$$D_x = \frac{\partial}{\partial x} + A$$

Covariant Derivatives and Vision

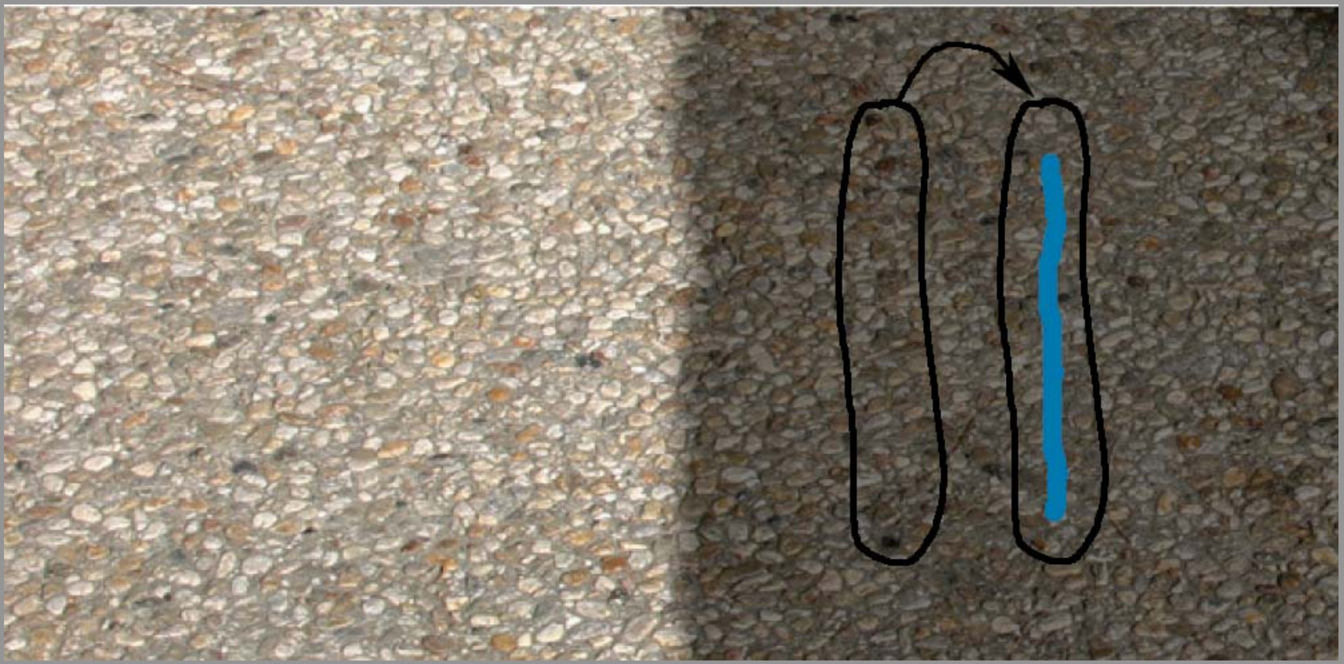
Todor Georgiev

Adobe Photoshop

Presentation at ECCV 2006

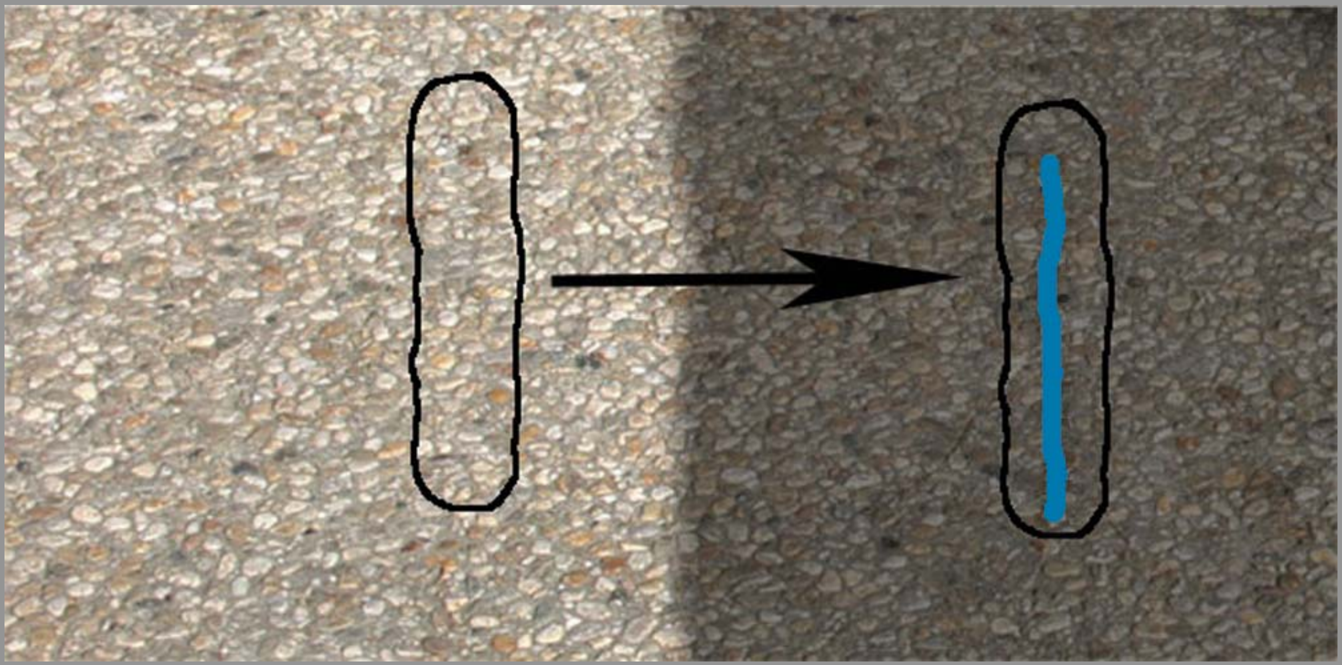
Original

Selection to clone



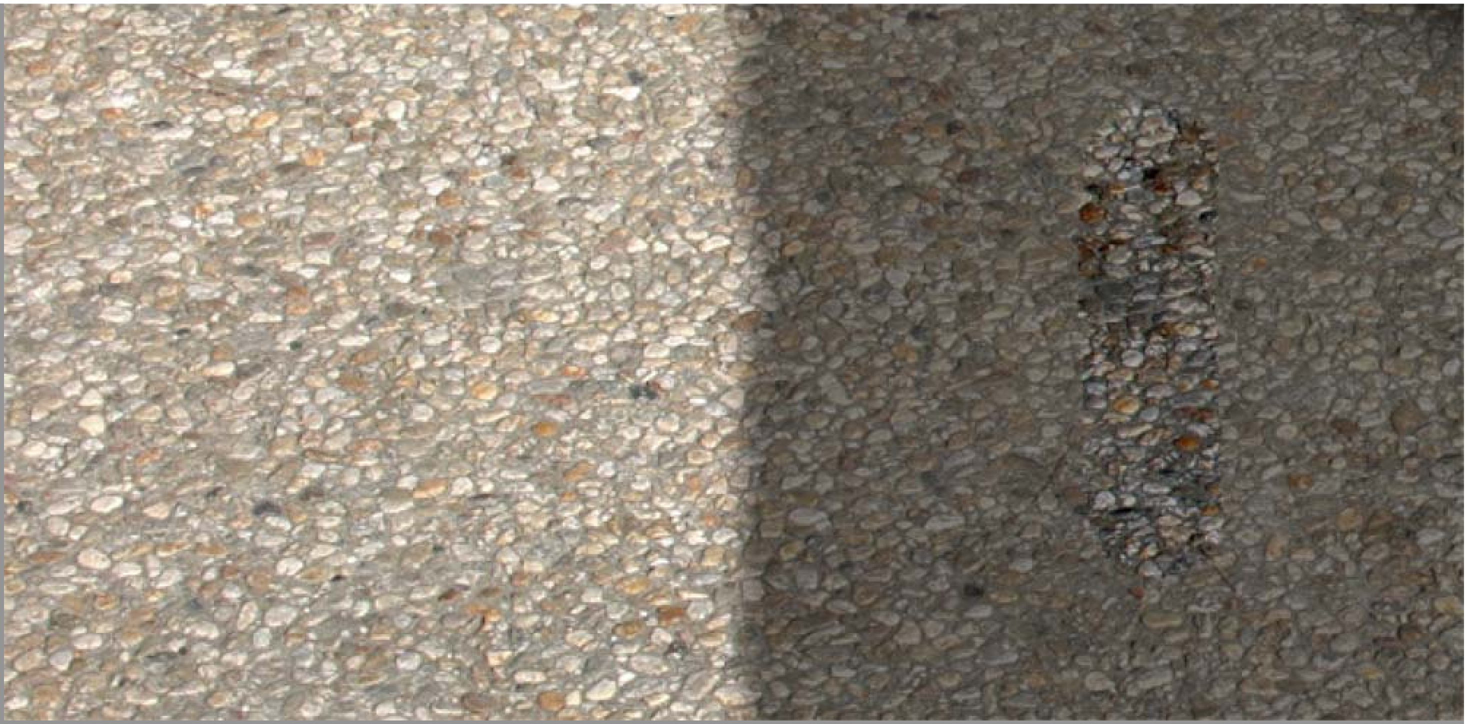
Poisson cloning from dark area

Selection to clone



Poisson cloning from illuminated area

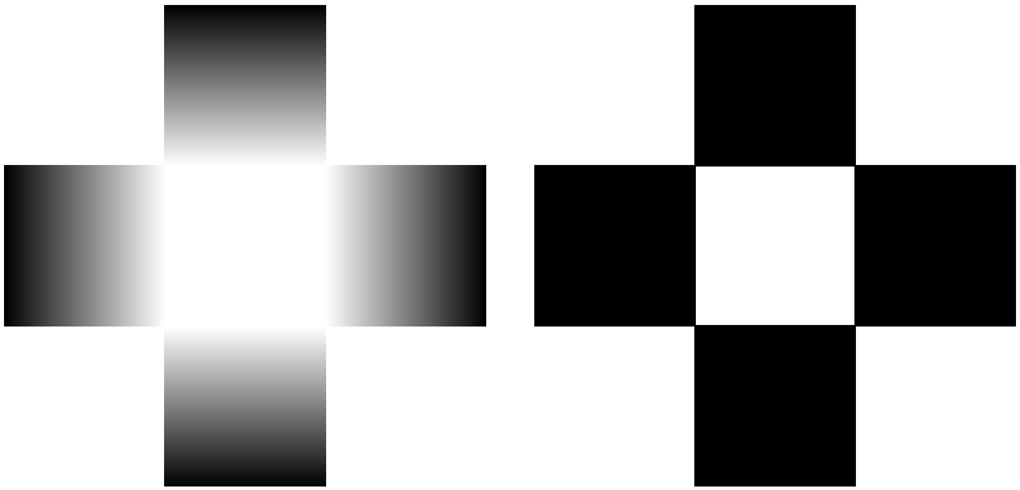
Poisson cloning



Covariant cloning (see next)

Poisson cloning can be viewed as an approximation to covariant cloning.

Outline of our theory:



Thanks to Jan Koenderink

Retina / Cortex Adaptation

- **The image is *just a record* of pixel values.**
- **We do not see pixel values directly.**
- **What we see is *an illusion* generated from the above record through internal adaptation. We can not compare pixels.**

Models of Image Space:

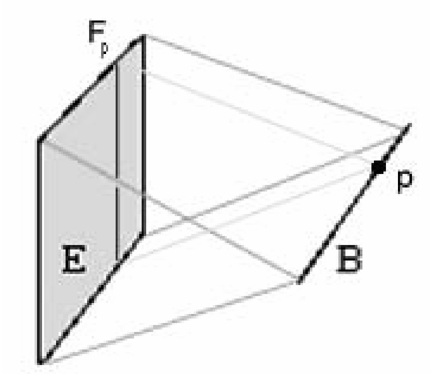
(1) Cartesian Product

- **A pair (location, intensity)**
- **Multiple copies of the intensity line.**
- **We can compare intensities. The image is a function that specifies an intensity at each point.**

(2) Fiber Bundle

- **Two spaces and a mapping (vertical projection)**

Total space E



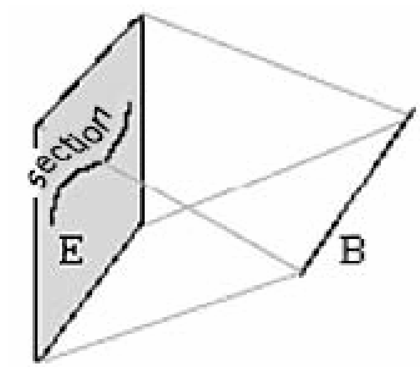
Base space B

- **Fiber is the set of points that map to a single point. We will use vector bundles, where fibers are vector spaces.**

Section in a Fiber Bundle

- **Mapping from base B to total space E**

Sections
replace
functions



- **We can not compare intensities. Horizontal projection is not defined. We have forgotten it.**

- **Perceptually correct model of the image**
  Image = graph of a section

Derivatives in a Fiber Bundle

Definition:

Derivative is a mapping from one section to another that satisfies the Leibniz rule relative to multiplication by functions:

$$D_x(f\sigma) = (\tfrac{\partial}{\partial x} f)\sigma + f D_x \sigma$$

In the Cartesian product space this definition is equivalent to the conventional derivative.

Derivatives in a Fiber Bundle

If we express a section as a linear combination of some basis sections

$$\sigma = \Sigma f^i \sigma_i$$

then the derivative will be:

$$\begin{aligned} D_x\sigma = D_x\Sigma(f^i\sigma_i) &= \Sigma((\frac{\partial}{\partial x}f^i)\sigma_i + f^i D_x\sigma_i) \\ &= \Sigma((\frac{\partial}{\partial x}f^i)\sigma_i + \Sigma f^i A^j{}_{ix}\sigma_j) \end{aligned}$$

Derivatives in a Fiber Bundle

If we represent the section in terms of the functions f^i that define it in a given basis (not writing the basis vectors), the last equation can be written as:

$$D_x f^i = \frac{\partial}{\partial x} f^i + \Sigma A^i{}_{j\,x} f^j$$

The functions f^i are called "color channels" in Photoshop, and D is called "*Covariant Derivative*". It corresponds to the derivative in the Cartesian product model.

The *covariant derivative* is a rigorous mathematical tool for perceptual pixel comparison in the fiber bundle model of image space. It replaces the conventional derivative of the Cartesian product model as:

$$\frac{\partial}{\partial x} \rightarrow \frac{\partial}{\partial x} + A_x(x,y)$$

$$\frac{\partial}{\partial y} \rightarrow \frac{\partial}{\partial y} + A_y(x,y)$$

$$\frac{\partial}{\partial x}\frac{\partial}{\partial x}f + \frac{\partial}{\partial y}\frac{\partial}{\partial y}f = 0$$

$$\frac{\partial}{\partial x} \rightarrow \frac{\partial}{\partial x} + A_x(x,y)$$

$$\frac{\partial}{\partial y} \rightarrow \frac{\partial}{\partial y} + A_y(x,y)$$

$$(\frac{\partial}{\partial x} + A_x)(\frac{\partial}{\partial x} + A_x)f + (\frac{\partial}{\partial y} + A_y)(\frac{\partial}{\partial y} + A_y)f = 0$$

$$\triangle f + f div\mathbf{A} + 2\mathbf{A}\cdot grad f + \mathbf{A}\cdot\mathbf{A}f = 0 \qquad \triangle = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

- Reconstructing images with the covariant Laplace equation

$$\triangle f + f div\mathbf{A} + 2\mathbf{A} \cdot grad f + \mathbf{A} \cdot \mathbf{A} f = 0,$$

based on adaptation vector field, ***A***.

- Reconstructing surfaces based on gradient field. Recent work by R. Raskar et. al. Covariant Laplace should produce better results than Poisson.

How can we know A?

It can be extracted based on the idea of ***covariantly constant section, next***:

Assume perceived gradient of image $g(x, y)$ is zero.
This means complete adaptation:

$$(\frac{\partial}{\partial x} + A_x(x,y))g(x,y) = 0$$

$$(\frac{\partial}{\partial y} + A_y(x,y))g(x,y) = 0$$

$$\mathbf{A}(x,y) = -\frac{gradg}{g}$$

Substitute in covariant Laplace:

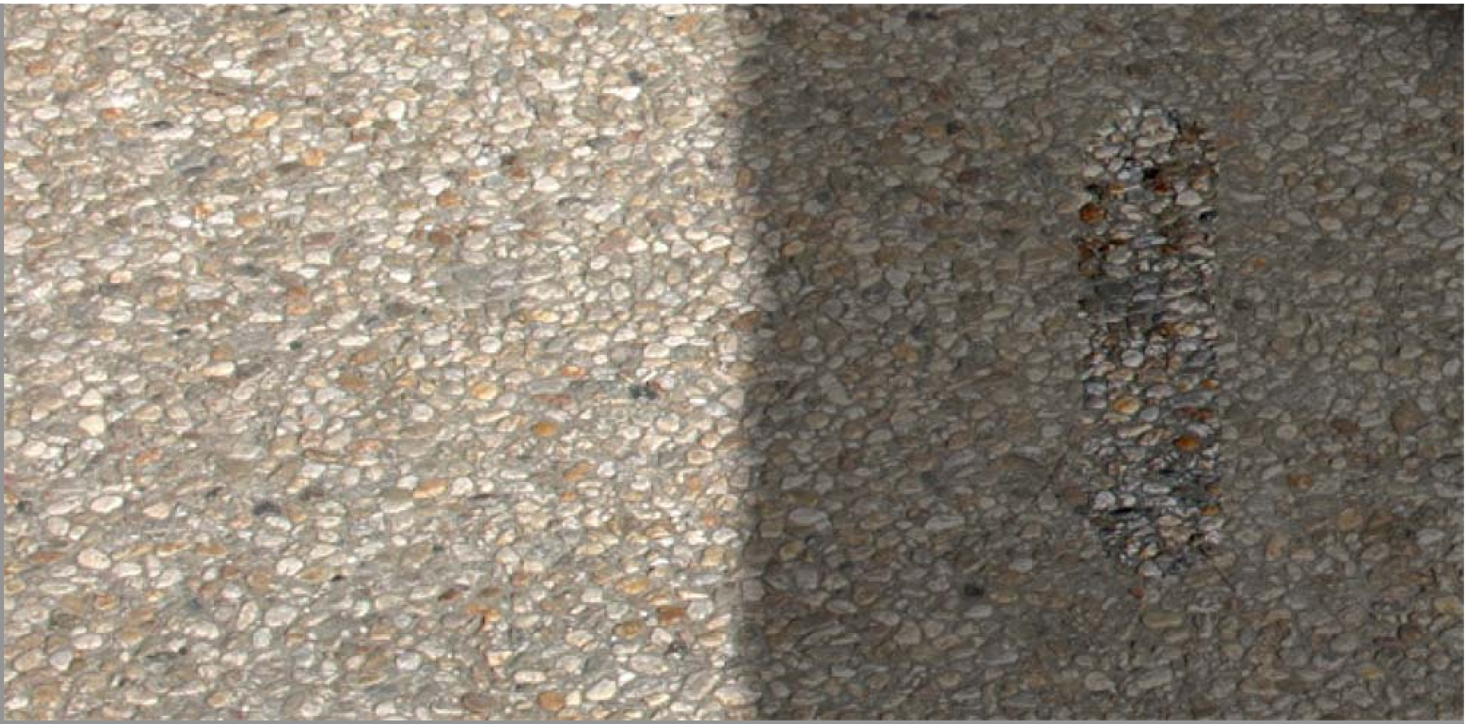
$$\frac{\triangle f}{f} - 2\frac{gradf}{f} \cdot \frac{gradg}{g} - \frac{\triangle g}{g} + 2\frac{(gradg)\cdot(gradg)}{g^2} = 0$$

Covariant cloning

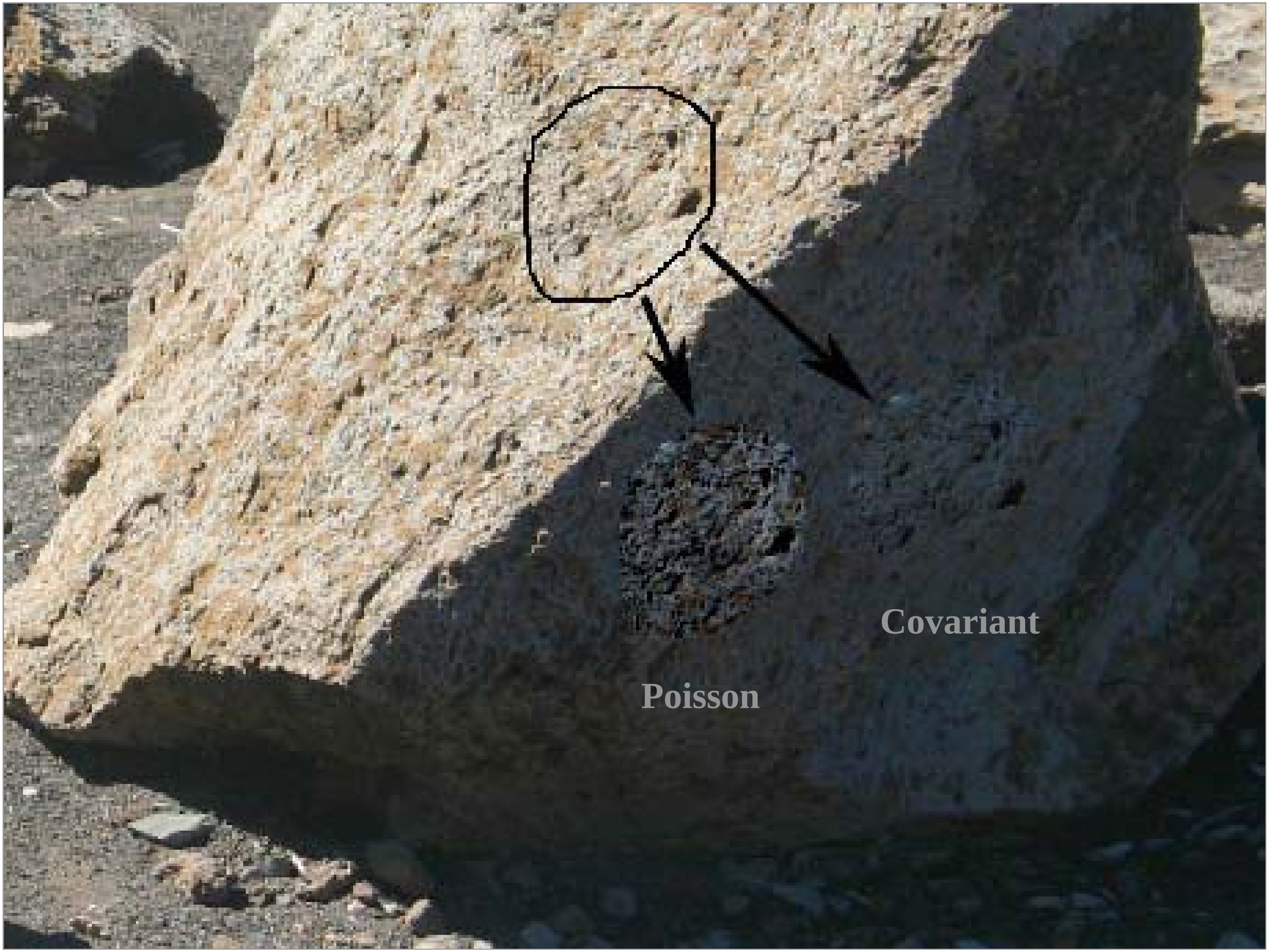
$$\triangle f(x,y) = \triangle g(x,y)$$

Poisson equation

Poisson cloning

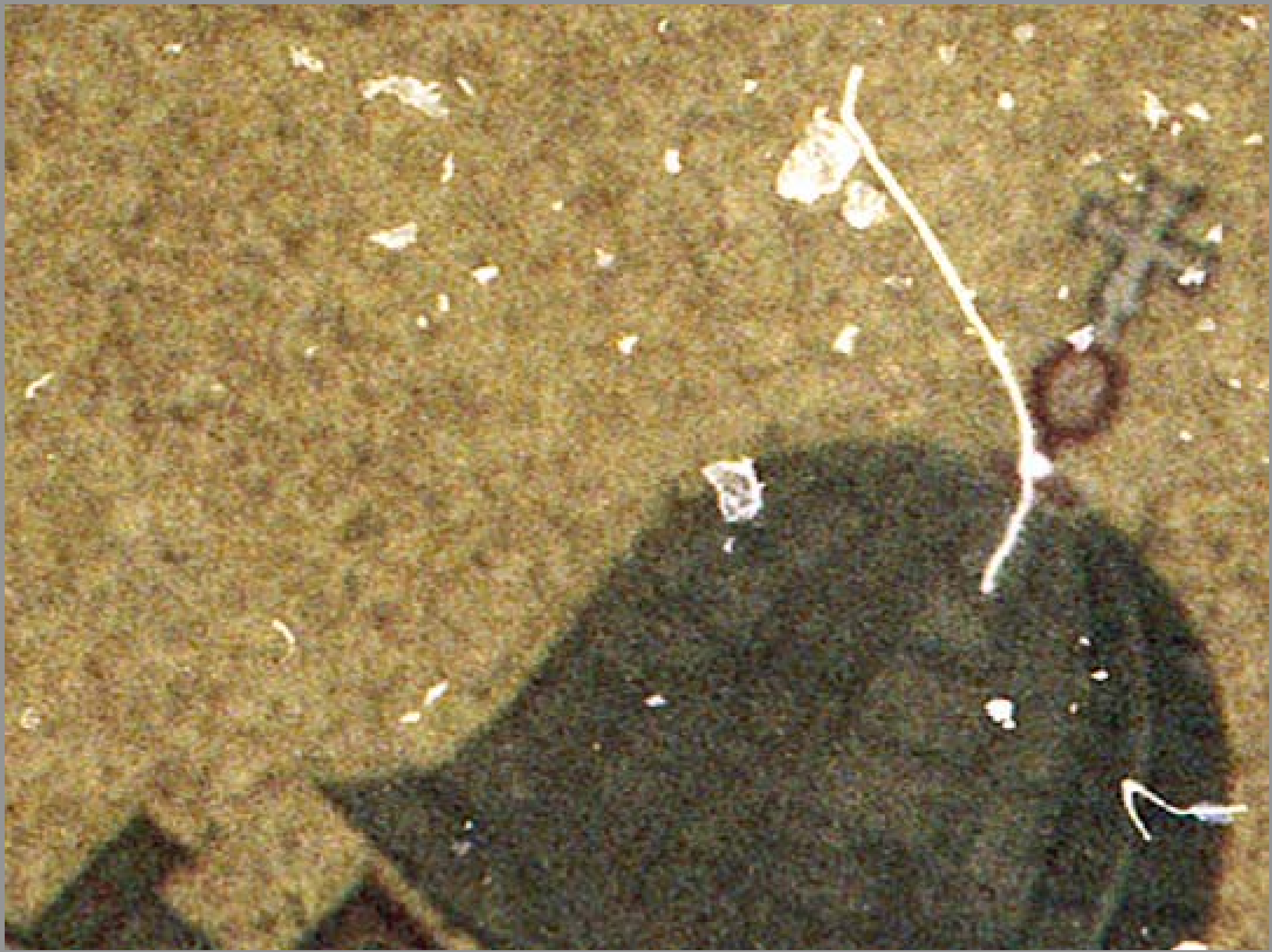


Covariant cloning

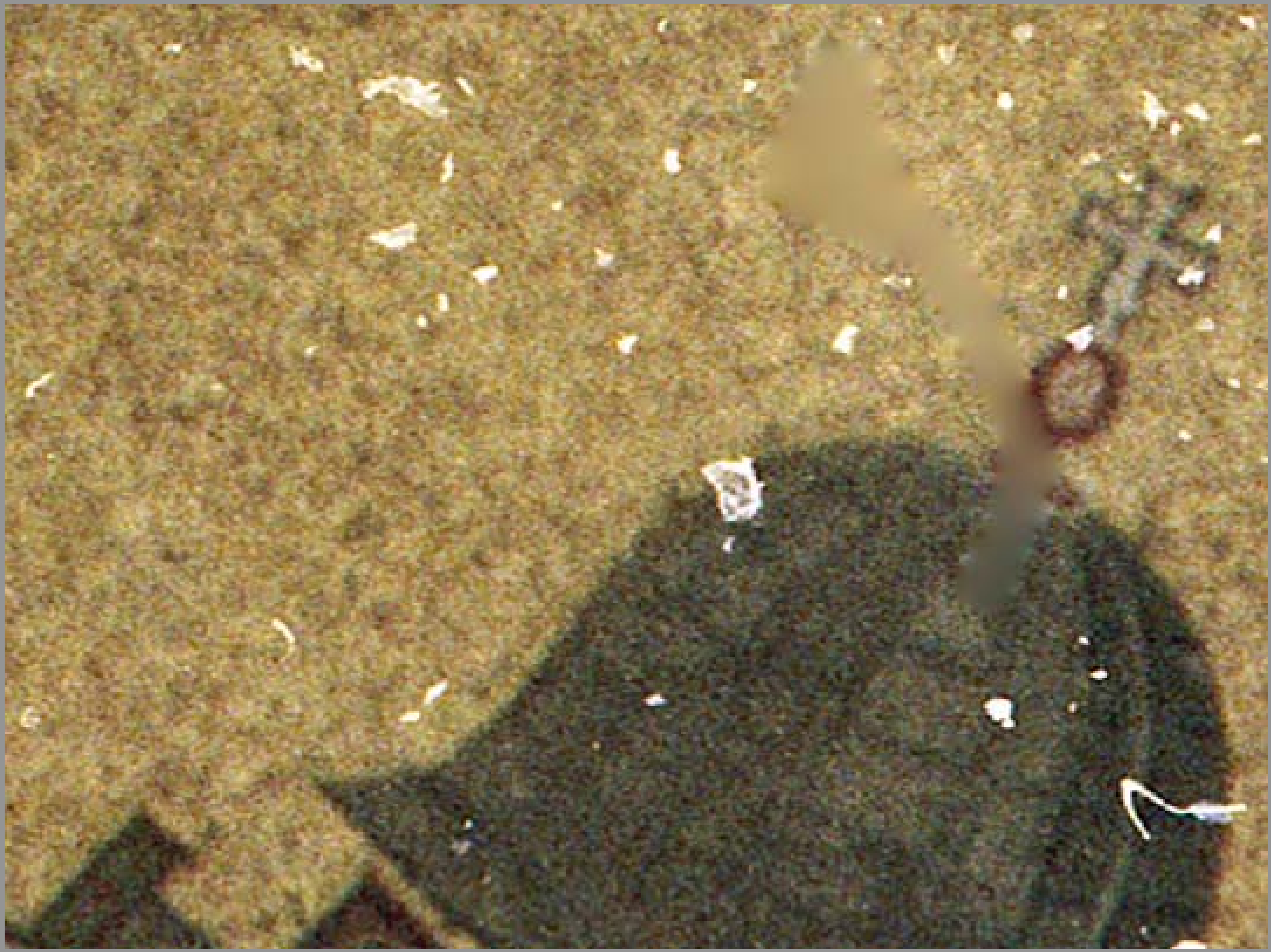

Covariant
Poisson

Detailed Example:

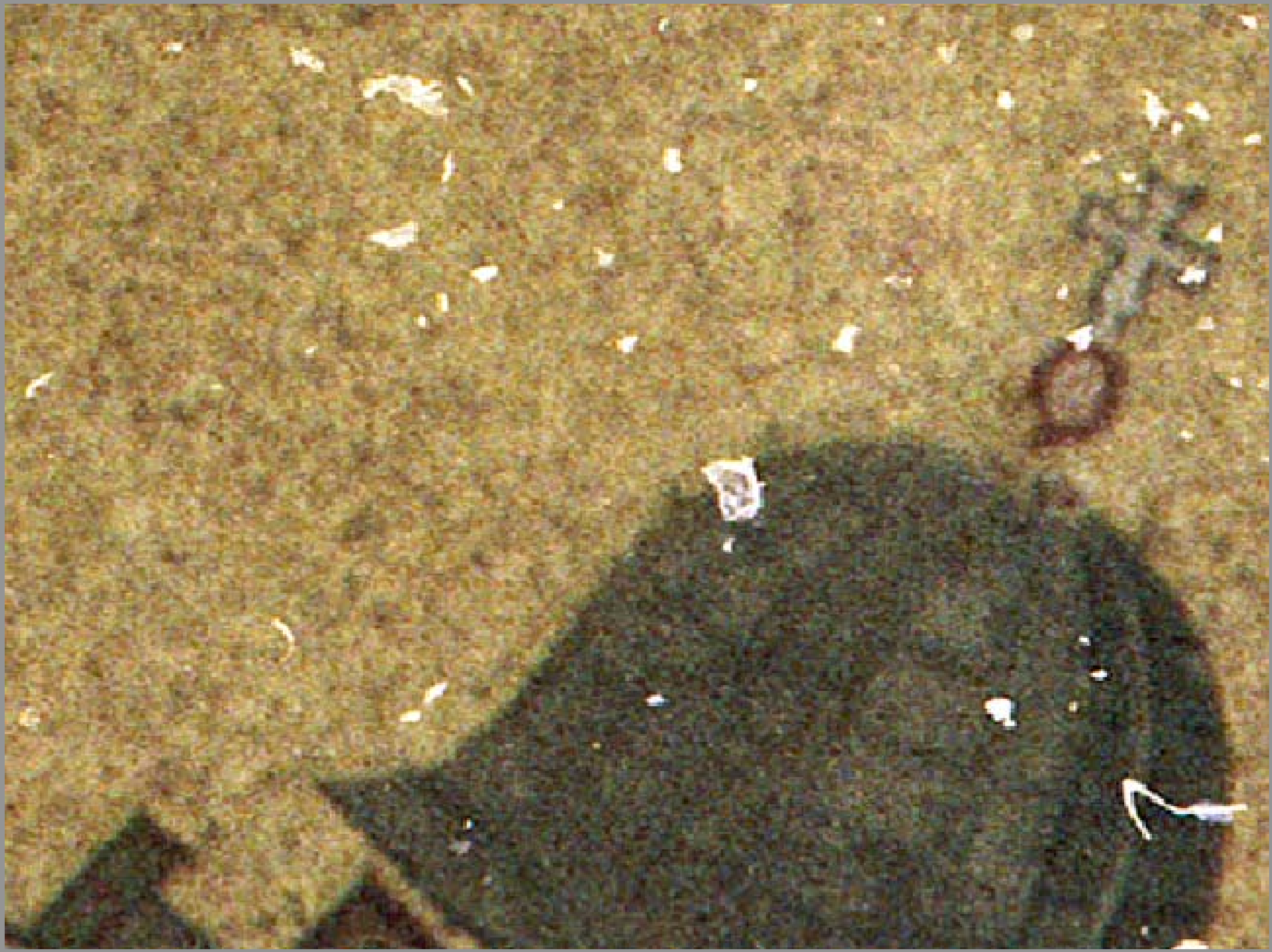
Original Damaged Area



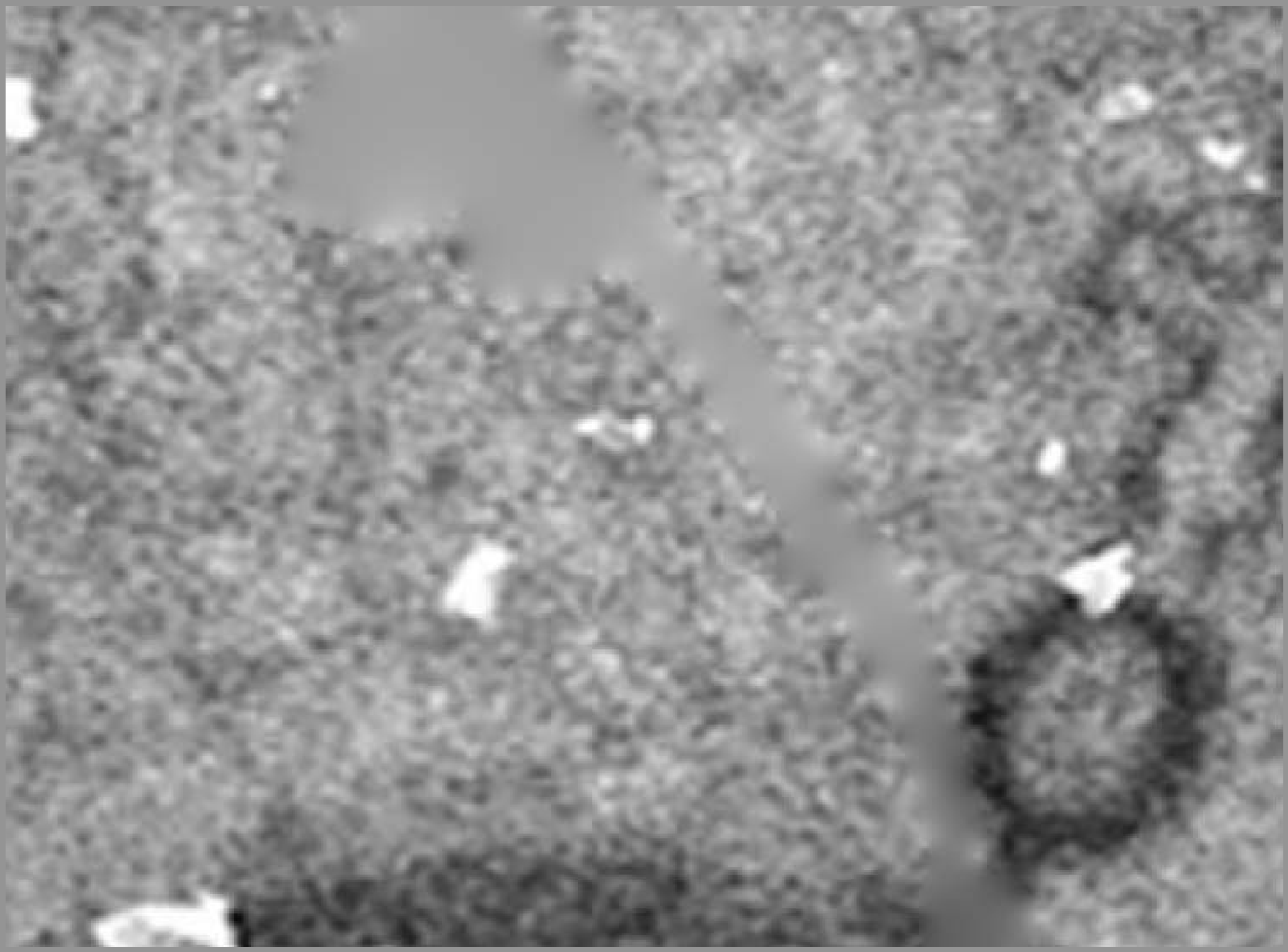
Laplace



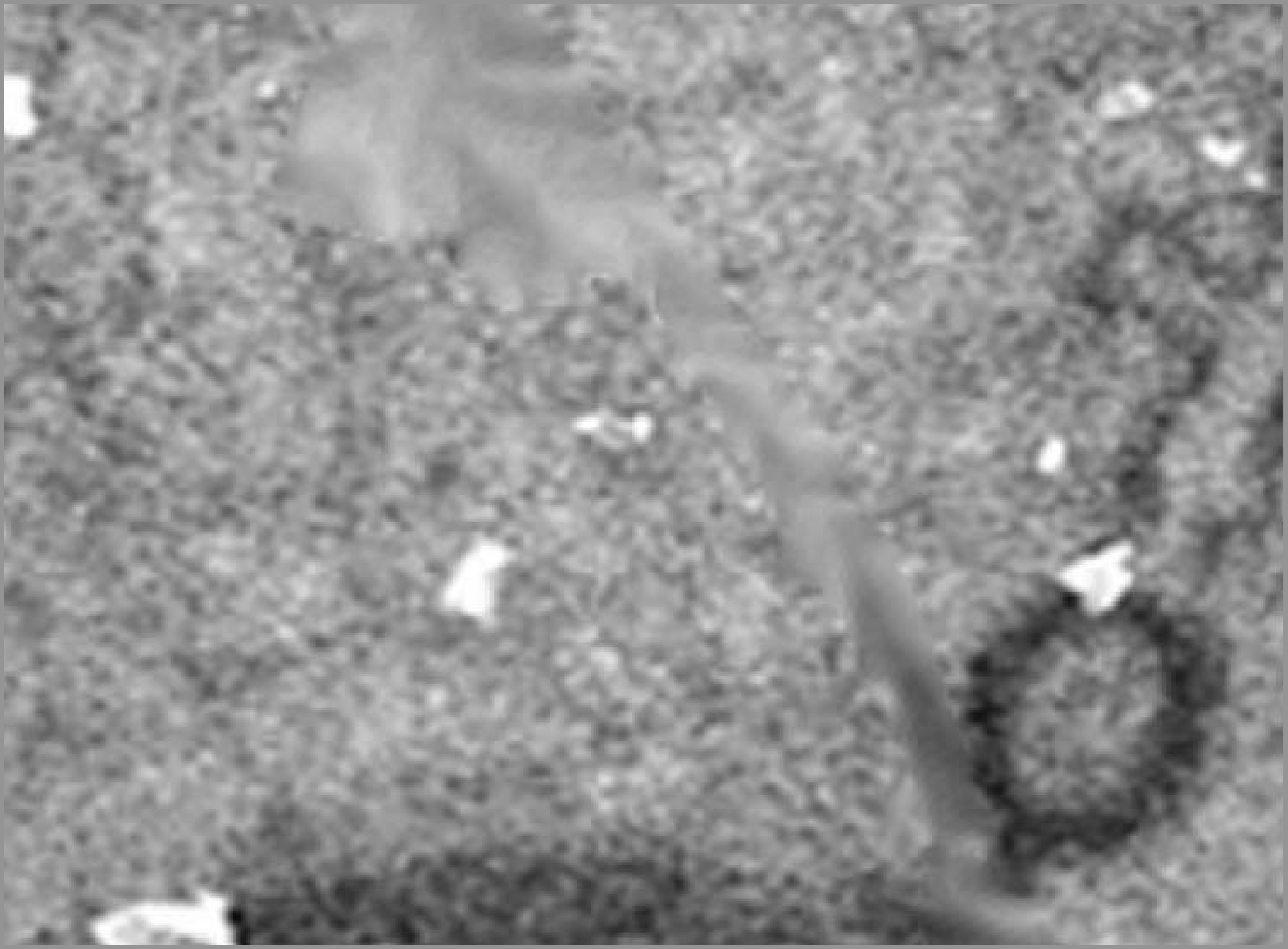
Poisson



Laplace

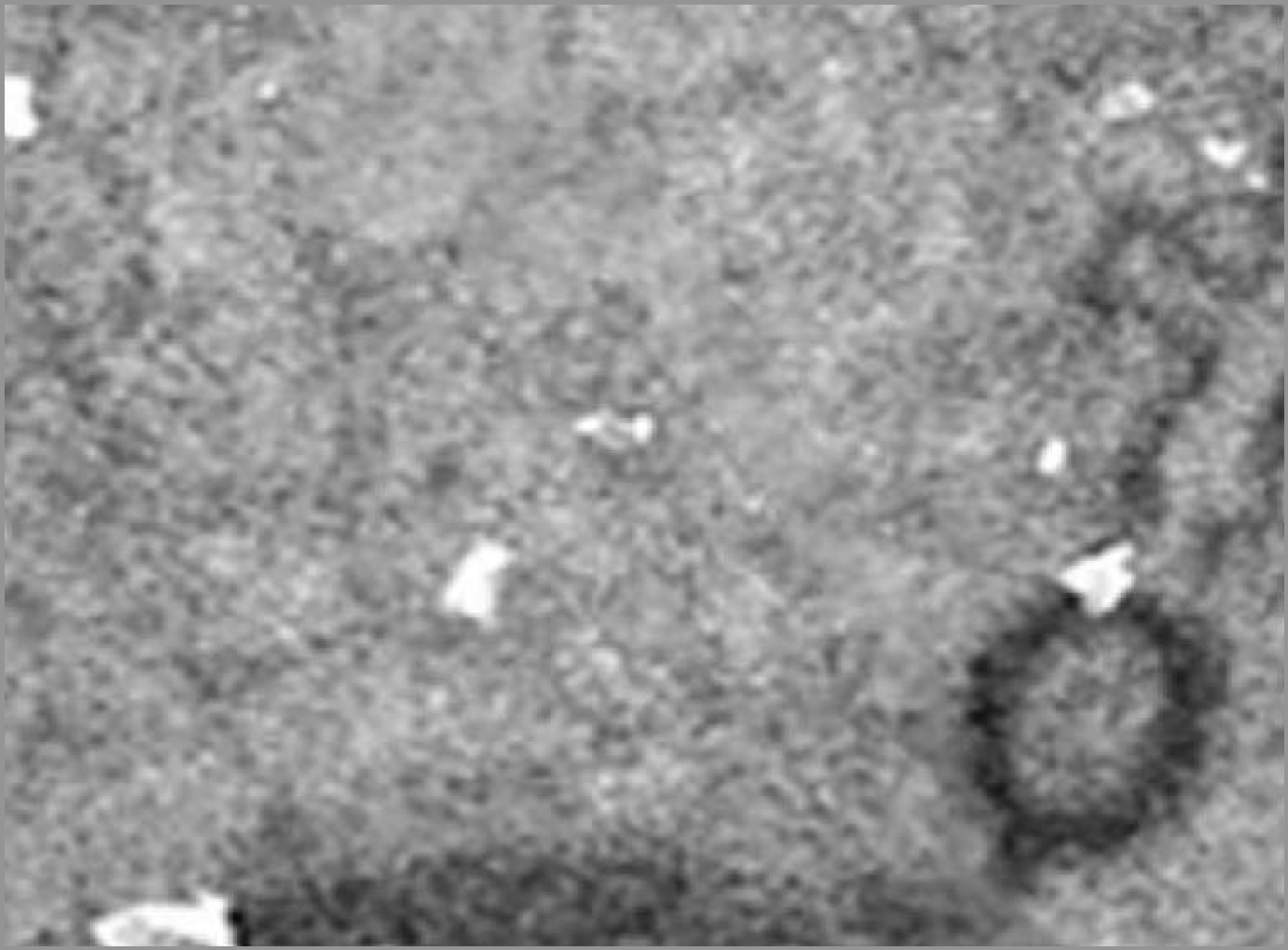


Inpainting

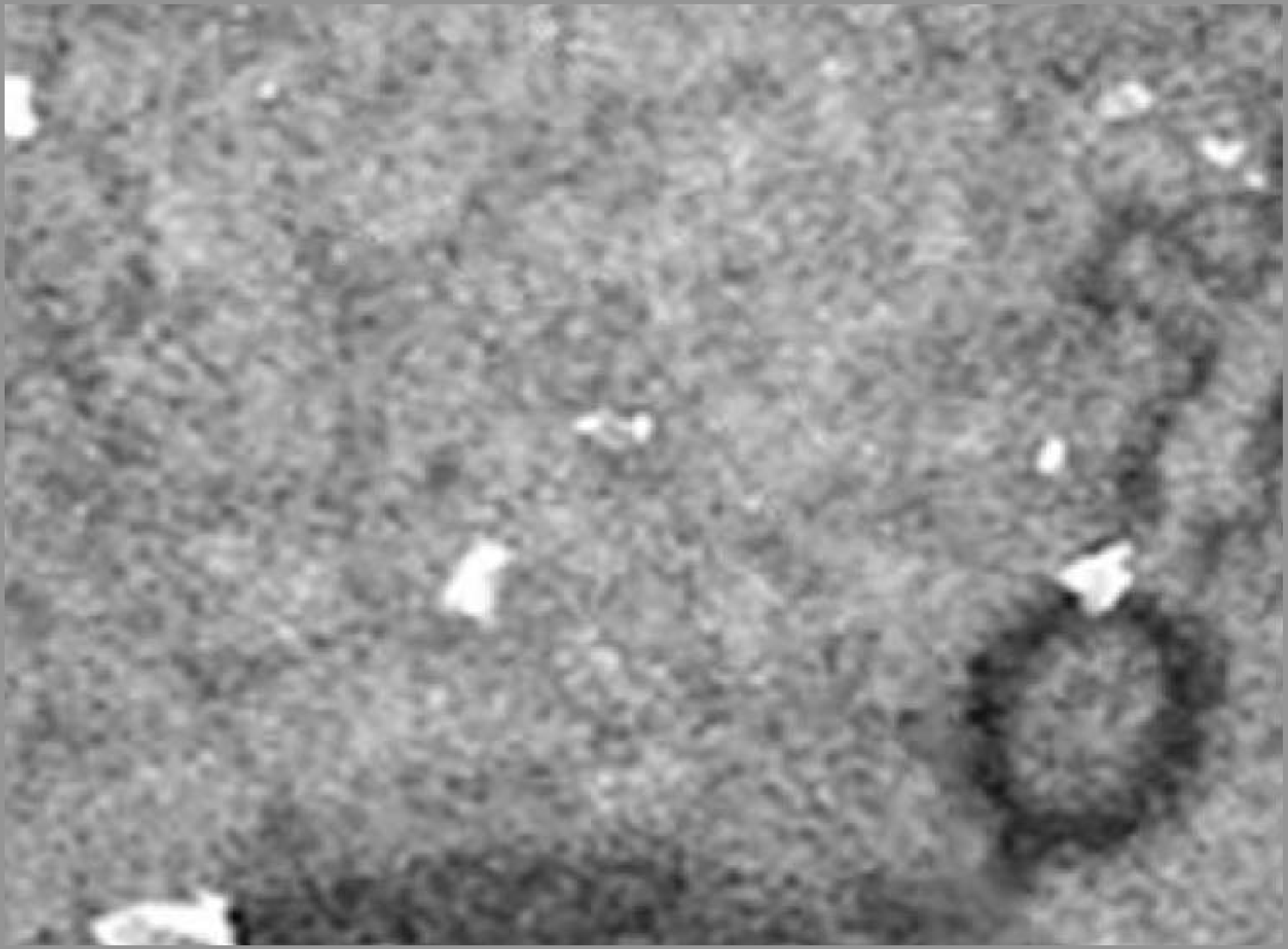


Thanks to Guillermo Sapiro and Kedar Patwardhan

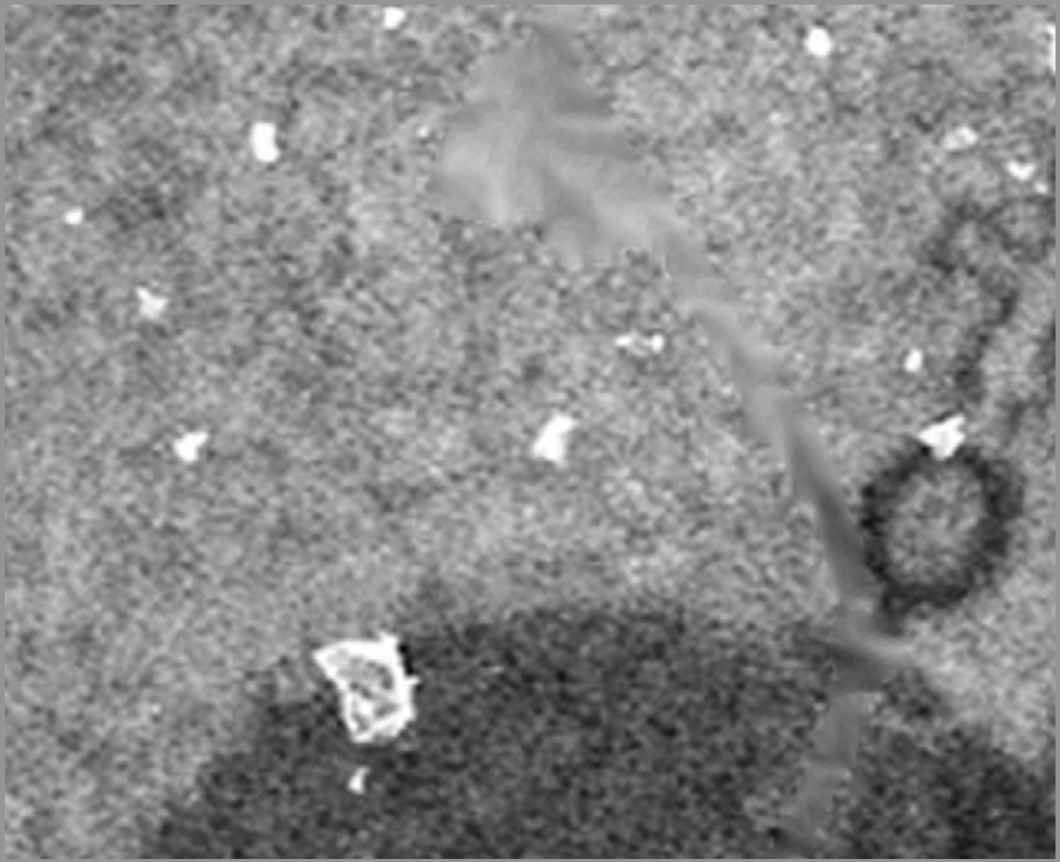
Poisson



Covariant

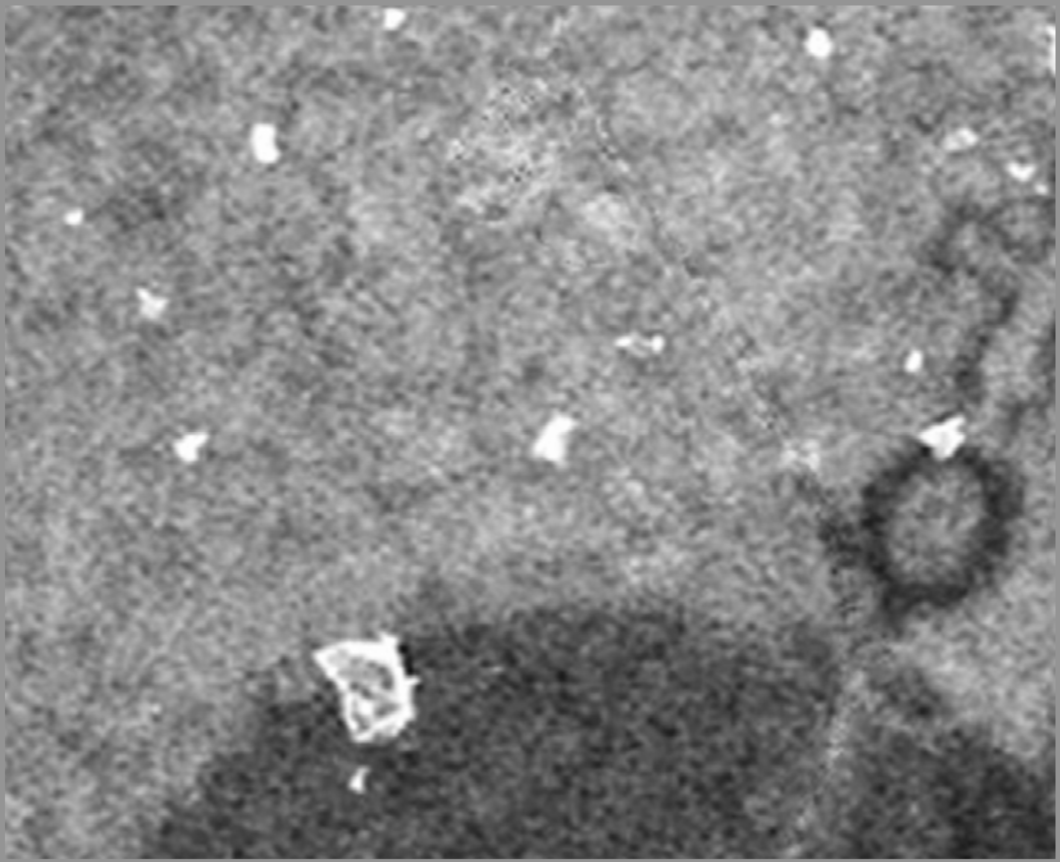


Inpainting



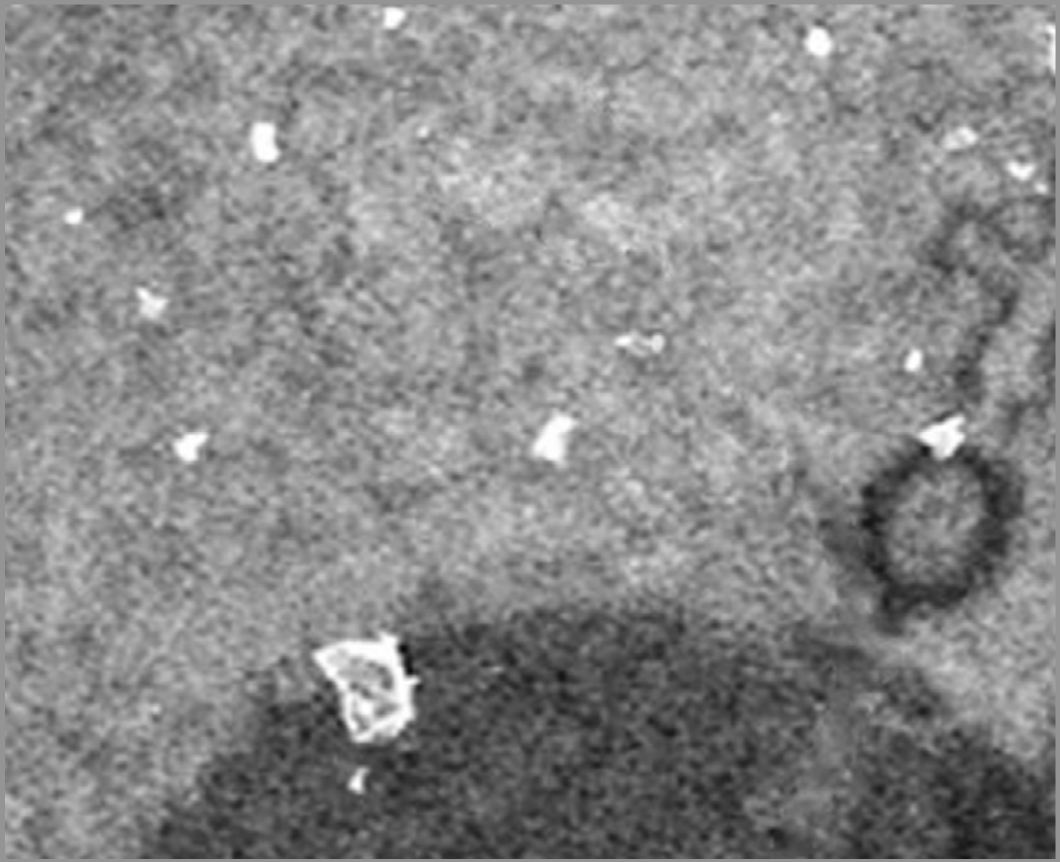
Thanks to Guillermo Sapiro and Kedar Patwardhan

Structure and Texture Inpainting



Bertalmio – Vese – Sapiro – Osher

Covariant Inpainting



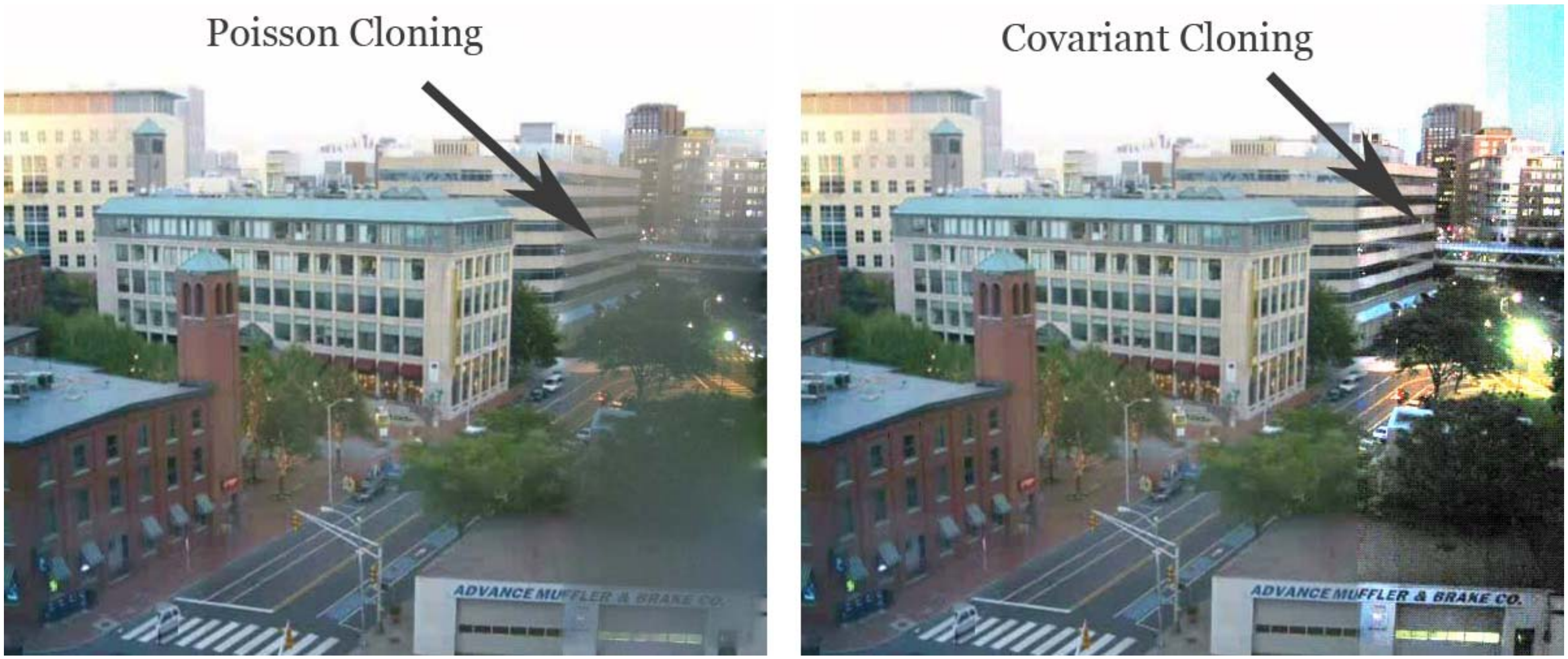
Day

Night

Covariant cloning from day

Poisson cloning from day

Cloning from night to day



Thanks to R. Raskar and J. Yu

Gradient Domain HDR Compression

Changing the lighting conditions. The visual system is robust. It compensates for the changes in illumination by adaptation vector field A:

$$f \rightarrow gf$$

$$\mathbf{A} \rightarrow \mathbf{A} - \frac{grad g}{g}$$

Simplest energy invariant to those transforms is:

$$\int \frac{((\frac{\partial}{\partial x} + A_x)f)^2 + ((\frac{\partial}{\partial y} + A_y)f)^2}{f^2} dxdy$$

Gradient Domain HDR Compression

$$\int \frac{((\frac{\partial}{\partial x} + A_x)f)^2 + ((\frac{\partial}{\partial y} + A_y)f)^2}{f^2} dxdy$$

Euler-Lagrange equation for the above energy:

$$\triangle \ln f = \triangle \ln g$$

Exactly reproduces the result of the Fattal-Lischinski-Werman paper. They assume log; we derive log.

Any good visual system needs to be logarithmic!

Conclusion:

The covariant (adapted) derivative provides
a way to perform perceptual image processing
according to how we see images as opposed to -
how images are recorded by the camera.

Useful for Poisson editing, inpainting or any PDE,
HDR compression, surface reconstruction from
gradients, night/day cloning, graph cuts, bilateral
and trilateral filters in terms of jet bundles, and
practically any perceptual image editing.

Appendix:

Bilateral interpreted in 3D image space

The image $z = f(x, y)$ **is a distribution in 3D:**

$\delta(z - f(x, y))$ **or** $\delta(\ln z - \ln f(x, y))$ **-perceptual?**

Integrate the following 3D filter expression over z

$$\int \delta(z - f(x, y))c(x - u, y - v)s(z - w)dxdydz$$

and evaluate it on the original surface. Result:

$$\int c(x - u, y - v)s(f(x, y) - f(u, v))dxdy \qquad \textbf{(1)}$$

Bilateral

$$\int c(x-u, y-v)s(f(x,y)-f(u,v))dxdy \qquad \textbf{(1)}$$

Same procedure on the logarithmic expression

$$\int \delta(\ln z - \ln f(x,y))c(x-u, y-v)s(z-w)dxdydz$$

produces (using "delta function of function" formula):

$$\int f(x,y)c(x-u, y-v)s(f(x,y)-f(u,v))dxdy \qquad \textbf{(2)}$$

Now, bilateral filter is exactly expression (2) divided by expression (1). Paris-Durand paper derives a similar result (based on intuition) and a speed up algorithm.