LA-UR-12-22230
Approved for public release; distribution is unlimited.

Title:	Image processing and reconstruction
Author(s):	Chartrand, Rick
Intended for:	Colloquium talk, U. Wisconsin-Stout, 6/26/12.

Abstract:

This talk will examine some mathematical methods for image processing and the solution of underdetermined, linear inverse problems. The talk will have a tutorial flavor, mostly accessible to undergraduates, while still presenting research results. The primary approach is the use of optimization problems. We will find that relaxing the usual assumption of convexity will give us much better results.

Image processing and reconstruction

Rick Chartrand

Los Alamos National Laboratory

June 26, 2012

In theory, there's no difference between theory and practice. In practice, there is.

–Yogi Berra

Outline

Mathematical image processing

Sparse image reconstruction

Convexity, nonconvexity

Examples

Summary

Mathematical image processing

Mathematics is used for many imaging tasks, such as denoising:

noisy image

denoised image

Noisy images are produced by photography in low-light conditions, some kinds of microscopy, ultrasound imaging, radiography of dense objects, and many other scientific applications.

Mathematical image processing

Mathematics is used for many imaging tasks, such as denoising, deblurring:

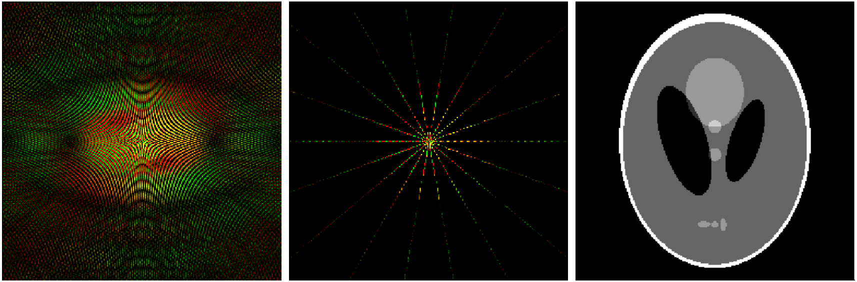
blurry image

deblurred image

Blurry images arise in unsteady photography, imaging with imperfect optics, and very-long-baseline radio astronomy.

Mathematical image processing

Mathematics is used for many imaging tasks, such as denoising, deblurring, and reconstruction:

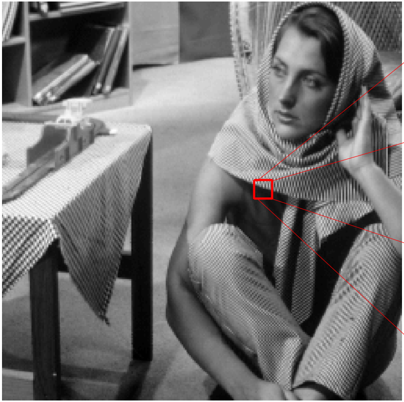


MRI data | 3.5% sampled | reconstruction

Inversion of measurement processes is necessary in radiography, MRI, interferometric astronomy, and many other applications.

Image processing as multivariable calculus

An $M \times N$-pixel image can be regarded as a point in $\mathbb{R}^{M \times N}$ or $\mathbb{R}^{MN}$.

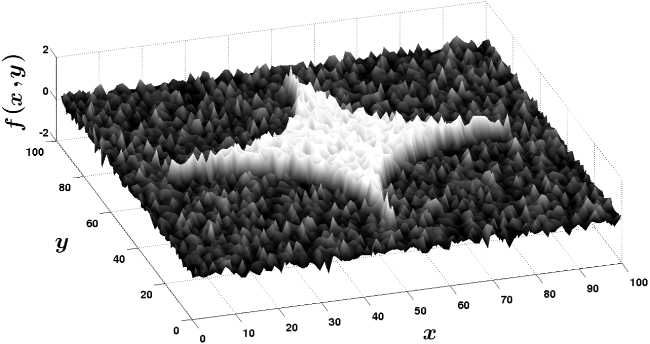


0.718	0.524	0.638	0.575	0.571	0.601	0.492	0.625
0.553	0.619	0.490	0.694	0.436	0.700	0.391	0.691
0.305	0.502	0.382	0.705	0.370	0.716	0.381	0.688
0.184	0.207	0.241	0.433	0.363	0.565	0.444	0.572
0.174	0.173	0.174	0.170	0.211	0.279	0.349	0.407
0.172	0.171	0.168	0.164	0.169	0.162	0.185	0.205
0.171	0.169	0.157	0.161	0.160	0.160	0.166	0.163
0.189	0.175	0.162	0.160	0.153	0.155	0.160	0.162

Images as functions

For modeling purposes, it can be more convenient to regard an image as a function of a continuous, 2-D variable.

100×100 noisy image



function on $[0, 100] \times [0, 100]$

Variational image denoising

A common approach in image processing is to formulate an *optimization problem* whose solution will have desirable properties.

Variational image denoising

A common approach in image processing is to formulate an *optimization problem* whose solution will have desirable properties.

For example, for image denoising, we can penalize oscillations:

$$\min_u \frac{1}{2}\int |\nabla u(x,y)|^2\,dx\,dy + \frac{\lambda}{2}\int |u(x,y)-f(x,y)|^2\,dx\,dy.$$

The second term penalizes discrepancies between u and f, with the value of λ controlling the balance.

Variational image denoising

A common approach in image processing is to formulate an *optimization problem* whose solution will have desirable properties.

For example, for image denoising, we can penalize oscillations:

$$\min_u \frac{1}{2}\int |\nabla u(x,y)|^2 \, dx \, dy + \frac{\lambda}{2}\int |u(x,y) - f(x,y)|^2 \, dx \, dy.$$

The second term penalizes discrepancies between u and f, with the value of λ controlling the balance.

The discrete analog is to minimize the following:

$$F(\vec{u}) = \frac{1}{2}\sum_{i=1}^{MN} \left[(D_x\vec{u})_i^2 + (D_y\vec{u})_i^2\right] + \frac{\lambda}{2}\sum_{i=1}^{MN}(u_i - f_i)^2.$$

D_x and D_y are matrices for computing finite-difference approximations of derivatives.

High-dimensional calculus

We can compute derivatives of F:

$$\nabla F(\vec{u}) = \left(D_x^T D_x + D_y^T D_y + \lambda I\right)\vec{u} - \lambda \vec{f},$$

$$\nabla^2 F(\vec{u}) = D_x^T D_x + D_y^T D_y + \lambda I.$$

High-dimensional calculus

We can compute derivatives of F:

$$\nabla F(\vec{u}) = \left(D_x^T D_x + D_y^T D_y + \lambda I\right)\vec{u} - \lambda \vec{f},$$

$$\nabla^2 F(\vec{u}) = D_x^T D_x + D_y^T D_y + \lambda I.$$

Since the Hessian of F is positive definite, we can solve our optimization problem by solving the linear system $\nabla F(\vec{u}) = 0$.

High-dimensional calculus

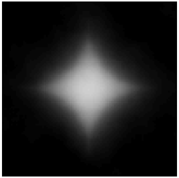
We can compute derivatives of $\boldsymbol{F}$:

$$\nabla F(\vec{u}) = \left(D_x^T D_x + D_y^T D_y + \lambda I\right)\vec{u} - \lambda \vec{f},$$

$$\nabla^2 F(\vec{u}) = D_x^T D_x + D_y^T D_y + \lambda I.$$

Since the Hessian of $\boldsymbol{F}$ is positive definite, we can solve our optimization problem by solving the linear system $\nabla F(\vec{u}) = 0$.

The linear diffusion results in edges being blurred:



Total variation

An alternative is to use *total variation* (TV) regularization:

$$\min_{u} \int |\nabla u| + \frac{\lambda}{2} \int |u - f|^2, \text{ or}$$

$$\min_{\vec{u}} F(\vec{u}) := \frac{1}{2} \sum_{i=1}^{MN} \sqrt{(D_x \vec{u})_i^2 + (D_y \vec{u})_i^2} + \frac{\lambda}{2} \sum_{i=1}^{MN} (u_i - f_i)^2;$$

Total variation

An alternative is to use *total variation* (TV) regularization:

$$\min_u \int |\nabla u| + \frac{\lambda}{2}\int |u-f|^2, \text{ or}$$

$$\min_{\vec{u}} F(\vec{u}) := \frac{1}{2}\sum_{i=1}^{MN}\sqrt{(D_x\vec{u})_i^2 + (D_y\vec{u})_i^2} + \frac{\lambda}{2}\sum_{i=1}^{MN}(u_i - f_i)^2;$$

$$\nabla F(\vec{u}) = D_x^T \frac{D_x\vec{u}}{|D\vec{u}|} + D_y^T \frac{D_y\vec{u}}{|D\vec{u}|} + \lambda(\vec{u} - \vec{f}).$$

Total variation

An alternative is to use *total variation* (TV) regularization:

$$\min_u \int |\nabla u| + \frac{\lambda}{2}\int |u - f|^2, \text{ or}$$

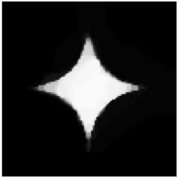
$$\min_{\vec{u}} F(\vec{u}) := \frac{1}{2}\sum_{i=1}^{MN}\sqrt{(D_x\vec{u})_i^2 + (D_y\vec{u})_i^2} + \frac{\lambda}{2}\sum_{i=1}^{MN}(u_i - f_i)^2;$$

$$\nabla F(\vec{u}) = D_x^T \frac{D_x\vec{u}}{|D\vec{u}|} + D_y^T \frac{D_y\vec{u}}{|D\vec{u}|} + \lambda(\vec{u} - \vec{f}).$$

- Let $\vec{u}^{(0)} = \vec{f}$.
- For any $k \geq 0$, given $\vec{u}^{(k)}$, let Q_k be the diagonal matrix having the values $1/|D\vec{u}^{(k)}|$. Then solve $\left(D_x^T Q_k D_x + D_y^T Q_k D_y + \lambda I\right)\vec{u}^{(k+1)} = \lambda\vec{f}$.

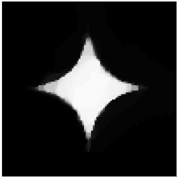
Edge (non)preservation

The intensity diffuses along edges, but not across them. We get sharp edges, but the shape is not preserved:



Edge (non)preservation

The intensity diffuses along edges, but not across them. We get sharp edges, but the shape is not preserved:



This is because TV penalizes the length of edges in the image. For example, if $u = \chi_E$ is the characteristic function of a set E,

$$\int |\nabla \chi_E| = \mathrm{length}(\partial E).$$

Consequently, long edges surrounding little area will not be preserved.

Preserving noisy shapes

Our fix is to modify the regularization term by introducing an exponent $p \in (0, 1)$, thereby softening the jump penalty:

$$\min_{u} \int |\nabla u|^p + \frac{\lambda}{2} \int |u - f|^2,$$

or

$$\min_{\vec{u}} \frac{1}{2} \sum_{i=1}^{MN} \left[(D_x \vec{u})_i^2 + (D_y \vec{u})_i^2\right]^{p/2} + \frac{\lambda}{2} \sum_{i=1}^{MN} (u_i - f_i)^2.$$

Preserving noisy shapes

Our fix is to modify the regularization term by introducing an exponent $p \in (0, 1)$, thereby softening the jump penalty:

$$\min_u \int |\nabla u|^p + \frac{\lambda}{2} \int |u - f|^2,$$

or

$$\min_{\vec{u}} \frac{1}{2} \sum_{i=1}^{MN} \left[(D_x \vec{u})_i^2 + (D_y \vec{u})_i^2\right]^{p/2} + \frac{\lambda}{2} \sum_{i=1}^{MN} (u_i - f_i)^2.$$

In the continuous setting, this penalizes a $(2 - p)$-dimensional measure of edges. The length of edges is not penalized.

Preserving noisy shapes

Our fix is to modify the regularization term by introducing an exponent $p \in (0,1)$, thereby softening the jump penalty:

$$\min_u \int |\nabla u|^p + \frac{\lambda}{2} \int |u - f|^2,$$

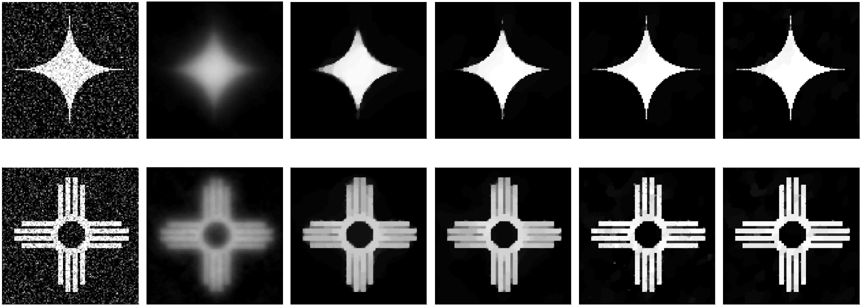
or

$$\min_{\vec{u}} \frac{1}{2} \sum_{i=1}^{MN} \left[(D_x \vec{u})_i^2 + (D_y \vec{u})_i^2 \right]^{p/2} + \frac{\lambda}{2} \sum_{i=1}^{MN} (u_i - f_i)^2.$$

In the continuous setting, this penalizes a $(2-p)$-dimensional measure of edges. The length of edges is not penalized.

The discrete implementation penalizes edge length less for smaller p. Noise is still removed.

Examples



noisy image | $p = 2$ | $p = 1$ | $p = 3/4$ | $p = 1/2$ | $p = 1/4$

Outline

Linear inverse problems

In many scientific applications, we have data $\vec{b}$ that constitute linear measurements $A\vec{s}$ of some object or state $\vec{s}$, so that we need to solve a linear system of equations, $A\vec{x} = \vec{b}$.

There are many cases where we don't have (or don't want to have) as many measurements as unknowns, making our linear system *underdetermined*. What then?

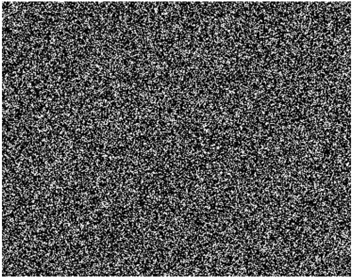
X-ray CT: less radiation

MRI: less time, less $

remote sensing: fewer sensors

Prior knowledge

Key idea: real images are *sparse* (or compressible).



Almost all solutions consistent with the data will be like this.

Only a minuscule fraction might look like this.

All we need to exploit this is a way to extract sparse solutions to linear systems.

Sparsity

A vector $\vec{s}$ (or matrix) is called *sparse* if most of its entries are zero.

Sparsity

A vector $\vec{s}$ (or matrix) is called *sparse* if most of its entries are zero.

More general notions of sparsity include:

- Transform sparsity: $\mathbf{\Psi}\vec{s}$ is sparse, for some linear transformation $\mathbf{\Psi}$. Examples: gradient; wavelet transforms.

Sparsity

A vector $\vec{s}$ (or matrix) is called *sparse* if most of its entries are zero.

More general notions of sparsity include:

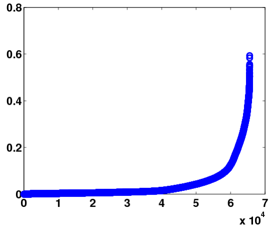
- Transform sparsity: $\mathbf{\Psi}\vec{s}$ is sparse, for some linear transformation $\mathbf{\Psi}$. Examples: gradient; wavelet transforms.
- Sparse representation: $\vec{s} = \mathbf{\Phi}\vec{x}$, for some sparse vector $\vec{x}$ and linear transformation $\mathbf{\Phi}$. Examples: wavelet transforms, *learned dictionaries*.

Sparsity

A vector $\vec{s}$ (or matrix) is called *sparse* if most of its entries are zero.

More general notions of sparsity include:

- Transform sparsity: $\Psi\vec{s}$ is sparse, for some linear transformation Ψ. Examples: gradient; wavelet transforms.
- Sparse representation: $\vec{s} = \Phi\vec{x}$, for some sparse vector $\vec{x}$ and linear transformation Φ. Examples: wavelet transforms, *learned dictionaries*.

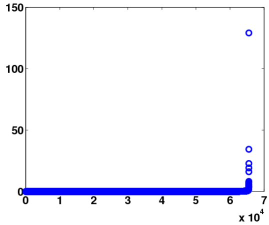


Sparsity

A vector $\vec{s}$ (or matrix) is called *sparse* if most of its entries are zero.

More general notions of sparsity include:

- Transform sparsity: $\Psi\vec{s}$ is sparse, for some linear transformation Ψ. Examples: gradient; wavelet transforms.
- Sparse representation: $\vec{s} = \Phi\vec{x}$, for some sparse vector $\vec{x}$ and linear transformation Φ. Examples: wavelet transforms, *learned dictionaries*.



Exploiting sparsity

We can find sparse solutions to linear systems by solving optimization problems with a sparsity-promoting penalty function.

$\min_{\vec{x}} P(\Psi\vec{x}),$ subject to $A\vec{x} = \vec{b},$

Exploiting sparsity

We can find sparse solutions to linear systems by solving optimization problems with a sparsity-promoting penalty function.

$\min_{\vec{x}} P(\Psi\vec{x}),$ subject to $A\vec{x} = \vec{b},$ or $\min_{\vec{x}} P(\vec{x}),$ subject to $A\Phi\vec{x} = \vec{b}.$

Exploiting sparsity

We can find sparse solutions to linear systems by solving optimization problems with a sparsity-promoting penalty function.

$$\min_{\vec{x}} P(\Psi\vec{x}), \text{ subject to } A\vec{x} = \vec{b}, \text{ or } \min_{\vec{x}} P(\vec{x}), \text{ subject to } A\Phi\vec{x} = \vec{b}.$$

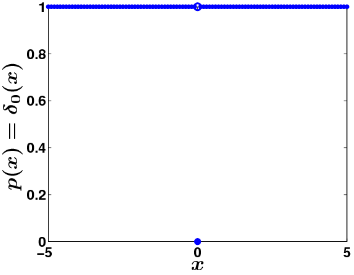
A good penalty function puts a premium on being nonzero.
Generally $P(\vec{x}) = \sum_{i=1}^{N} p(x_i)$ for some function p. Examples:

Exploiting sparsity

We can find sparse solutions to linear systems by solving optimization problems with a sparsity-promoting penalty function.

$$\min_{\vec{x}} P(\Psi \vec{x}), \text{ subject to } A\vec{x} = \vec{b}, \text{ or } \min_{\vec{x}} P(\vec{x}), \text{ subject to } A\Phi\vec{x} = \vec{b}.$$

A good penalty function puts a premium on being nonzero. Generally $P(\vec{x}) = \sum_{i=1}^{N} p(x_i)$ for some function p. Examples:



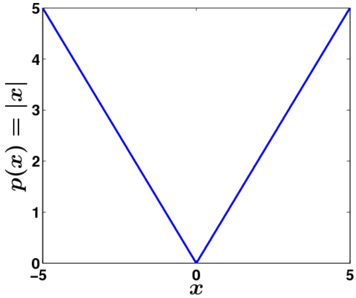
ideal, but can't solve

Exploiting sparsity

We can find sparse solutions to linear systems by solving optimization problems with a sparsity-promoting penalty function.

$\min_{\vec{x}} P(\Psi\vec{x}),$ subject to $A\vec{x} = \vec{b},$ or $\min_{\vec{x}} P(\vec{x}),$ subject to $A\Phi\vec{x} = \vec{b}.$

A good penalty function puts a premium on being nonzero.
Generally $P(\vec{x}) = \sum_{i=1}^{N} p(x_i)$ for some function p. Examples:



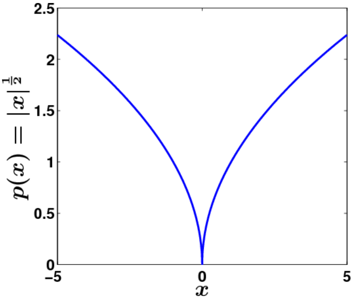
good, can solve fast

Exploiting sparsity

We can find sparse solutions to linear systems by solving optimization problems with a sparsity-promoting penalty function.

$$\min_{\vec{x}} P(\Psi\vec{x}), \text{ subject to } A\vec{x} = \vec{b}, \text{ or } \min_{\vec{x}} P(\vec{x}), \text{ subject to } A\Phi\vec{x} = \vec{b}.$$

A good penalty function puts a premium on being nonzero. Generally $P(\vec{x}) = \sum_{i=1}^{N} p(x_i)$ for some function p. Examples:



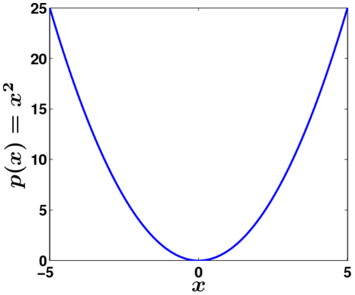
better, still can solve fast

Exploiting sparsity

We can find sparse solutions to linear systems by solving optimization problems with a sparsity-promoting penalty function.

$\min_{\vec{x}} P(\Psi \vec{x})$, subject to $A\vec{x} = \vec{b}$, or $\min_{\vec{x}} P(\vec{x})$, subject to $A\Phi\vec{x} = \vec{b}$.

A good penalty function puts a premium on being nonzero. Generally $P(\vec{x}) = \sum_{i=1}^{N} p(x_i)$ for some function p. Examples:



bad

Abstracting length

The penalty functions $P(\vec{x}) = \sum_{i=1}^{N} |x_i|^p$ can all be regarded as measuring some notion of “size” of $\vec{x}$.

Abstracting length

The penalty functions $P(\vec{x}) = \sum_{i=1}^{N} |x_i|^p$ can all be regarded as measuring some notion of “size” of $\vec{x}$.

When $p = 2$, this is the square of the length of the vector $\vec{x}$, often written $\|\vec{x}\|$. Following this example, we define a more general notion of length:

$$\|\vec{x}\|_p := \left(\sum_{i=1}^{N} |x_i|^p \right)^{1/p}.$$

This is called the ℓ^p *norm* of $\vec{x}$.

Abstracting length

The penalty functions $P(\vec{x}) = \sum_{i=1}^{N} |x_i|^p$ can all be regarded as measuring some notion of "size" of $\vec{x}$.

When $p = 2$, this is the square of the length of the vector $\vec{x}$, often written $||\vec{x}||$. Following this example, we define a more general notion of length:

$$||\vec{x}||_p := \left(\sum_{i=1}^{N} |x_i|^p \right)^{1/p}.$$

This is called the ℓ^p *norm* of $\vec{x}$.

(When $N = \infty$, ℓ^p itself is the set of sequences $\vec{x}$ where the infinite series $||\vec{x}||_p$ converges. This is a vector space, so one talks about "ℓ^p spaces." The ℓ is for Henri Lebesgue.)

Abstracting length

The penalty functions $P(\vec{x}) = \sum_{i=1}^{N} |x_i|^p$ can all be regarded as measuring some notion of “size” of $\vec{x}$.

When $p = 2$, this is the square of the length of the vector $\vec{x}$, often written $\|\vec{x}\|$. Following this example, we define a more general notion of length:

$$\|\vec{x}\|_p := \left(\sum_{i=1}^{N} |x_i|^p\right)^{1/p}.$$

This is called the ℓ^p *norm* of $\vec{x}$.

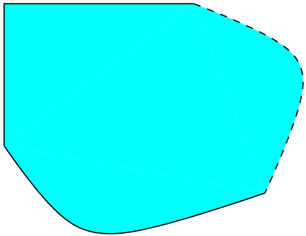
(When $N = \infty$, ℓ^p itself is the set of sequences $\vec{x}$ where the infinite series $\|\vec{x}\|_p$ converges. This is a vector space, so one talks about “ℓ^p spaces.” The ℓ is for Henri Lebesgue.)

Not quite one of these cases: $\|\vec{x}\|_0$ is the number of nonzero components of $\vec{x}$. Note $\|\vec{x}\|_0 = \lim_{p\to 0} \|x\|_p^p$.

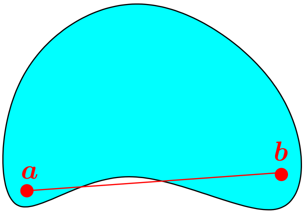
Outline

Convex sets

A set E is *convex* iff E contains the line segment joining any two points in E.

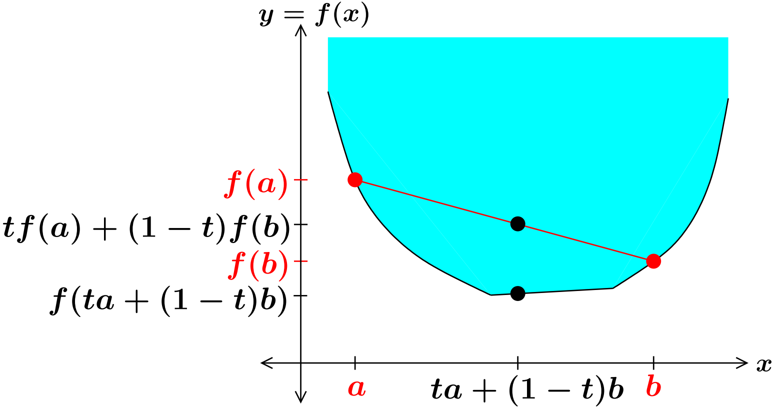


a convex set



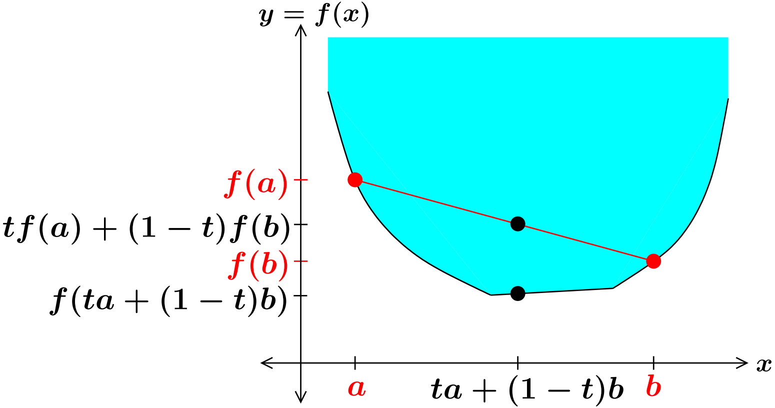
a nonconvex set

Convex functions



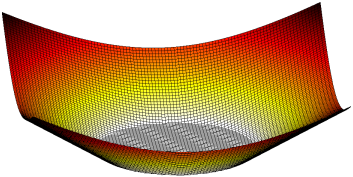
A function $f : X \to \bar{\mathbb{R}}$ is convex iff its *epigraph* is a convex set, where $\mathrm{epi}\, f = \{(x, y) \in X \times \bar{\mathbb{R}} : y \geq f(x)\}$,

Convex functions

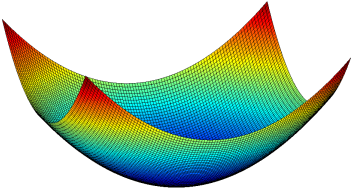


A function $f : X \to \bar{\mathbb{R}}$ is convex iff its *epigraph* is a convex set, where $\operatorname{epi} f = \{(x, y) \in X \times \bar{\mathbb{R}} : y \geq f(x)\}$, or, iff for all $a, b \in X$ and $t \in [0, 1]$, $f(ta + (1-t)b) \leq tf(a) + (1-t)f(b)$.

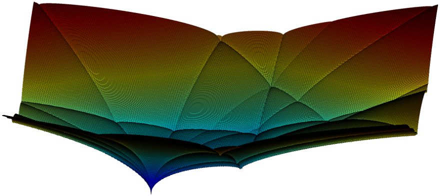
Convexity and optimization



a convex function



a *strictly* convex function



a nonconvex function

Convexity is desirable for optimization, as convex functions do not have nonglobal minimizers. *Strictly* convex functions have unique minimizers. Nonconvex functions can be very difficult to minimize.

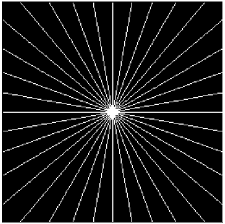
Outline

Phantom example

We reconstruct an image from samples of its Fourier transform:

$$\min_{\vec{x}} \|\nabla \vec{x}\|_p, \text{ subject to } (\mathcal{F}\vec{x})_i = (\mathcal{F}\vec{s})_i, \text{ for } i \in S.$$

test image $\vec{s}$



18 lines/6.9% sampled

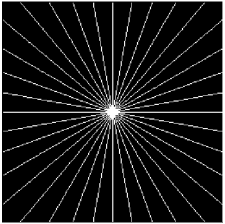
$p = 1$

Phantom example

We reconstruct an image from samples of its Fourier transform:

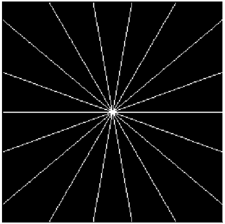
$$\min_{\vec{x}} \|\nabla \vec{x}\|_p, \text{ subject to } (\mathcal{F}\vec{x})_i = (\mathcal{F}\vec{s})_i, \text{ for } i \in S.$$

test image $\vec{s}$



18 lines/6.9% sampled

$p = 1$



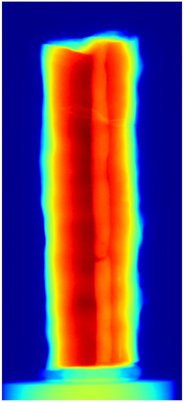
9 lines/3.5% sampled

$p = 1$

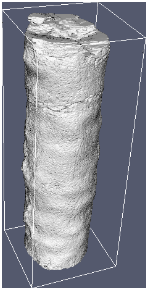
$p = 1/2$

3-D tomography

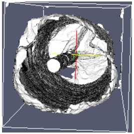
Six radiographs allow reconstruction of a stalagmite segment:



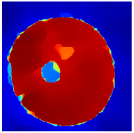
radiograph



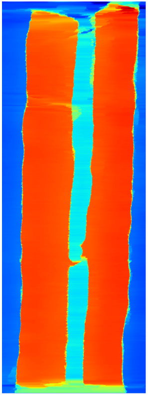
isosurface



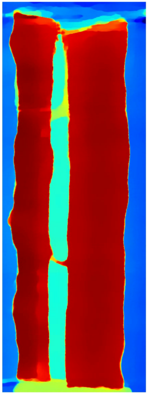
iso from end



z slice



x slice



y slice

(with Gary Sandine, LANL)

Shortening MRI scans

Synthetic MRI example, 256×256, 20% sampling, using both wavelet-transform and gradient penalty terms:

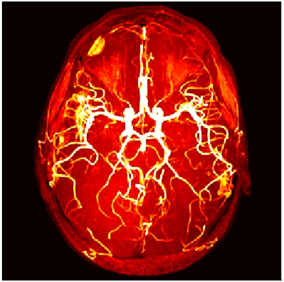
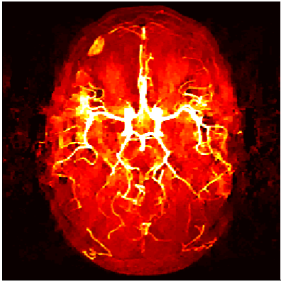
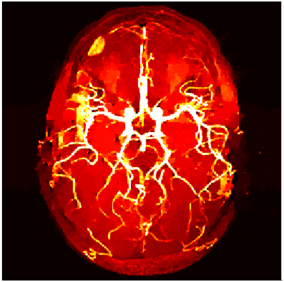


image synthetically sampled



reconstruction with $p = 1$



reconstruction with $p = 1/2$

Interferometric imaging

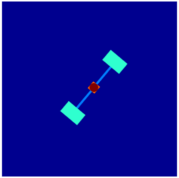
Given a network of N telescopes, the correlation between the electric field at each pair gives us $\binom{N}{2}$ Fourier-transform samples.

Interferometric imaging

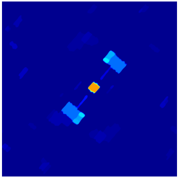
Given a network of N telescopes, the correlation between the electric field at each pair gives us $\binom{N}{2}$ Fourier-transform samples. Radio astronomers use Earth-rotation synthesis to increase the sampling. But what about geostationary objects?

Interferometric imaging

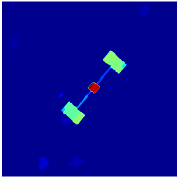
Given a network of N telescopes, the correlation between the electric field at each pair gives us $\binom{N}{2}$ Fourier-transform samples. Radio astronomers use Earth-rotation synthesis to increase the sampling. But what about geostationary objects?



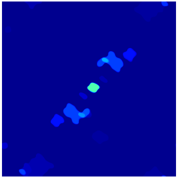
test image



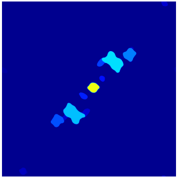
16 telescopes, $p = 1$



16 telescopes, $p = 1/2$



10 telescopes, $p = 1$



10 telescopes, $p = 1/2$

Why might global minimization be possible?

Consider a smoothed penalty function, restricted to the feasible plane $A\vec{x} = \vec{b}$:

$$\sum_{i=1}^{N} (x_i^2 + \epsilon)^{p/2}.$$

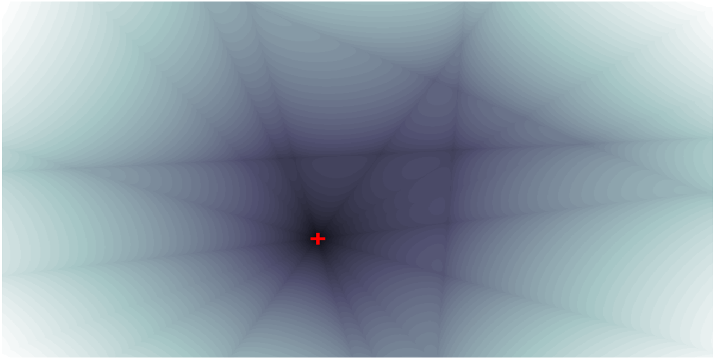
A moderate ϵ fills in the local minima.

Why might global minimization be possible?

Consider a smoothed penalty function, restricted to the feasible plane $A\vec{x} = \vec{b}$:

$$\sum_{i=1}^{N} (x_i^2 + \epsilon)^{p/2}.$$

A moderate ϵ fills in the local minima.



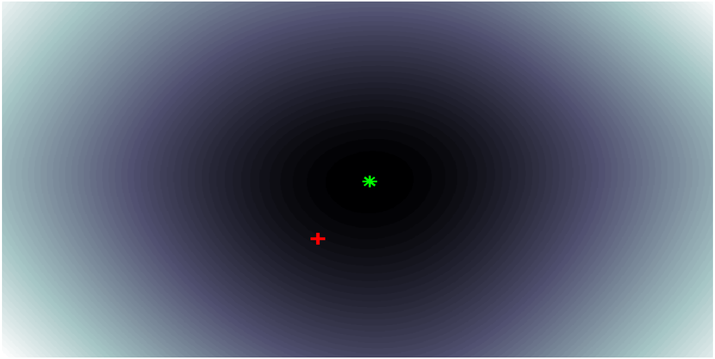
$\epsilon = 0$

Why might global minimization be possible?

Consider a smoothed penalty function, restricted to the feasible plane $A\vec{x} = \vec{b}$:

$$\sum_{i=1}^{N} (x_i^2 + \epsilon)^{p/2}.$$

A moderate ϵ fills in the local minima.



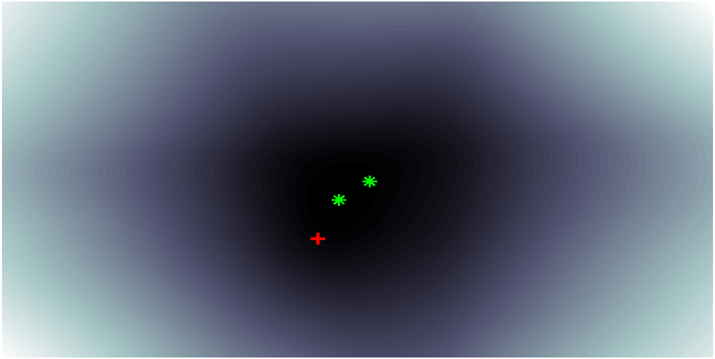
$\epsilon = 1$

Why might global minimization be possible?

Consider a smoothed penalty function, restricted to the feasible plane $A\vec{x} = \vec{b}$:

$$\sum_{i=1}^{N} (x_i^2 + \epsilon)^{p/2}.$$

A moderate ϵ fills in the local minima.



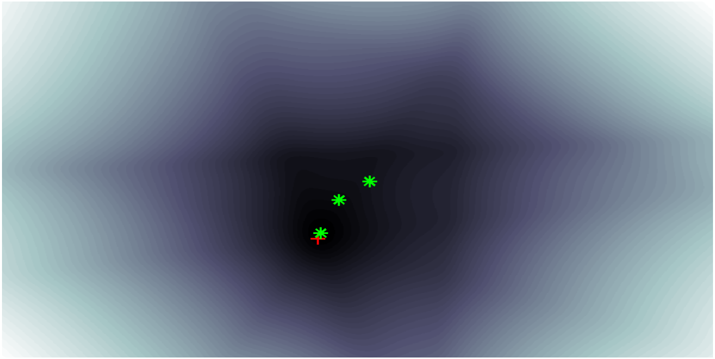
$\epsilon = 0.1$

Why might global minimization be possible?

Consider a smoothed penalty function, restricted to the feasible plane $A\vec{x} = \vec{b}$:

$$\sum_{i=1}^{N} (x_i^2 + \epsilon)^{p/2}.$$

A moderate ϵ fills in the local minima.



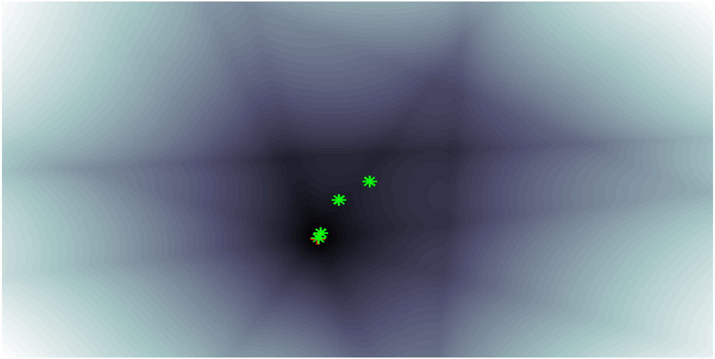
$\epsilon = 0.01$

Why might global minimization be possible?

Consider a smoothed penalty function, restricted to the feasible plane $A\vec{x} = \vec{b}$:

$$\sum_{i=1}^{N} (x_i^2 + \epsilon)^{p/2}.$$

A moderate ϵ fills in the local minima.



$\epsilon = 0.001$

Summary

- Many imaging problems can be tackled mathematically.
- Denoising with nonconvex penalty functions preserves shapes better.
- Nonconvex penalty functions also allow more severely underdetermined linear inverse problems to be solved.
- Smoothing the penalty function appears to keep algorithms from converging to nonglobal minima.

`math.lanl.gov/~rick`

Summary

- Many imaging problems can be tackled mathematically.
- Denoising with nonconvex penalty functions preserves shapes better.
- Nonconvex penalty functions also allow more severely underdetermined linear inverse problems to be solved.
- Smoothing the penalty function appears to keep algorithms from converging to nonglobal minima.

`math.lanl.gov/~rick`

Summary

- Many imaging problems can be tackled mathematically.
- Denoising with nonconvex penalty functions preserves shapes better.
- Nonconvex penalty functions also allow more severely underdetermined linear inverse problems to be solved.
- Smoothing the penalty function appears to keep algorithms from converging to nonglobal minima.

`math.lanl.gov/~rick`

Summary

- Many imaging problems can be tackled mathematically.
- Denoising with nonconvex penalty functions preserves shapes better.
- Nonconvex penalty functions also allow more severely underdetermined linear inverse problems to be solved.
- Smoothing the penalty function appears to keep algorithms from converging to nonglobal minima.

`math.lanl.gov/~rick`