ACPR 19 Tutorial

Digital Geometry in Pattern Recognition: Extracting Geometric Features with DGtal and Applications – Part II –

Bertrand Kerautret[1] Jacques-Olivier Lachaud [2]

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

[1]LIRIS (UMR CNRS 5205) Université de Lyon 2, F-69676, France

[2]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

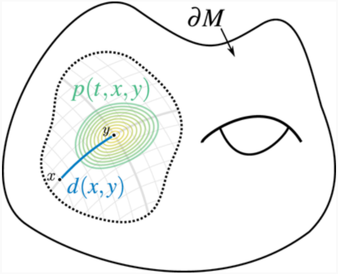
https://kerautret.github.io/ACPR19-DGPRTutorial

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
- analagous to $d-1$-manifold in $\mathbb{R}^d$: locally $d-1$ directions

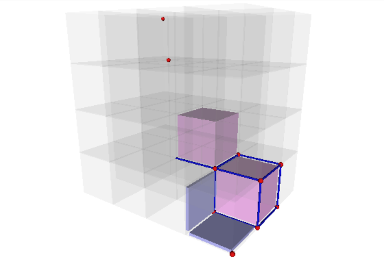


Digital Surfaces

Sound definition for digital surfaces

Digital surfaces require cell/interpixel topology

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

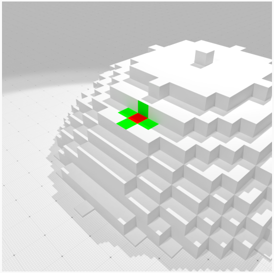


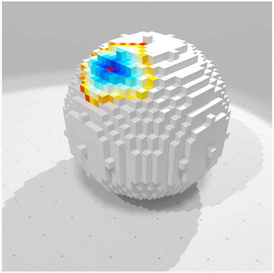
Digital Surfaces

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



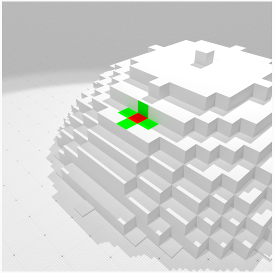


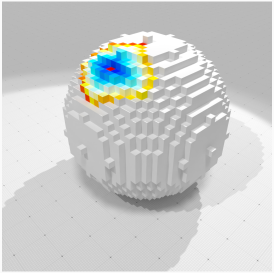
Digital Surfaces

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



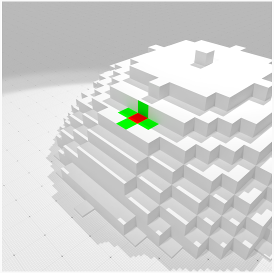


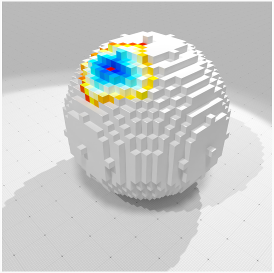
Digital Surfaces

Sound definition for digital surfaces

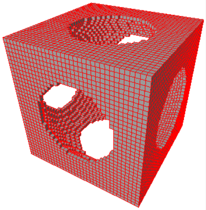
Digital surfaces require cell/interpixel topology

- 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



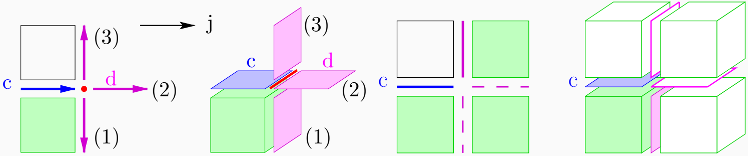
- For now, a surface is a set of surfels

Questions ?

Can we define local neighborhood relations so that

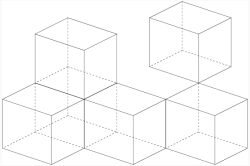
- a whole connected surface can be extracted by their tracking,
- Jordan separation property is satisfied

Bel adjacency in a picture i

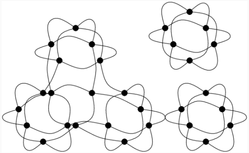


- shape / binary picture I: finite subset X of $\mathbb{Z}^n$
- boundary element or *bel* in I = surfel between X and X^c
- For each direction j ($n-1$ directions for each bel)
  - interior bel-adjacency from c along j = first follower of c which is a bel
  - exterior bel-adjacency from c along j = last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

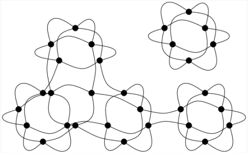
Bel adjacency graph i



bels



interior graph



exterior graph

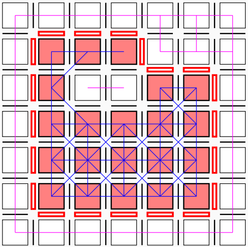
- For each direction, choose interior/exterior $\Rightarrow 2^{\frac{n(n-1)}{2}}$ bel-adjacencies

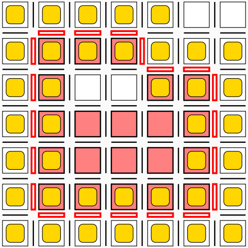
Theorem (3D [Herman,Webster83])

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





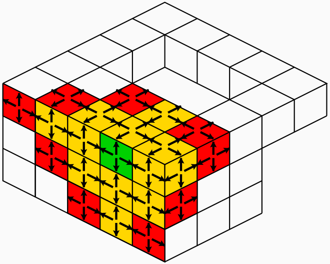
Theorem (nD, $n \geq 2$, [Udupa94])

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.

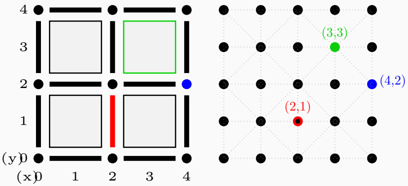
Tracking digital boundaries i



- boundary in parallepiped 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

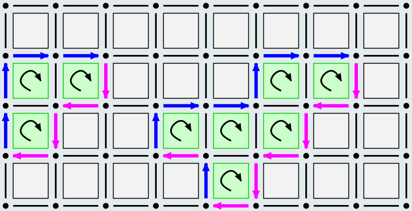


Classical trick for representing cell space

- 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

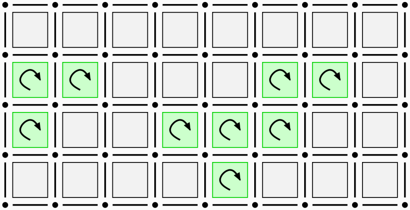
Cubical chain complex ii

Construction of a chain complex



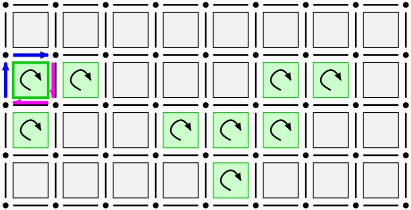
- oriented k-cells form k-dimensional bases
- k-chains are formal sums of k-cells (coefficient $\mathbb{Z}$)
  $[\sum_i +o_i^n]$ is a digital shape
  $\sum[+s_j^{n-1}] + \sum[-s_{j'}^{n-1}]$ is a digital surface
- boundary operator Δ, with $\Delta\Delta = 0$, based on cell parities

Application to digital boundaries



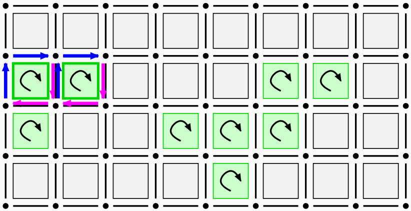
- digital shape is a subset X of $\mathbb{Z}^n$ (odd parities)
- its boundary $= n - 1$-chain $\Delta \sum_{x \in X} +x$
- it is a cycle since $\Delta\Delta = 0$

Application to digital boundaries



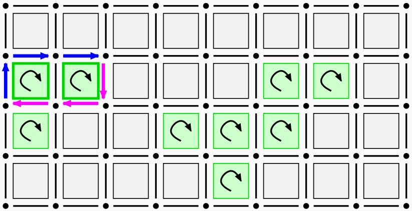
- digital shape is a subset X of $\mathbb{Z}^n$ (odd parities)
- its boundary = $n - 1$-chain $\Delta \sum_{x \in X} +x$
- it is a cycle since $\Delta\Delta = 0$

Application to digital boundaries



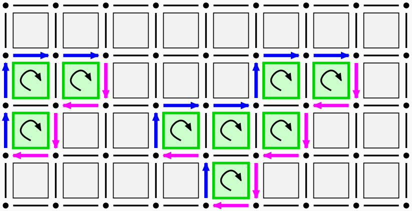
- digital shape is a subset X of $\mathbb{Z}^n$ (odd parities)
- its boundary = $n-1$-chain $\Delta \sum_{x \in X} +x$
- it is a cycle since $\Delta\Delta = 0$

Application to digital boundaries



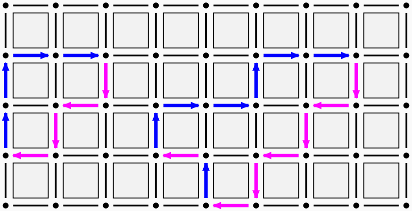
- digital shape is a subset X of $\mathbb{Z}^n$ (odd parities)
- its boundary $= n - 1$-chain $\Delta \sum_{x \in X} +x$
- it is a cycle since $\Delta\Delta = 0$

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

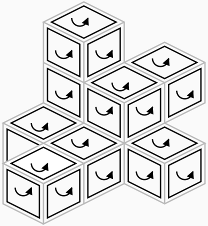


- digital shape is a subset X of $\mathbb{Z}^n$ (odd parities)
- its boundary $= n - 1$-chain $\Delta \sum_{x \in X} +x$
- it is a cycle since $\Delta\Delta = 0$

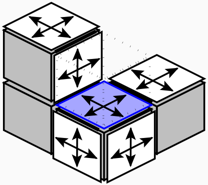
Theorem

The boundary of a volume is a closed digital surface

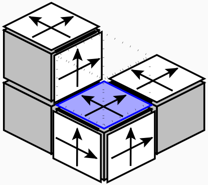
Oriented boundary tracking i



ΔX

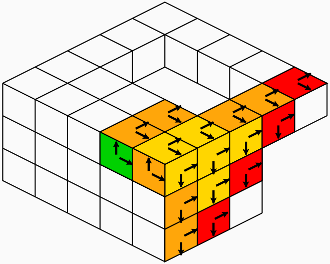


neighbors

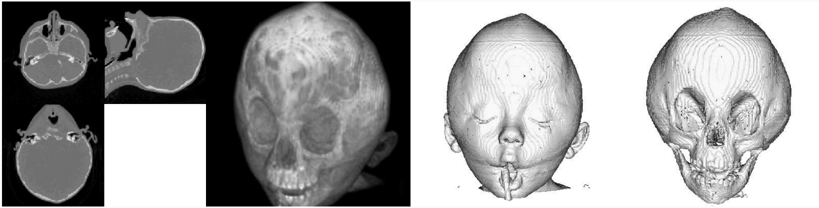


“positive” neighbors

- 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



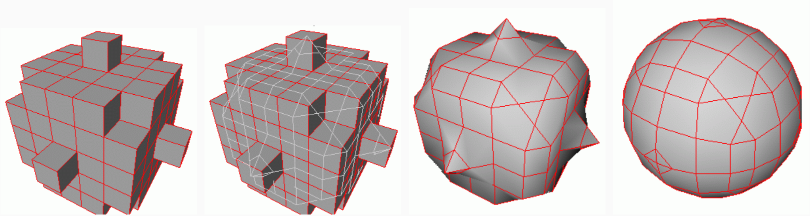
Definition (Isosurface)

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

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

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

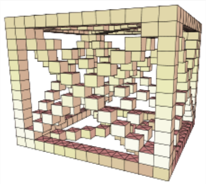
Duality isosurfaces / digital surface



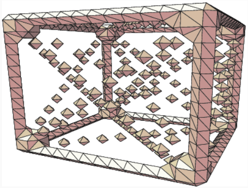
$$X = \{\vec{x} \in \mathbb{Z}^3, I(\vec{x}) \geq s\}$$

Theorem [L. Montanvert 2000]

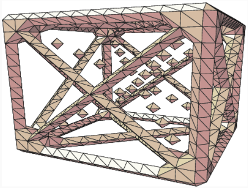
bel-adjacency complex defines a $n-1$-pseudomanifold without boundary



shape

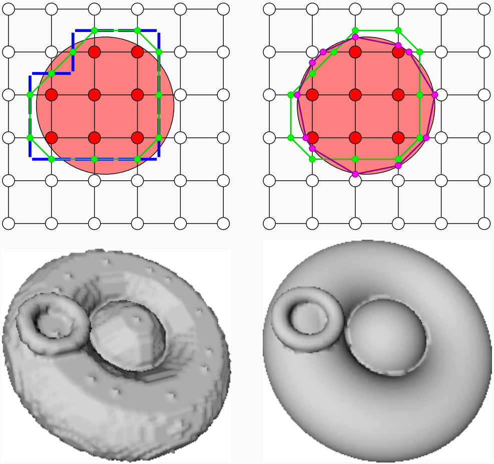


interior



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

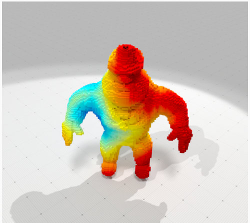
Generic representation of digital surfaces

- 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

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



Digital surfaces in DGtal

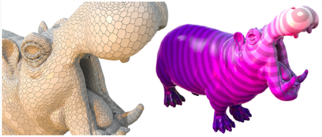
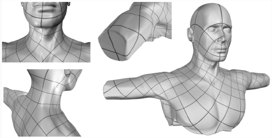
Usual usages: use Shortcuts module

- image $\rightarrow$ digital surface / indexed digital surface
- extraction of connected components / main component
- polynomial shape $\rightarrow$ digital surface / indexed surface
- digital surface $\leftrightarrow$ indexed digital surface
- (indexed) digital surface $\rightarrow$ dual triangulated surface (MC)
- (indexed) digital surface $\rightarrow$ primal polygonal surface
- (indexed) digital surface $\rightarrow$ 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

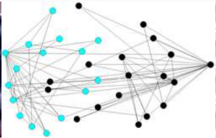
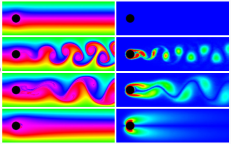
Computer graphics, geometry processing, shape optimization

(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, ...]

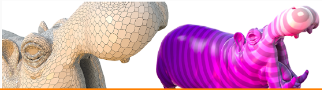
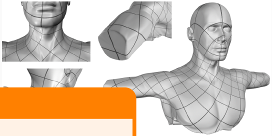
Graph and network analysis, image processing, fluid simul.

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

II.2 Discrete (exterior) Calculus

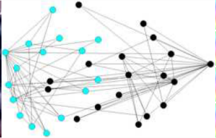
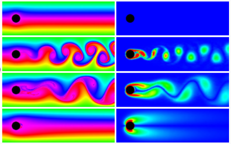
Computer graphics, geometry processing, shape optimization

Discrete exterior calculus (DEC)

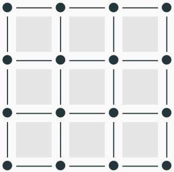
- no discretization, discrete by nature
- keep algebraic properties of calculus, exact Stokes' theorem
- reduces to matrix/vectors
- works without embedding, just metric
- "any" cell complex, arbitrary dimension

Graph and network analysis, image processing, fluid simul.

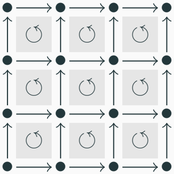
(Images: Bugeau et al. 2014, couprie et al. 2014, Elcott et al. 2006)

Cell complex, chains, boundary, forms



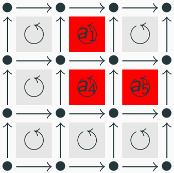
- *cell complex* K: vertices, edges, faces (pixels/surfels)

Cell complex, chains, boundary, forms



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

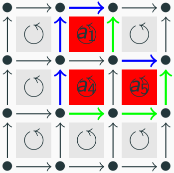
Cell complex, chains, boundary, forms



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

- *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



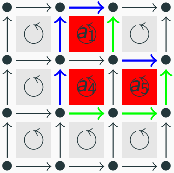
$$\sigma := a_1 + a_4 + a_5 \in C_2(K)$$

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

$$\eta = \partial_2 \sigma$$

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

Cell complex, chains, boundary, forms



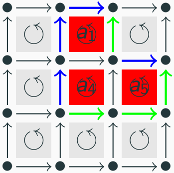
$\sigma := a_1 + a_4 + a_5 \in C_2(K)$

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

$\eta = \partial_2 \sigma$

- *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 $C^k(K) := \mathrm{Hom}(C_k(K), \mathbb{R})$
  - 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 $C^k(K) := \mathrm{Hom}(C_k(K), \mathbb{R})$
  - 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 α with k-chain σ

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

Exterior derivative, Stokes theorem

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

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

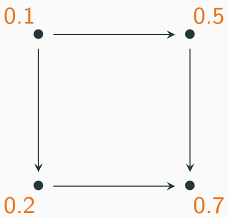
thus incidence relations define derivative by duality

Exterior derivative, Stokes theorem

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

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

thus incidence relations define derivative by duality



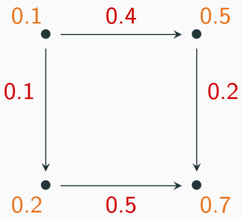
- Function or discrete 0-form : $\alpha = (0.2, 0.7, 0.1, 0.5)$

Exterior derivative, Stokes theorem

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

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

thus incidence relations define derivative by duality



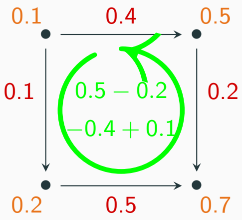
- Function or discrete 0-form : $\alpha = (0.2, 0.7, 0.1, 0.5)$
- 1-form $\mathbf{d}_0(\alpha) = \beta = (0.5, 0.1, 0.2, 0.4)$

Exterior derivative, Stokes theorem

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

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

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: $\mathbf{d}_k : C^k(K) \to C^{k+1}(K)$

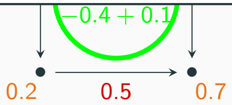
$$(\mathbf{d}_k \alpha^k)(\sigma_{k+1}) := \alpha^k(\partial_{k+1}\sigma_{k+1})$$

thu

(discrete) Stokes theorem is trivial by definition

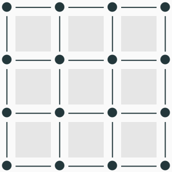
$$\int_\sigma \mathbf{d}\alpha = \int_{\partial\sigma} \alpha$$

for σ any k-chain and α any $k-1$-form

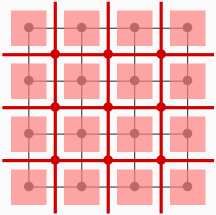


- Function or discrete 0-form : $\alpha = (0.2, 0.7, 0.1, 0.5)$
- 1-form $\mathbf{d}_0(\alpha) = \beta = (0.5, 0.1, 0.2, 0.4)$
- 2-form $\mathbf{d}_1(\beta) = 0$, since $\mathbf{d}_1\mathbf{d}_0 = 0$

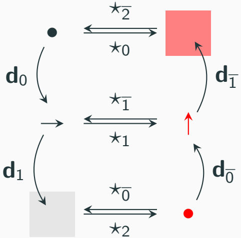
Dual cell complex, Hodge star, calculus



complex K



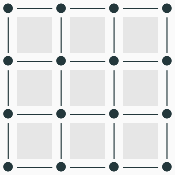
dual complex $\overline{K}$



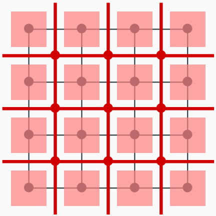
primal dual

- Hodge duality created with dual/orthogonal structure

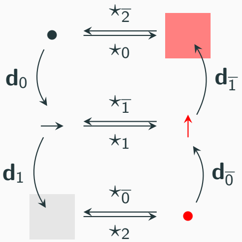
Dual cell complex, Hodge star, calculus



complex K



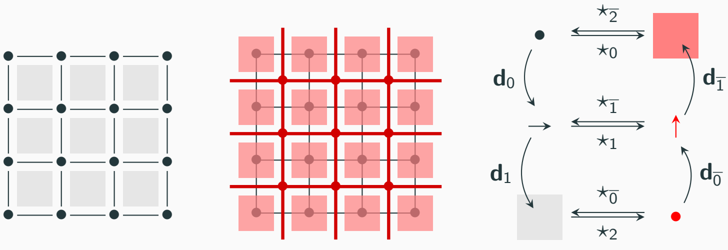
dual complex $\overline{K}$



primal dual

- Hodge duality created with dual/orthogonal structure
- anti-derivatives $\mathbf{d}_{\overline{k}}$ are derivatives in dual complex
    - in matrix form $\mathbf{d}_{\overline{k}}^{\mathsf{T}} := \mathbf{d}_{n-1-k}$

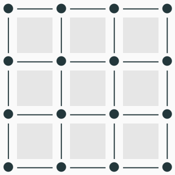
Dual cell complex, Hodge star, calculus



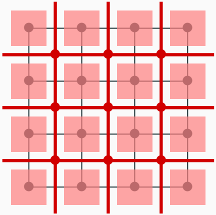
complex K — dual complex $\overline{K}$ — primal dual

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

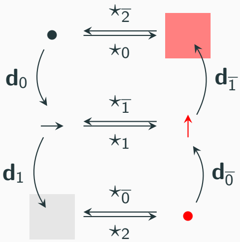
Dual cell complex, Hodge star, calculus



complex K



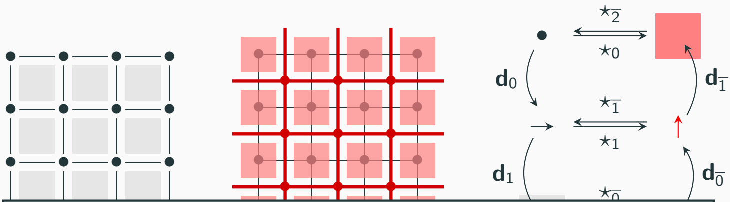
dual complex $\overline{K}$



primal dual

- Hodge duality created with dual/orthogonal structure
- anti-derivatives $\mathbf{d}_{\overline{k}}$ are derivatives in dual complex
  - in matrix form $\mathbf{d}_{\overline{k}}^{\mathsf{T}} := \mathbf{d}_{n-1-k}$
- 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 $\mathrm{d}A$
- wedge products satisfy algebraic properties (Leibniz rules ...)
  - $\alpha \wedge \beta := \mathrm{diag}(\alpha)\beta$, for $\alpha \in C^k(K), \beta \in C^{2-k}(\overline{K})$,
  - $f \wedge \gamma := \mathrm{diag}(\mathbf{M}_{01} f)\gamma$, 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 $\delta_1 := -\star_{\bar{2}}\mathbf{d}_{\bar{1}}\star_1$, $\delta_2 := -\star_{\bar{1}}\mathbf{d}_{\bar{0}}\star_2$,
- Laplacian $\Delta := \delta_1 \mathbf{d}_0$
- Edge Laplacian $\Delta_1 := \mathbf{d}_0\delta_1 + \delta_2\mathbf{d}_1$,
- musical ops : Vector field $\xrightarrow{\flat}$ 1-form $\xrightarrow{\sharp}$ Vector field
- gradient $\nabla f := (\mathbf{d}_0 f)^\sharp$
- divergence $\mathrm{div}\mathbf{V} := \delta_1 \mathbf{V}^\flat$
- L^2 inner-product $(\alpha, \beta)_{\Omega,k} := \int_\Omega \alpha \wedge \star_k \beta$, for α, β k-forms

- $f \wedge \gamma := \mathrm{diag}(\mathbf{M}_{01} f)\gamma$, for $f \in C^0(K), \gamma \in C^1(K)$ ...

Formulate PDE/variational problems with digital calculus

Anisotropic diffusion on image I

- $\frac{\partial u}{\partial t} = \mathrm{div}(f(\|\nabla u\|)\nabla u)$ 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 $\mathbf{A} := \mathbf{d}_0$, $\mathbf{G}_k := \star_k$:

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

  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)

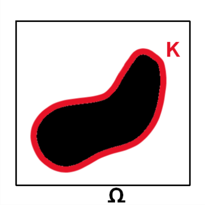
An important example: Mumford-Shah functional

Mumford-Shah functional for image restoration

We minimize

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

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



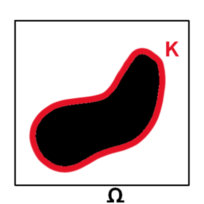
An important example: Mumford-Shah functional

Mumford-Shah functional for image restoration

We minimize

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

- Ω the image domain
- g the input image
- u a piecewise smooth approximation of g
- ✗ K the set of discontinuities
- ✗ $\mathcal{H}^1$ 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 [?]

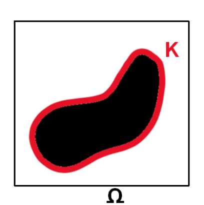
Ambrosio-Tortorelli functional

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

- Ω 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



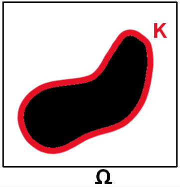
Ambrosio-Tortorelli functional

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

- Ω 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

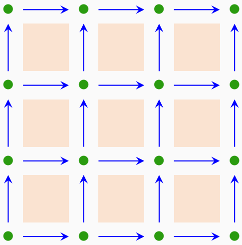
✓ no Hausdorff measure



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

Discrete formulation of AT

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

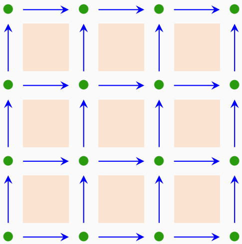


We choose :

- 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

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



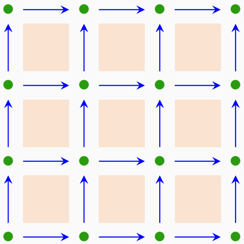
We choose :

- 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

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

Discrete formulation of AT

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



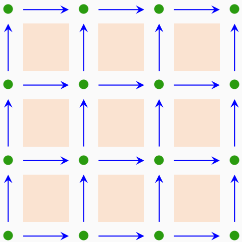
We choose :

- 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

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

Discrete formulation of AT

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



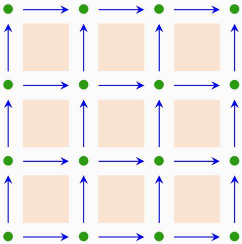
We choose :

- 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

$$\mathrm{AT}_\epsilon^{2,0}(\mathrm{u},\mathrm{v}) = \alpha\,(\mathrm{u}-\mathrm{g},\mathrm{u}-\mathrm{g})_{\Omega,2}$$
$$+ \ \lambda\epsilon\,(\mathbf{d}_0\mathrm{v},\mathbf{d}_0\mathrm{v})_{\Omega,1} + \frac{\lambda}{4\epsilon}\,(1-\mathrm{v},1-\mathrm{v})_{\Omega,0}$$

Discrete formulation of AT

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



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

$$\mathrm{AT}_\epsilon^{2,0}(\mathrm{u},\mathrm{v}) = \alpha\,(\mathrm{u}-\mathrm{g},\mathrm{u}-\mathrm{g})_{\Omega,2} + (?,?)_\Omega$$

$$+ \lambda\epsilon\,(\mathbf{d}_0\mathrm{v},\mathbf{d}_0\mathrm{v})_{\Omega,1} + \frac{\lambda}{4\epsilon}(1-\mathrm{v},1-\mathrm{v})_{\Omega,0}$$

Discrete formulation of AT

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

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

Discrete formulation of AT

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

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

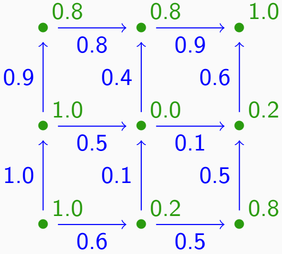
0.8	0.8	1.0
1.0	0.0	0.2
1.0	0.2	0.8

- 0-form v

Discrete formulation of AT

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

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

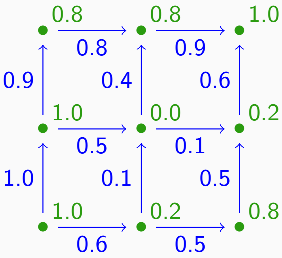


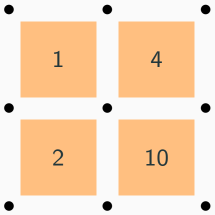
- 0-form v
- 1-form $\mathbf{M}_{01}\mathsf{v}$

Discrete formulation of AT

$$\int_\Omega v^2 |\nabla u|^2 \ \mathrm{d}x \quad = (\mathrm{v}\delta_2 \mathrm{u}, \mathrm{v}\delta_2 \mathrm{u})_{\Omega,1}$$

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





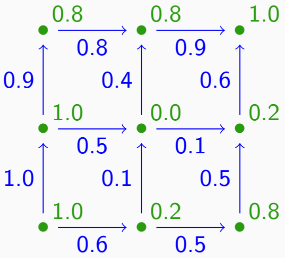
- 2-form u

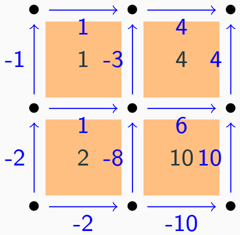
- 0-form v
- 1-form $\mathbf{M}_{01}\mathrm{v}$

Discrete formulation of AT

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

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





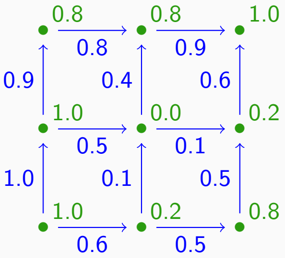
- 0-form v
- 1-form $\mathbf{M}_{01}\mathsf{v}$

- 2-form u
- 1-form $\delta_2\mathsf{u}$

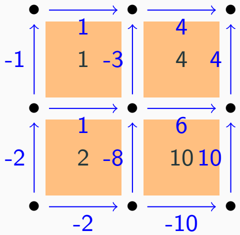
Discrete formulation of AT

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

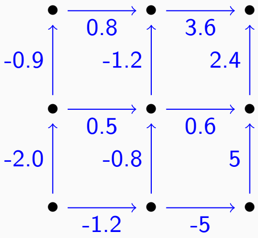
- $\mathsf{v}\delta_2\mathsf{u} = \mathsf{v} \wedge \delta_2 \mathsf{u} = \mathrm{diag}(\mathbf{M}_{01}\mathsf{v})\delta_2\mathsf{u}$



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



- 2-form u
- 1-form $\delta_2\mathsf{u}$



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

Discrete formulation of AT

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

Discrete formulation of AT

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

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

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

Discrete formulation of AT

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

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

$$\mathrm{AT}_\epsilon^{2,0}(\mathsf{u},\mathsf{v}) = \alpha(\mathsf{u}-\mathsf{g})^\intercal\mathbf{G}_2(\mathsf{u}-\mathsf{g}) + \mathsf{u}^\intercal\mathbf{B}'^\intercal\mathrm{diag}(\mathbf{M}_{01}\mathsf{v})^2\mathbf{G}_1\mathbf{B}'\mathsf{u} + \lambda\epsilon\mathsf{v}^\intercal\mathbf{A}^\intercal\mathbf{G}_1\mathbf{A}\mathsf{v} + \frac{\lambda}{4\epsilon}(1-\mathsf{v})^\intercal\mathbf{G}_0(1-\mathsf{v})$$

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

Discrete formulation of AT

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

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

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

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

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

Discrete formulation of AT: vectorial data

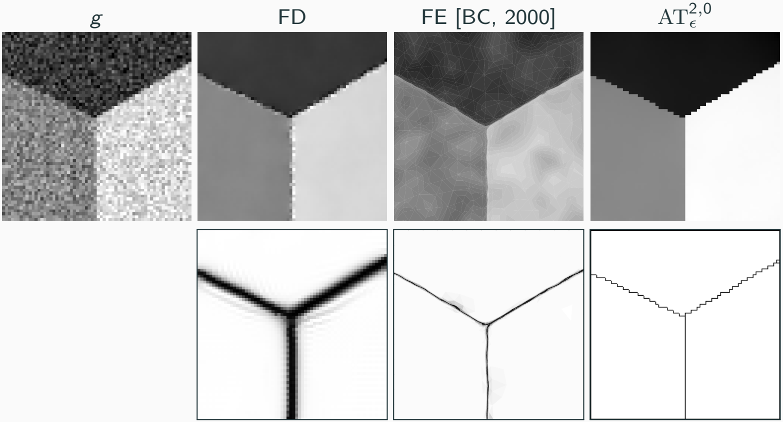
$$
\begin{aligned}
\mathrm{AT}_{\epsilon}^{2,0}(u_1,\ldots,u_n,\mathrm{v}) = \alpha \sum_i (u_i - g_i, u_i - g_i)_{\Omega,2} \\
+ \sum_i (\mathrm{diag}(\mathbf{M}_{01}\mathrm{v})\delta_2 u_i, \mathrm{diag}(\mathbf{M}_{01}\mathrm{v})\delta_2 u_i)_{\Omega,1} \\
+ \lambda\epsilon\,(\mathbf{d}_0\mathrm{v}, \mathbf{d}_0\mathrm{v})_{\Omega,1} + \frac{\lambda}{4\epsilon}(1-\mathrm{v}, 1-\mathrm{v})_{\Omega,0}
\end{aligned}
$$

- We solve alternatively for the u_i and v the sparse linear systems:

$$
\left\{
\begin{array}{rcl}
\forall i \in \{1,\ldots,n\}, \left[\alpha \mathbf{G}_2 - \mathbf{B}'^{\mathsf{T}}\mathrm{diag}\left(\mathbf{M}_{01}\mathrm{v}\right)^2 \mathbf{G}_1 \mathbf{B}'\right] u_i & = & \alpha \mathbf{G}_2 g_i, \\
\left[\frac{\lambda}{4\varepsilon}\mathbf{G}_0 + \lambda\varepsilon \mathbf{A}^{\mathsf{T}}\mathbf{G}_1\mathbf{A} + \mathbf{M}_{01}{}^{\mathsf{T}}(\sum_i \mathrm{diag}\left(\mathbf{B}' u_i\right)^2)\mathbf{G}_1\mathbf{M}_{01}\right]\mathrm{v} & = & \frac{\lambda}{4\varepsilon}\mathbf{G}_0 1.
\end{array}
\right.
$$

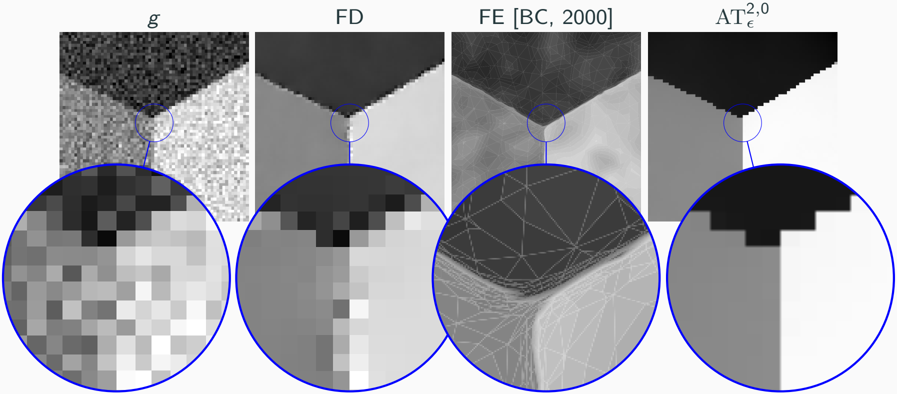
- Our algorithm progressively decreases ϵ to get a better chance of capturing the optimum
  - ϵ follows typically sequence $2, 1, 0.5, 0.25$ (for $h = 1$ sampling)
  - results on u and v are starting point for next ϵ

Image restoration on toy examples

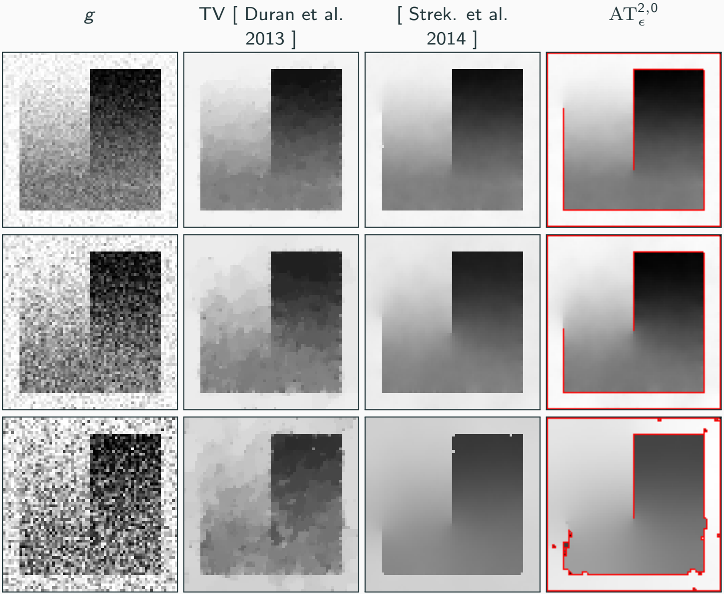


- systems are solved using Cholesky decomposition (Eigen)
- ϵ takes the successive values $2, 1, 0.5, 0.25$, for sampling step $h = 1$.

Image restoration on toy examples



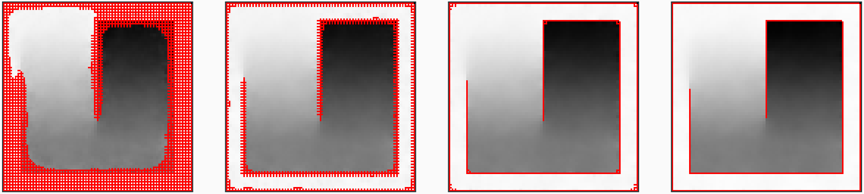
- systems are solved using Cholesky decomposition (Eigen)
- ϵ takes the successive values $2, 1, 0.5, 0.25$, for sampling step $h = 1$.


g
TV [Duran et al. 2013]
[Strek. et al. 2014]
$\mathrm{AT}_{\epsilon}^{2,0}$

Influence of parameter ϵ

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

- Γ-convergence parameter
- Controls the thickness of the contours
    - large ϵ convexifies AT and helps to detect the discontinuities;
    - as ϵ goes to 0, the discontinuities become thinner and thinner.



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

Image restoration / denoising

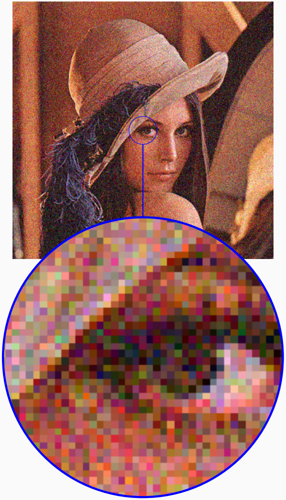
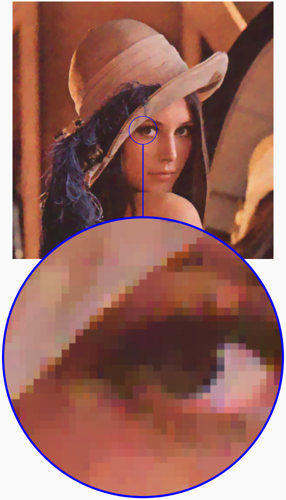
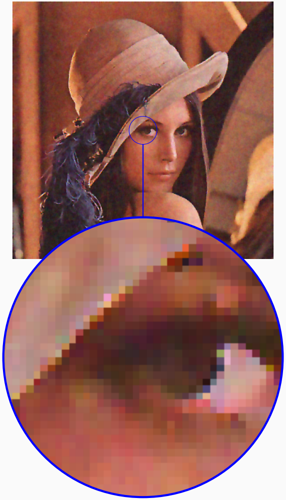
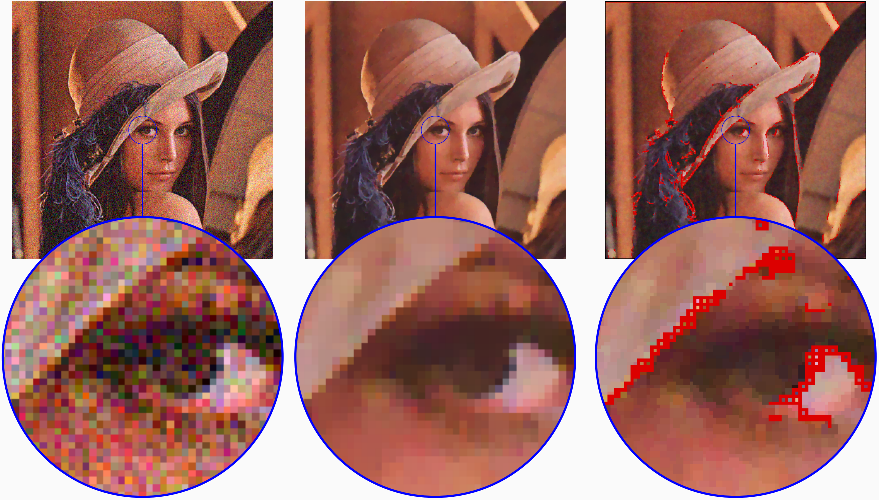
g (PSNR = 20.23 dB)	TV (PSNR = 29.36 dB)	$\mathrm{AT}_\epsilon^{2,0}$ (PSNR = 29.03 dB)
		

Image restoration / denoising

g	TV	$\mathrm{AT}_\epsilon^{2,0}$
(PSNR = 20.23 dB)	(PSNR = 29.36 dB)	(PSNR = 29.03 dB)

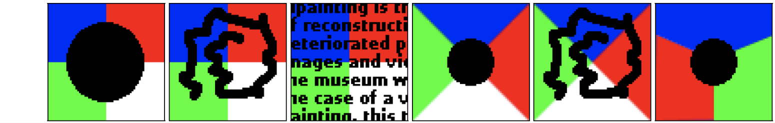


Scale-space given by α and λ and image segmentation

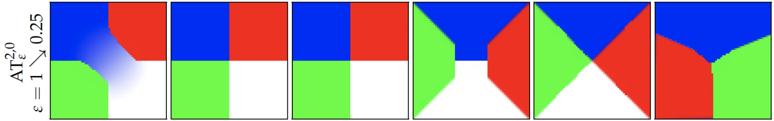
for decreasing λ

Image inpainting (on toy example)

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



$\mathrm{AT}^{2,0}_{\epsilon}$ with ϵ from **1** to 0.25



$\mathrm{AT}^{2,0}_{\epsilon}$ with ϵ from **4** to 0.25

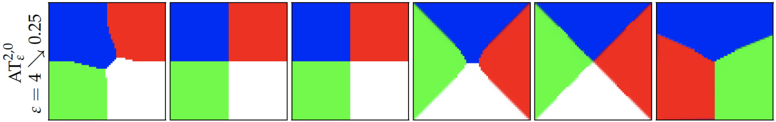
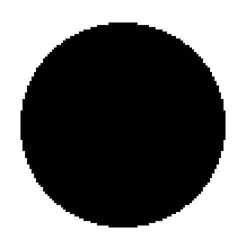
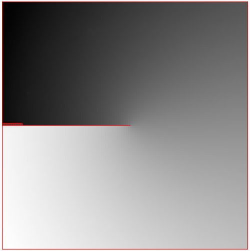


Image inpainting (on classical crack-tip example)

g



mask M

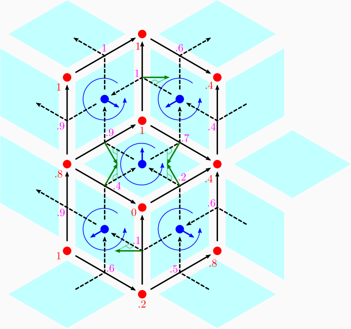


$\mathrm{AT}_{\epsilon}^{2,0}$, $\alpha = 1$, $\lambda = 0.0024$

- Decreasing sequence of λ (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 ϵ is big enough (ϵ from 4 to 0.25 here, for image of size 110×110).

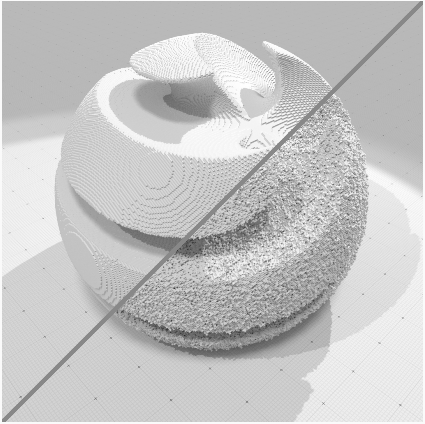
Feature delineation on digital surfaces

digital surface = boundary of set of voxels



same discrete calculus

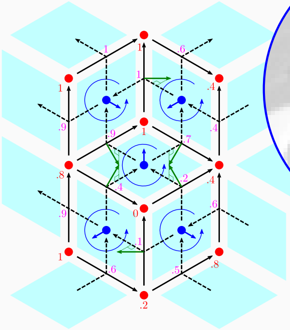
same $\mathrm{AT}_{\epsilon}^{2,0}$



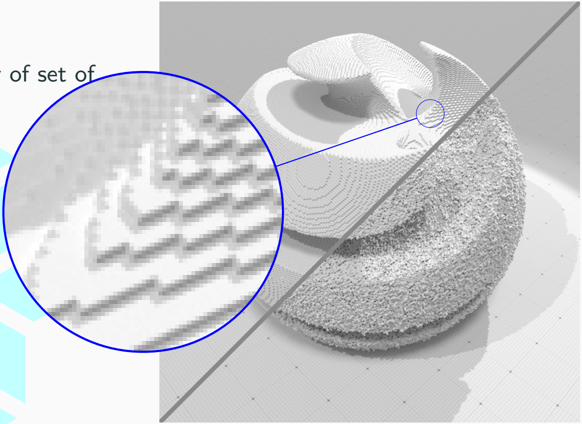
Input: normal vector field g estimated by Integral Invariant digital normal estimator.

Feature delineation on digital surfaces

digital surface = boundary of set of voxels



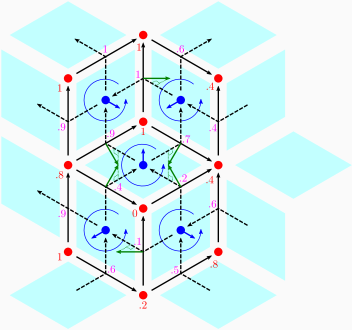
same discrete calculus
same $\mathrm{AT}_{\epsilon}^{2,0}$



Input: normal vector field g estimated by Integral Invariant digital normal estimator.

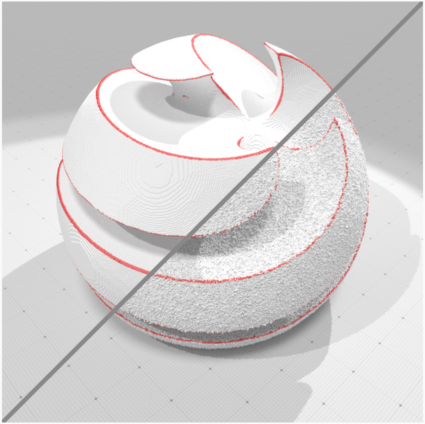
Feature delineation on digital surfaces

digital surface = boundary of set of voxels



same discrete calculus

same $\mathrm{AT}_{\epsilon}^{2,0}$

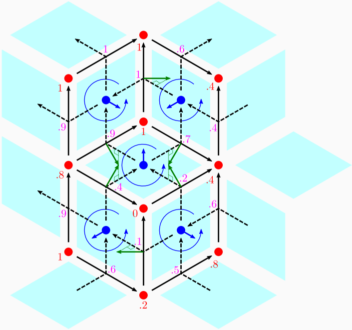


Input: normal vector field g estimated by Integral Invariant digital normal estimator.

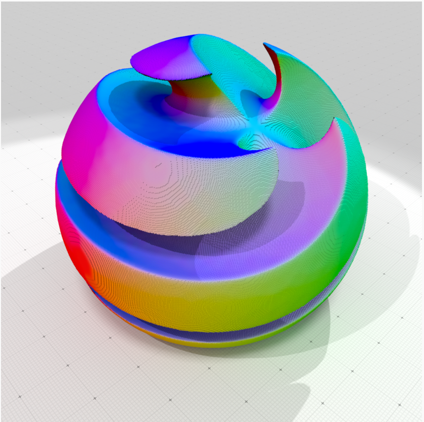
Output: piecewise smooth normals $(u_i)_{i=1,2,3}$ and features v

Feature delineation on digital surfaces

digital surface = boundary of set of voxels



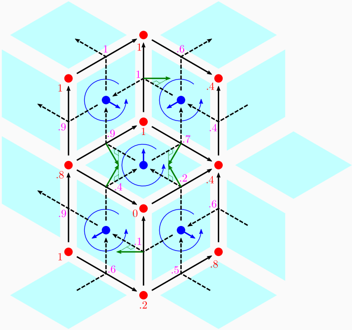
same discrete calculus
same $\mathrm{AT}_{\epsilon}^{2,0}$



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

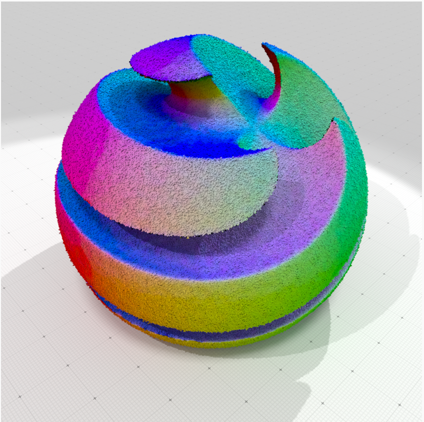
Feature delineation on digital surfaces

digital surface = boundary of set of voxels



same discrete calculus

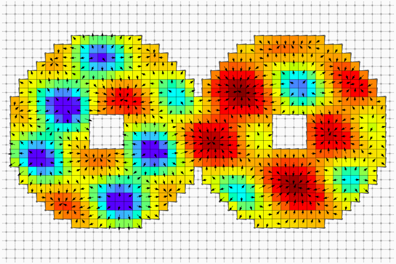
same $\mathrm{AT}_{\epsilon}^{2,0}$



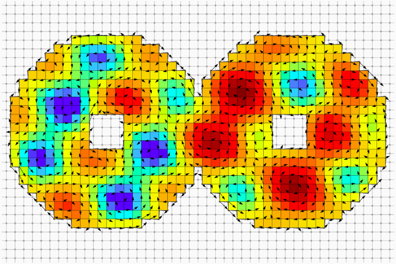
Input: normal vector field g estimated by Integral Invariant digital normal estimator.

Output: piecewise smooth normals $(u_i)_{i=1,2,3}$ and features 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;

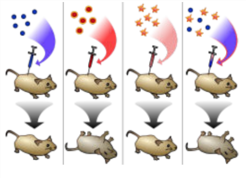
3. Reproducible Research

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;



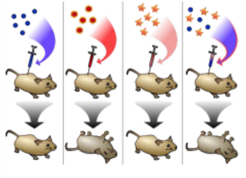
3. Reproducible Research

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... ?*



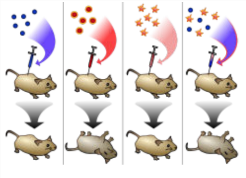
3. Reproducible Research

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... ?*



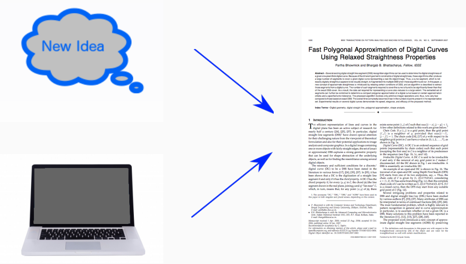
Computer Science:

- 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.



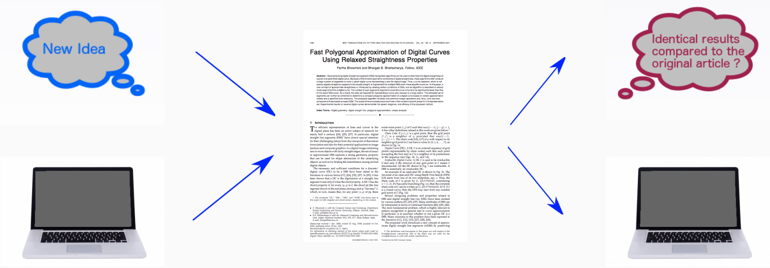
3.1 (short) Introduction to Reproducible Research (1)

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.
  $\rightarrow$ 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.
  $\rightarrow$ 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.
  $\rightarrow$ 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.
  $\rightarrow$ 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:

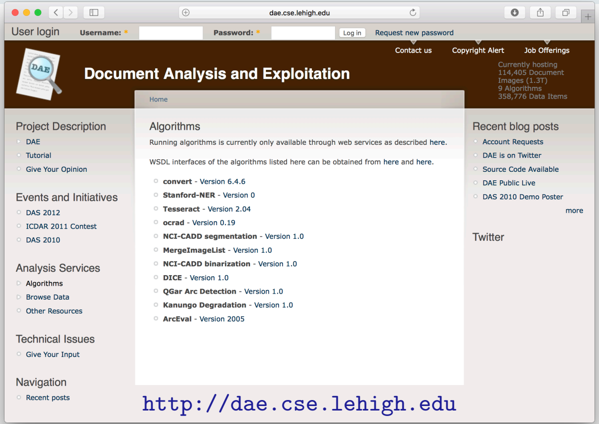
- Stated in 2014 as part *Runway Startup Postdoc Program* at the Jacobs Technion Cornell Institute.
- Sponsored by IEEE.
- Defined by themselves as a *computational reproducibility platform*.
- Not a journal itself. It only runs code, but not publishes articles.
- Assigns a DOI to each source code.
- 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 **D**ocument **A**nalysis and **E**xploitation.
- Allows to run document analysis algorithms and apply comparisons.

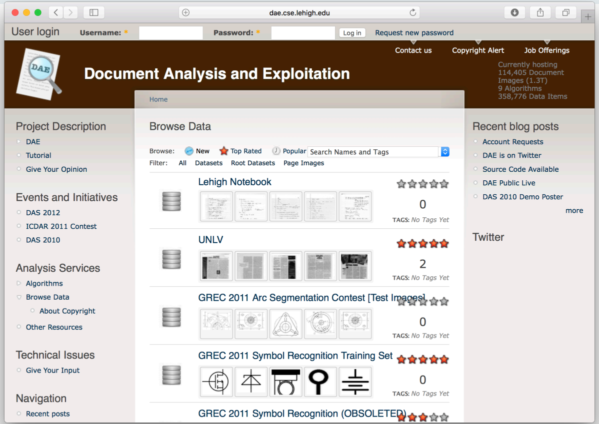


http://dae.cse.lehigh.edu

3.2 Rapid overview of the RR platforms: DAE

Description: [Lamiroy & Lopresti 16]

- Platform for **D**ocument **A**nalysis and **E**xploitation.
- Allows to run document analysis algorithms and apply comparisons.
- RR framework for document analysis with image data base.



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 platforms.
- 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.

ipol.im

IPOL Journal · Image Processing On Line

HOME · ABOUT · ARTICLES · PREPRINTS · WORKSHOPS · NEWS · SEARCH

Automatic Detection of Internal Copy-Move Forgeries in Images

Thibaud Ehret

article | demo | archive

published · 2018-07-25 →BibTeX

reference · Thibaud Ehret, *Automatic Detection of Internal Copy-Move Forgeries in Images*, Image Processing On Line, 8 (2018), pp. 167–191. https://doi.org/10.5201/ipol.2018.213

Communicated by Loïc Simon
Demo edited by Thibaud Ehret

Abstract

This article presents an implementation and discussion of the recently proposed 'Efficient Dense-Field Copy-Move Forgery Detection' by Cozzolino et al. This method is a forgery detection based on a dense field of descriptors chosen to be invariant by rotation. Zernike moments were suggested in the original article. An efficient matching of the descriptors is then performed using PatchMatch, which is extremely efficient to find duplicate regions. Regions matched by PatchMatch are processed to find the final detections. This allows a precise and accurate detection of copy-move forgeries inside a single suspicious image. We also extend successfully the method to the use of dense SIFT descriptors and show that they are better at detecting forgeries using Poisson editing.

Download

- full text manuscript: PDF (5.6M)
- source code: TAR/GZ

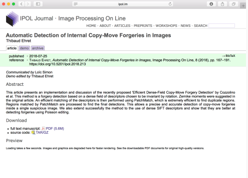
Preview

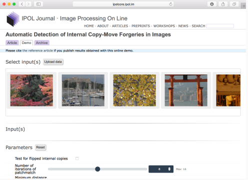
Loading takes a few seconds. Images and graphics are degraded here for faster rendering. See the downloadable PDF documents for original high-quality versions.

3.3 New Ways of Publications: Image Processing On Line

Short overview

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

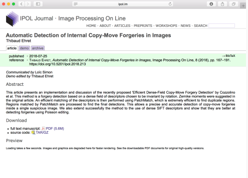


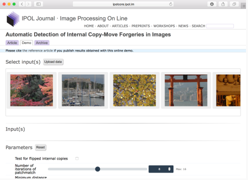


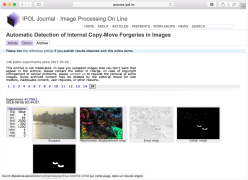
3.3 New Ways of Publications: Image Processing On Line

Short overview

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



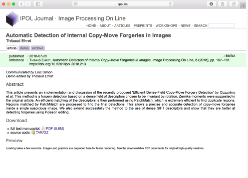


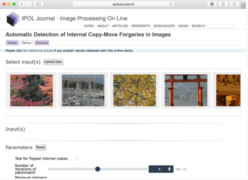


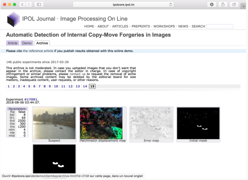
3.3 New Ways of Publications: Image Processing On Line

Short overview

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



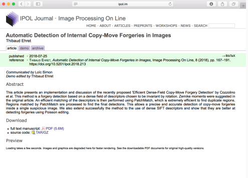


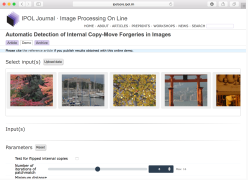


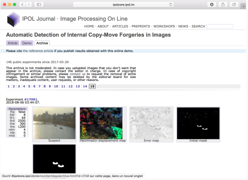
3.3 New Ways of Publications: Image Processing On Line

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.



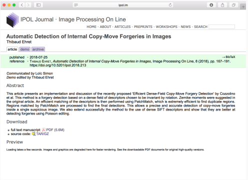


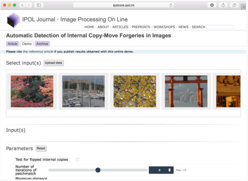


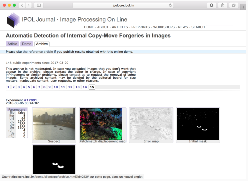
3.3 New Ways of Publications: Image Processing On Line

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 demontrations (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].

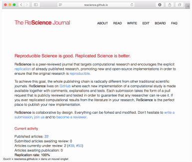
Details:

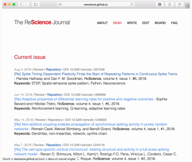
- Origin: first volume in 2015.
- Editorial Board:
  - Editors-in-Chief:
    - Konrad Hinsen (Molecular Biophysics - Python, C, Racket, Clojure).
    - Nicolas P. Rougier (Comp. Neuroscience, Computer Science - Python, C/C++).
  - 11 Associate Editors with roles in: Bioinformatics; Cognitive Modelling;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

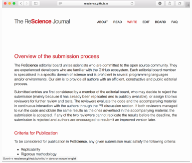
3.2 ReScience Journal: short overview

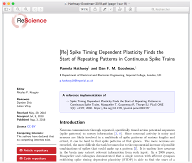
Characteristics

- Same presentation as in a “classic” journal.





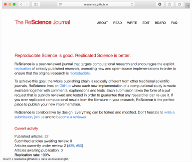


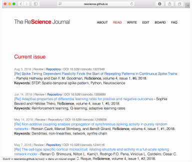


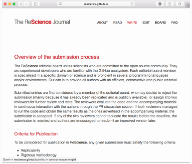
3.2 ReScience Journal: short overview

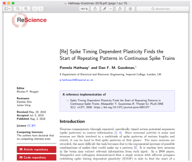
Characteristics

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





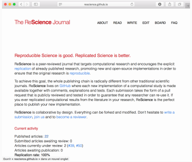


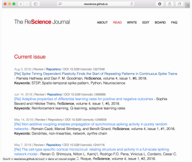


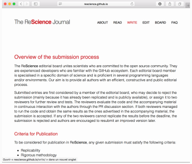
3.2 ReScience Journal: short overview

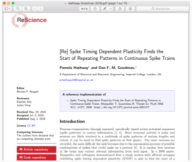
Characteristics

- 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).









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"*.

The Journal of Open Source Software

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:
  $\Rightarrow$ from a 2014 survey 90% mention to use software and 70% indicates that they were obligatory [Hettrick *et al.* 14].
- Software leak of scholarity support: no echosystem of publication, citation, acknowledge.
- JOSS is the contribution to offer mordern computational research results.

3.3 JOSS Journal: motivations

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

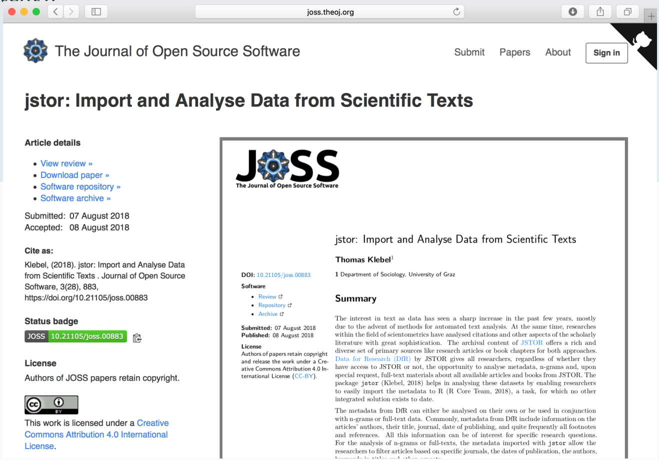
- Software more and more present in numerous disciplines:
  $\Rightarrow$ from a 2014 survey 90% mention to use software and 70% indicates that they were obligatory [Hettrick *et al.* 14].
- Software leak of scholarity support: no echosystem of publication, citation, acknowledge.
- JOSS is the contribution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

As mentionned by Buckheit and Dohono [Buckheit & Donoho 95]:
"An article about computational science in a scientic 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).



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
  $\Rightarrow$ collaboration with classic developement 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
  $\Rightarrow$ collaboration with classic developement tools (based on *GitHub*).

Content requested:

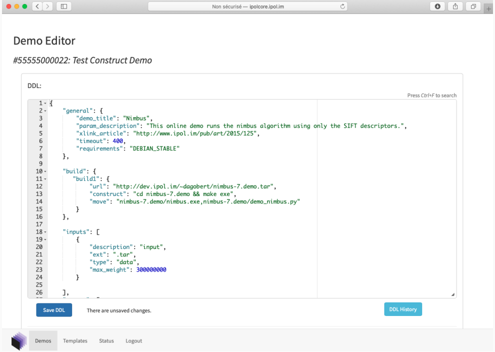
- Software need to be open source.
- Research application.
- Submitter needs to be main software contributor.
- Significant new contribution.
- Feature-complete (not partial).

4. Practical Session: Hands on IPOL Demos

4. Practical session: hands on the IPOL Demonstration System

Focus on online demonstration

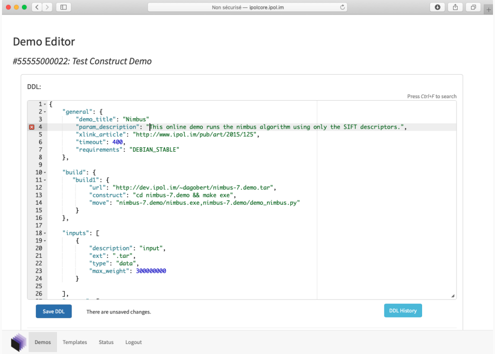
- Construction of demonstration from your own code.
- New Demo Editor panel:
  $\rightarrow$ to construct your demonstration without hard coding the demo.



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.

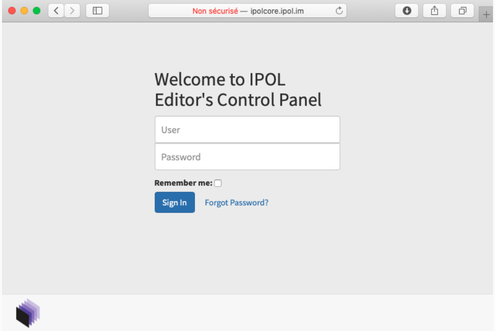


4. Practical session: hands on the IPOL Demonstration System

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.

`http://ipolcore.ipol.im/cp/login/`



4. Practical session: first step

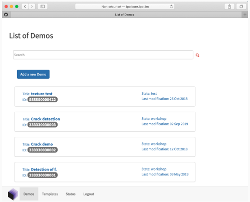
Step 1: main demo repository construction

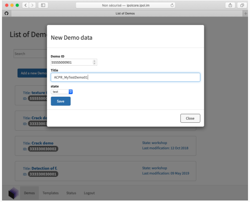
1. Login on : `http://ipolcore.ipol.im/cp/login`

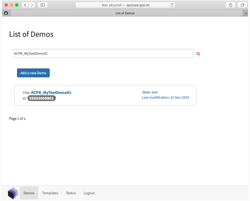
   (login: acpr2019 password: 222hhh888k)

2. Add a new Demo.

3. Select an number ID and titles using this convention:
   DemoID: 555559000XX Title: ACPRMyTestDemoXX





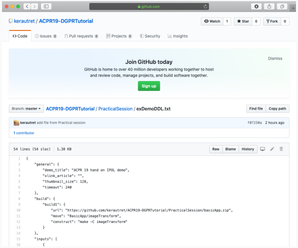


4. Practical session: second step

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`



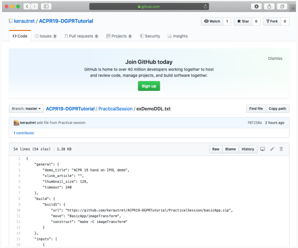
4. Practical session: second step

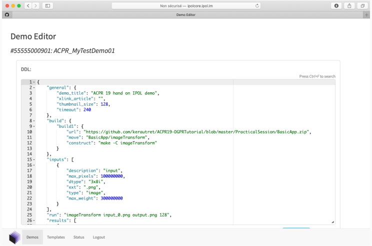
Step 2: Edit configuration file

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.





4. Practical session: second step

Step 2: Edit configuration file

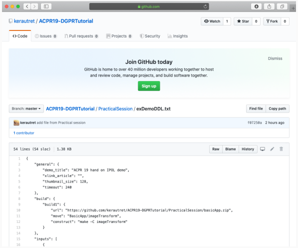
1. Start from the base example available from the tutorial repository:

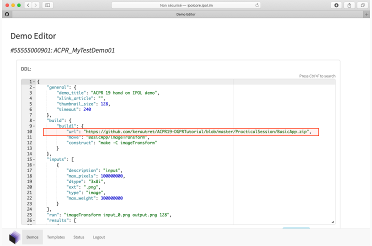
   `https://github.com/kerautret/ACPR19-DGPRTutorial` file: `PracticalSession/exDemoDDL.txt`

2. Copy the content in the demo editor panel.

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

   `https://github.com/kerautret/ACPR19-DGPRTutorial` file: `PracticalSession/BasicApp.zip`

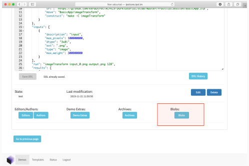


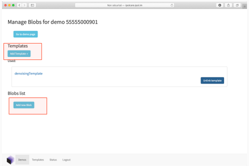


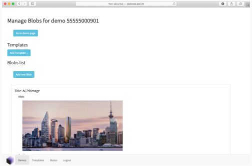
4. Practical session: last step

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).





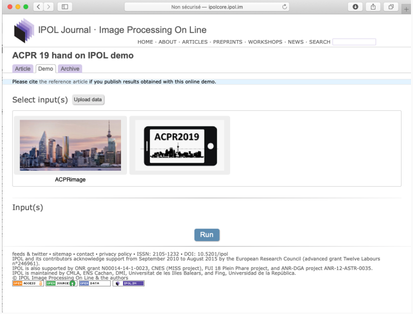


4. Practical session: last step

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)

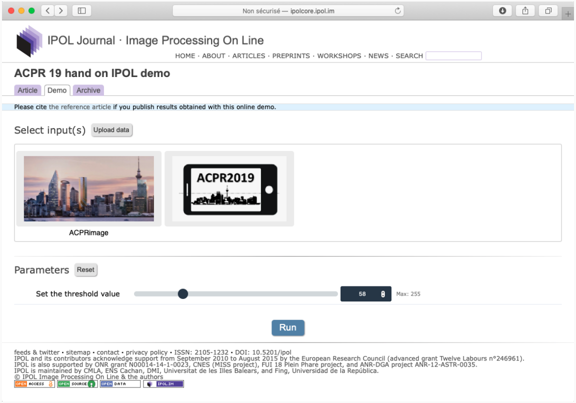
   `http://ipolcore.ipol.im/demo/clientApp/demo.html?id=555559000XX`



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.



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.

```
    "params": [
        {
            "id": "parameterID",
            "label": "Label description...",
            "type": "range",
            "values": {
                "default": 50,
                "max": 100,
                "min": 0,
                "step": 1
            }
        },
```

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)
- [Hettrick *et al.* 14] Hettrick, 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