Terrain generation by iterative process

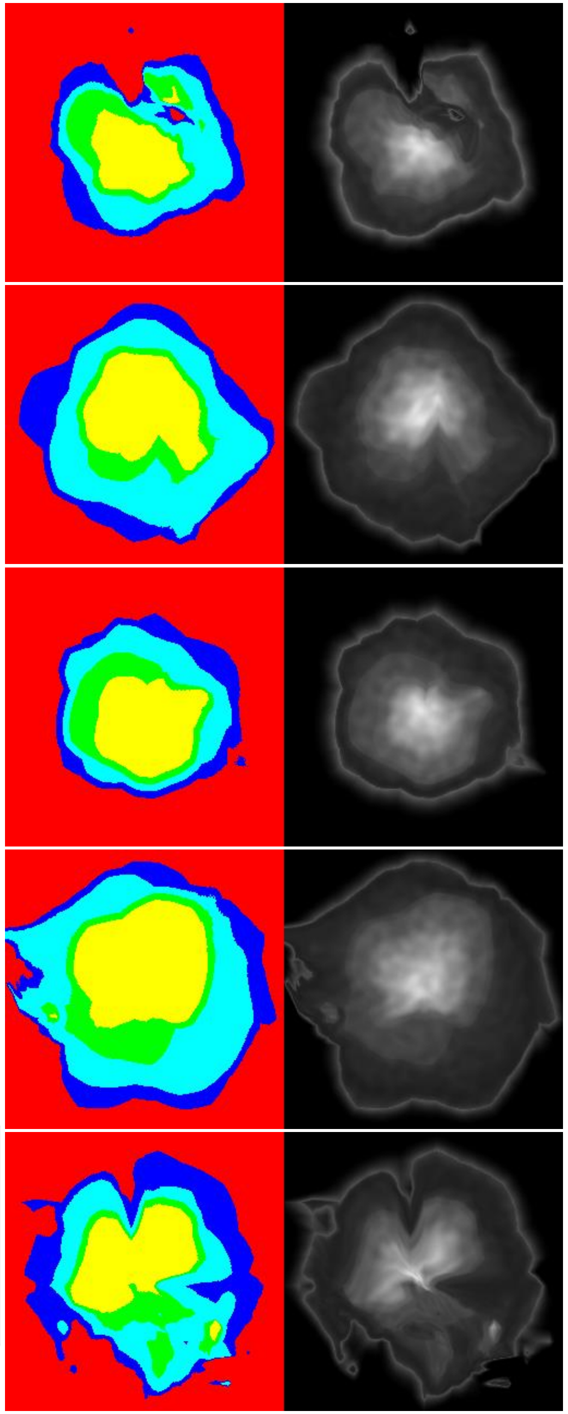
Overleaf link : https://www.overleaf.com/9817881829vdswhfvpkzww

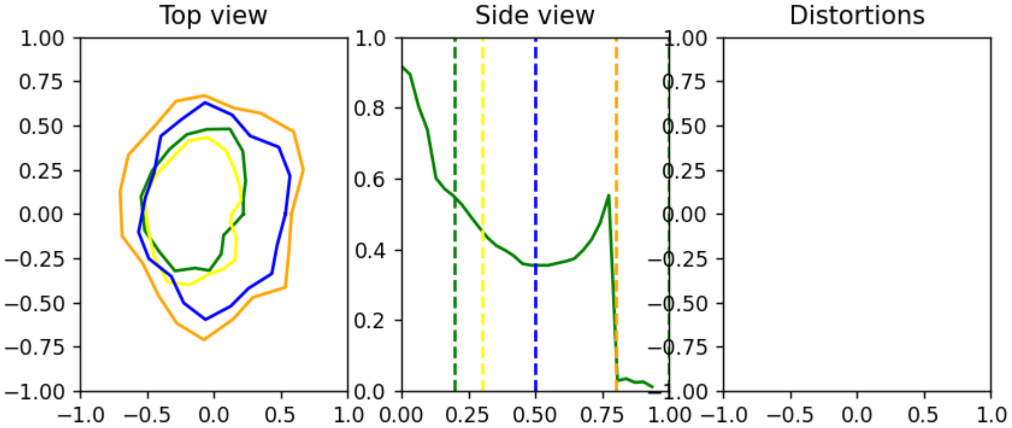
Main pipeline

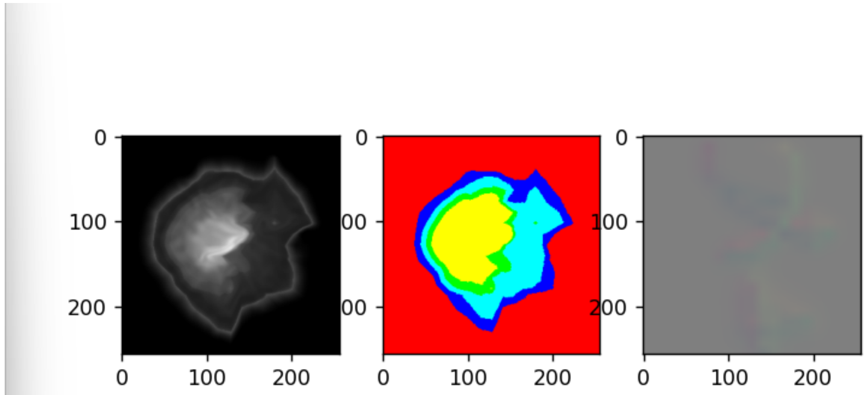
- Initial terrain generation
- Iterative augmentation
- Final augmentations

Initial terrain generation
Possible solution: Deep learning

- Creation of a synthetic terrain dataset from top-view and side-view sketches + distorsions from vector field
- Outputs: heightmap + labeled image
- Use of out-of-the-box cGAN («pix2pix») for interactive generation
- Related work: Guerin et al. «Interactive Example-Based Terrain Authoring with Conditional Generative Adversarial Networks» (2017)
    - But without any code required and very low learning time (transfer learning)







Iterative augmentation

- Process:
  - Compute properties at surface (depth, normals, water flow, and distance to other structures)
  - Find best fitting generation rule
  - Instantiate new structure
  - Repeat

Iterative augmentation – best fitting rule

- Compute a score at each point of the surface based on the defined rules
  - Here, used a **completely** ad-hoc fitting function to minimize

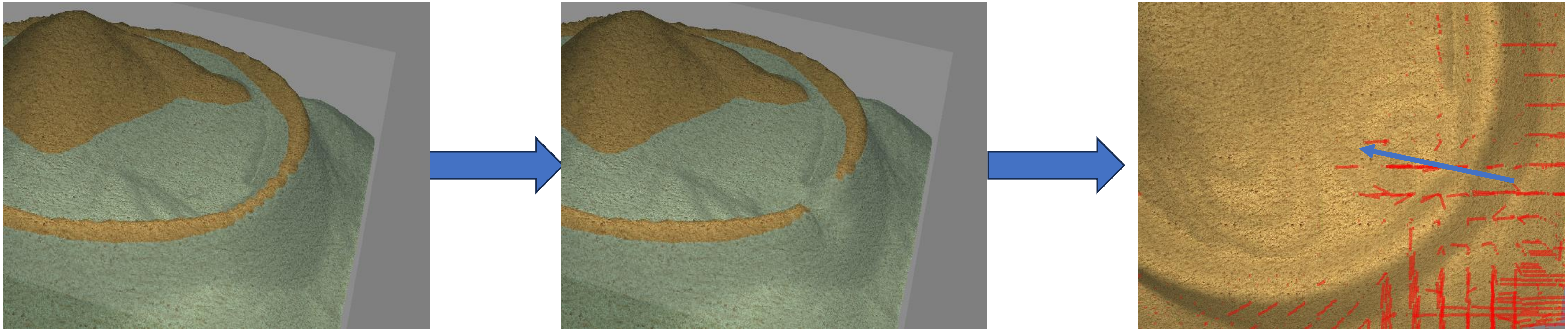
$$f(p) = 100 * \langle \hat{w}_f, \vec{n} \rangle + dist(barrier)^2 - ||\vec{w}_f||$$

- (Should be also compared with the score of all other rules)

Compute best fitting score

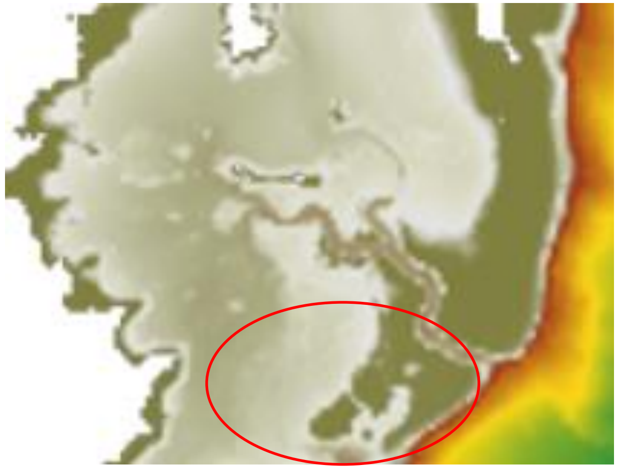
Instantiate structure (here, «passe»)

Direct effect on the water flow computation



Iterative augmentation – current state

- Everything is hard-coded, different rules don’t «compete» yet
- 3 first cases to create :
  - Passe, Delta, Motu



- Once those are done, I’m confident we have everything for many other structures (≠ coral areas, water inlets, caves, coral boulders, fractures, sand beds, etc.)

Final augmentation

- 3D erosion process based on water flow
- Coral models instantiation from the gen. rules