Tetris :: Training Image Generator

* a SGeMS plugin

Alexandre Boucher
Jef Caers
Rahul Gupta
Addy Satija
Design Constraints

**Aim:**
- Allow complex parameterization of geometries and rules
- Strong core algorithm with expandable libraries
- Mix and match modules (increased combinatorial possibilities)

**Avoid:**
- Long procedural algorithm (no long list of *if - else*)
- Code repetitiveness (hard to maintain)
Modeling Dichotomy

**Geo-object**
- Define a geological entity
- A 3D volume
- Possibly complex geometry
- Unspecified location
- Unspecified relationships with other objects

**Simulation**
- Geo-object density
- Interactions between objects
- Stacking parameters
Building geo-object from elements

- A geo-object is a composition of elements
- An element is a unit of a larger ensemble (building blocks)
A flexible framework to build elements

Library of shapes

Modify shapes

Operations on Shape:

- shear( dx_dz, dy_dz )
- rotate( az, ax, ay )
- translate( dx, dy, dz )

Operations between Shapes:

- difference( Shape )
- merge( Shape )
- intersection( Shape )
Creating an element

There are no restrictions on the complexity of the element builder
Parameterization of the elements

Every parameter can be defined as a constant or a distribution.

Furthermore, parameters of the distribution can either be constant or be taken from a locally varying property (LVP).

Includes:

- Dimension of geo-objects
- Shearing, Rotation, Translation
- Moments of a distribution
Building complex Element

Union: 6 rotated and translated kernels
1 Sphere

Exponential Kernels
Gaussian Kernels
Creating a Geological Object

inner = HalfEllipsoid( 20, 20, 20 )  
outer = HalfEllipsoid( 30, 30, 15 )

inner.intersection( outer )  
outer.difference( inner )

Resulting geo-object
Asymmetrical mound with drapes

inner = HalfEllipsoid( 10, 25, 40 )
outer = HalfEllipsoid( 30, 30, 35 )
inner.shear( 0, 0.3 )
outer.shear( 0, 0.5 )

// Create the Drapes
drapes_top = HalfEllipsoid( 40, 40, 15 )
drapes_bot = HalfEllipsoid( 35, 35, 9 )

// Constraint by the envelope
inner.intersection( outer )
outer.difference( inner )

// Remove the core from the envelope
outer.difference( inner )
drapes.difference( inner )

// Remove the core from the drape
outer.difference( drapes )
drapes.difference( drapes_bot )

// Remove the drape from the envelope

outer.difference( drapes )
Faulted and Tilted Mound

// Create one Mound : inner and outer
inner.shear(-0.2, -0.5)
outer.shear(-0.2, -0.5)

// Create two Cuboids to partition the Mound
inner_faulted = Cuboid(80, 80, 50)  // Partition inner
outer_faulted = Cuboid(80, 80, 50)  // Partition outer

inner_faulted.translate(50, 0, 25)
inner_faulted.translate(50, 0, 25)
inner_faulted.intersection(inner)
outer_faulted.intersection(outer)
inner.difference(inner_faulted)
outer.difference(outer_faulted)

inner_faulted.rotate(0, 20, 0)  // tilt
outer_faulted.rotate(0, 20, 0)  // tilt
inner_faulted.translate(0, 7, 5)  // throw
outer_faulted.translate(0, 7, 5)  // throw
Building complex inner structures

Mimic the building of a carbonate mound through phases of accretion

The flank are debris
- steeper at the summit,
- gentler at the base

Build stochastically from 8 shapes and 12 operations

The structure can be saved in a file
Complex geological objects

Pre-configured geo-objects
Ensure geometrical and spatial relationships

Example Channel manager:
Manager that controls
crevasse splay placement
lobe placement

Example Carbonate Mound:
Manager that controls
Inner-outer-debris Element operations
Simulation parameters

- Stopping Criterion (*density*)
- Stacking (*simulate a set of object*)
- Positioning
  - Random
  - Intensity field
  - Bottom to Top (*simulate from the bottom to the top*)
- Interactions
  - No overlap (*Object1 never touches Object2*)
  - Attach (*Object1 touches Object2*)
  - Full Overlap (*Object1 is fully enclosed in Object2*)

*Can be used with all object types*
Positioning

Position the geological objects on the grid

- Random
- Intensity
- Bottom to Top
- Bottom to Top with intensity

independent of the objects: re-usable
Intensity field positioning
Bottom to top intensity positioning

Stochastic arrangement of six elements
- 2 (sheared) radial kernels for the core
- 3 half kernels for the debris

Contributor: Andre Jung, Tuebingen University,
Correlated spatially variable parameters

Anticline example:

- **Angles**
  - Outer Envelope Size
  - Inner Core Size

- **Shear**
  - Steeper angles:
    - Increased in shearing
    - Decreasing outer envelope
    - Fast decreasing of inner core size
Carbonate mounds on a anticline

More angular slopes:

- Increased in shearing
- Decreasing outer envelope size
- Fast decreasing of inner core size

MPS implications:

- Cannot be solved with a probability field
- The geometries characteristics are spatially dependent
Stacking objects

- Displacement vector \((dx, dy, dz)\)
- The displacement can be constant, spatially varying or stochastic
- Stack one geo-object type
- Build a sequence of stacked geo-objects
Stacking :: Geological objects
Interactions

- No overlap: Objects cannot touch
- Attach: up to 25% of the volume overlapped
- Full Overlap: 100% volume

*If objects are stacked: interactions are applied on the stack*
Remarks on locally varying parameters

Complexity and specificity is introduced through LVP

*Consider the geological settings*

Every parameter (geometries and interactions) can be locally variable

*Kriging/Simulation are options to generate LVP*

Danger of conflicting inputs

Need geologist involvement

*Analog information as stored on database will be key in parameterizing the geo-objects*
User’s perspective

Flexible / Expandable / Powerful Training Image Generator

User is shielded from the design through a User Interface

The UI can also be changed for local terminology

Need analog information in database
  • Object and Element parameterization
  • Locally varying parameters (LVP) fields

When the available shapes/rules are not sufficient:
  • new Positioning classes and/or
  • new Element classes
  • new Interaction classes
Boolean Simulator

- Currently testing as Object-based simulator for streamlines history matching (ppm)

- Perturb the intensity field driving the preferential locations of geo-objects

- Conditioning to hard data would require a new stopping criterion class
TODO list

- Expand elements library
- Stacking options
- Specialized objects library (e.g. turbidites)
- Additional checks for parameter consistency
- Expand interaction and positioning options
- Utility programs
Acknowledgements to Marco Pontiggia, Sergio Nardon and Giuseppe Serani at Eni for their support and valuable suggestions.

Questions?