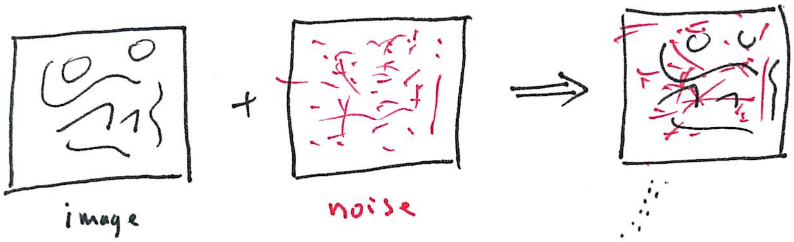
PCMI Lecture #6

Image Processing and graphs: diffusion, spanning trees, etc.

we return now to the problem of image denoising as a motivational starting point.

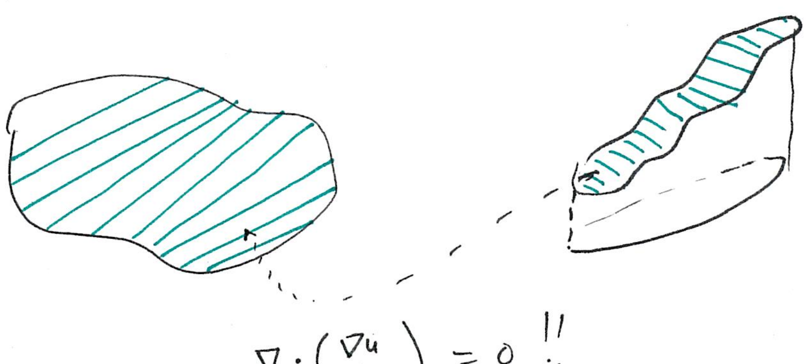


$\int |\nabla u|^2$ does not work so well

So ... we changed the variational energy (differential operator) to get something that didn't simply smooth.

E.g. $\int |\nabla u| \Longrightarrow \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right)$ small is good

Fact:

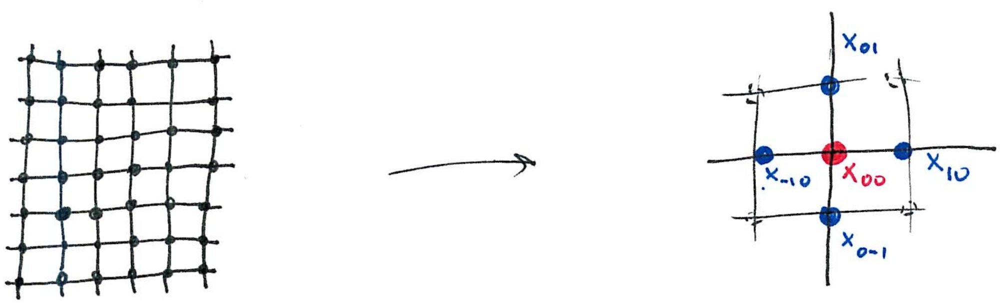


$$\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) = 0 \;!!$$

This operator only diffuses "along" level sets not across them.
That is curvature induces evolution trying to reduce curvature of level sets.

how about modifying domain instead of operator?

For this we need to look at precisely what the domain is and how the operators are expressed.



how smooth something is can be measured by how big its laplacian is. I.e. how smooth f is = how big is $|\Delta f|$.

$$\left(f(x_{10}) - f(x_{00})\right) - \left(f(x_{00}) - f(x_{-10})\right)$$
$$+ \left(f(x_{01}) - f(x_{00})\right) - \left(f(x_{00}) - f(x_{0-1})\right)$$

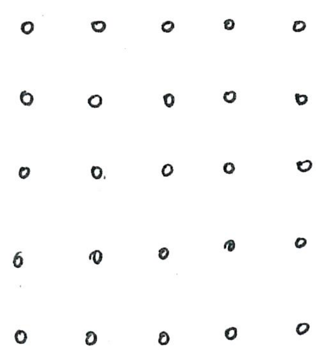
(where we have assumed $\Delta x = \Delta y = 1$)

$$\Delta f \underset{\text{discrete}}{=} f(x_{10}) + f(x_{-10}) + f(x_{01}) + f(x_{0-1}) - 4 f(x_{00})$$

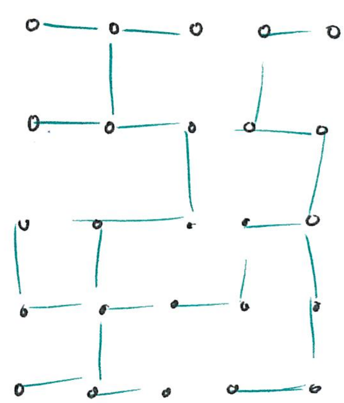
For $\Delta x, \Delta y$ variable, $\neq 1$ we get some weights; in particular when $\Delta x = \Delta y = h \Rightarrow$ a factor of $\frac{1}{h^2}$ out front.

$\Delta f = 0 \Rightarrow f$ harmonic $\Rightarrow f =$ it's average.

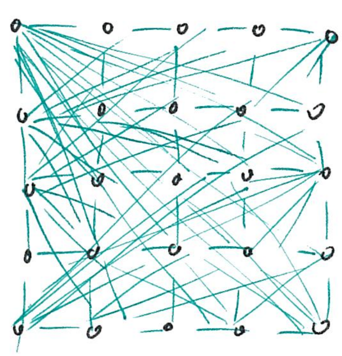
Idea: Change the domain by changing the weights / connections
(this is what we will mean by changing the domain)



MST



Non-local



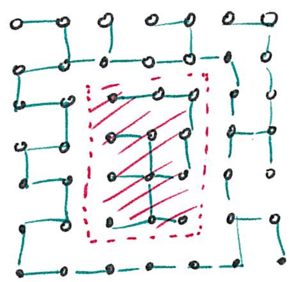
Both are subsets of full graph.

minimal spanning tree

MST idea:

Find a spanning tree that is a subgraph of the usual 4-neighbor graph, taking into account the weights and looking for one that minimizes weights on connections.

weights = pixel differences.



$\Rightarrow$ now run the discrete version of $\Delta u = u_f$ on it... i.e. something like

$$\int |\nabla u|^2 dx + \lambda \int |u-f|^2 dx$$

Actual Implementation: we make non-parametric modifications to the MST

Things like

- halt tree building when every node is connected to at least one other node; now add every edge with less weight than the biggest weight added so far.

(show some results from paper)

NL-means

Non-Local idea:

Go in the other direction... use fully connected graphs and prune it or weight it by how similar the pixels are.

More precisely:

$x_{i=1,\dots,N}$

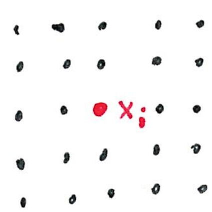
N pixels

$$\tilde{f}(x_i) = \sum_{j=1}^{N} w_{ij}\, f(x_j)$$

where: example

$$w_{ij} = \left| \frac{v(x_i)\cdot v(x_j)}{\|v(x_i)\|\,\|v(x_j)\|} \right| \Bigg/ \sum_j \left| \frac{v_i \cdot v_j}{\|v_i\|\,\|v_j\|} \right|$$

and v_i = ~~f~~ on the 25 pixels centered on x_i $= \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_i) \\ \vdots \\ f(x_n) \end{pmatrix}$ 25



We might, to speed things up, ignore all w_{ij} (set them to zero) if $\left| \dfrac{v_i \cdot v_j}{\|v_i\|\,\|v_j\|} \right| < \varepsilon$

This was introduced by Buades, Coll and Morel and is very good at denoising. (show pictures from Big paper)

~~Use a [illegible]~~ same lecture !!

Comparison:

The Asaki et al. paper looks at how good we can do with smoothing using a minimal, non-parametric perturbation to the usual 4-neighbor graph.

In particular, the point of the method is <u>not</u> to solve once and for all the denoising problem: it is more of a careful, surprising study.

The Buades et al. paper says to heck with small perturbations of the usual 4-neighbor graph: let's let the data tell us who the neighbors should be.

The results indicate that this is, for some purposes, the right approach.

It makes sense... if you are going to average, average with pixels who "look" alike.

Back to smoothing

$$\Delta f = f_{10} + f_{01} + f_{-10} + f_{0-1} - 4 f_{00}$$

identity

$$= C f - 4 I f$$

connectivity matrix

$$= \underbrace{(C - 4 I)}_{\text{symmetric}} f$$

... and non-positive definite*

* actually, I have only checked this for the continuous analog and the 1-D case... but it should work.

Detour

This leads off on anoth tangent (normal?)

Diffusion Maps

Key idea is that we can easily define a diffusion on any graph using the connections & weights on edges to build an operator.

[Coifman et al. al before that a bunch of people d work]

Graph Laplacian: (I would call it $-\Delta G$)

$$L = \{k_{ij}\} = \begin{cases} 1 & \text{if } i = j \\ -\frac{1}{\sqrt{d_i d_j}} & \text{if } i \text{ \& } j \text{ are connected} \\ 0 & \text{if } i \text{ \& } j \text{ are not connected} \end{cases}$$

$\Rightarrow$ has complete eigenspace, ~~positive ei~~ non-negative eigenvalues.

one can simply compute eigenfunctions $\phi_{1,\dots,N}$

and then

$$\text{map } x_i \longrightarrow \{\phi_1(x_i), \phi_2(x_i) \dots\}$$

Coifman, Lafon, et al ~~map~~ construct this map by starting with a kernel

$$K(x,y)$$ (symmetric, nonnegative definite)

measures similarity. Then

note that $\frac{K(x,y)}{\nu(x)}$ is a markov process.

$$\nu(x) = \int k(x,y)\, d\mu(y)$$

$$\bar{k}(x,y) = \frac{k(x,y)}{\sqrt{\nu(x)\,\nu(y)}}$$

now note that

$$K f(x) \equiv \int \bar{K}(x,y)\, f(y)\, dy$$

~~this~~ will have an orthonormal basis with non-negative eigenvalues

$$\Rightarrow \quad \bar{K}(x,y) = \sum_i \lambda_i^2\, \phi_i(x)\, \phi_i(y)$$

$$x \Rightarrow \begin{pmatrix} \lambda_1^2\, \phi_1(x) \\ \lambda_2^2\, \phi_2(x) \\ \lambda_3^2\, \phi_3(x) \\ \vdots \\ \lambda_N^2\, \phi_N(x) \end{pmatrix}$$

Diffusion map

Diffusion distance → Euclidean distance in the Range space of the Diffusion map

Note: There is nothing in the Diffusion map approach particular to images, even though it has been applied to images.

End of Detour

Image application: patches in images

$\Downarrow$

this connects us to the NL means approach.

E.g. we could compute similarities of patches and then run diffusion with noisy image as initial condition.