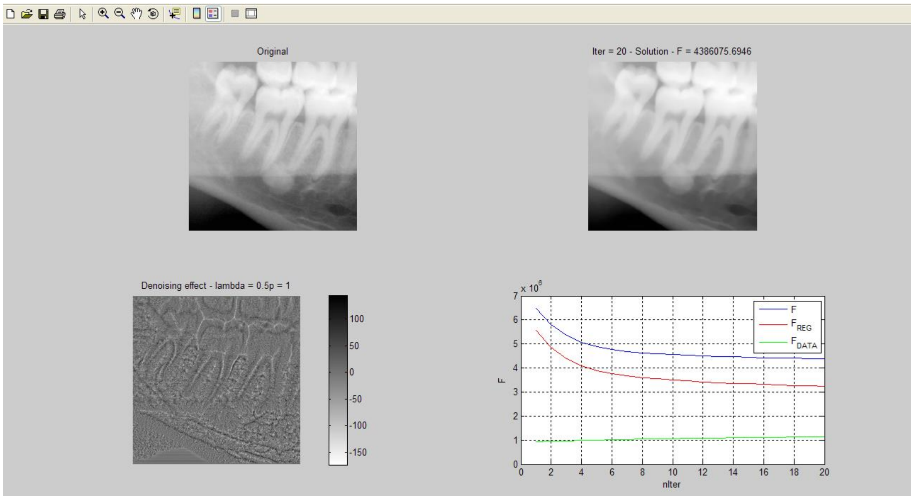
For the regularization term: ∇J_R the derivatives of the potential function have to be considered (Bertero et al., in preparation) and grouped into positive and strictly positive values.

Faster convergence for large number of iterates (from Bertero et al. 2008)



Computational time: 54.5s, 7.7s, 4.0s for a 256 x 256 image, in Matlab.
Results obtained only with Jo → EM solution.

Real-time filtering of panoramic images



No appreciable edge smoothing with total variation
Poisson noise model - $\lambda = 0.5$
P is the gradient operator

Application for intensive algebraic methods

Denoising – Bayesian filtering
Deconvolution (tomosynthesis, volumetric reconstruction from limited angle of view)
Deconvolution (CB-CT, FanBeam CT)
….

Amenable to be implemented on CUDA architectures → Real-time volumetric reconstruction.

Overview

Filtering images

MAP, Tikhonov and Poisson model of the noise

Poisson case

A-priori and Markov Random Fields

Cost function minimization