

ACPR 19 Tutorial

# Digital Geometry in Pattern Recognition: Extracting Geometric Features with DGtal and Applications

– Part II –

---

Bertrand Kerautret<sup>1</sup> Jacques-Olivier Lachaud <sup>2</sup>

5th Asian Conference on Pattern Recognition  
26 November, 2019, Auckland, New Zealand



<sup>1</sup>LIRIS (UMR CNRS 5205) Université de Lyon 2, F-69676, France

<sup>2</sup>LAMA (UMR CNRS 5127), Université Savoie Mont Blanc, F-73376, France



## Overview of the presentation - Part II -

### 1. Digital Surfaces

Digital surface tracking and algebraic topology

Duality with isosurfaces

### 2. Discrete Exterior Calculus

Foundations of discrete calculus

Discrete calculus model of Ambrosio-Tortorelli functional

Some applications of AT functional

### 3. Reproducible Research

### 4. Practical session: Hands on the IPOL Demonstration System

<https://kerautret.github.io/ACPR19-DGPTutorial>



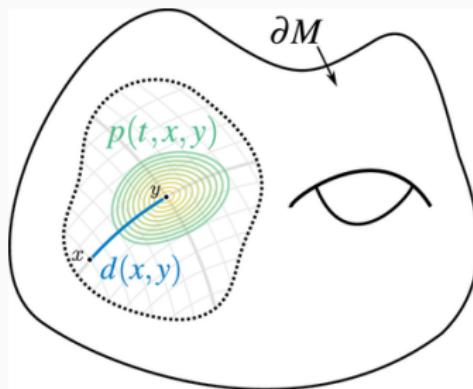
## **1. Digital Surfaces**

---

# Digital Surfaces

## Sound definition for digital surfaces

- boundary of volumes must be digital surfaces
- closed connected digital surfaces have a connected interior
- valid in arbitrary dimension
- open digital surfaces are orientable
- analogous to  $d - 1$ -manifold in  $\mathbb{R}^d$ : locally  $d - 1$  directions

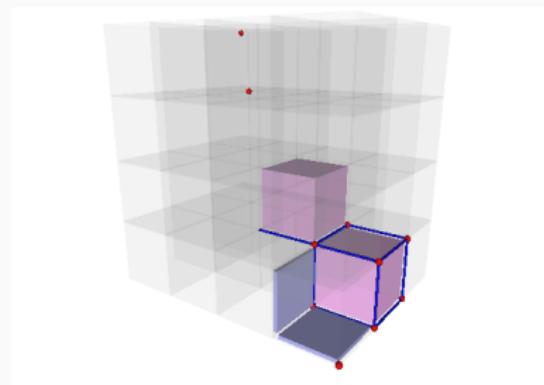


# Digital Surfaces

## Sound definition for digital surfaces

### Digital surfaces require cell/interpixel topology

- volume elements (**voxel**) are  $d$ -cells
- surface elements (**surfel**) are  $d - 1$ -cells

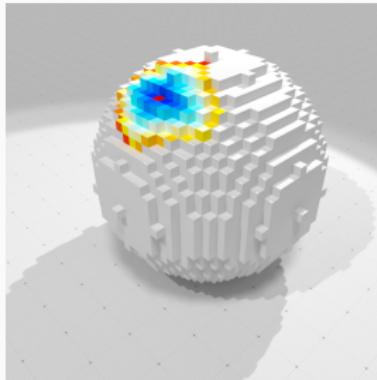
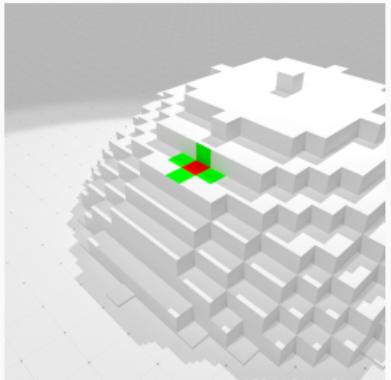


# Digital Surfaces

## Sound definition for digital surfaces

### Digital surfaces require cell/interpixel topology

- volume elements (**voxel**) are  $d$ -cells
- surface elements (**surfel**) are  $d - 1$ -cells
- surfels have **adjacent** surfels; surfels + adjacencies form a graph

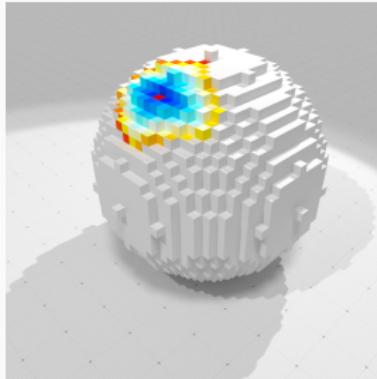
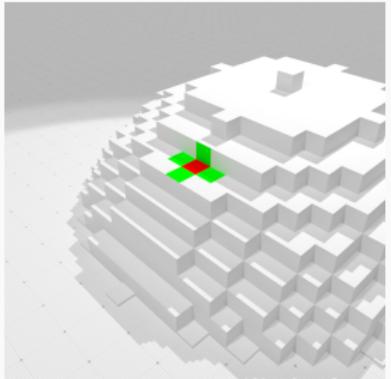


# Digital Surfaces

## Sound definition for digital surfaces

### Digital surfaces require cell/interpixel topology

- volume elements (**voxel**) are  $d$ -cells
- surface elements (**surfel**) are  $d - 1$ -cells
- surfels have adjacent surfels; surfels + adjacencies form a graph
- surfels have  $2d - 2$  neighbor surfels, two per dimension

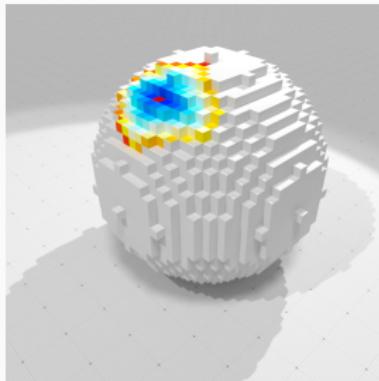
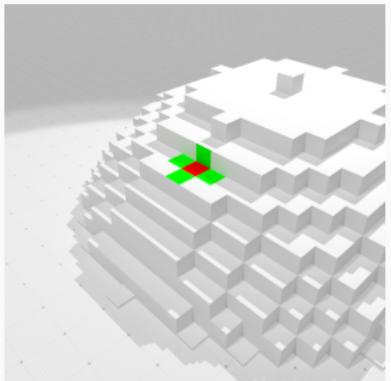


# Digital Surfaces

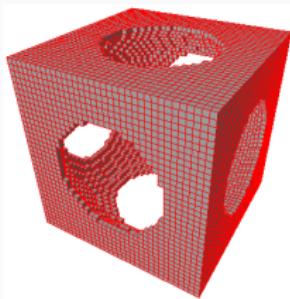
## Sound definition for digital surfaces

### Digital surfaces require cell/interpixel topology

- volume elements (**voxel**) are  $d$ -cells
- surface elements (**surfel**) are  $d - 1$ -cells
- surfels have **adjacent** surfels; surfels + adjacencies form a graph
- surfels have  $2d - 2$  **neighbor** surfels, two per dimension
- surfel adjacency graph depends on chosen **connectednesses** for interior/exterior.



# Topology on digital surfaces ? i



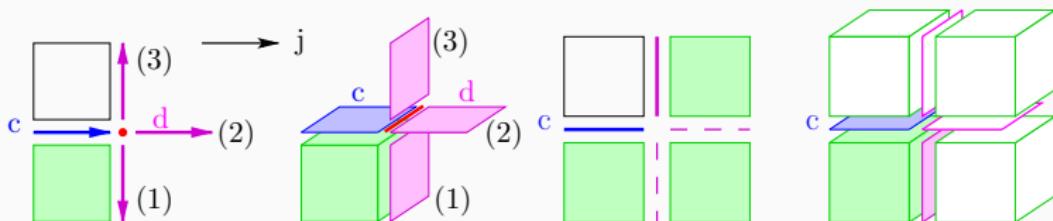
- For now, a surface is a set of surfels

## Questions ?

Can we define local neighborhood relations so that

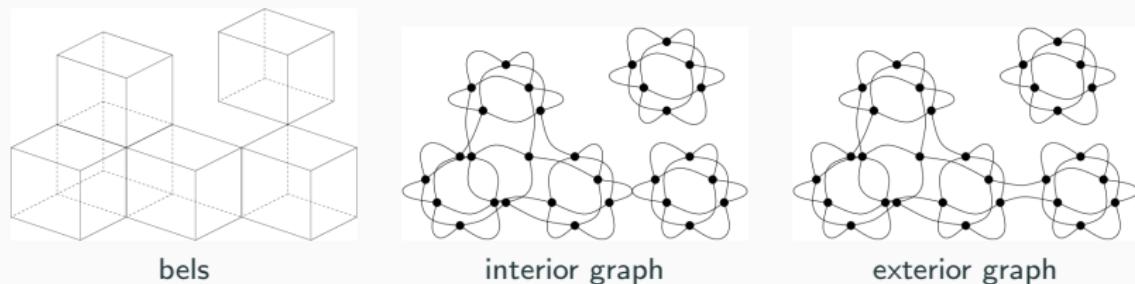
- a whole connected surface can be extracted by their **tracking**,
- **Jordan separation** property is satisfied

# Bel adjacency in a picture $i$



- shape / binary picture  $I$ : finite subset  $X$  of  $\mathbb{Z}^n$
- boundary element or *bel* in  $I$  = surfel between  $X$  and  $X^c$
- For each direction  $j$  ( $n - 1$  directions for each bel)
  - **interior** bel-adjacency from  $c$  along  $j$  = first follower of  $c$  which is a bel
  - **exterior** bel-adjacency from  $c$  along  $j$  = last follower of  $c$  which is a bel

# Bel adjacency graph i



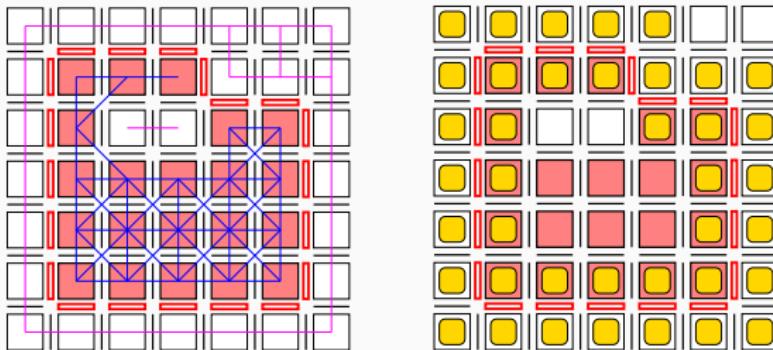
- For each direction, choose interior/exterior  $\Rightarrow 2^{\frac{n(n-1)}{2}}$  bel-adjacencies

## Theorem (3D [Herman,Webster83])

Let  $O \subset X$  6-connected,  $Q \subset X^c$  18-connected.  $c$  any bel.

The **all-interior** bel-adjacency graph component containing  $c$  is the boundary surface between  $O$  and  $Q$ .

## Bel adjacency graph ii



**Theorem ( $nD$ ,  $n \geq 2$ , [Udupa94])**

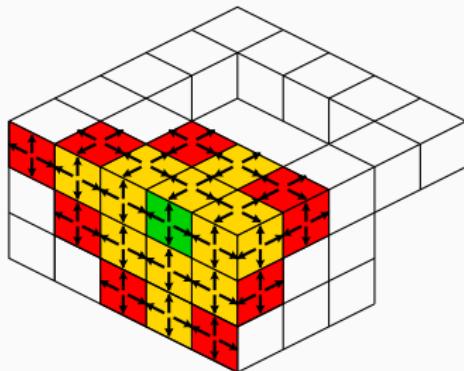
Let  $O \subset X$   $2n$ -connected,  $Q \subset X^c$   $2n^2$ -connected.  $c$  a bel.

The *all-interior* bel-adjacency graph component containing  $c$  is the boundary surface between  $O$  and  $Q$ .

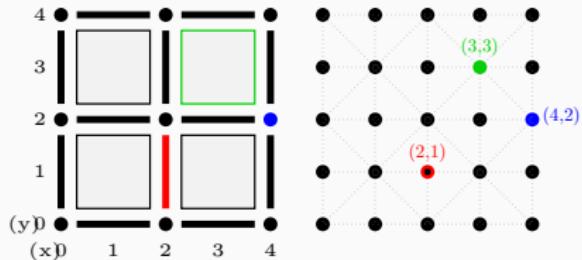
- To extract a boundary component  $\Rightarrow$  track it.

## Tracking digital boundaries i

- boundary in parallelepiped  $N^n$
- number of bels is  $V = O(N^{n-1})$
- degree of each vertex is  $2n - 2$
- breadth-first traversal of bel-adjacency graph
- each bel is visited  $2n - 2$  times
- time complexity  $\approx (2n - 2)V$



# Cubical chain complex i

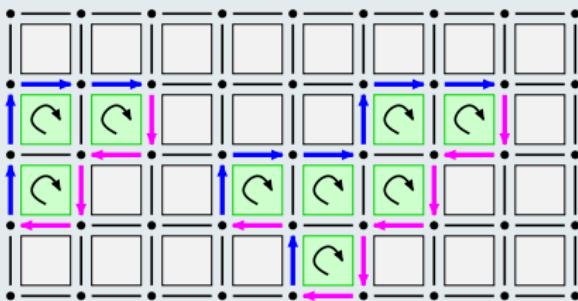


## Classical trick for representing cell space

- isomorphism “grid” and “Khalimsky’s space”
- cells can be represented by integer points
- a cell is an element of  $\mathbb{Z}^n$ , parities = topology
- pixels, voxels,  $n$ -cells have odd parities

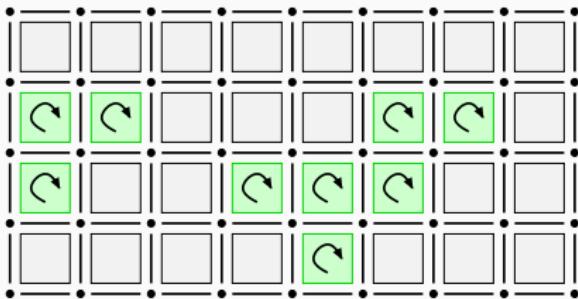
## Cubical chain complex ii

### Construction of a chain complex



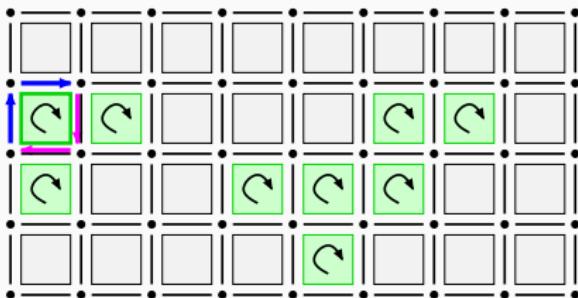
- oriented  $k$ -cells form  $k$ -dimensional bases
- $k$ -chains are formal sums of  $k$ -cells (coefficient  $\mathbb{Z}$ )  
 $[\sum_i +o_i^n]$  is a digital shape  
 $\sum [+s_j^{n-1}] + \sum [-s_{j'}^{n-1}]$  is a digital surface
- boundary operator  $\Delta$ , with  $\Delta\Delta = 0$ , based on cell parities

## Application to digital boundaries



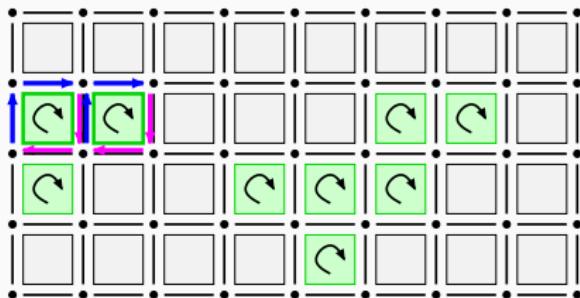
- digital shape is a subset  $X$  of  $\mathbb{Z}^n$  (odd parities)
- its boundary =  $n - 1$ -chain  $\Delta \sum_{x \in X} +x$
- it is a cycle since  $\Delta\Delta = 0$

## Application to digital boundaries



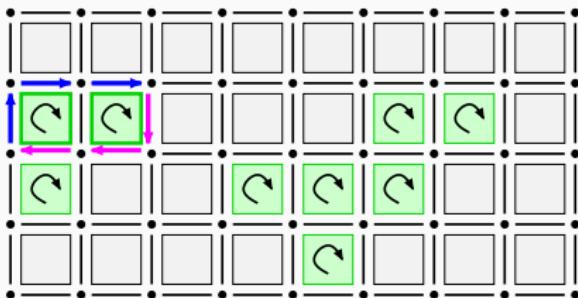
- digital shape is a subset  $X$  of  $\mathbb{Z}^n$  (odd parities)
- its boundary =  $n - 1$ -chain  $\Delta \sum_{x \in X} +x$
- it is a cycle since  $\Delta\Delta = 0$

## Application to digital boundaries



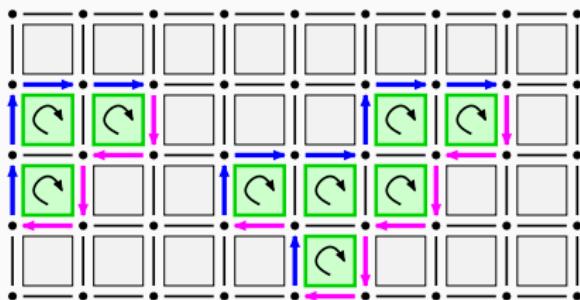
- digital shape is a subset  $X$  of  $\mathbb{Z}^n$  (odd parities)
- its boundary =  $n - 1$ -chain  $\Delta \sum_{x \in X} +x$
- it is a cycle since  $\Delta\Delta = 0$

## Application to digital boundaries



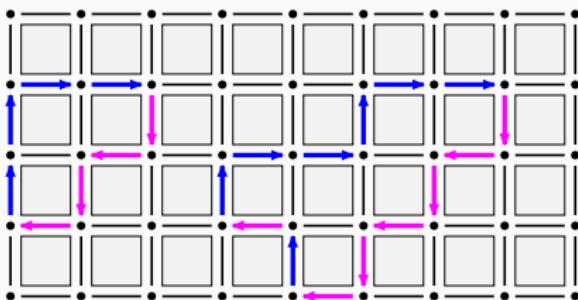
- digital shape is a subset  $X$  of  $\mathbb{Z}^n$  (odd parities)
- its boundary =  $n - 1$ -chain  $\Delta \sum_{x \in X} +x$
- it is a cycle since  $\Delta\Delta = 0$

## Application to digital boundaries



- digital shape is a subset  $X$  of  $\mathbb{Z}^n$  (odd parities)
- its boundary =  $n - 1$ -chain  $\Delta \sum_{x \in X} +x$
- it is a cycle since  $\Delta\Delta = 0$

## Application to digital boundaries

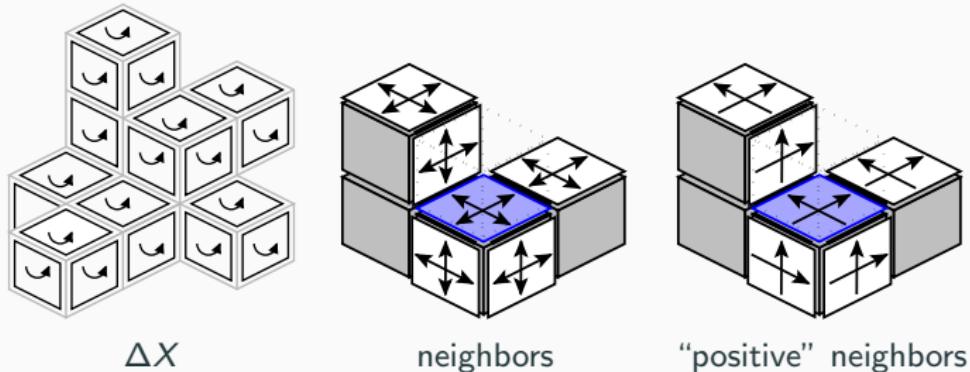


- digital shape is a subset  $X$  of  $\mathbb{Z}^n$  (odd parities)
- its boundary =  $n - 1$ -chain  $\Delta \sum_{x \in X} +x$
- it is a cycle since  $\Delta\Delta = 0$

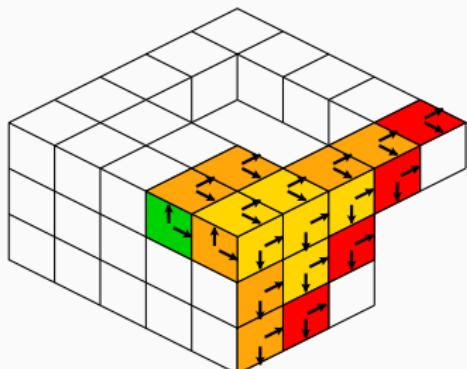
### Theorem

The boundary of a volume is a closed digital surface

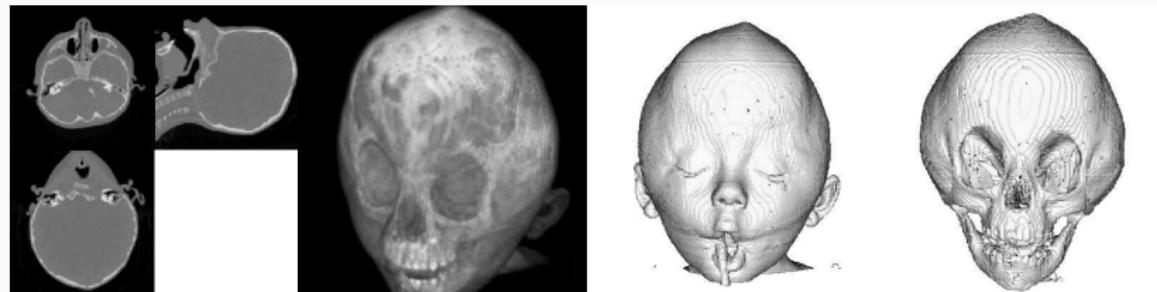
## Oriented boundary tracking i



- since  $\Delta\Delta X = 0$
- breadth-first traversal of **directed** bel-adjacency graph
- each bel is visited  $\frac{2n-2}{2}$  times
- time complexity  $\approx (n-1)V$



# Isosurfaces i



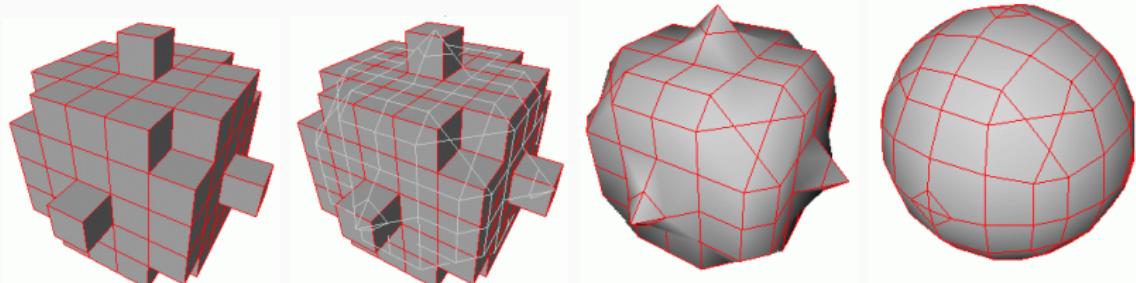
## Definition (Isosurface)

Let  $I : \mathbb{R}^3 \rightarrow \mathbb{R}$ .

**Isosurface** of value  $s$  in  $I = \{(x, y, z) \in \mathbb{R}^3, I(x, y, z) = s\}$ .

- *marching-cubes* [Lorensen,Cline87], by scanning

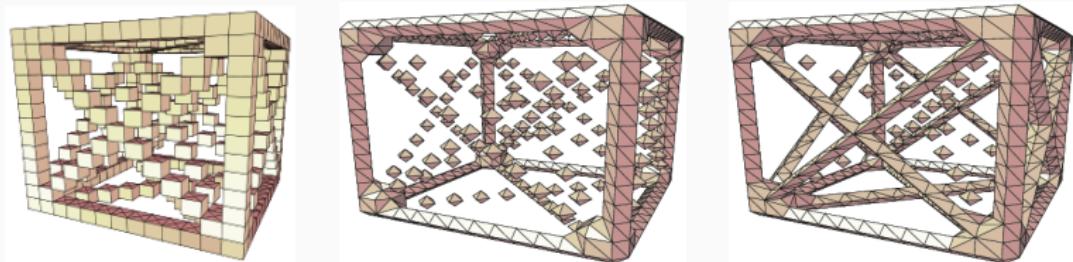
# Duality isosurfaces / digital surface



$$X = \{\vec{x} \in \mathbb{Z}^3, I(\vec{x}) \geq s\}$$

## Theorem [L. Montanvert 2000]

bel-adjacency complex defines a  $n - 1$ -pseudomanifold without boundary

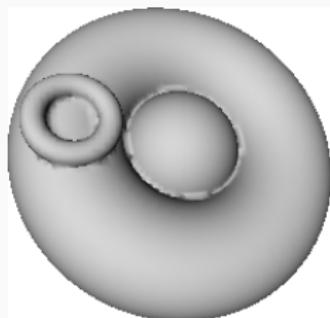
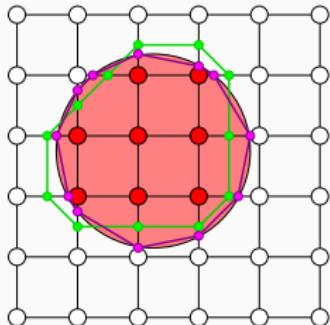
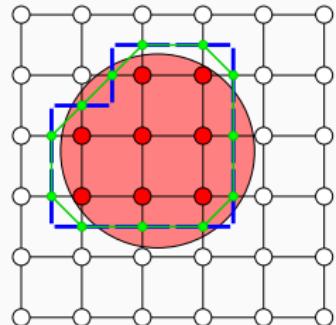


shape

interior

exterior

# Making isosurfaces nice



1.  $X = \{\vec{x} \in \mathbb{Z}^3, I(\vec{x}) \geq s\}$
2. track  $\Delta \sum_{x \in X} +x$
3. local triangulation
4. move vertices

# Digital surfaces in DGtal

## Generic representation of digital surfaces

- One interface `DigitalSurface`, many implementations
  - `DigitalSetBoundary` implicit boundary of volumes
  - `SetOfSurfels` explicit set of surfels
  - `LightImplicitDigitalSurface` implicit digital surface from interior/exterior predicate
  - `ImplicitDigitalSurface` same as above with boundary precomputation
  - `LightExplicitDigitalSurface` explicit digital surface from surfel predicate
  - `ExplicitDigitalSurface` same as above with boundary precomputation

# Digital surfaces in DGtal

## Generic representation of digital surfaces

- One interface `DigitalSurface`, many implementations
  - `DigitalSetBoundary` implicit boundary of volumes
  - `SetOfSurfels` explicit set of surfels
  - `LightImplicitDigitalSurface` implicit digital surface from interior/exterior predicate
  - `ImplicitDigitalSurface` same as above with boundary precomputation
  - `LightExplicitDigitalSurface` explicit digital surface from surfel predicate
  - `ExplicitDigitalSurface` same as above with boundary precomputation
- everything generic, arbitrary dimensional

# Digital surfaces in DGtal

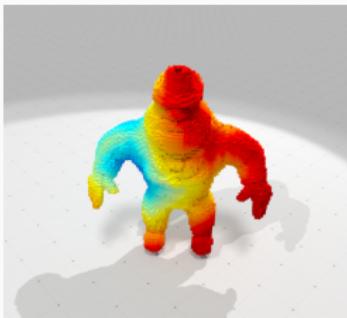
## Generic representation of digital surfaces

- One interface `DigitalSurface`, many implementations
  - `DigitalSetBoundary` implicit boundary of volumes
  - `SetOfSurfels` explicit set of surfels
  - `LightImplicitDigitalSurface` implicit digital surface from interior/exterior predicate
  - `ImplicitDigitalSurface` same as above with boundary precomputation
  - `LightExplicitDigitalSurface` explicit digital surface from surfel predicate
  - `ExplicitDigitalSurface` same as above with boundary precomputation
- everything generic, arbitrary dimensional
- In 3D, one interface `IndexedDigitalSurface`, same many implementations
  - all cells are numbered from 0,
  - based on half-edge data structure,
  - faster if process requires several pass, many neighborhood comput.

# Digital surfaces in DGtal

## Usual usages: use **Shortcuts module**

```
1 #include "DGtal/helpers/Shortcuts.h"
...
3 typedef Shortcuts<Z3i::KSpace> SH3;
4 auto params = SH3::defaultParameters();
5 params("surfaceTraversal", "BreadthFirst") // specifies breadth-first traversal
     ("colormap", "Jet"); // specifies the colormap
6 auto vol      = SH3::makeBinaryImage( "samples/A1.100.vol", params ); //load vol file
7 auto K        = SH3::getKSpace( vol );                                // define grid cell space
8 auto surface  = SH3::makeLightDigitalSurface( vol, K, params );//define digital surface
9 auto surfels  = SH3::getSurfelRange( surface, params );           //track surfels with BFT
10 auto cmap     = SH3::getColorMap( 0, surfels.size(), params );//paint surfels from rank
    SH3::Colors colors( surfels.size() );
11 for ( int i = 0; i < surfels.size(); ++i ) colors[ i ] = cmap( i );
12 bool ok     = SH3::saveOBJ( surface, SH3::RealVectors(), colors, "al-primal-bft.obj" );
```



# Digital surfaces in DGtal

## Usual usages: use **Shortcuts module**

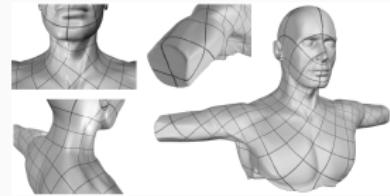
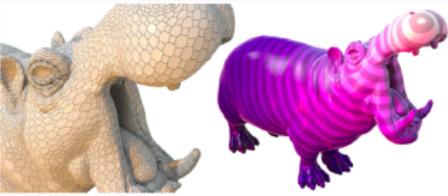
- image → digital surface / indexed digital surface
- extraction of connected components / main component
- polynomial shape → digital surface / indexed surface
- digital surface ↔ indexed digital surface
- (indexed) digital surface → dual triangulated surface (MC)
- (indexed) digital surface → primal polygonal surface
- (indexed) digital surface → dual polygonal surface
- traversals/visitors: breadth-first, depth-first
- many export to OBJ
- many geometric estimation: area, normals, curvatures

## 2. Discrete Exterior Calculus

---

## II.2 Discrete (exterior) Calculus

Computer graphics, geometry processing, shape optimization



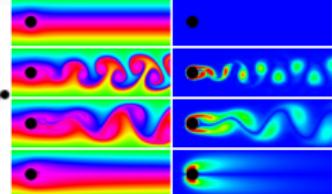
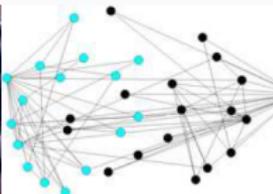
(Images: Knöppel et al. 2015, Crane et al. 2013, Springborn et al. 2010)

Discrete exterior calculus [[Desbrun, Hirani, Leok, ...](#)]

Discrete differential calculus [[Polthier, Pinkall, Bobenko, ...](#)]

Discrete calculus [[Grady, Polimeni, ...](#)]

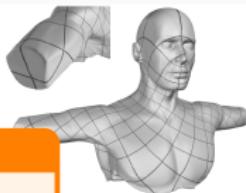
Graph and network analysis, image processing, fluid simul.



(Images: Bugeau et al. 2014, couprie et al. 2014, Elcott et al. 2006)

## II.2 Discrete (exterior) Calculus

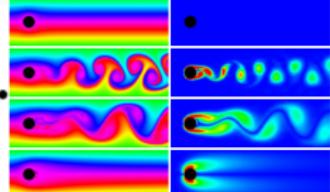
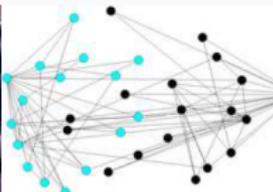
Computer graphics, geometry processing, shape optimization



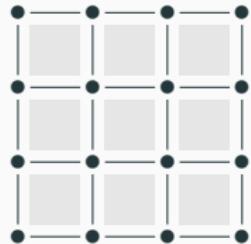
### Discrete exterior calculus (DEC)

- no discretization, discrete by nature
- keep algebraic properties of calculus, exact Stokes' theorem
- reduces to matrix/vectors
- works without embedding, just metric
- “any” cell complex, arbitrary dimension

Graph and network analysis, image processing, fluid simul.

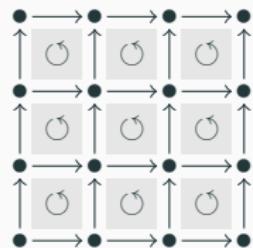


## Cell complex, chains, boundary, forms



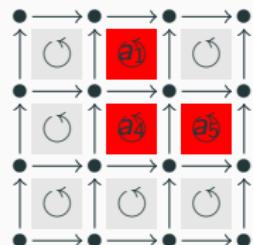
- *cell complex  $K$ :* vertices, edges, faces (pixels/surfels)

## Cell complex, chains, boundary, forms



- *cell complex  $K$ :* vertices, edges, faces (pixels/surfels) with orientation

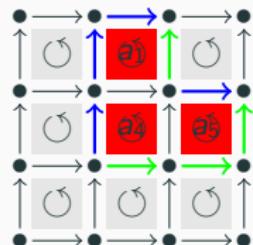
## Cell complex, chains, boundary, forms



$$\sigma := a_1 + a_4 + a_5 \in C_2(K)$$

- *cell complex  $K$ :* vertices, edges, faces (pixels/surfels) with orientation
- *$k$ -chains:*  $C_k(K)$  are integral formal sums of oriented cells

## Cell complex, chains, boundary, forms



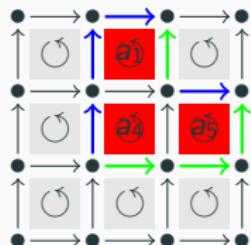
$$\sigma := \textcolor{red}{a_1} + \textcolor{red}{a_4} + \textcolor{red}{a_5} \in C_2(K)$$

$$\eta := \sum \textcolor{green}{e_i} - \sum \textcolor{blue}{e_j} \in C_1(K)$$

$$\eta = \partial_2 \sigma$$

- *cell complex*  $K$ : vertices, edges, faces (pixels/surfels) with orientation
- *$k$ -chains*:  $C_k(K)$  are integral formal sums of oriented cells
- *boundary operators*:  $\cdots C_2(K) \xrightarrow{\partial_2} C_1(K) \xrightarrow{\partial_1} C_0(K) \xrightarrow{\partial_0} 0$

## Cell complex, chains, boundary, forms



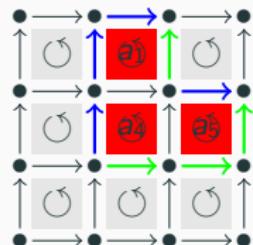
$$\sigma := a_1 + a_4 + a_5 \in C_2(K)$$

$$\eta := \sum e_i - \sum e_j \in C_1(K)$$

$$\eta = \partial_2 \sigma$$

- *cell complex*  $K$ : vertices, edges, faces (pixels/surfels) with orientation
- *$k$ -chains*:  $C_k(K)$  are integral formal sums of oriented cells
- *boundary operators*:  $\cdots C_2(K) \xrightarrow{\partial_2} C_1(K) \xrightarrow{\partial_1} C_0(K) \xrightarrow{\partial_0} 0$
- *discrete  $k$ -forms*: elements of  $C^k(K) := \text{Hom}(C_k(K), \mathbb{R})$ 
  - 0-forms: functions, i.e. a value per vertex
  - 1-forms: differential forms/vector field, i.e. a value per edge
  - 2-forms: area forms, i.e. a value per face

## Cell complex, chains, boundary, forms



$$\sigma := a_1 + a_4 + a_5 \in C_2(K)$$

$$\eta := \sum e_i - \sum e_j \in C_1(K)$$

$$\eta = \partial_2 \sigma$$

- *cell complex*  $K$ : vertices, edges, faces (pixels/surfels) with orientation
- *$k$ -chains*:  $C_k(K)$  are integral formal sums of oriented cells
- *boundary operators*:  $\cdots C_2(K) \xrightarrow{\partial_2} C_1(K) \xrightarrow{\partial_1} C_0(K) \xrightarrow{\partial_0} 0$
- *discrete  $k$ -forms*: elements of  $C^k(K) := \text{Hom}(C_k(K), \mathbb{R})$ 
  - 0-forms: functions, i.e. a value per vertex
  - 1-forms: differential forms/vector field, i.e. a value per edge
  - 2-forms: area forms, i.e. a value per face
- *Integral*  $\int_{\sigma} \alpha =$  pairing  $k$ -form  $\alpha$  with  $k$ -chain  $\sigma$

$$\int_{\sigma} \alpha := \alpha(\sigma) = \sum_i a_i \alpha(c_i) \quad \text{if } \sigma = \sum_i a_i c_i$$

## Exterior derivative, Stokes theorem

- exterior derivative defined by duality:  $\mathbf{d}_k : C^k(K) \rightarrow C^{k+1}(K)$

$$(\mathbf{d}_k \alpha^k)(\sigma_{k+1}) := \alpha^k(\partial_{k+1} \sigma_{k+1})$$

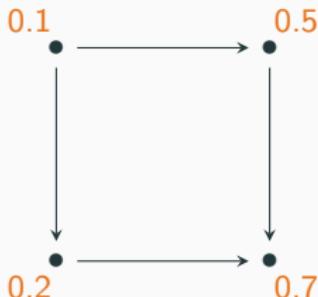
thus incidence relations define derivative by duality

## Exterior derivative, Stokes theorem

- exterior derivative defined by duality:  $\mathbf{d}_k : C^k(K) \rightarrow C^{k+1}(K)$

$$(\mathbf{d}_k \alpha^k)(\sigma_{k+1}) := \alpha^k(\partial_{k+1} \sigma_{k+1})$$

thus incidence relations define derivative by duality



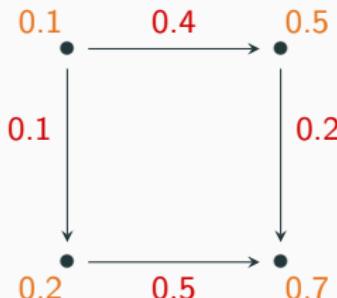
- Function or discrete 0-form :  $\alpha = (0.2, 0.7, 0.1, 0.5)$

## Exterior derivative, Stokes theorem

- exterior derivative defined by duality:  $\mathbf{d}_k : C^k(K) \rightarrow C^{k+1}(K)$

$$(\mathbf{d}_k \alpha^k)(\sigma_{k+1}) := \alpha^k(\partial_{k+1} \sigma_{k+1})$$

thus incidence relations define derivative by duality



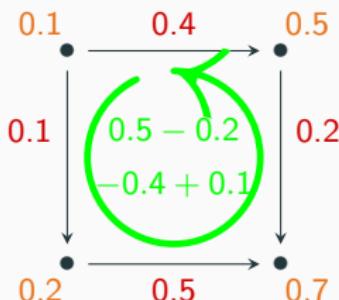
- Function or discrete 0-form :  $\alpha = (0.2, 0.7, 0.1, 0.5)$
- 1-form  $\mathbf{d}_0(\alpha) = \beta = (0.5, 0.1, 0.2, 0.4)$

## Exterior derivative, Stokes theorem

- exterior derivative defined by duality:  $\mathbf{d}_k : C^k(K) \rightarrow C^{k+1}(K)$

$$(\mathbf{d}_k \alpha^k)(\sigma_{k+1}) := \alpha^k(\partial_{k+1} \sigma_{k+1})$$

thus incidence relations define derivative by duality



- Function or discrete 0-form :  $\alpha = (0.2, 0.7, 0.1, 0.5)$
- 1-form  $\mathbf{d}_0(\alpha) = \beta = (0.5, 0.1, 0.2, 0.4)$
- 2-form  $\mathbf{d}_1(\beta) = 0$ , since  $\mathbf{d}_1 \mathbf{d}_0 = 0$

## Exterior derivative, Stokes theorem

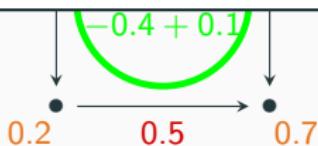
- exterior derivative defined by duality:  $\mathbf{d}_k : C^k(K) \rightarrow C^{k+1}(K)$

$$(\mathbf{d}_k \alpha^k)(\sigma_{k+1}) := \alpha^k(\partial_{k+1} \sigma_{k+1})$$

thus

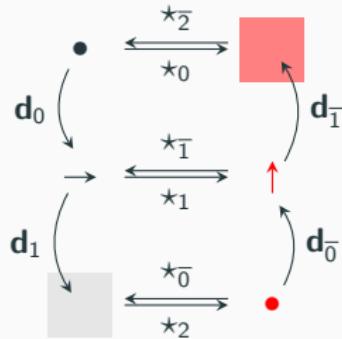
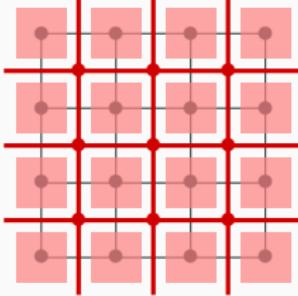
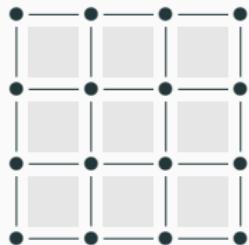
(discrete) Stokes theorem is trivial by definition

$$\int_{\sigma} \mathbf{d}\alpha = \int_{\partial\sigma} \alpha$$

for  $\sigma$  any  $k$ -chain and  $\alpha$  any  $k - 1$ -form

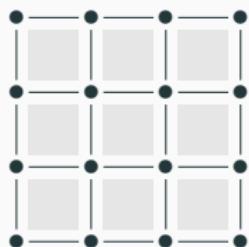
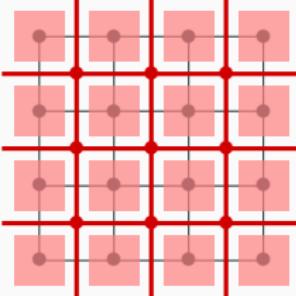
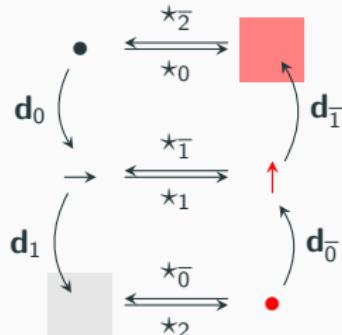
- Function or discrete 0-form :  $\alpha = (0.2, 0.7, 0.1, 0.5)$
- 1-form  $\mathbf{d}_0(\alpha) = \beta = (0.5, 0.1, 0.2, 0.4)$
- 2-form  $\mathbf{d}_1(\beta) = 0$ , since  $\mathbf{d}_1 \mathbf{d}_0 = 0$

## Dual cell complex, Hodge star, calculus



- Hodge duality created with dual/orthogonal structure

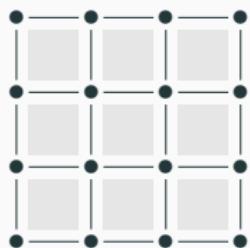
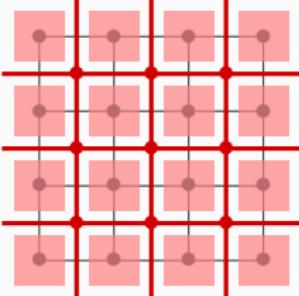
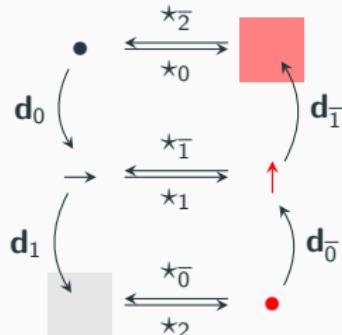
# Dual cell complex, Hodge star, calculus

complex  $K$ dual complex  $\bar{K}$ 

primal      dual

- Hodge duality created with dual/orthogonal structure
- anti-derivatives  $d_{\bar{k}}$  are derivatives in dual complex
  - in matrix form  $d_{\bar{k}}^T := d_{n-1-k}$

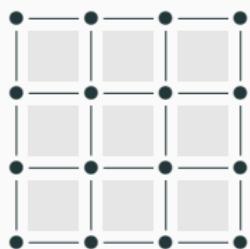
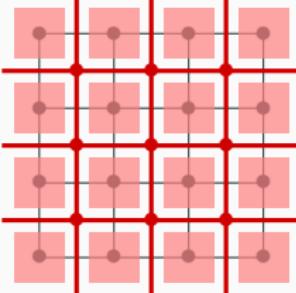
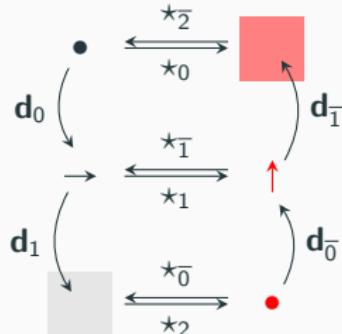
# Dual cell complex, Hodge star, calculus

complex  $K$ dual complex  $\bar{K}$ 

primal      dual

- Hodge duality created with dual/orthogonal structure
- anti-derivatives  $d_{\bar{k}}$  are derivatives in dual complex
  - in matrix form  $d_{\bar{k}}^{-T} := d_{n-1-k}$
- Hodge stars  $\star_k$  transport  $k$ -forms to dual  $2 - k$ -forms
  - diagonal matrices incorporating metric information
  - e.g.  $\star_k \mathbf{1} = \alpha$  is the area 2-form  $dA$

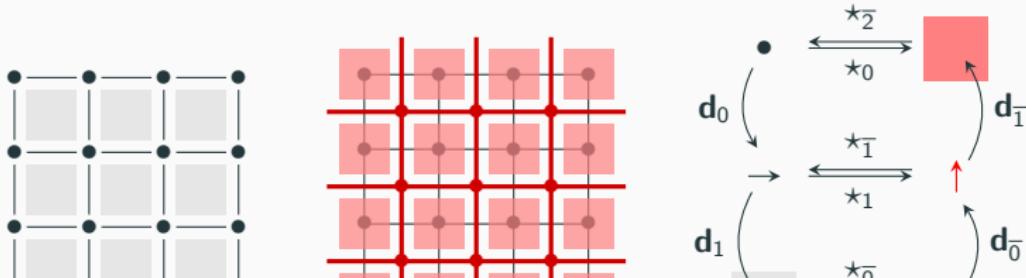
# Dual cell complex, Hodge star, calculus

complex  $K$ dual complex  $\bar{K}$ 

primal      dual

- Hodge duality created with dual/orthogonal structure
- anti-derivatives  $d_{\bar{k}}$  are derivatives in dual complex
  - in matrix form  $d_{\bar{k}}^{-T} := d_{n-1-k}$
- Hodge stars  $\star_k$  transport  $k$ -forms to dual  $2 - k$ -forms
  - diagonal matrices incorporating metric information
  - e.g.  $\star_k \mathbf{1} = \alpha$  is the area 2-form  $dA$
- wedge products satisfy algebraic properties (Leibniz rules . . .)
  - $\alpha \wedge \beta := \text{diag}(\alpha)\beta$ , for  $\alpha \in C^k(K), \beta \in C^{2-k}(\bar{K})$ ,
  - $f \wedge \gamma := \text{diag}(\mathbf{M}_{01}f)\gamma$ , for  $f \in C^0(K), \gamma \in C^1(K)$  . . .

## Dual cell complex, Hodge star, calculus



Almost all the calculus is built from the previous operators

- codifferentials  $\delta_1 := -\star_{\bar{2}}d_{\bar{1}}\star_1$ ,  $\delta_2 := -\star_{\bar{1}}d_{\bar{0}}\star_2$ ,
- Laplacian  $\Delta := \delta_1 d_0$
- Edge Laplacian  $\Delta_1 := d_0 \delta_1 + \delta_2 d_1$ ,
- musical ops : Vector field  $\xrightarrow{\flat}$  1-form  $\xrightarrow{\sharp}$  Vector field
- gradient  $\nabla f := (d_0 f)^\sharp$
- divergence  $\text{div } \mathbf{V} := \delta_1 \mathbf{V}^\flat$
- $L^2$  inner-product  $(\alpha, \beta)_{\Omega, k} := \int_{\Omega} \alpha \wedge \star_k \beta$ , for  $\alpha, \beta$   $k$ -forms
  - $t \wedge \gamma := \text{diag}(\mathbf{M}_{01} t) \gamma$ , for  $t \in C^0(K)$ ,  $\gamma \in C^1(K)$  ...

## Formulate PDE/variational problems with digital calculus

### Anisotropic diffusion on image /

- $\frac{\partial u}{\partial t} = \operatorname{div}(f(\|\nabla u\|)\nabla u)$  with  $u_0 = I$ ,  $f$  decreasing from 1 to 0.
- $\frac{u^{t+dt} - u^t}{dt} = \delta_1(F \wedge \mathbf{d}_0 u)$ , with function  $F$  defined for all edge  $e$  as  $F(e) = f(\|u(v_j) - u(v_i)\|)$ , with  $\partial_1 e = v_j - v_i$ .
- in matrix form, with matrices  $\mathbf{A} := \mathbf{d}_0$ ,  $\mathbf{G}_k := \star_k$ :

$$\mathbf{u}^{t+dt} = (\mathbf{Id} + dt \mathbf{G}_{\bar{2}} \mathbf{A}^T \mathbf{G}_1 \operatorname{diag}(\mathbf{f}) \mathbf{A}) \mathbf{u}^t,$$

with  $\mathbf{f}$  computed per edge from value of  $\mathbf{u}^t$ .

- exactly the “strangely stable” algorithm of Perona-Malik for rectangular domains
- more general formulation in digital calculus (any complex)

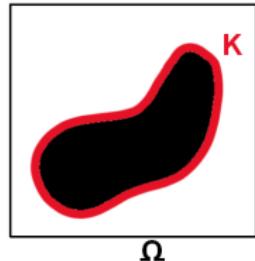
# An important example: Mumford-Shah functional

## Mumford-Shah functional for image restoration

We minimize

$$\mathcal{MS}(K, \textcolor{green}{u}) = \underbrace{\alpha \int_{\Omega \setminus K} |\textcolor{green}{u} - g|^2 \, dx}_{\text{fidelity term}} + \underbrace{\int_{\Omega \setminus K} |\nabla \textcolor{green}{u}|^2 \, dx}_{\text{smoothness term}} + \lambda \underbrace{\mathcal{H}^1(K \cap \Omega)}_{\text{discontinuities length}}$$

- $\Omega$  the image domain
- $g$  the input image
- $\textcolor{green}{u}$  a piecewise smooth approximation of  $g$
- $K$  the set of discontinuities
- $\mathcal{H}^1$  the Hausdorff measure



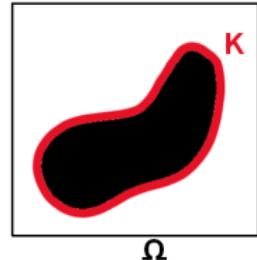
# An important example: Mumford-Shah functional

## Mumford-Shah functional for image restoration

We minimize

$$\mathcal{MS}(K, u) = \underbrace{\alpha \int_{\Omega \setminus K} |u - g|^2 \, dx}_{\text{fidelity term}} + \underbrace{\int_{\Omega \setminus K} |\nabla u|^2 \, dx}_{\text{smoothness term}} + \lambda \underbrace{\mathcal{H}^1(K \cap \Omega)}_{\text{discontinuities length}}$$

- $\Omega$  the image domain
- $g$  the input image
- $u$  a piecewise smooth approximation of  $g$
- $\times$   $K$  the set of discontinuities
- $\times$   $\mathcal{H}^1$  the Hausdorff measure



# Mumford-Shah functional

## Notably difficult to minimize

Many relaxations and convexifications have been proposed.

- Total Variation [?] and its variants
- Multi-phase level sets [?] and follow-ups
- Discrete graph approaches [?, ?]
- Calibration method [?] and associated algorithms [?, ?]
- Ambrosio-Tortorelli functional [?]
- convex relaxations of AT [?]

# Mumford-Shah functional

## Notably difficult to minimize

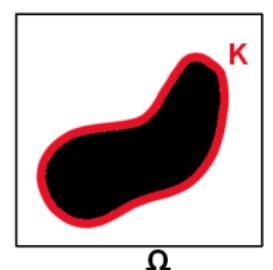
Many relaxations and convexifications have been proposed.

- Total Variation [?] and its variants
- Multi-phase level sets [?] and follow-ups
- Discrete graph approaches [?, ?]
- Calibration method [?] and associated algorithms [?, ?]
- Ambrosio-Tortorelli functional [?]
- convex relaxations of AT [?]

## Ambrosio-Tortorelli functional

$$AT_\epsilon(u, v) = \alpha \int_{\Omega} |u - g|^2 \, dx + \int_{\Omega} v^2 |\nabla u|^2 \, dx + \lambda \int_{\Omega} \epsilon |\nabla v|^2 + \frac{1}{\epsilon} \frac{(1 - v)^2}{4} \, dx$$

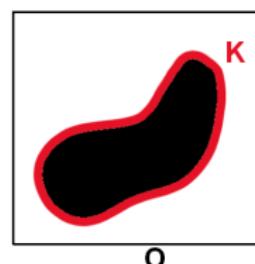
- $\Omega$  the image domain
- $g$  the input image
- $u$  a piecewise smooth approximation of  $g$
- $v$  a smooth approximation of  $1 - \chi_K$
- ✓ whole domain integration
- ✓ no Hausdorff measure



## Ambrosio-Tortorelli functional

$$AT_\epsilon(u, v) = \alpha \int_{\Omega} |u - g|^2 \, dx + \int_{\Omega} v^2 |\nabla u|^2 \, dx + \lambda \int_{\Omega} \epsilon |\nabla v|^2 + \frac{1}{\epsilon} \frac{(1-v)^2}{4} \, dx$$

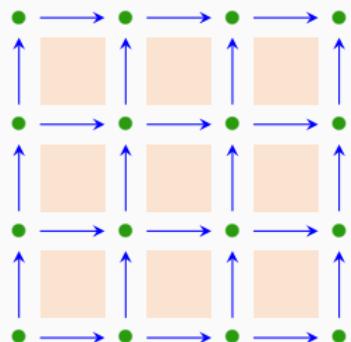
- $\Omega$  the image domain
- $g$  the input image
- $u$  a piecewise smooth approximation of  $g$
- $v$  a smooth approximation of  $1 - \chi_K$
- ✓ whole domain integration
- ✓ no Hausdorff measure



$\Gamma$ -convergence:  $AT_\epsilon \xrightarrow[\epsilon \rightarrow 0]{} \mathcal{MS}$

# Discrete formulation of AT

$$AT_\epsilon(u, v) = \alpha \int_{\Omega} |u - g|^2 \, dx + \int_{\Omega} v^2 |\nabla u|^2 \, dx + \lambda \int_{\Omega} \epsilon |\nabla v|^2 + \frac{1}{4\epsilon} (1 - v)^2 \, dx$$



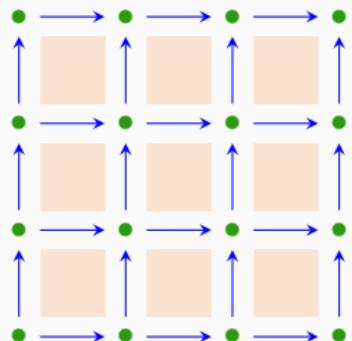
We choose :

- functions  $u, g$  to live on faces
  - $u, g$  are **2-forms**
  - equivalently dual **0-forms**
- function  $v$  to live on vertices
  - $v$  is a **0-form**



# Discrete formulation of AT

$$AT_{\epsilon}(u, v) = \alpha \int_{\Omega} |u - g|^2 \, dx + \int_{\Omega} v^2 |\nabla u|^2 \, dx + \lambda \int_{\Omega} \epsilon |\nabla v|^2 + \frac{1}{4\epsilon} (1 - v)^2 \, dx$$



We choose :

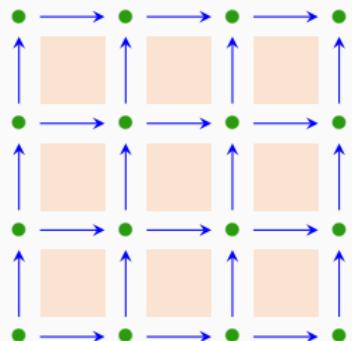
- functions  $u, g$  to live on faces
  - $u, g$  are **2-forms**
  - equivalently dual **0-forms**
- function  $v$  to live on vertices
  - $v$  is a **0-form**



$$AT_{\epsilon}^{2,0}(u, v) = \alpha (u - g, u - g)_{\Omega, 2}$$

## Discrete formulation of AT

$$AT_\epsilon(u, v) = \alpha \int_{\Omega} |u - g|^2 \, dx + \int_{\Omega} v^2 |\nabla u|^2 \, dx + \lambda \int_{\Omega} \epsilon |\nabla v|^2 + \frac{1}{4\epsilon} (1 - v)^2 \, dx$$



We choose :

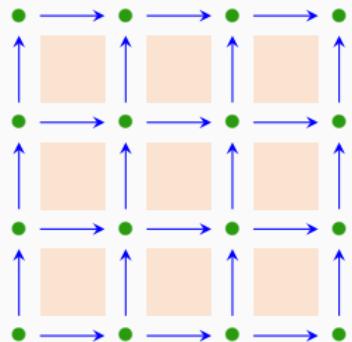
- functions  $u, g$  to live on faces
  - $u, g$  are **2-forms**
  - equivalently dual **0-forms**
- function  $v$  to live on vertices
  - $v$  is a **0-form**



$$AT_\epsilon^{2,0}(u, v) = \alpha (u - g, u - g)_{\Omega, 2} + \lambda \epsilon (\mathbf{d}_0 v, \mathbf{d}_0 v)_{\Omega, 1}$$

# Discrete formulation of AT

$$AT_\epsilon(u, v) = \alpha \int_{\Omega} |u - g|^2 \, dx + \int_{\Omega} v^2 |\nabla u|^2 \, dx + \lambda \int_{\Omega} \epsilon |\nabla v|^2 + \frac{1}{4\epsilon} (1 - v)^2 \, dx$$



We choose :

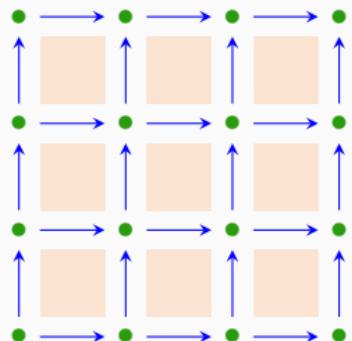
- functions  $u, g$  to live on faces
  - $u, g$  are **2-forms**
  - equivalently dual **0-forms**
- function  $v$  to live on vertices
  - $v$  is a **0-form**



$$\begin{aligned} AT_\epsilon^{2,0}(u, v) &= \alpha (u - g, u - g)_{\Omega, 2} \\ &+ \lambda \epsilon (\mathbf{d}_0 v, \mathbf{d}_0 v)_{\Omega, 1} + \frac{\lambda}{4\epsilon} (1 - v, 1 - v)_{\Omega, 0} \end{aligned}$$

# Discrete formulation of AT

$$AT_\epsilon(u, v) = \alpha \int_{\Omega} |u - g|^2 \, dx + \int_{\Omega} v^2 |\nabla u|^2 \, dx + \lambda \int_{\Omega} \epsilon |\nabla v|^2 + \frac{1}{4\epsilon} (1 - v)^2 \, dx$$



We choose:

- functions  $u, g$  to live on faces
  - $u, g$  are **2-forms**
  - equivalently dual **0-forms**
- function  $v$  to live on vertices
  - $v$  is a **0-form**



$$\begin{aligned} AT_\epsilon^{2,0}(u, v) &= \alpha (u - g, u - g)_{\Omega, 2} + (v, v)_{\Omega, 0} \\ &\quad + \lambda \epsilon (\mathbf{d}_0 v, \mathbf{d}_0 v)_{\Omega, 1} + \frac{\lambda}{4\epsilon} (1 - v, 1 - v)_{\Omega, 0} \end{aligned}$$

## Discrete formulation of AT

$$\int_{\Omega} v^2 |\nabla u|^2 \, dx = (\textcolor{green}{v} \delta_2 \textcolor{orange}{u}, \textcolor{green}{v} \delta_2 \textcolor{orange}{u})_{\Omega,1}$$

- $\textcolor{green}{v} \delta_2 \textcolor{orange}{u} = \textcolor{green}{v} \wedge \delta_2 \textcolor{orange}{u} = \text{diag}(\mathbf{M}_{01} \textcolor{green}{v}) \delta_2 \textcolor{orange}{u}$

## Discrete formulation of AT

$$\int_{\Omega} v^2 |\nabla u|^2 \, dx = (\mathbf{v} \delta_2 \mathbf{u}, \mathbf{v} \delta_2 \mathbf{u})_{\Omega,1}$$

- $\mathbf{v} \delta_2 \mathbf{u} = \mathbf{v} \wedge \delta_2 \mathbf{u} = \text{diag}(\mathbf{M}_{01} \mathbf{v}) \delta_2 \mathbf{u}$

• 0.8      0.8      1.0

• 1.0      0.0      0.2

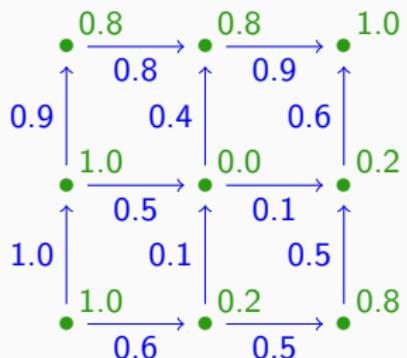
• 1.0      0.2      0.8

- 0-form  $\mathbf{v}$

## Discrete formulation of AT

$$\int_{\Omega} v^2 |\nabla u|^2 \, dx = (\textcolor{green}{v} \delta_2 \textcolor{orange}{u}, \textcolor{green}{v} \delta_2 \textcolor{orange}{u})_{\Omega,1}$$

- $\textcolor{green}{v} \delta_2 \textcolor{orange}{u} = \textcolor{green}{v} \wedge \delta_2 \textcolor{orange}{u} = \text{diag}(\mathbf{M}_{01} \textcolor{green}{v}) \delta_2 \textcolor{orange}{u}$

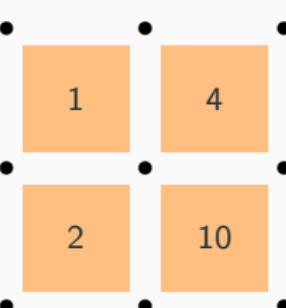
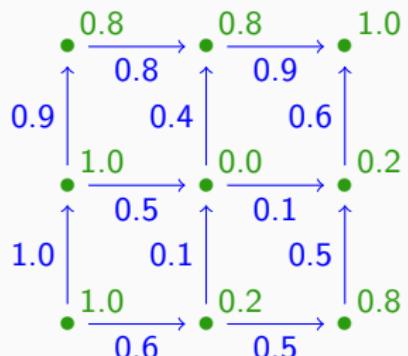


- 0-form  $v$
- 1-form  $\mathbf{M}_{01}v$

## Discrete formulation of AT

$$\int_{\Omega} v^2 |\nabla u|^2 \, dx = (\textcolor{green}{v} \delta_2 \textcolor{orange}{u}, \textcolor{green}{v} \delta_2 \textcolor{orange}{u})_{\Omega,1}$$

- $\textcolor{green}{v} \delta_2 \textcolor{orange}{u} = \textcolor{green}{v} \wedge \delta_2 \textcolor{orange}{u} = \text{diag}(\mathbf{M}_{01} \textcolor{green}{v}) \delta_2 \textcolor{orange}{u}$



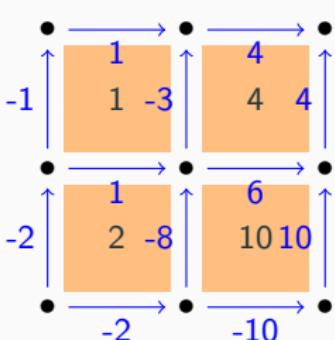
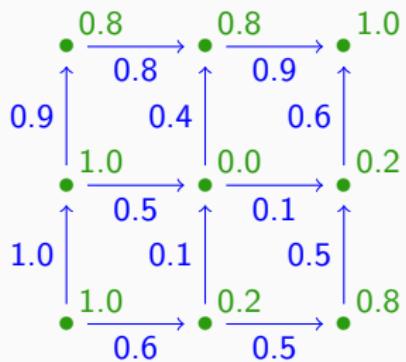
- 2-form  $\textcolor{orange}{u}$

- 0-form  $\textcolor{green}{v}$
- 1-form  $\mathbf{M}_{01} \textcolor{green}{v}$

## Discrete formulation of AT

$$\int_{\Omega} v^2 |\nabla u|^2 \, dx = (\textcolor{green}{v} \delta_2 \textcolor{orange}{u}, \textcolor{green}{v} \delta_2 \textcolor{orange}{u})_{\Omega,1}$$

- $\textcolor{green}{v} \delta_2 \textcolor{orange}{u} = \textcolor{green}{v} \wedge \delta_2 \textcolor{orange}{u} = \text{diag}(\mathbf{M}_{01} \textcolor{green}{v}) \delta_2 \textcolor{orange}{u}$

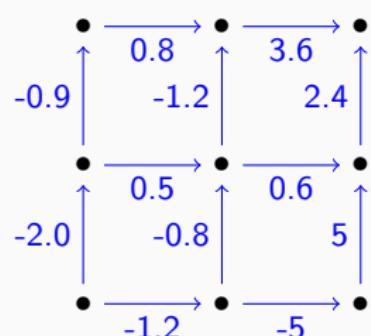
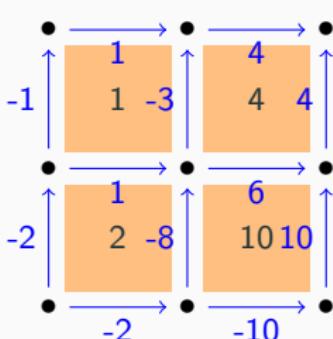
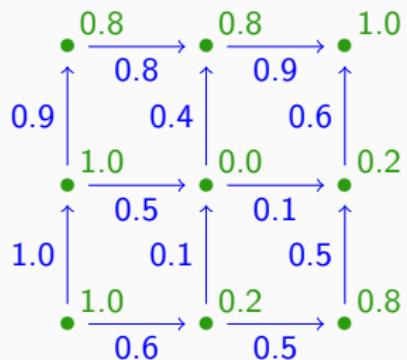


- 0-form  $v$
- 1-form  $\mathbf{M}_{01} \textcolor{green}{v}$
- 2-form  $u$
- 1-form  $\delta_2 \textcolor{orange}{u}$

## Discrete formulation of AT

$$\int_{\Omega} v^2 |\nabla u|^2 \, dx = (\mathbf{v} \delta_2 \mathbf{u}, \mathbf{v} \delta_2 \mathbf{u})_{\Omega,1}$$

- $\mathbf{v} \delta_2 \mathbf{u} = \mathbf{v} \wedge \delta_2 \mathbf{u} = \text{diag}(\mathbf{M}_{01} \mathbf{v}) \delta_2 \mathbf{u}$



- 0-form  $v$
- 1-form  $\mathbf{M}_{01} \mathbf{v}$
- 2-form  $\mathbf{M}_{01} \mathbf{M}_{01} \mathbf{v}$
- 2-form  $\mathbf{v}$
- 1-form  $\delta_2 \mathbf{u}$

- 1-form  $\text{diag}(\mathbf{M}_{01} \mathbf{v}) \delta_2 \mathbf{u}$

## Discrete formulation of AT

$$\begin{aligned} \text{AT}_{\epsilon}^{2,0}(\mathbf{u}, \mathbf{v}) &= \alpha (\mathbf{u} - \mathbf{g}, \mathbf{u} - \mathbf{g})_{\Omega,2} + (\text{diag}(\mathbf{M}_{01}\mathbf{v})\delta_2\mathbf{u}, \text{diag}(\mathbf{M}_{01}\mathbf{v})\delta_2\mathbf{u})_{\Omega,1} \\ &\quad + \lambda\epsilon (\mathbf{d}_0\mathbf{v}, \mathbf{d}_0\mathbf{v})_{\Omega,1} + \frac{\lambda}{4\epsilon} (\mathbf{1} - \mathbf{v}, \mathbf{1} - \mathbf{v})_{\Omega,0} \end{aligned}$$

## Discrete formulation of AT

$$\begin{aligned} \text{AT}_\epsilon^{2,0}(\mathbf{u}, \mathbf{v}) &= \alpha (\mathbf{u} - \mathbf{g}, \mathbf{u} - \mathbf{g})_{\Omega,2} + (\text{diag}(\mathbf{M}_{01}\mathbf{v})\delta_2\mathbf{u}, \text{diag}(\mathbf{M}_{01}\mathbf{v})\delta_2\mathbf{u})_{\Omega,1} \\ &\quad + \lambda\epsilon (\mathbf{d}_0\mathbf{v}, \mathbf{d}_0\mathbf{v})_{\Omega,1} + \frac{\lambda}{4\epsilon} (\mathbf{1} - \mathbf{v}, \mathbf{1} - \mathbf{v})_{\Omega,0} \end{aligned}$$

- with matrices  $\mathbf{A} := \mathbf{d}_0$ ,  $\mathbf{B}' := \delta_2$ ,  $\mathbf{G}_k := \star_k$ .

$$\begin{aligned} \text{AT}_\epsilon^{2,0}(\mathbf{u}, \mathbf{v}) &= \alpha(\mathbf{u} - \mathbf{g})^\top \mathbf{G}_2(\mathbf{u} - \mathbf{g}) + \mathbf{u}^\top \mathbf{B}'^\top \text{diag}(\mathbf{M}_{01}\mathbf{v})^2 \mathbf{G}_1 \mathbf{B}' \mathbf{u} \\ &\quad + \lambda\epsilon \mathbf{v}^\top \mathbf{A}^\top \mathbf{G}_1 \mathbf{A} \mathbf{v} + \frac{\lambda}{4\epsilon} (\mathbf{1} - \mathbf{v})^\top \mathbf{G}_0 (\mathbf{1} - \mathbf{v}) \end{aligned}$$

## Discrete formulation of AT

$$\begin{aligned} \text{AT}_{\epsilon}^{2,0}(\mathbf{u}, \mathbf{v}) &= \alpha (\mathbf{u} - \mathbf{g}, \mathbf{u} - \mathbf{g})_{\Omega,2} + (\text{diag}(\mathbf{M}_{01}\mathbf{v})\delta_2\mathbf{u}, \text{diag}(\mathbf{M}_{01}\mathbf{v})\delta_2\mathbf{u})_{\Omega,1} \\ &\quad + \lambda\epsilon (\mathbf{d}_0\mathbf{v}, \mathbf{d}_0\mathbf{v})_{\Omega,1} + \frac{\lambda}{4\epsilon} (\mathbf{1} - \mathbf{v}, \mathbf{1} - \mathbf{v})_{\Omega,0} \end{aligned}$$

- with matrices  $\mathbf{A} := \mathbf{d}_0$ ,  $\mathbf{B}' := \delta_2$ ,  $\mathbf{G}_k := \star_k$ .

$$\begin{aligned} \text{AT}_{\epsilon}^{2,0}(\mathbf{u}, \mathbf{v}) &= \alpha(\mathbf{u} - \mathbf{g})^T \mathbf{G}_2(\mathbf{u} - \mathbf{g}) + \mathbf{u}^T \mathbf{B}'^T \text{diag}(\mathbf{M}_{01}\mathbf{v})^2 \mathbf{G}_1 \mathbf{B}' \mathbf{u} \\ &\quad + \lambda\epsilon \mathbf{v}^T \mathbf{A}^T \mathbf{G}_1 \mathbf{A} \mathbf{v} + \frac{\lambda}{4\epsilon} (\mathbf{1} - \mathbf{v})^T \mathbf{G}_0 (\mathbf{1} - \mathbf{v}) \end{aligned}$$

- Euler-Lagrange:  $\min_{\mathbf{u}, \mathbf{v}} \text{AT}_{\epsilon}^{2,0} \Rightarrow \frac{d\text{AT}_{\epsilon}^{2,0}}{d\mathbf{u}} = 0$  and  $\frac{d\text{AT}_{\epsilon}^{2,0}}{d\mathbf{v}} = 0$
- $\text{AT}_{\epsilon}^{2,0}$  is **quadratic** in  $\mathbf{u}$  and in  $\mathbf{v}$

## Discrete formulation of AT

$$\begin{aligned} \text{AT}_{\epsilon}^{2,0}(\mathbf{u}, \mathbf{v}) &= \alpha(\mathbf{u} - \mathbf{g}, \mathbf{u} - \mathbf{g})_{\Omega,2} + (\text{diag}(\mathbf{M}_{01}\mathbf{v})\delta_2\mathbf{u}, \text{diag}(\mathbf{M}_{01}\mathbf{v})\delta_2\mathbf{u})_{\Omega,1} \\ &\quad + \lambda\epsilon(\mathbf{d}_0\mathbf{v}, \mathbf{d}_0\mathbf{v})_{\Omega,1} + \frac{\lambda}{4\epsilon}(\mathbf{1} - \mathbf{v}, \mathbf{1} - \mathbf{v})_{\Omega,0} \end{aligned}$$

- with matrices  $\mathbf{A} := \mathbf{d}_0$ ,  $\mathbf{B}' := \delta_2$ ,  $\mathbf{G}_k := \star_k$ .

$$\begin{aligned} \text{AT}_{\epsilon}^{2,0}(\mathbf{u}, \mathbf{v}) &= \alpha(\mathbf{u} - \mathbf{g})^T \mathbf{G}_2(\mathbf{u} - \mathbf{g}) + \mathbf{u}^T \mathbf{B}'^T \text{diag}(\mathbf{M}_{01}\mathbf{v})^2 \mathbf{G}_1 \mathbf{B}' \mathbf{u} \\ &\quad + \lambda\epsilon \mathbf{v}^T \mathbf{A}^T \mathbf{G}_1 \mathbf{A} \mathbf{v} + \frac{\lambda}{4\epsilon}(\mathbf{1} - \mathbf{v})^T \mathbf{G}_0(\mathbf{1} - \mathbf{v}) \end{aligned}$$

- Euler-Lagrange:  $\min_{\mathbf{u}, \mathbf{v}} \text{AT}_{\epsilon}^{2,0} \Rightarrow \frac{d\text{AT}_{\epsilon}^{2,0}}{d\mathbf{u}} = 0$  and  $\frac{d\text{AT}_{\epsilon}^{2,0}}{d\mathbf{v}} = 0$
- $\text{AT}_{\epsilon}^{2,0}$  is quadratic in  $\mathbf{u}$  and in  $\mathbf{v}$
- We solve alternatively for  $\mathbf{u}$  and  $\mathbf{v}$  the sparse linear systems:

$$\left\{ \begin{array}{l} [\alpha \mathbf{G}_2 - \mathbf{B}'^T \text{diag}(\mathbf{M}_{01}\mathbf{v})^2 \mathbf{G}_1 \mathbf{B}'] \mathbf{u} = \alpha \mathbf{G}_2 \mathbf{g}, \\ \left[ \frac{\lambda}{4\epsilon} \mathbf{G}_0 + \lambda\epsilon \mathbf{A}^T \mathbf{G}_1 \mathbf{A} + \mathbf{M}_{01}^T \text{diag}(\mathbf{B}' \mathbf{u})^2 \mathbf{G}_1 \mathbf{M}_{01} \right] \mathbf{v} = \frac{\lambda}{4\epsilon} \mathbf{G}_0 \mathbf{1}. \end{array} \right.$$

## Discrete formulation of AT: vectorial data

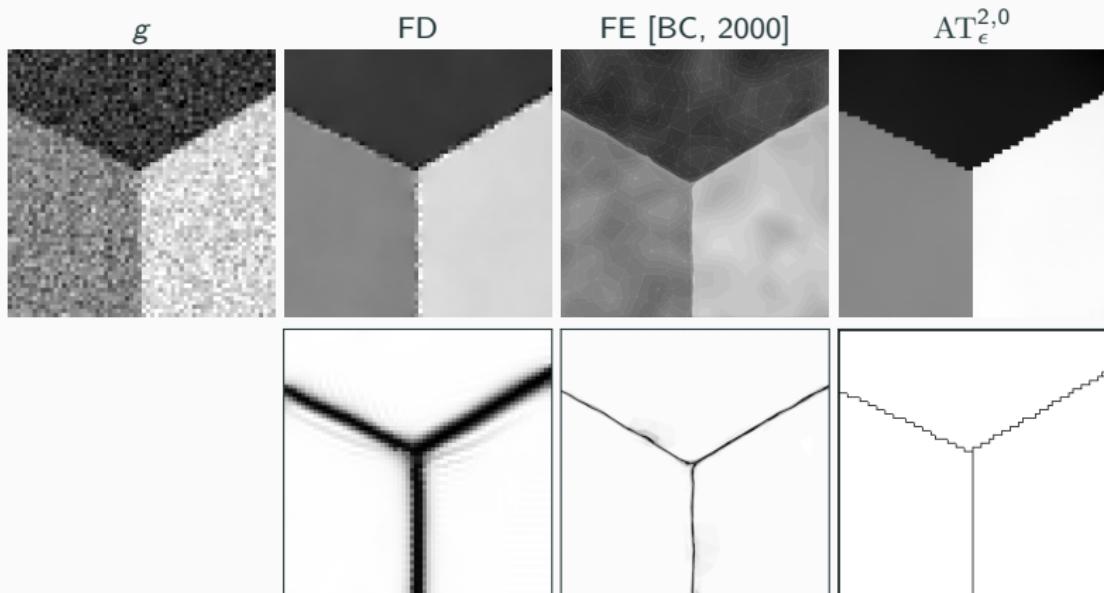
$$\begin{aligned}
 \text{AT}_{\epsilon}^{2,0}(\mathbf{u}_1, \dots, \mathbf{u}_n, \mathbf{v}) = & \alpha \sum_i (\mathbf{u}_i - \mathbf{g}_i, \mathbf{u}_i - \mathbf{g}_i)_{\Omega,2} \\
 & + \sum_i (\text{diag}(\mathbf{M}_{01}\mathbf{v})\delta_2 \mathbf{u}_i, \text{diag}(\mathbf{M}_{01}\mathbf{v})\delta_2 \mathbf{u}_i)_{\Omega,1} \\
 & + \lambda \epsilon (\mathbf{d}_0 \mathbf{v}, \mathbf{d}_0 \mathbf{v})_{\Omega,1} + \frac{\lambda}{4\epsilon} (\mathbf{1} - \mathbf{v}, \mathbf{1} - \mathbf{v})_{\Omega,0}
 \end{aligned}$$

- We solve alternatively for the  $\mathbf{u}_i$  and  $\mathbf{v}$  the sparse linear systems:

$$\left\{
 \begin{array}{lcl}
 \forall i \in \{1, \dots, n\}, [\alpha \mathbf{G}_2 - \mathbf{B}'^T \text{diag}(\mathbf{M}_{01}\mathbf{v})^2 \mathbf{G}_1 \mathbf{B}'] \mathbf{u}_i & = & \alpha \mathbf{G}_2 \mathbf{g}_i, \\
 \left[ \frac{\lambda}{4\epsilon} \mathbf{G}_0 + \lambda \epsilon \mathbf{A}^T \mathbf{G}_1 \mathbf{A} + \mathbf{M}_{01}^T (\sum_i \text{diag}(\mathbf{B}' \mathbf{u}_i)^2) \mathbf{G}_1 \mathbf{M}_{01} \right] \mathbf{v} & = & \frac{\lambda}{4\epsilon} \mathbf{G}_0 \mathbf{1}.
 \end{array}
 \right.$$

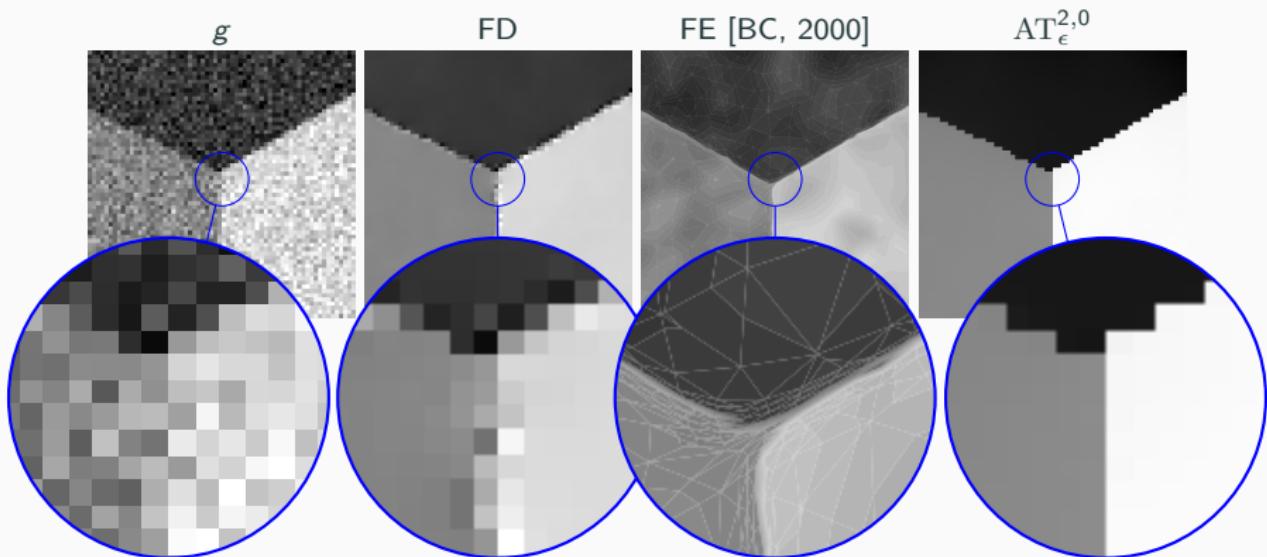
- Our algorithm progressively decreases  $\epsilon$  to get a better chance of capturing the optimum
  - $\epsilon$  follows typically sequence 2, 1, 0.5, 0.25 (for  $h = 1$  sampling)
  - results on  $\mathbf{u}$  and  $\mathbf{v}$  are starting point for next  $\epsilon$

## Image restoration on toy examples

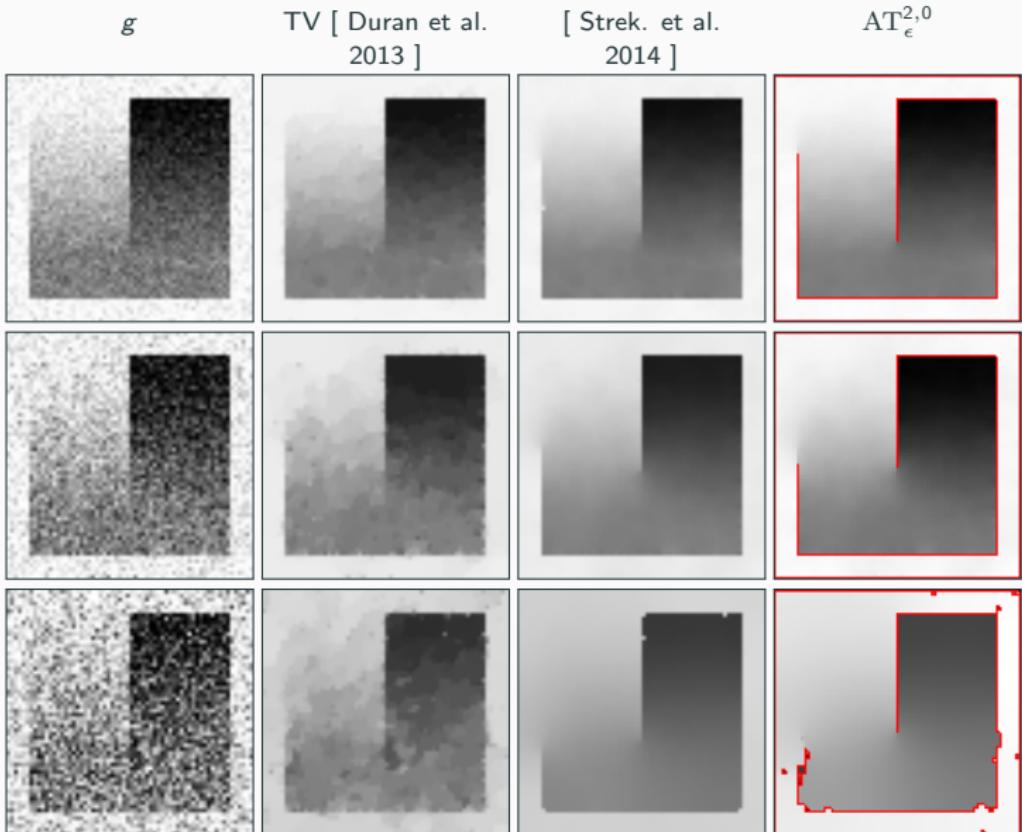


- systems are solved using Cholesky decomposition (Eigen)
- $\epsilon$  takes the successive values 2, 1, 0.5, 0.25, for sampling step  $h = 1$ .

## Image restoration on toy examples



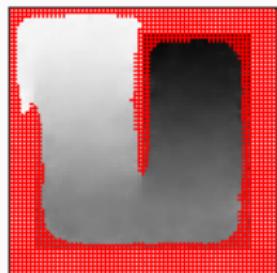
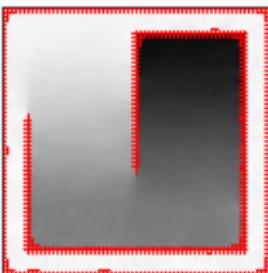
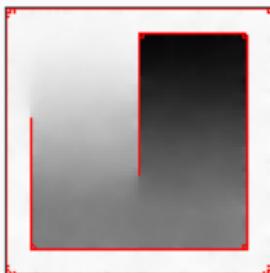
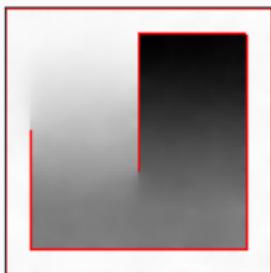
- systems are solved using Cholesky decomposition (Eigen)
- $\epsilon$  takes the successive values 2, 1, 0.5, 0.25, for sampling step  $h = 1$ .



## Influence of parameter $\epsilon$

$$AT_\epsilon(u, v) = \alpha \int_{\Omega} |u - g|^2 \, dx + \int_{\Omega} v^2 |\nabla u|^2 \, dx + \lambda \int_{\Omega} \epsilon |\nabla v|^2 + \frac{1}{\epsilon} \frac{(1-v)^2}{4} \, dx$$

- $\Gamma$ -convergence parameter
- Controls the **thickness of the contours**
  - large  $\epsilon$  convexifies  $AT$  and helps to detect the discontinuities;
  - as  $\epsilon$  goes to 0, the discontinuities become thinner and thinner.

 $\epsilon = 2 \searrow 2$  $\epsilon = 2 \searrow 1$  $\epsilon = 2 \searrow 0.5$  $\epsilon = 2 \searrow 0.25$

## Image restoration / denoising

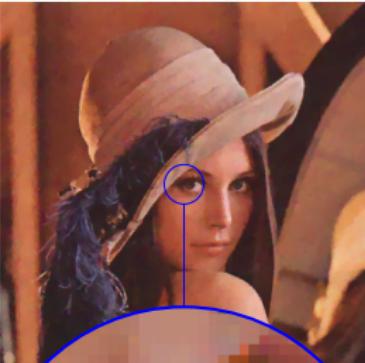
 $g$ 

(PSNR = 20.23 dB)

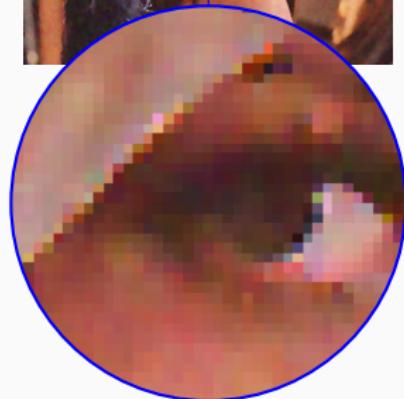
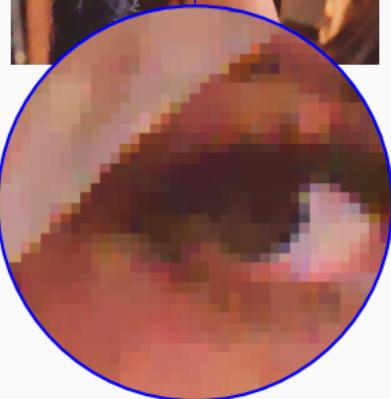
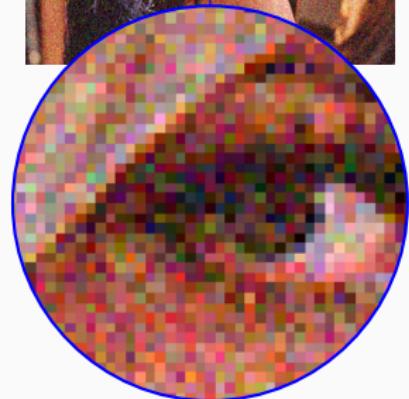


TV

(PSNR = 29.36 dB)

 $\text{AT}_{\epsilon}^{2,0}$ 

(PSNR = 29.03 dB)



## Image restoration / denoising

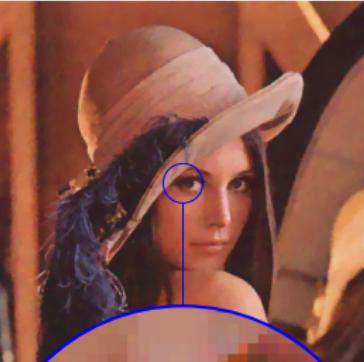
 $g$ 

(PSNR = 20.23 dB)

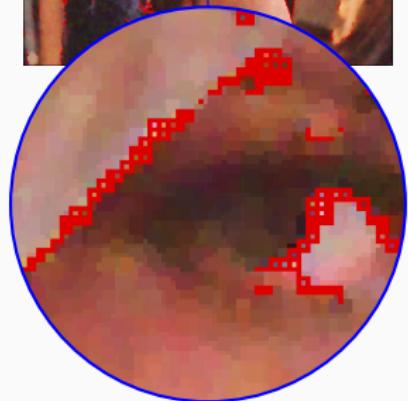
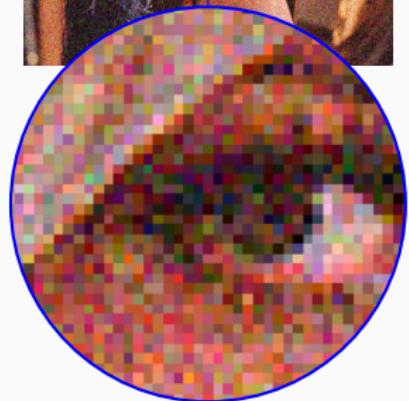


TV

(PSNR = 29.36 dB)

 $\text{AT}_{\epsilon}^{2,0}$ 

(PSNR = 29.03 dB)



## Scale-space given by $\alpha$ and $\lambda$ and image segmentation



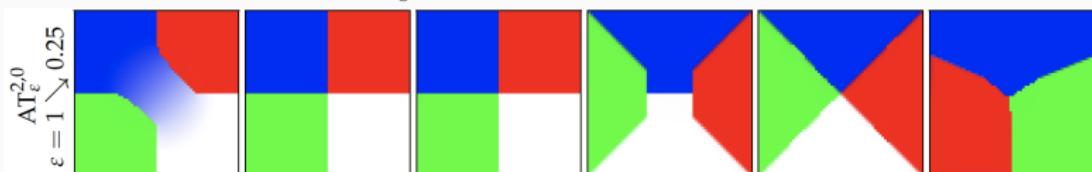
for decreasing  $\lambda$

# Image inpainting (on toy example)

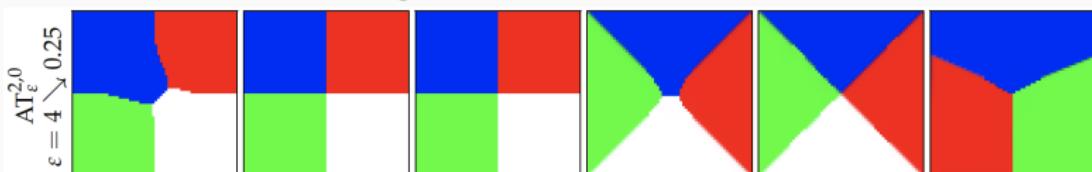
- mask (in black) : domain  $M$  where data  $g$  (in color) is unknown
- $\alpha(x) := \{\alpha \in \Omega \setminus M, 0 \text{ elsewhere}\}$
- initialization:  $u$  random in  $M$ ,  $= g$  in  $\Omega \setminus M$



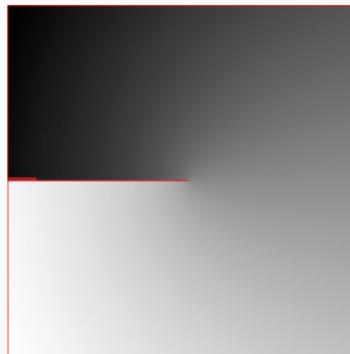
$AT_\epsilon^{2,0}$  with  $\epsilon$  from 1 to 0.25



$AT_\epsilon^{2,0}$  with  $\epsilon$  from 4 to 0.25



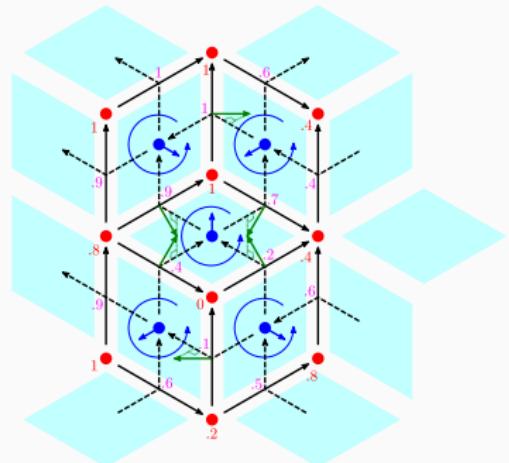
## Image inpainting (on classical crack-tip example)

 $g$ mask  $M$  $\text{AT}_\epsilon^{2,0}, \alpha = 1, \lambda = 0.0024$ 

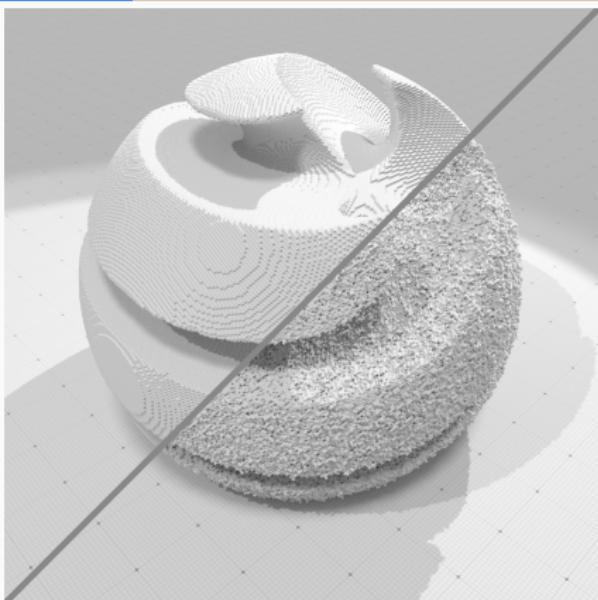
- Decreasing sequence of  $\lambda$  (irreversibility !?)
- same result as [Pock, Bishof, Cremers, Pock 2009], based on MS relaxation of [Alberti, Bouchitté, Dal Maso 2003]
- result independent of initialization as long as first  $\epsilon$  is big enough ( $\epsilon$  from 4 to 0.25 here, for image of size  $110 \times 110$ ).

## Feature delineation on digital surfaces

digital surface = boundary of set of voxels



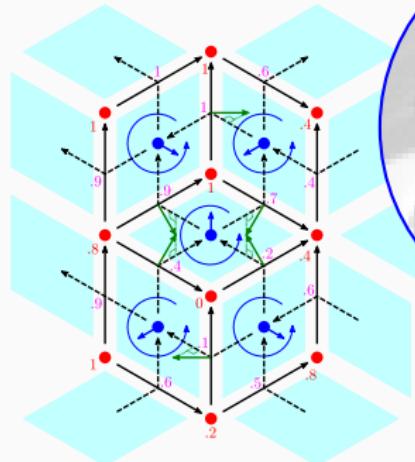
same discrete calculus  
same  $AT_{\epsilon}^{2,0}$



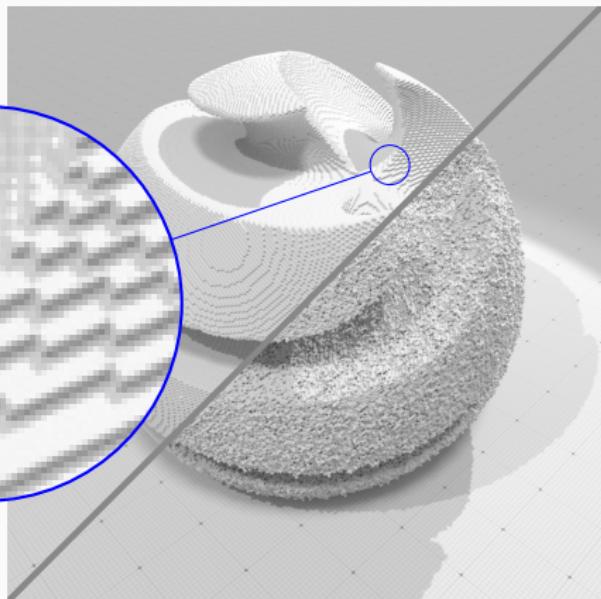
Input: normal vector field  $\mathbf{g}$  estimated by Integral Invariant digital normal estimator.

## Feature delineation on digital surfaces

digital surface = boundary of set of voxels



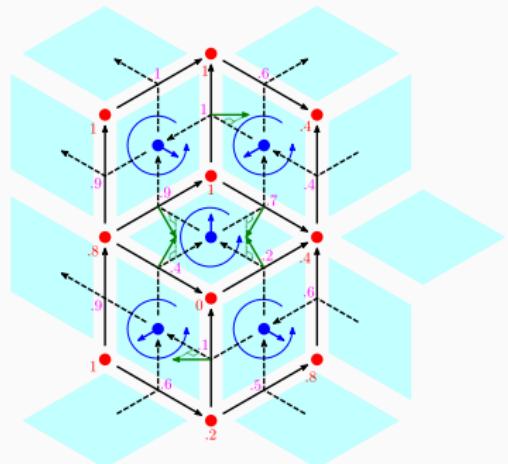
same discrete calculus  
same  $AT_{\epsilon}^{2,0}$



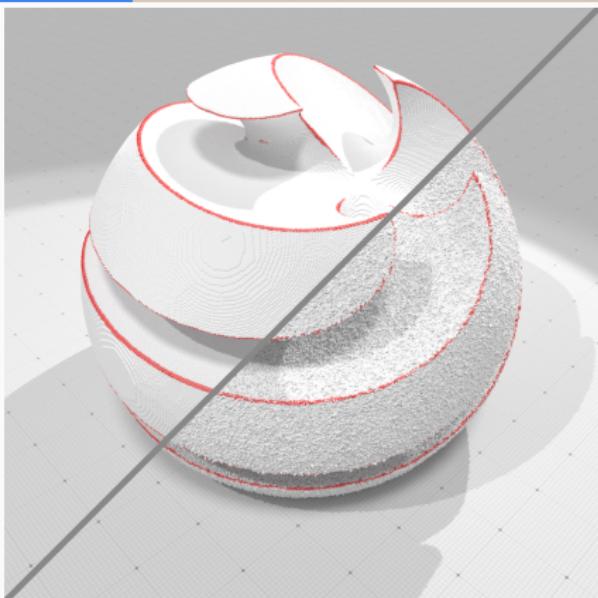
Input: normal vector field  $\mathbf{g}$  estimated by Integral Invariant digital normal estimator.

## Feature delineation on digital surfaces

digital surface = boundary of set of voxels



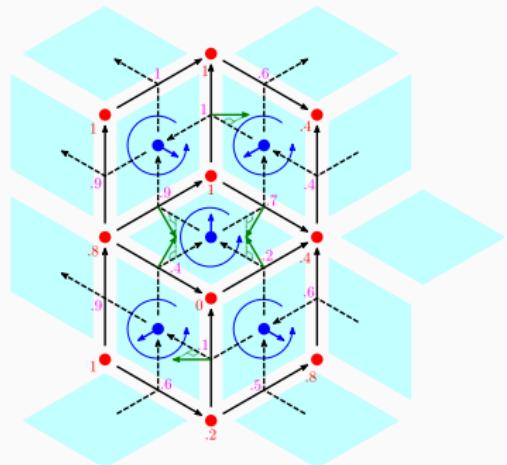
same discrete calculus  
same  $AT_{\epsilon}^{2,0}$



Input: normal vector field  $\mathbf{g}$  estimated by Integral Invariant digital normal estimator.  
Output: piecewise smooth normals  $(\mathbf{u}_i)_{i=1,2,3}$  and features  $\mathbf{v}$

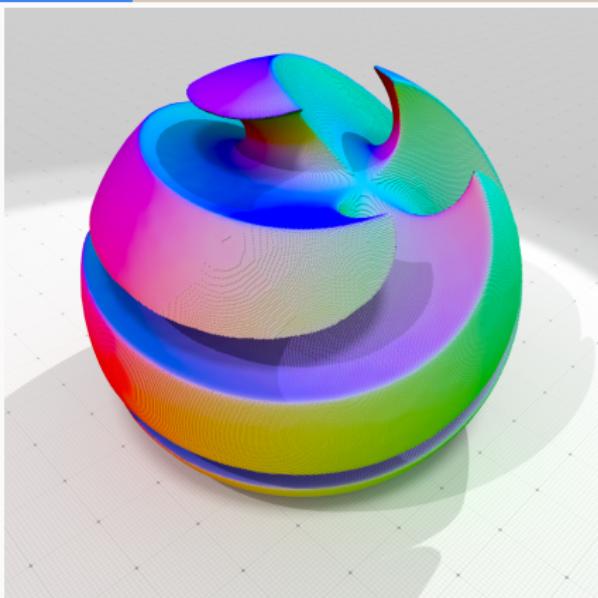
## Feature delineation on digital surfaces

digital surface = boundary of set of voxels



same discrete calculus

same  $AT_{\epsilon}^{2,0}$

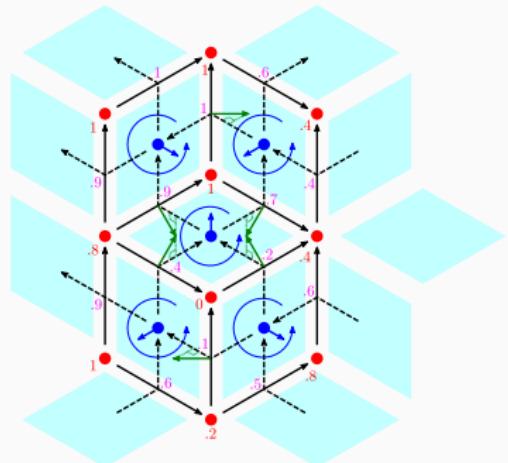


Input: normal vector field  $\mathbf{g}$  estimated by Integral Invariant digital normal estimator.

Output: piecewise smooth normals  $(\mathbf{u}_i)_{i=1,2,3}$  and features  $\mathbf{v}$

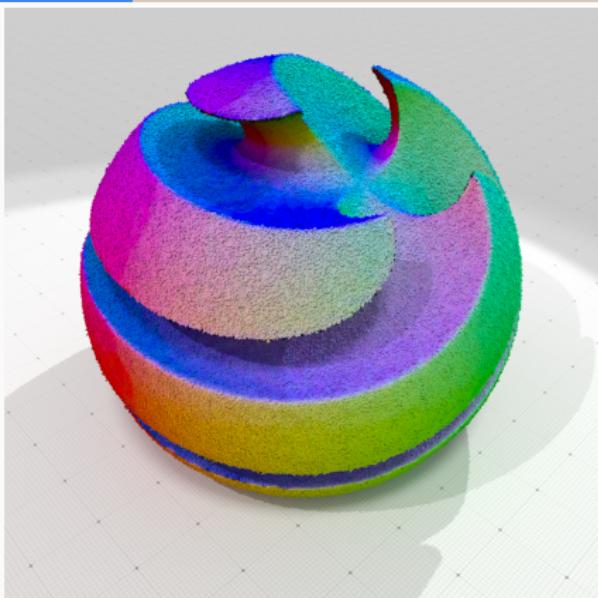
## Feature delineation on digital surfaces

digital surface = boundary of set of voxels



same discrete calculus

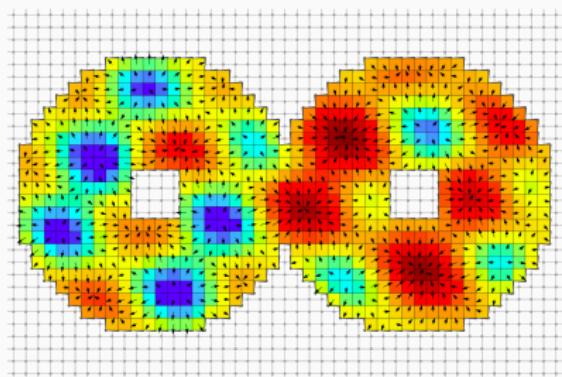
same  $AT_{\epsilon}^{2,0}$



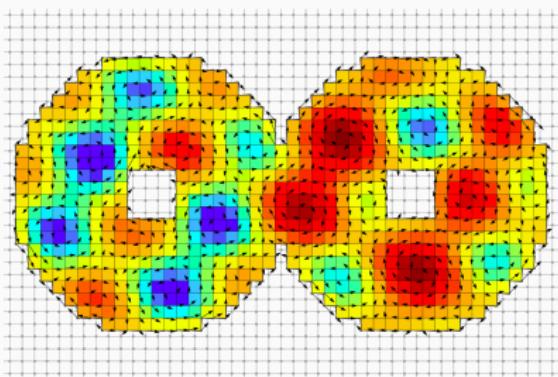
Input: normal vector field  $\mathbf{g}$  estimated by Integral Invariant digital normal estimator.

Output: piecewise smooth normals  $(\mathbf{u}_i)_{i=1,2,3}$  and features  $\mathbf{v}$

## Discrete Exterior Calculus: conclusion



curl-free  $\mathbf{d}_1 \mathbf{v} = 0$



div-free  $\delta_1 \mathbf{v} = 0$

- a sound framework for variational modeling / PDE
- well suited to digital domains: images, digital surfaces
- reduces to linear algebra computations
- visit [DEC package](#) on [dgtal.org](#) documentation

### **3. Reproducible Research**

---

### 3. Reproducible Research

#### Overview

- 3.1 (short) Introduction to Reproducible Research.
- 3.2 Rapid overview of the RR platforms.
- 3.3 New ways of publications.

### 3. Reproducible Research

#### Overview

- 3.1 (short) Introduction to Reproducible Research.
- 3.2 Rapid overview of the RR platforms.
- 3.3 New ways of publications.

#### Reproducible research in sciences:

- *Theoretical scientists share demonstrations;*

$$\begin{aligned}
 \text{CPT}^n &= 2\alpha_1 + C(T\alpha^2) \\
 &= 2\alpha_1 T + 2\alpha_2 T^2 + C(T^2\alpha^3) \\
 &= 2\alpha_1 T + 2\alpha_2 T^2 + 2\alpha_3 T^3 + C(T^3\alpha^4) \dots \text{etc.} \\
 CPO^n &= 2\alpha_1 + 2\alpha_2 + 2\alpha_3 + \dots + 2\alpha_n
 \end{aligned}$$

The recurrence equation shows that we can relate the summation equation to the previous one by adding the first term of the sequence to both sides:

$$2\alpha_1 (1 + \alpha_2 + \alpha_3 + \dots + \alpha_n) = 2\alpha_1 + 2\alpha_2 + 2\alpha_3 + \dots + 2\alpha_n$$

Since we know the converging infinite geometric series states:

$$\sum_{k=0}^{\infty} r^k = \frac{r}{1-r}, |r| < 1$$

This in turn leads us to the closed form formula:

$$2\alpha_1 + 2\alpha_2 + 2\alpha_3 + \dots + 2\alpha_n = 2\alpha_1 \cdot \frac{1 - \alpha^n}{1 - \alpha}$$

Proof by induction:

$$\begin{aligned}
 &\text{Inductive hypothesis: } 2\alpha_1 + 2\alpha_2 + \dots + 2\alpha_k = 2\alpha_1 \cdot \frac{1 - \alpha^k}{1 - \alpha} \quad \forall k < K \\
 &\text{Inductive Step:} \\
 &\quad 2\alpha_1 + 2\alpha_2 + \dots + 2\alpha_k + 2\alpha_{K+1} = 2\alpha_1 \cdot \frac{1 - \alpha^k}{1 - \alpha} + 2\alpha_{K+1} \\
 &\quad 2\alpha_1 \cdot \frac{1 - \alpha^k}{1 - \alpha} + 2\alpha_{K+1} = \frac{2\alpha_1 - 2\alpha_1 \alpha^k + 2\alpha_{K+1}}{1 - \alpha} \\
 &\quad \frac{2\alpha_1 - 2\alpha_1 \alpha^k + 2\alpha_{K+1}}{1 - \alpha} = \frac{2\alpha_1(1 - \alpha^k) + 2\alpha_{K+1}}{1 - \alpha} \\
 &\quad \frac{2\alpha_1(1 - \alpha^k) + 2\alpha_{K+1}}{1 - \alpha} = \frac{2\alpha_1 - 2\alpha_1 \alpha^k + 2\alpha_{K+1}}{1 - \alpha} \\
 &\quad \frac{2\alpha_1 - 2\alpha_1 \alpha^k + 2\alpha_{K+1}}{1 - \alpha} = \frac{2\alpha_1 - 2\alpha_1 \alpha^k + 2\alpha_{K+1}}{1 - \alpha}
 \end{aligned}$$

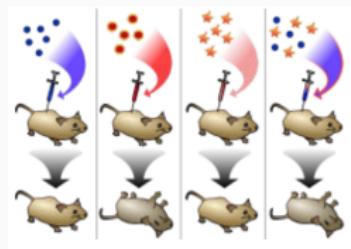
### 3. Reproducible Research

#### Overview

- 3.1 (short) Introduction to Reproducible Research.
- 3.2 Rapid overview of the RR platforms.
- 3.3 New ways of publications.

#### Reproducible research in sciences:

- *Theoretical scientists share demonstrations;*
- *Experimental scientists share procedures;*



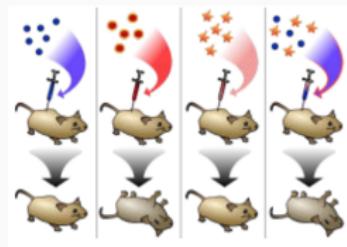
### 3. Reproducible Research

#### Overview

- 3.1 (short) Introduction to Reproducible Research.
- 3.2 Rapid overview of the RR platforms.
- 3.3 New ways of publications.

#### Reproducible research in sciences:

- *Theoretical scientists* share demonstrations;
- *Experimental scientists* share procedures;
- *Computational scientists... ?*



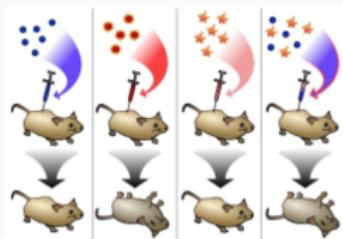
### 3. Reproducible Research

#### Overview

- 3.1 (short) Introduction to Reproducible Research.
- 3.2 Rapid overview of the RR platforms.
- 3.3 New ways of publications.

#### Reproducible research in sciences:

- *Theoretical scientists* share demonstrations;
- *Experimental scientists* share procedures;
- *Computational scientists... ?*



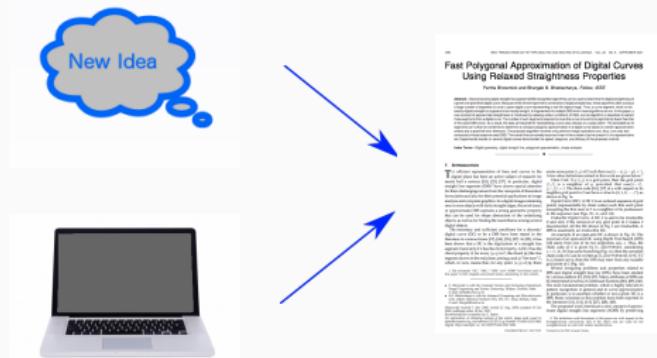
#### Computer Science:

- Description of methods/algorithms;
- description often limited (constraints on page limits);
- parameters not given or not well described;
- steps of pre/post processing missing.

## 3.1 (short) Introduction to Reproducible Research (1)

### Research in Computer Science:

- 1 New idea;
- 2 demonstration, implementation;
- 3 article publication.



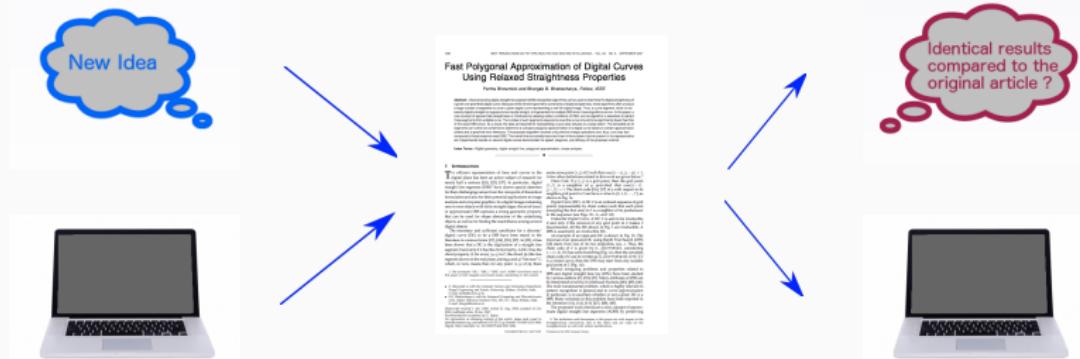
### 3.1 (short) Introduction to Reproducible Research (1)

#### Research in Computer Science:

- 1 New idea;
- 2 demonstration, implementation;
- 3 article publication.

#### Reusable Research:

- 1 Article which seems interesting;
- 2 re-implement the algorithm;
- 3 result conformity with the original.



## 3.1 (short) Introduction to Reproducible Research (2)

- RR redefines the result of the research. Not just a paper, but also all the procedures needed to obtain the same published results.

## 3.1 (short) Introduction to Reproducible Research (2)

- RR redefines the result of the research. Not just a paper, but also all the procedures needed to obtain the same published results.
- Which items? The article itself, the source code, and the data.

### 3.1 (short) Introduction to Reproducible Research (2)

- RR redefines the result of the research. Not just a paper, but also all the procedures needed to obtain the same published results.
- Which items? The article itself, the source code, and the data.
- The paper can be seen just as the advertising of the scholarship:  
Definition of a “reproducible scientific publication” given by Claerbout and followed also by Buckheit and Donoho [Buckheit & Donoho 95]:

*"An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship.*

*The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures."*

### 3.1 (short) Introduction to Reproducible Research (2)

- RR redefines the result of the research. Not just a paper, but also all the procedures needed to obtain the same published results.
- Which items? The article itself, the source code, and the data.
- The paper can be seen just as the advertising of the scholarship:  
Definition of a “reproducible scientific publication” given by Claerbout and followed also by Buckheit and Donoho [Buckheit & Donoho 95]:

*"An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship.*

*The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures."*

#### Recent survey from Nature [Baker et al. 16]

Mention that more than 70% of researchers have already failed to reproduce results from other researchers.

### 3.1 (short) Introduction to Reproducible Research (2)

- RR redefines the result of the research. Not just a paper, but also all the procedures needed to obtain the same published results.
- Which items? The article itself, the source code, and the data.
- The paper can be seen just as the advertising of the scholarship:  
Definition of a “reproducible scientific publication” given by Claerbout and followed also by Buckheit and Donoho [Buckheit & Donoho 95]:

*"An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship.*

*The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures."*

#### Recent survey from Nature [Baker et al. 16]

Mention that more than 70% of researchers have already failed to reproduce results from other researchers.

⇒ more than the half affirm to also fail to reproduce their own results.

## 3.1 (short) Introduction to Reproducible Research (3)

### Benefit for authors

- Less prone to errors: since all the methods are clear, public, and therefore auditable.

## 3.1 (short) Introduction to Reproducible Research (3)

### Benefit for authors

- Less prone to errors: since all the methods are clear, public, and therefore auditable.
- Good practices.  
→ Reliable science, the descriptions match what the published content.

## 3.1 (short) Introduction to Reproducible Research (3)

### Benefit for authors

- Less prone to errors: since all the methods are clear, public, and therefore auditable.
- Good practices.  
→ Reliable science, the descriptions match what the published content.
- The descriptions, methodology, source code, and data is available to the scientific community.

## 3.1 (short) Introduction to Reproducible Research (3)

### Benefit for authors

- Less prone to errors: since all the methods are clear, public, and therefore auditable.
- Good practices.  
→ Reliable science, the descriptions match what the published content.
- The descriptions, methodology, source code, and data is available to the scientific community.
- Comparison of methods easier (or even possible!).

## 3.1 (short) Introduction to Reproducible Research (3)

### Benefit for authors

- Less prone to errors: since all the methods are clear, public, and therefore auditable.
- Good practices.  
→ Reliable science, the descriptions match what the published content.
- The descriptions, methodology, source code, and data is available to the scientific community.
- Comparison of methods easier (or even possible!).

### Benefit for authors

- possibility of publishing research of high quality.
- increase of the visibility of the publications

## 3.2 Rapid overview of the RR platforms

- **Galaxy** - <https://galaxyproject.org>
- IPython - <https://ipython.org>
- Jupyter - <http://jupyter.org>
- **Code Ocean** - <https://codeocean.com>
- Research Compendia - <http://researchcompendia.science>
- RunMyCode - <http://www.runmycode.org>
- **DAE** - <http://dae.cse.lehigh.edu/DAE>
- **IPOL** - <https://www.ipol.im>

## 3.2 Rapid overview of the RR platforms: Galaxy

### Description:

- Platform for **genomic research**.
- It makes available **tools** which can be used by **non-expert** users too.
- Galaxy defines a **workflow** as a **reusable templates** which contains different algorithms applied to the input data.
- In order to achieve **reproducibility** the system stores:
  - the **input dataset**,
  - the **tools** and **algorithms** which were applied to the data along the **chain**,
  - the **parameters**,
  - the **output** result.

<https://galaxyproject.org>

## 3.2 Rapid overview of the RR platforms: Jupyter

### Description:

- Spin-off of IPython in 2014.
- Main goal: separate the Python language used in IPython from all the other functionalities needed to run the notebooks (for example, the notebook format, the web framework, or the message protocols).
- Languages: execution kernels in Jupyter.
- Nowadays it supports more than 40 languages that can be used as kernels.

<http://jupyter.org>

## 3.2 Rapid overview of the RR platforms: Code Ocean

### Description:

- Stated in **2014** as part *Runway Startup Postdoc Program* at the Jacobs Technion Cornell Institute.
- Sponsored by **IEEE**.
- Defined by themselves as a **computational reproducibility platform**.
- **Not a journal** itself. It only runs code, but not publishes articles.
- Assigns a **DOI** to each source code.
- **Several languages** accepted: Python, R, Julia, Matlab, Octave, C++, Fortran, Perl, Java
- They claim *to view and download for everyone for free*. The free plan is **limited** and the other are **paid**.
- **Plans** based on **CPU time** and **storage** usage. For example, the *researcher* plan allows 1h CPU and 5GB of storage per month.
- **No statistics** on usage. **Seems low** by website inspection.

<https://codeocean.com/>

## 3.2 Rapid overview of the RR platforms: DAE

### Description: [Lamiroy & Lopresti 16]

- Platform for Document Analysis and Exploitation.
- Allows to run document analysis algorithms and apply comparisons.

The screenshot shows the homepage of the DAE platform at [dae.cse.lehigh.edu](http://dae.cse.lehigh.edu). The page has a dark header with the title "Document Analysis and Exploitation". The left sidebar contains links for "Project Description", "Events and Initiatives", "Analysis Services", "Technical Issues", and "Navigation". The main content area features a section titled "Algorithms" with a list of available tools like convert, Stanford-NER, Tesseract, ocrad, NCI-CADD segmentation, MergeImageList, NCI-CADD binarization, DICE, QGar Arc Detection, Kanungo Degradation, and ArcEval. To the right, there's a sidebar with "Recent blog posts" and a "Twitter" feed.

<http://dae.cse.lehigh.edu>

## 3.2 Rapid overview of the RR platforms: DAE

### Description: [Lamiroy & Lopresti 16]

- Platform for Document Analysis and Exploitation.
- Allows to run document analysis algorithms and apply comparisons.
- RR framework for document analysis with image data base.

The screenshot shows the DAE platform interface. At the top, there is a navigation bar with links for User login, Contact us, Copyright Alert, and Job Offerings. A sidebar on the left contains sections for Project Description, Events and Initiatives, Analysis Services, Technical Issues, and Navigation. The main content area features a "Browse Data" section displaying various datasets with thumbnails, titles, ratings, and tags. A sidebar on the right shows recent blog posts and a Twitter feed.

**User login**   Username:  Password:  Log In Request new password

Contact us Copyright Alert Job Offerings

**Document Analysis and Exploitation**

**Home**

**Project Description**

- DAE
- Tutorial
- Give Your Opinion

**Events and Initiatives**

- DAS 2012
- ICDAR 2011 Contest
- DAS 2010

**Analysis Services**

- Algorithms
- Browse Data
  - About Copyright
- Other Resources

**Technical Issues**

- Give Your Input

**Navigation**

- Recent posts

**Browse Data**

Browse: New Top Rated Popular Search Names and Tags Filter: All Datasets Root Datasets Page Images

**Lehigh Notebook** 0 TAGS: No Tags Yet

**UNLV** 2 TAGS: No Tags Yet

**GREC 2011 Arc Segmentation Contest [Test Images]** 0 TAGS: No Tags Yet

**GREC 2011 Symbol Recognition Training Set** 0 TAGS: No Tags Yet

**GREC 2011 Symbol Recognition (OBSOLETE)**

**Recent blog posts**

- Account Requests
- DAE is on Twitter
- Source Code Available
- DAE Public Live
- DAS 2010 Demo Poster

more

**Twitter**

## 3.2 Rapid overview of the RR platforms: IPOL Journal

### Description: [Arevalo et al. 16]

- A complete peer-reviewed journal can be considered as a platform.
- Image Processing domain.
- Focused on mathematical rigorness. Detailed descriptions.
- Fast to create new demos for editors: automatic system.
- Accepted languages: C/C++, Python, MATLAB, Octave.
- Free to use/submit.
- Next move: machine learning applications. Servers with GPU.

<https://www.ipol.im>

### 3.3 New Ways of Publications

#### Recent original journals

- IPOL

<https://www.ipol.im>

- ReScience

<http://rescience.github.io>

- JOSS

<https://joss.theoj.org>

### 3.3 New Ways of Publications: Image Processing On Line

Origin: <http://www.ipol.im>

- Journal started in October 2009.
- Initiative of Nicolas Limare and Jean-Michel Morel (CMLA).
- First article published in 2010.
- Domain of Image Processing.



### 3.3 New Ways of Publications: Image Processing On Line

**Origin:** <http://www.ipol.im>

- Journal started in October 2009.
- Initiative of Nicolas Limare and Jean-Michel Morel (CMLA).
- First article published in 2010.
- Domain of Image Processing.

#### Motivations

- Reproducible Research.
- New way to publish research results.
- Allows everybody to test the algorithms (with their own images).
- Free online demonstration (user-platform independant) and source code.

## 3.3 New Ways of Publications: Image Processing On Line

### Short overview

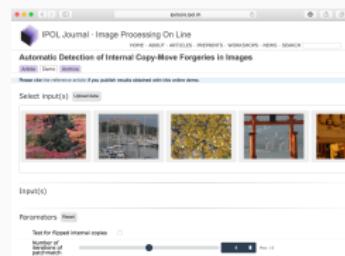
- Journal publishing **algorithm** description, **source code**, online demonstration with experiment archives.



## 3.3 New Ways of Publications: Image Processing On Line

### Short overview

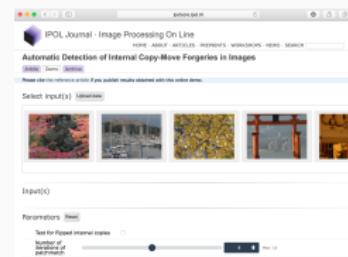
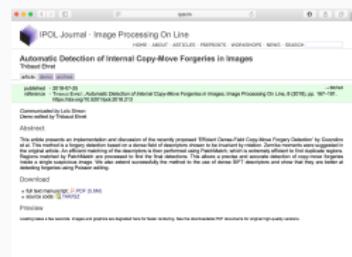
- Journal publishing algorithm description, source code, **online demonstration** with experiment archives.



### 3.3 New Ways of Publications: Image Processing On Line

#### Short overview

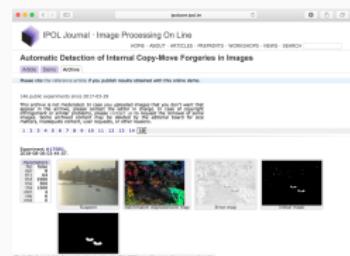
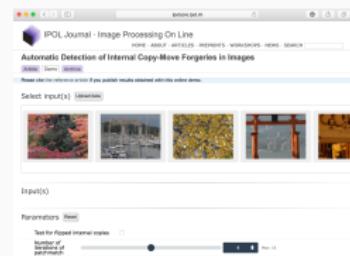
- Journal publishing algorithm description, source code, online demonstration with **experiment archives**.



### 3.3 New Ways of Publications: Image Processing On Line

#### Short overview

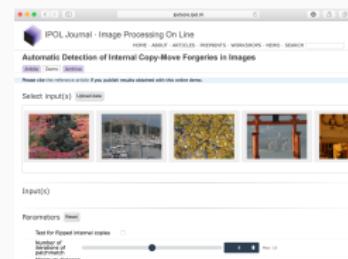
- Journal publishing algorithm description, source code, online demonstration with **experiment archives**.
- The peer-review process includes the article, and source code.



### 3.3 New Ways of Publications: Image Processing On Line

#### Short overview

- Journal publishing algorithm description, source code, online demonstration with **experiment archives**.
- The peer-review process includes the article, and source code.
- Open Science journal and Reproducible Research.



### 3.3 New Ways of Publications: Image Processing On Line

#### Short overview

- Journal publishing algorithm description, source code, online demonstration with **experiment archives**.
- The peer-review process includes the article, and source code.
- Open Science journal and Reproducible Research.
- Like classic journal: ISSN, DOI, indexed by:  
*SCOPUS, DBLP, Scirus, Google Scholar, DOAJ, SHERPA/RoMEO, Héloïse, WorldCat, CrossRef, Ulrich, Index Copernicus, PBN, JGate, VisionBib, CVonline, JournalSeek and NewJour.*



### 3.3 New Ways of Publications: Image Processing On Line

#### Short overview

- Journal publishing algorithm description, source code, online demonstration with **experiment archives**.
- The peer-review process includes the article, and source code.
- Open Science journal and Reproducible Research.
- Like classic journal: ISSN, DOI, indexed by:  
*SCOPUS, DBLP, Scirus, Google Scholar, DOAJ, SHERPA/RoMEO, Héloïse, WorldCat, CrossRef, Ulrich, Index Copernicus, PBN, JGate, VisionBib, CVonline, JournalSeek and NewJour.*

#### Recent evolution

- New system to automatically create our own online demonstrations (see next session).
- Extended topics to **sound** and **video** processing with **3D** processing.

## 3.2 ReScience Journal

### Philosophy (<http://rescience.github.io>)

- Context of Reproducible research [Buckheit & Donoho 95].
- Explicit replication: propose a new implementation of an existing work.
- Motivated from replication problems in computational science [Hinsen 15], [Topalidou *et al.* 15], [Hinsen 14].



## 3.2 ReScience Journal

### Philosophy (<http://rescience.github.io>)

- Context of Reproducible research [Buckheit & Donoho 95].
- Explicit replication: propose a new implementation of an existing work.
- Motivated from replication problems in computational science [Hinsen 15], [Topalidou *et al.* 15], [Hinsen 14].

### Details:

- Origin: first volume in 2015.
- Editorial Board:
  - Editors-in-Chief:
    - Konrad Hinsen (Molecular Biophysics - Python, C, Racket, Clojure).
    - Nicolas P. Rougier (Comp. Neuroscience, Computer Science - Python, C/C++).
  - 11 Associate Editors with roles in: Bioinformatics; Cognitive Modelling; Computational Ecology; Computational Physics; Image processing; Ecology, High-Performance Computing; Physics; Robotics; Signal Processing

## 3.2 ReScience Journal: short overview

### Characteristics

- Same presentation as in a “classic” journal.



**Reproducible Science is good. Replicated Science is better.**

ReScience is a peer-reviewed journal that seeks computational research and encourages the explicit sharing of code and data, pushing new and open-source IPOL methods to others for anyone to use and extend. The original research is [here](#).

To achieve this goal, the while paper-journal is entirely different from other traditional journals. It is a digital publication that is freely available online. All research papers are made reproducible by providing complete source code and data, making it possible for anyone to verify the results and replicate the work. Each submission must include a self-contained executable file that can be run on a standard computer system. This file contains the code and data needed to reproduce all parts of the paper. We encourage authors to use well-documented, well-tested, and well-maintained open-source software. The journal also includes a section for comments, discussions and tests. Each submission takes the form of a pull request to our GitHub repository. The editorial process follows the same workflow as GitHub pull requests, so anyone can contribute to the review and improvement of your work. ReScience is the perfect place for you to publish your research.

ReScience is committed to design. Everything can be copied and modified. Don't hesitate to write a paper, even if you are not a scientist or engineer.

**Submit a paper.**  
 Accepted articles: [11](#)  
 Submitted articles awaiting review: [0](#)  
 Total articles: [11](#) | [Accepted: 11](#) | [Pending Review: 0](#) | [In Progress: 0](#)  
 Article priority queue: [0](#)  
 Accepted articles priority queue: [0](#)  
 In progress articles priority queue: [0](#) | [Total: 0](#)



**Current issue**

July 14, 2018 | **Volume 1, Issue 1** | [View contents](#)

“[Re] Spike Timing Dependent Plasticity Finds the Start of Repeating Patterns in Caudate-Spiny Trains”  
 Author: [Pascal Kettner](#) and [Ben F. M. Groenewald](#)  
 DOI: [https://doi.org/10.4236/rs.2018011001](#) | [PDF](#) | [HTML](#)

July 14, 2018 | **Volume 1, Issue 2** | [View contents](#)

“[Re] A New Method for Computing the Optimal Spacing Strategy in Point-to-Point Traveling Salesman Problems”  
 Author: [Dmitrii Gavrilov](#), [Nikolai Slobodkin](#), and [Boris Gromov](#)  
 DOI: [https://doi.org/10.4236/rs.201802002](#) | [PDF](#) | [HTML](#)

July 14, 2018 | **Volume 1, Issue 3** | [View contents](#)

“[Re] Implementing Reinforcement Learning: Sketching adaptive softmax”  
 Author: [Raphael Lämmle](#)  
 DOI: [https://doi.org/10.4236/rs.201803003](#) | [PDF](#) | [HTML](#)



**Overview of the submission process**

The ReScience editorial board consists of scientists who are committed in the open science community. They are experienced researchers who are familiar with the IPOL approach. Each editor is based in her/his institution and has a specific responsibility for a certain type of paper. This ensures that the review is done by people who are experts in their field and have an interest in the topic.

Submissions are first considered by a member of the editorial board, who may decide to reject the manuscript or send it to another editor for further consideration. If the editor decides to send the manuscript to another editor, they will provide feedback and reasons for doing so. The referees evaluate the code and the accompanying material. If the code and material are acceptable, the manuscript is accepted. If the code and material are not acceptable, the editor can either accept the manuscript or ask for changes. If the changes are accepted, the manuscript is rejected and authors are encouraged to submit an improved version later.

**Critics to Publication**

To be considered for publication in ReScience, any given submission must satisfy the following criteria:

- Replicability
- Reproductive
- Executable



Page 1 / 10 | [View Article](#) | [Download PDF](#)

## 3.2 ReScience Journal: short overview

### Characteristics

- Same presentation as in a “classic” journal.
- Journal living on *Github*.
- Original submission process through *Pull Request* on *Github*.

The figure consists of four screenshots arranged horizontally, illustrating the ReScience Journal's submission process:

- Screenshot 1 (Top Left):** The homepage of "The ReScience Journal" on GitHub. It features a header with "The ReScience Journal" and navigation links: MELT, READ, PRINT, EDIT, BOARD, PING. Below the header is a main content area with the heading "Reproducible Science is good. Replicated Science is better." and a paragraph about the journal's mission.
- Screenshot 2 (Top Middle):** A screenshot of the "Current issue" page. It shows a list of articles from the "July 16, 2013 Volume 1, Issue 1" issue of "IPOL Journal". The first article listed is "Spike Timing Dependent Plasticity Finds the Start of Repeating Patterns in Continuous Spike Trains" by Peter M. Hahnloser and Ben F. M. de Gennaro.
- Screenshot 3 (Top Right):** A screenshot of the "Overview of the submission process" page. It provides a brief description of the peer review process, mentioning that submissions are considered by a member of the editorial board who may decide to reject the submission or send it to a referee. It also notes that the journal uses GitHub pull requests for code and accompanying material.
- Screenshot 4 (Bottom Right):** A screenshot of a GitHub pull request titled "[#6] Spike Timing Dependent Plasticity Finds the Start of Repeating Patterns in Continuous Spike Trains". The pull request is from "Peter M. Hahnloser" and "Ben F. M. de Gennaro" to "The ReScience Journal". The commit message is "Initial Pull Request Based on the Start of Repeating Patterns in Continuous Spike Trains". The pull request has been merged and closed.

## 3.2 ReScience Journal: short overview

### Characteristics

- Same presentation as in a “classic” journal.
- Journal living on *Github*.
- Original submission process through *Pull Request* on *Github*.
- Peer reviewed journal (**reviews** and **reviewer name** given in the paper).

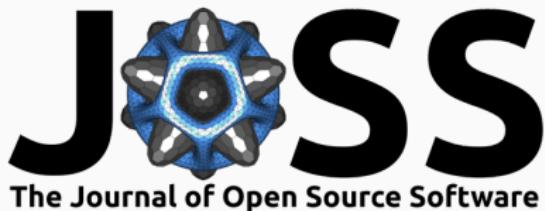
The figure consists of four screenshots of a computer screen displaying the ReScience Journal website and its GitHub repository. The top part shows the GitHub repository interface with a pull request titled "[Re] Spine Timing Dependent Plasticity Finds the Start of Repairing Patterns in Cnidarian Spide Trains". The bottom part shows four browser windows displaying different pages of the ReScience Journal:

- Reproducible Science is good. Replicated Science is better.**: A welcome message.
- Current issue**: A list of articles from the current issue, including "Spine Timing Dependent Plasticity Finds the Start of Repairing Patterns in Cnidarian Spide Trains" by Peter Harkay and Ben F. M. Graafland, published in volume 1, issue 1, 2018.
- Overview of the submission process**: Instructions for authors, mentioning that submissions are considered by a member of the editorial board, who may decide to reject the manuscript or send it to a referee. It also mentions the peer review process and the submission of revised manuscripts.
- Criteria for publication**: Guidelines for what constitutes a good submission, such as being self-contained, having a clear introduction, and so on.

### 3.3 JOSS Journal

#### Journal of Open Source Software (<https://joss.theoj.org>)

- Origin: founded by Arfon M. Smith in May 2016.
- Free and Open Access.
- Peer reviewing.
- Motivated by the fact that [Smith et al. 17]: "*Current publishing and citation do not acknowledge software as a first-class research output*".



### 3.3 JOSS Journal

#### Journal of Open Source Software (<https://joss.theoj.org>)

- Origin: founded by Arfon M. Smith in May 2016.
- Free and Open Access.
- Peer reviewing.
- Motivated by the fact that [Smith et al. 17]: "*Current publishing and citation do not acknowledge software as a first-class research output*".

#### Details

- Design: defined in the current merit system of science.
- Aim: can be considered as a "journal for research software packages".
- Editorial Board:
  - Arfon Smith (@arfon), Editor-in-Chief.
  - 19 Topic Editors: representing: Astronomy; Biodiversity Informatics; Bioinformatics; Computational Science; Data Science; Engineering; Computational Combustion; Computational Social Science; Fluid Dynamics; Energy Engineering; Geophysics; Geoscience; High Perf. Computing; Image; Information Sciences; Machine Learning; Neuroimaging; Nuclear Engineering; Open Science; Psychology; Semantic Web; Social Sciences; Software Deployment; Reproducible Research.

### 3.3 JOSS Journal: motivations

#### Motivation of JOSS Editor in chief [Smith et al. 17]:

- Software more and more present in numerous disciplines:  
⇒ from a 2014 survey 90% mention to use software and 70% indicates that they were obligatory [Hettrick et al. 14].
- Software leak of scholarly support: no ecosystem of publication, citation, acknowledge.
- JOSS is the contribution to offer modern computational research results.

### 3.3 JOSS Journal: motivations

#### Motivation of JOSS Editor in chief [Smith et al. 17]:

- Software more and more present in numerous disciplines:  
⇒ from a 2014 survey 90% mention to use software and 70% indicates that they were obligatory [Hettrick et al. 14].
- Software leak of scholarly support: no ecosystem of publication, citation, acknowledge.
- JOSS is the contribution to offer modern computational research results.

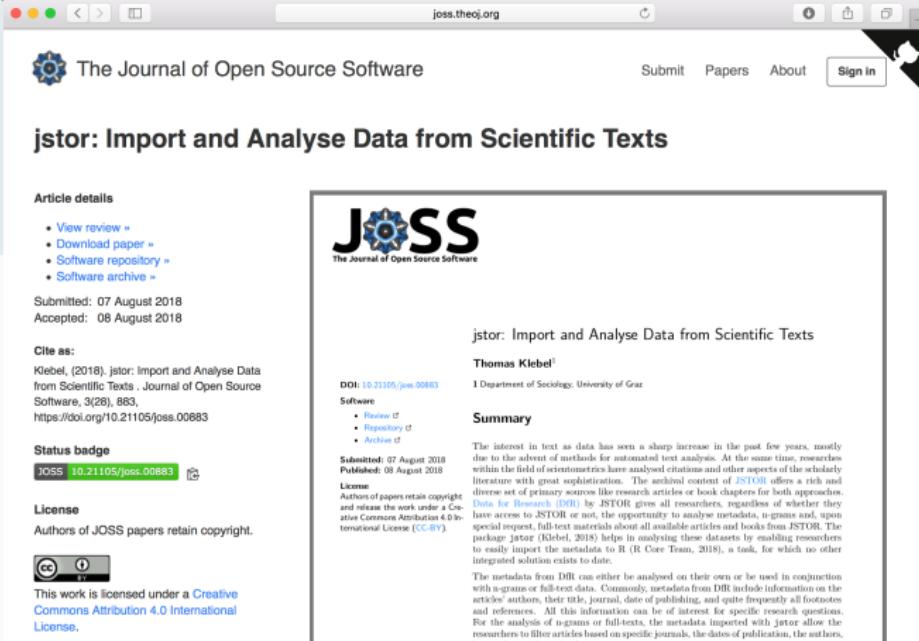
As mentioned by Buckheit and Dohono [Buckheit & Donoho 95]:

*"An article about computational science in a scientific publication is not the scholarship itself it is merely advertising of the scholarship"*

## 3.3 JOSS Journal: characteristics

### Specific form:

- Form: **voluntary short**: short abstract length (author names, list of key references, a link to software repository and a short description of the content)



The screenshot shows a web browser displaying the JOSS (The Journal of Open Source Software) website. The URL in the address bar is `joss.theoj.org`. The page title is "jstor: Import and Analyse Data from Scientific Texts". The main content area features the JOSS logo and the article details.

**Article details**

- View review »
- Download paper »
- Software repository »
- Software archive »

Submitted: 07 August 2018  
Accepted: 08 August 2018

**Cite as:**  
 Klebel, (2018). jstor: Import and Analyse Data from Scientific Texts . Journal of Open Source Software, 3(28), 883,  
<https://doi.org/10.21105/joss.00683>

**Status badge**  


**License**  
 Authors of JOSS papers retain copyright.  
  
 This work is licensed under a [Creative Commons Attribution 4.0 International License](#).

**DOI: 10.21105/joss.00683**

**Software**

- Review
- Repository
- Archive

Submitted: 07 August 2018  
Published: 08 August 2018

**License**  
 Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

**Summary**  
 The interest in text as data has seen a sharp increase in the past few years, mostly due to the advent of methods for automated text analysis. At the same time, researchers within the field of econometrics have analysed citations and other aspects of the scholarly literature with great sophistication. The archival content of JSTOR offers a rich and diverse set of primary sources like research articles or book chapters for both approaches. Data for Research (DfR) by JSTOR gives all researchers, regardless of whether they have access to the full-text or not, the ability to analyse the text in granular detail upon request, full-text analysis about all available articles and books from JSTOR. The package `jstor` (Klebel, 2018) helps in analysing these datasets by enabling researchers to easily import the metadata to R (R Core Team, 2018), a task, for which no other integrated solution exists to date.

The metadata from DfR can either be analysed on their own or be used in conjunction with n-grams or full-text data. Commonly, metadata from DfR include information on the article's authors, their title, journal, date of publishing, and quite frequently all footnotes and references. This information can be of interest for specific research questions. For the analysis of n-grams or full-texts, the metadata imported with `jstor` allow the researchers to filter articles based on specific journals, the dates of publication, the authors,

### 3.3 JOSS Journal: characteristics

#### Specific form:

- Form: **voluntary short**: short abstract length (author names, list of key references, a link to software repository and a short description of the content).
- **Not allowed**: **API** or **novel research** descriptions.

### 3.3 JOSS Journal: characteristics

#### Specific form:

- Form: **voluntary short**: short abstract length (author names, list of key references, a link to software repository and a short description of the content).
- **Not allowed**: **API** or **novel research** descriptions.
- Same characteristics than other journals: ISSN, Crossref DOI.
- Code Review: **direct visibility**  
⇒ collaboration with classic development tools (based on *GitHub*).

### 3.3 JOSS Journal: characteristics

#### Specific form:

- Form: **voluntary short**: short abstract length (author names, list of key references, a link to software repository and a short description of the content).
- **Not allowed**: **API** or **novel research** descriptions.
- Same characteristics than other journals: ISSN, Crossref DOI.
- Code Review: **direct visibility**  
⇒ collaboration with classic development tools (based on *GitHub*).

#### Content requested:

- Software need to be **open source**.
- **Research** application.
- Submitter needs to be **main software contributor**.
- Significant new contribution.
- **Feature-complete** (not partial).

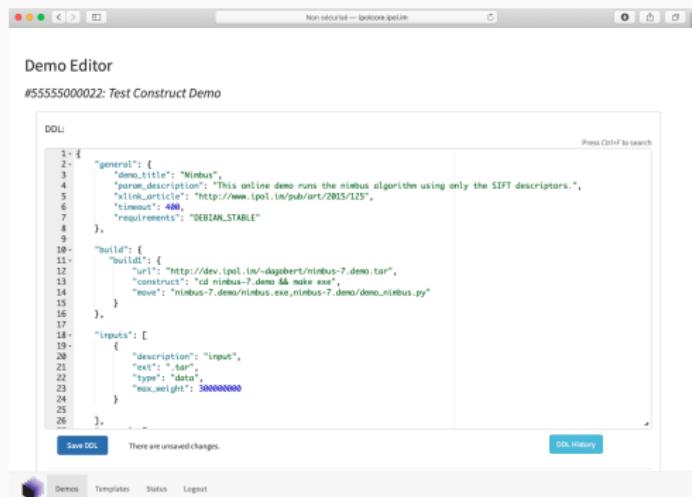
## **4. Practical Session: Hands on IPOL Demos**

---

## 4. Practical session: hands on the IPOL Demonstration System

### Focus on online demonstration

- Construction of demonstration from your own code.
- New Demo Editor panel:
  - to construct your demonstration without hard coding the demo.



The screenshot shows a web-based application window titled "Demo Editor". The URL in the address bar is "Non secured — ipolone.ipol.eu". The main content area displays a JSON configuration file for a demonstration:

```
1- {
2-   "general": {
3-     "demo_title": "Nimbus",
4-     "param_description": "This online demo runs the nimbus algorithm using only the SIFT descriptors.",
5-     "url_ncs": "http://www.ipol.im/pub/art/2013/l25",
6-     "size": 400,
7-     "requirements": "DEBIAN_STABLE"
8-   },
9-   "build": {
10-     "buildit": {
11-       "url": "http://dev.ipol.im/~dagobert/nimbus-7.demo.tar",
12-       "construct": "cd nimbus-7.demo && make exec",
13-       "move": "nimbus-7.demo/nimbus.exe,nimbus-7.demo/demo_nimbus.py"
14-     }
15-   },
16-   "inputs": [
17-     {
18-       "description": "input",
19-       "ext": ".tar",
20-       "type": "data",
21-       "max_weight": 300000000
22-     }
23-   ]
24- }
```

At the bottom of the editor, there are two buttons: "Save DOL" and "DOL History". Below the editor, a navigation bar includes links for "Demos", "Templates", "Status", and "Logout".

## 4. Practical session: hands on the IPOL Demonstration System

### Focus on online demonstration

- Construction of demonstration from your own code.
- New Demo Editor panel:
  - to construct your demonstration without hard coding the demo.
- Interpreted in live from the online editor.

The screenshot shows the IPOL Demo Editor interface. At the top, there's a title bar with the text "Non secured — ipolcore.ipol.in". Below it is a toolbar with standard window controls. The main area is titled "Demo Editor" and contains the identifier "#55555000022: Test Construct Demo". The interface is divided into two main sections: a code editor on the left and a preview pane on the right.

The code editor displays the following DDL (Domain Definition Language) code:

```
1- {
2-   "general": {
3-     "demo_title": "Nimbus"
4-   },
5-   "param_description": "This online demo runs the nimbus algorithm using only the SIFT descriptors.",
6-   "link_article": "http://www.ipol.in/pub/art/2015/125",
7-   "timeout": 400,
8-   "requirements": "DEBIAN_STABLE"
9- },
10- "build": {
11-   "build": {
12-     "url": "http://dev.ipol.in/~dogobert/nimbus-7-demo.tar",
13-     "construct": "cd nimbus-7-demo && make exec",
14-     "move": "nimbus-7-demo/nimbus.exe,nimbus-7-demo/demo_nimbus.py"
15-   }
16- },
17- "inputs": [
18-   {
19-     "description": "input",
20-     "ext": ".tar",
21-     "type": "data",
22-     "max_weight": 30000000
23-   }
24- ],
25- },
26- }
```

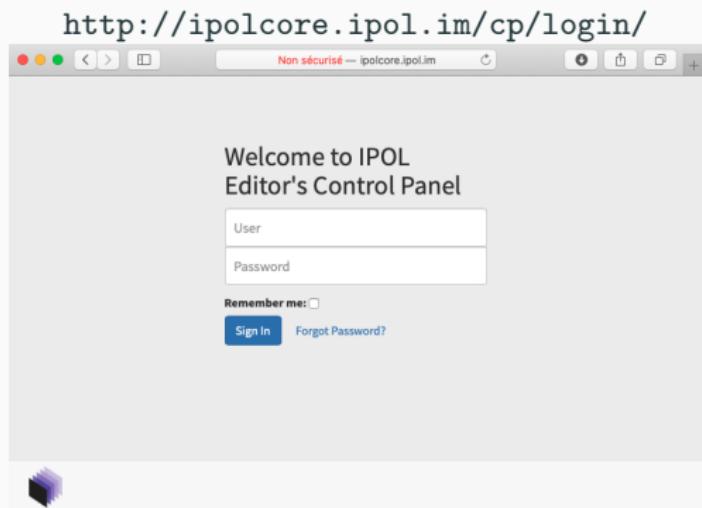
Below the code editor, there are buttons for "Save DDL" and "DB History". A message at the bottom says "There are unsaved changes." The preview pane is currently empty.

At the bottom of the interface, there are navigation links: "Demos", "Templates", "Status", and "Logout".

## 4. Practical session: hands on the IPOL Demonstration System

### What you need to construct your demonstration

1. Request a demo editor account (when submitting a paper to IPOL)  
→ use temporary account to experiment the construction.
2. Image based source code archive (to be available online).
3. Typical command line/parameters.



## 4. Practical session: first step

### Step 1: main demo repository construction

1. Login on : <http://ipolcore.ipol.im/cp/login>

(login: acpr2019 password: 222hhh888k)

2. Add a new Demo.

3. Select an number ID and titles using this convention:

DemoID: 555559000XX Title: ACPRMyTestDemoXX

List of Demos

Search

Add a new Demo

|                    |  |               |                                |
|--------------------|--|---------------|--------------------------------|
| My texture test    |  | Date test     | Last modification: 28 Oct 2018 |
| My Crash detection |  | Date workshop | Last modification: 30 Sep 2018 |
| My Crash demo      |  | Date workshop | Last modification: 12 Oct 2018 |
| My Extension of I  |  | Date workshop | Last modification: 04 May 2018 |

Demosa Templates Status Logout

New Demo data

Demo ID  ACPRMyTestDemoXX

Title  ACPRMyTestDemoXX

Status  Test  Demo

Close

List of Demos

|                    |  |               |                                |
|--------------------|--|---------------|--------------------------------|
| My texture test    |  | Date test     | Last modification: 28 Oct 2018 |
| My Crash detection |  | Date workshop | Last modification: 30 Sep 2018 |
| My Crash demo      |  | Date workshop | Last modification: 12 Oct 2018 |
| My Extension of I  |  | Date workshop | Last modification: 04 May 2018 |

Demosa Templates Status Logout

List of Demos

ACPRMyTestDemoXX

Date test Last modification: 31 Nov 2018

Page 1 of 1

List of Demos

|                    |  |               |                                |
|--------------------|--|---------------|--------------------------------|
| My texture test    |  | Date test     | Last modification: 28 Oct 2018 |
| My Crash detection |  | Date workshop | Last modification: 30 Sep 2018 |
| My Crash demo      |  | Date workshop | Last modification: 12 Oct 2018 |
| My Extension of I  |  | Date workshop | Last modification: 04 May 2018 |
| ACPRMyTestDemoXX   |  | Date test     | Last modification: 31 Nov 2018 |

Demosa Templates Status Logout

## 4. Practical session: second step

### Step 2: Edit configuration file

1. Start from the base example available from the tutorial repository:

<https://github.com/kerautret/ACPR19-DGPRTutorial> file: `PracticalSession/exDemoDDL.txt`



The screenshot shows a GitHub repository page for 'ACPR19-DGPRTutorial'. The 'PracticalSession' directory contains the file 'exDemoDDL.txt'. The file content is displayed as follows:

```
34 Lines | 2M Lines | 3.08 KB
1: {
2:   "parameters": [
3:     {
4:       "name": "NORM", "value": "NORM 2D field or 3D, 000",
5:       "type": "dropdown", "label": "Normale type", "value": "NORM 2D field or 3D, 000",
6:       "minValue": 100,
7:       "maxValue": 200
8:     },
9:     {
10:      "label": "URL"
11:    },
12:    {
13:      "label": "Server"
14:    },
15:    {
16:      "label": "URL", "value": "https://github.com/kerautret/ACPR19-DGPRTutorial/Lib/realData/geojson/shapefile.shp",
17:      "type": "dropdown", "label": "Shapefile URL", "value": "https://github.com/kerautret/ACPR19-DGPRTutorial/Lib/realData/geojson/shapefile.shp"
18:    }
19:  ],
20:  "inputs": [
21:  ]
22: }
```

## 4. Practical session: second step

### Step 2: Edit configuration file

1. Start from the base example available from the tutorial repository:

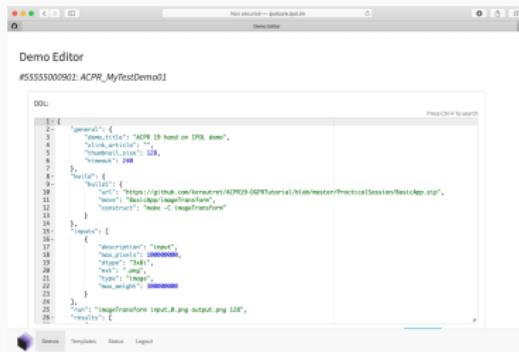
<https://github.com/kerautret/ACPR19-DGPRTutorial> file: `PracticalSession/exDemoDDL.txt`

2. Copy the content in the demo editor panel.



A screenshot of a GitHub browser interface. The URL is `https://github.com/kerautret/ACPR19-DGPRTutorial/PracticalSession/exDemoDDL.txt`. The page shows the contents of the `exDemoDDL.txt` file:

```
1  [general]
2  [name] = "ACPR 19 hand on IPOL demo"
3  [alias_activities] =
4  [description] = "IPOL demo"
5  [version] = 1.0
6  [language] = "IPOL"
7
8  [build] {
9    [url] = "https://github.com/kerautret/ACPR19-DGPRTutorial/blob/master/PracticalSession/TestApp.ipol"
10   [name] = "BuildApp"
11   [script] = "None"
12   [contract] = "None"
13 }
14
15 [inputs] {
16   [url] = "https://ipol.inria.fr/api/v1/demos/testapp/input"
17   [name] = "Input"
18   [type] = "Image"
19   [size] = "1000x1000"
20   [format] = "png"
21   [weight] = 1
22 }
23
24 [outputs] {
25   [url] = "https://ipol.inria.fr/api/v1/demos/testapp/output"
26   [name] = "Output"
27   [type] = "Image"
28   [size] = "1000x1000"
29   [format] = "png"
30   [weight] = 1
31 }
```



A screenshot of a Demo Editor window titled "65555000902: ACPR\_MyTestDemo01". The window displays the same configuration file content as the GitHub page, with syntax highlighting for code blocks.

```
1  [general]
2  [name] = "ACPR 19 hand on IPOL demo"
3  [alias_activities] =
4  [description] = "IPOL demo"
5  [version] = 1.0
6  [language] = "IPOL"
7
8  [build] {
9    [url] = "https://github.com/kerautret/ACPR19-DGPRTutorial/blob/master/PracticalSession/TestApp.ipol"
10   [name] = "BuildApp"
11   [script] = "None"
12   [contract] = "None"
13 }
14
15 [inputs] {
16   [url] = "https://ipol.inria.fr/api/v1/demos/testapp/input"
17   [name] = "Input"
18   [type] = "Image"
19   [size] = "1000x1000"
20   [format] = "png"
21   [weight] = 1
22 }
23
24 [outputs] {
25   [url] = "https://ipol.inria.fr/api/v1/demos/testapp/output"
26   [name] = "Output"
27   [type] = "Image"
28   [size] = "1000x1000"
29   [format] = "png"
30   [weight] = 1
31 }
```

## 4. Practical session: second step

### Step 2: Edit configuration file

- Start from the base example available from the tutorial repository:

<https://github.com/kerautret/ACPR19-DGPRTutorial> file: **PracticalSession/exDemoDDL.txt**

- Copy the content in the demo editor panel.

- Ajust parameters and your source code link: (or this toy example)

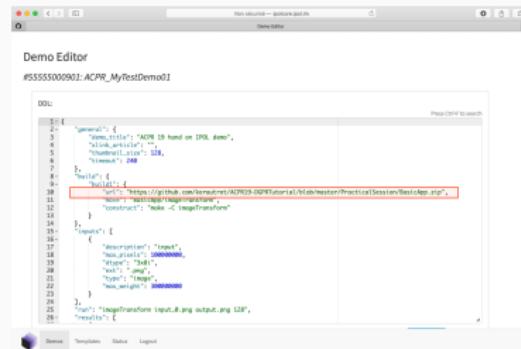
<https://github.com/kerautret/ACPR19-DGPRTutorial> file: **PracticalSession/BasicApp.zip**



```

 1: {
 2:   "general": {
 3:     "name": "exDemo",
 4:     "version": "0.1.0",
 5:     "description": "A simple DDC demo",
 6:     "author": "kerautret",
 7:     "email": "kerautret@inria.fr",
 8:     "url": "https://github.com/kerautret/ACPR19-DGPRTutorial"
 9:   },
10:   "source": {
11:     "code": "BasicApp.zip",
12:     "url": "https://github.com/kerautret/ACPR19-DGPRTutorial/blob/master/PracticalSession/BasicApp.zip"
13:   },
14:   "inputs": [
15:     {
16:       "description": "Input mesh",
17:       "mesh": "input.mesh",
18:       "format": "obj",
19:       "axis": "uv",
20:       "weight": "0.5"
21:     }
22:   ],
23:   "outputs": [
24:     {
25:       "url": "https://github.com/kerautret/ACPR19-DGPRTutorial/blob/master/PracticalSession/BasicApp.out"
26:     }
27:   ]
28: }

```



## 4. Practical session: last step

### Step 3: Select Blobs and start!

1. Need to select the default images to test your demo: use the "Blobs".
2. Customize (or not default blob).

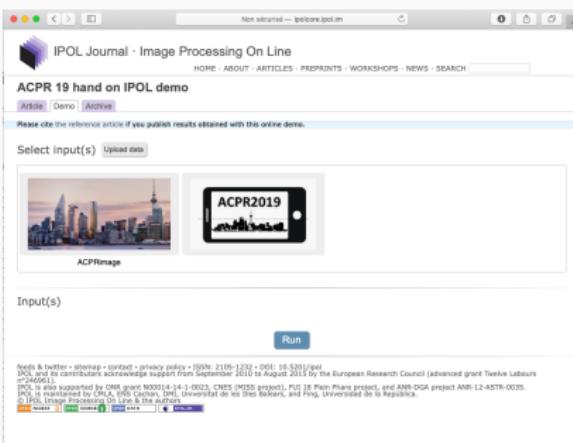


## 4. Practical session: last step

### Step 3: Select Blobs and start!

1. Need to select the default images to test your demo: use the "Blobs".
2. Customize (or not default blob).
3. Check the resulting online demo! (using following url)

<http://ipolcore.ipol.im/demo/clientApp/demo.html?id=555559000XX>



## 4. Practical session: customize parameters

### Exercice

1. Add a new parameter on the demonstration.

→ instead using the default parameter, add a 'params' section in the DDL.  
→ use \$ to transmit the parameter.

The screenshot shows a web browser window with the following details:

- Title Bar:** Non sécurisé — ipolcore.ipol.im
- Page Header:** IPOL Journal - Image Processing On Line
- Navigation:** HOME · ABOUT · ARTICLES · PREPRINTS · WORKSHOPS · NEWS · SEARCH
- Section:** ACPR 19 hand on IPOL demo
- Buttons:** Article | Demo | Archive
- Text:** Please cite the reference article if you publish results obtained with this online demo.
- Input:** Select input(s)
- Image:** Two images are displayed: "ACPRImage" (a city skyline at sunset) and "ACPR2019" (a smartphone displaying a city skyline silhouette).
- Parameters:** Parameters   
Set the threshold value  Max: 255
- Buttons:** Run
- Page Footer:** terms & conditions · sitemap · contact · privacy policy · ISSN: 2100-1232 · DOI: 10.5301/ipol  
IPOL and its contributors acknowledge support from September 2010 to August 2015 by the European Research Council (advanced grant Twelve Labours n°246961).  
IPOL is also supported by CNR grant N00014-14-1-0023, CNES (MISS project), FUI 18 Iren Phare project, and ANR-DGA project ANR-12-ASTR-0035.  
IPOL is maintained by CNRS, EPFL, Inria, DMF, Universitat de les Illes Balears, and Ping, Universidad de la Republica.  
© IPOL Journal - Image Processing On Line & the authors.
- Bottom Buttons:** OPEN SOURCE  OPEN SOURCE  OPEN DATA  IPOL DEMO

## 4. Practical session: customize parameters

### Exercice

1. Add a new parameter on the demonstration.

→ instead using the default parameter, add a 'params' section in the DDL.  
→ use \$ to transmit the parameter.

```
22 < "params": [
23 < {
24 <   "id": "parameterID",
25 <   "label": "Label description...",
26 <   "type": "range",
27 <   "values": {
28 <     "default": 50,
29 <     "max": 100,
30 <     "min": 0,
31 <     "step": 1
32 <   }
33 > },
```

## References

Thank you for your attention!

-  [Baker *et al.* 16] M 1,500 scientists lift the lid on reproducibility.  
**Nature** **533** (2016) 452.  
533 (2016) <https://doi.org/10.1038/533452a>
-  [Smith *et al.* 17] M Smith, A., E Niemeyer, K., Katz, D.S., A Barba, L., Githinji, G., Gymrek, M., Huff, K., Madan, C., Cabunoc Mayes, A., Moerman, K., Prins, P., Ram, K., Rokem, A., K Teal, T., Valls Guimera, R., T Vanderplas, J.:  
**Journal of open source software (joss): Design and first-year review.**  
4 (2017)
-  [Hettick *et al.* 14] Hettick, S., Antonioletti, M., Carr, L., Chue Hong, N., Crouch, S., De Roure, D., Emsley, I., Goble, C., Hay, A., Inupakutika, D., Jackson, M., Nenadic, A., Parkinson, T., Parsons, M.I., Pawlik, A., Peru, G., Proeme, A., Robinson, J., Sufi, S.:  
**UK Research Software Survey 2014.**  
(2014)
-  [Buckheit & Donoho 95] Buckheit, J.B., Donoho, D.L.  
**In: WaveLab and Reproducible Research. Springer New York, New York, NY (1995) 55–81**

## References (2)

-  [Hinsen 2015] Hinsen, K.:  
**Writing Software Specifications.**  
Computing in Science & Engineering (2015)
-  [Topalidou *et al.* 15] Topalidou, M., Leblois, A., Boraus, T., Rougier, N.:  
**A long journey into reproducible computational neuroscience.**  
Frontiers in Computational Neuroscience (2015)
-  [Hinsen 14] Hinsen, K.:  
**Computational science shifting the focus tools to models.**  
F1000Research (2014)
-  [Rougier 17] Rougier, P, N.:  
**Weighted Voronoi Stippling.**  
ReScience 3 (2017) 1–8

## References (3)

-  [Secord 02] Secord, A.:  
**Weighted Voronoi stippling.**  
In: Proceedings of the second international symposium on Non-photorealistic animation and rendering, ACM Press (2002) 37–43
-  [Lamiroy & Lopresti 16] Lamiroy, B., Lopresti, D.P.:  
**The DAE platform: A framework for reproducible research in document image analysis.**  
In: RRPR@ICPR. Volume 10214 of Lecture Notes in Computer Science. (2016) 17–29
-  [Arevalo et al. 16] Arévalo, M., Escobar, C., Monasse, P., Monzón, N., Colom, M.:  
**The IPOL demo system: A scalable architecture of microservices for reproducible research.**  
In: RRPR@ICPR. Volume 10214 of Lecture Notes in Computer Science. (2016) 3–16